Cyber Public School

OSCP PRACTICE PART 03

https://cyberpublicschool.com/

OSCP PRACTICE PART 03

OSCP PRACTICE PART 03

Passive Information Gathering

Passive information gathering is a type of information gathering technique in which information is collected without actively engaging with the target system or individuals. This technique involves collecting information from publicly available sources such as social media, websites, news articles, and other online platforms.

Passive information gathering is often used by individuals or organizations to gather intelligence on their competitors, potential clients, or other relevant parties. It can also be used to identify vulnerabilities in a system or to gather information about potential targets for social engineering attacks.

Some examples of passive information gathering techniques include:

1. Open-source intelligence (OSINT): This involves collecting information from publicly available sources such as social media, websites, and news articles.
2. Social engineering: This involves using deception or manipulation to trick individuals into revealing sensitive information.
3. Network reconnaissance: This involves scanning a network to identify hosts, services, and other information that can be used to gain unauthorized access.

4. Dumpster diving: This involves searching through trash or recycling bins for information that may be useful.

It is important to note that while passive information gathering techniques may be legal in some cases, they can also be illegal if they involve accessing unauthorized information or violating someone's privacy. Therefore, it is important to ensure that any information gathering activities are conducted within the boundaries of the law and ethical guidelines.

OSCP PRACTICE PART 03

Taking Notes

Taking notes is an important skill that can help individuals to retain information, organize their thoughts, and improve their memory. There are several effective strategies for taking notes, including:

1. Use a structured format: Structuring notes in a logical and organized way can make it easier to review and understand them later. One popular format is the Cornell Method, which involves dividing a page into three sections: a section for main points, a section for supporting details, and a summary section at the bottom.

2. Use keywords and abbreviations: Using keywords and abbreviations can help to condense information and make notes more concise. For example, using "w/" instead of "with" or "bc" instead of "because" can save time and space.

3. Stay focused: It is important to stay focused during lectures, meetings, or other situations where notes are being taken. Avoid distractions such as phones or other electronic devices, and try to stay engaged with the speaker or presenter.

4. Review and revise: Reviewing and revising notes after they have been taken can help to reinforce the information and identify any gaps or areas that need clarification.

5. Use technology: There are many tools available for taking notes, including digital note-taking apps such as Evernote, Google Keep, or OneNote. These apps can make it easy to organize and access notes across different devices.

6. Highlight and underline: Highlighting or underlining important information can help to draw attention to key points and make them easier to find later.

Remember, the goal of note-taking is to capture important information and make it easier to remember and review later. By using effective note-taking strategies, individuals can improve their ability to retain information and stay organized.

OSCP PRACTICE PART 03

Website Recon

Website reconnaissance, also known as website discovery, is the process of gathering information about a website or web application. This information can be used to identify potential vulnerabilities or to gain a better understanding of the target site's architecture and functionality.

Here are some common techniques for website reconnaissance:

Passive reconnaissance: Passive reconnaissance involves gathering information from publicly available sources such as search engines, social media, and other online resources. This can include identifying the site's owner, the site's IP address, and any subdomains associated with the site.

Active reconnaissance: Active reconnaissance involves interacting directly with the target site or application to gather information. This can include scanning the site for open ports, running automated vulnerability scans, or using web application testing tools to identify potential vulnerabilities.

Footprinting: Footprinting involves gathering information about the target site's infrastructure, including its operating system, web server software, and other details. This information can be used to identify potential vulnerabilities and to develop an attack strategy.

OSCP PRACTICE PART 03

Social engineering: Social engineering techniques can be used to gather information about the target site's employees or users. This can include phishing attacks or other forms of social engineering to obtain login credentials or other sensitive information.

Directory enumeration: Directory enumeration involves identifying directories and files on the target site. This can be done manually or using automated tools, and can provide information about the site's structure and functionality.

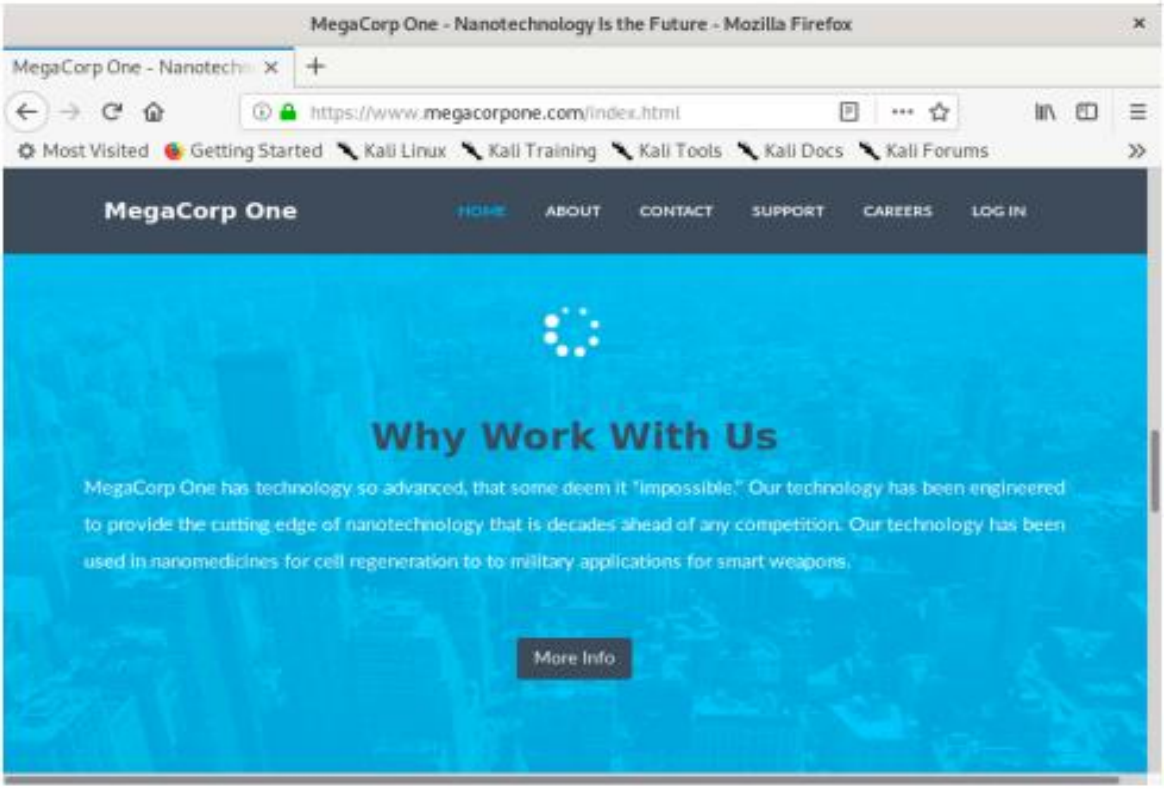
It is important to note that website reconnaissance should only be conducted with the owner's permission, or in the context of a legitimate security assessment. Unauthorized website reconnaissance can be illegal and can result in serious consequences.

It is important to note that gathering information about a target organization in this manner without proper authorization can be considered unethical and potentially illegal. It is crucial to obtain permission from the organization or to conduct such activities within the context of a legitimate security assessment.

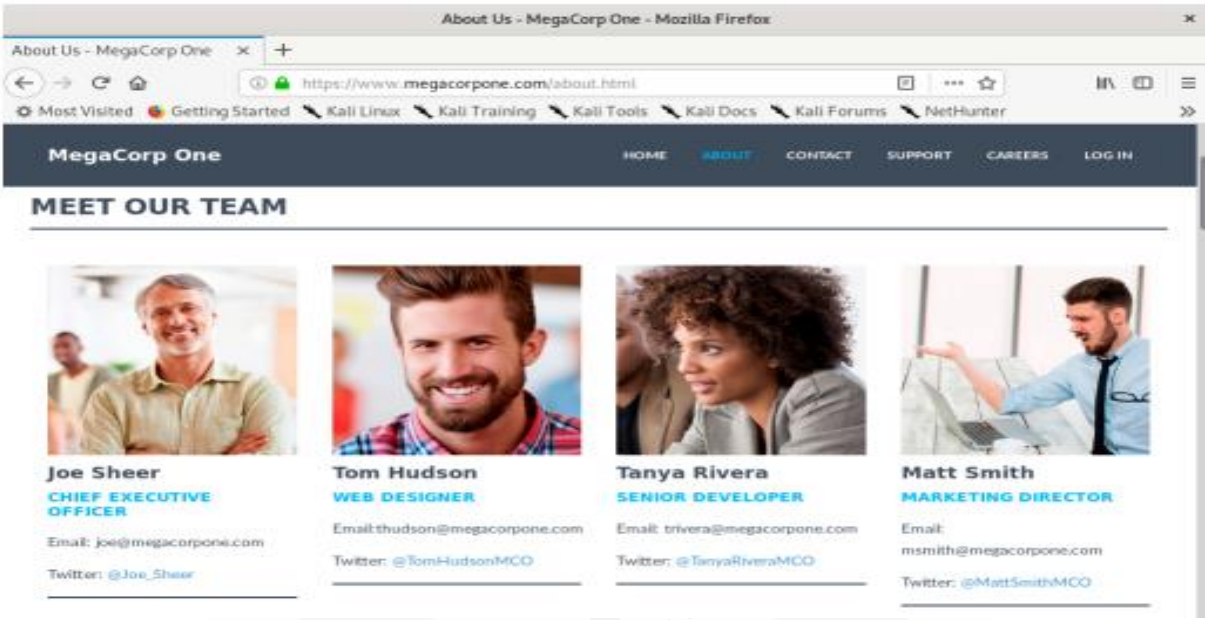
OSCP PRACTICE PART 03

Assuming proper authorization has been obtained, browsing a target organization's website can indeed provide valuable information. In this case, a quick review of MegaCorp One's website reveals that they are a nanotech company. The about page at https://www.megacorpone.com/about.html reveals email addresses and Twitter accounts of some of their employees.

This information can be useful in conducting further reconnaissance activities, such as social engineering attempts or targeted phishing attacks. However, it is important to ensure that any information gathered is used ethically and within the boundaries of the law.



OSCP PRACTICE PART 03

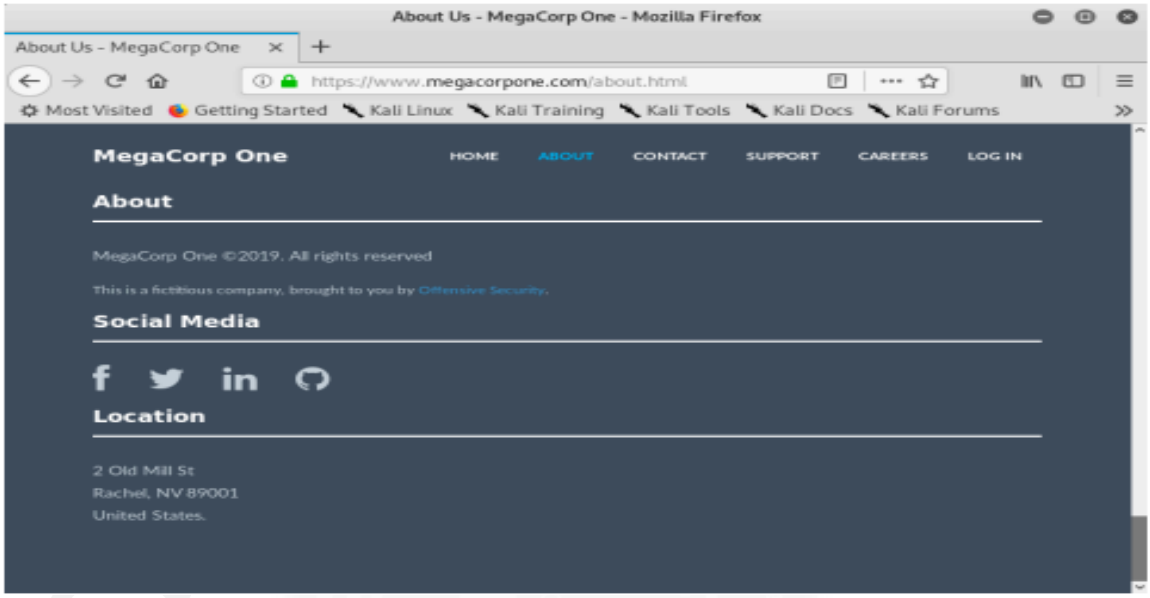


It is important to note that any social media information gathering campaign should be conducted within the boundaries of the law and with respect to the privacy of individuals. Gathering information through social media channels should not involve any unethical or illegal activities.

Regarding the email address format, this observation can indeed be useful in conducting further reconnaissance activities, such as guessing potential email addresses for individuals within the organization. However, it is important to ensure that any such activities are conducted within the context of a legitimate security assessment.

OSCP PRACTICE PART 03

Recording the corporate social media accounts can also provide valuable information, such as the types of content the organization posts and the engagement it receives from its followers. This information can be useful in developing social engineering or phishing campaigns, or in identifying potential weaknesses in the organization's security posture. Again, it is important to ensure that any information gathering is conducted ethically and within the boundaries of the law.



OSCP PRACTICE PART 03

Whois Enumeration

Whois enumeration is the process of gathering information about a target organization or website by querying public Whois databases. Whois is a protocol used to obtain information about domain names, IP addresses, and other network-related information. The information obtained through Whois enumeration can include the name and contact information of the domain owner, as well as technical information about the domain, such as the name servers and registrar.

Here are some common techniques for Whois enumeration:

1. Whois lookup tools: There are a number of online tools that allow you to perform a Whois lookup on a target domain. These tools can provide information about the domain owner, registration and expiration dates, and other details.

2. Command-line tools: Command-line tools such as "whois" can be used to perform a Whois lookup from a terminal. This can be useful for automating the process of querying multiple domains.

3. Whois servers: There are a number of Whois servers that can be queried directly using the Whois protocol. This can be useful for obtaining information about domain names that may not be publicly listed in Whois databases.

some examples of command-line tools that can be used for Whois enumeration:

1. "whois" command:

The "whois" command can be used to perform a Whois lookup from a terminal. Here's an example of how to use it:

```ruby
$ whois example.com
```

This will return the Whois information for the domain "example.com", including the registrar, registration date, and contact information.

2. "whois" utility in Windows:

If you're using Windows, you can use the "whois" utility to perform a Whois lookup. Here's an example:

```makefile
C:\> whois example.com
```

This will return the Whois information for the domain "example.com".

3. "whois" command in Linux:

In Linux, the "whois" command can be installed using the package manager. Here's an example:

```ruby
$ sudo apt-get install whois
$ whois example.com
```

This will install the "whois" command and then perform a Whois lookup for the domain "example.com".

It's important to note that not all Whois information is publicly available, and some domains may have privacy protections that hide the owner's contact information. Additionally, some Whois servers may impose rate limits or other restrictions on queries. Therefore, it's important to ensure that any Whois enumeration activities are conducted within the boundaries of the law and with respect for the privacy of individuals.

Example :

```
whois megacorpone.com
```

```
whois 38.100.193.70
```

OSCP PRACTICE PART 03

Google Hacking

Google hacking is a technique used to discover vulnerabilities and sensitive information about a target organization by using advanced search operators in Google search. Google hacking involves using specific search queries to uncover information that is not normally available through standard search methods.

Here are some common Google search operators used in Google hacking:

1. **"site:" operator:** This operator limits the search to a specific website or domain. For example, "site:example.com" will only return search results from the example.com domain.
2. **"filetype:" operator:** This operator limits the search to a specific file type. For example, "filetype:pdf" will only return search results that are in PDF format.
3. **"intitle:" operator:** This operator limits the search to pages with a specific word or phrase in the title. For example, "intitle:login" will only return search results with the word "login" in the title.
4. **"inurl:" operator:** This operator limits the search to pages with a specific word or phrase in the URL. For example, "inurl:admin" will only return search results with the word "admin" in the URL.

5. **"related:" operator:** This operator returns pages that are related to a specific website. For example, "related:example.com" will return pages that are related to the example.com website.

some examples of Google hacking Search query .

Site search: To search for pages on a specific website or domain, use the "site:" operator followed by the website or domain name. For example, to search for pages related to security on the website "example.com", you would use the following query:

```makefile
site:example.com security
```

Filetype search: To search for a specific file type, use the "filetype:" operator followed by the file extension. For example, to search for PDF documents related to a specific topic, you would use the following query:

```vbnet
filetype:pdf "topic of interest"
```

OSCP PRACTICE PART 03

Intitle search: To search for pages with a specific word or phrase in the title, use the "intitle:" operator followed by the search term. For example, to search for pages related to login pages that have "admin" in the title, you would use the following query:

```vbnet
intitle:"admin login"
```

Inurl search: To search for pages with a specific word or phrase in the URL, use the "inurl:" operator followed by the search term. For example, to search for pages related to web application vulnerabilities that have "php" in the URL, you would use the following query:

```makefile
inurl:php vulnerability
```

Related search: To search for pages related to a specific website, use the "related:" operator followed by the website or domain name. For example, to search for pages related to the website "example.com", you would use the following query:

```makefile
related:example.com
```

OSCP PRACTICE PART 03

It's important to use Google hacking techniques ethically and within the boundaries of the law. Additionally, it's important to note that Google may impose rate limits or other restrictions on search queries, so it's important to use these techniques sparingly and with caution.

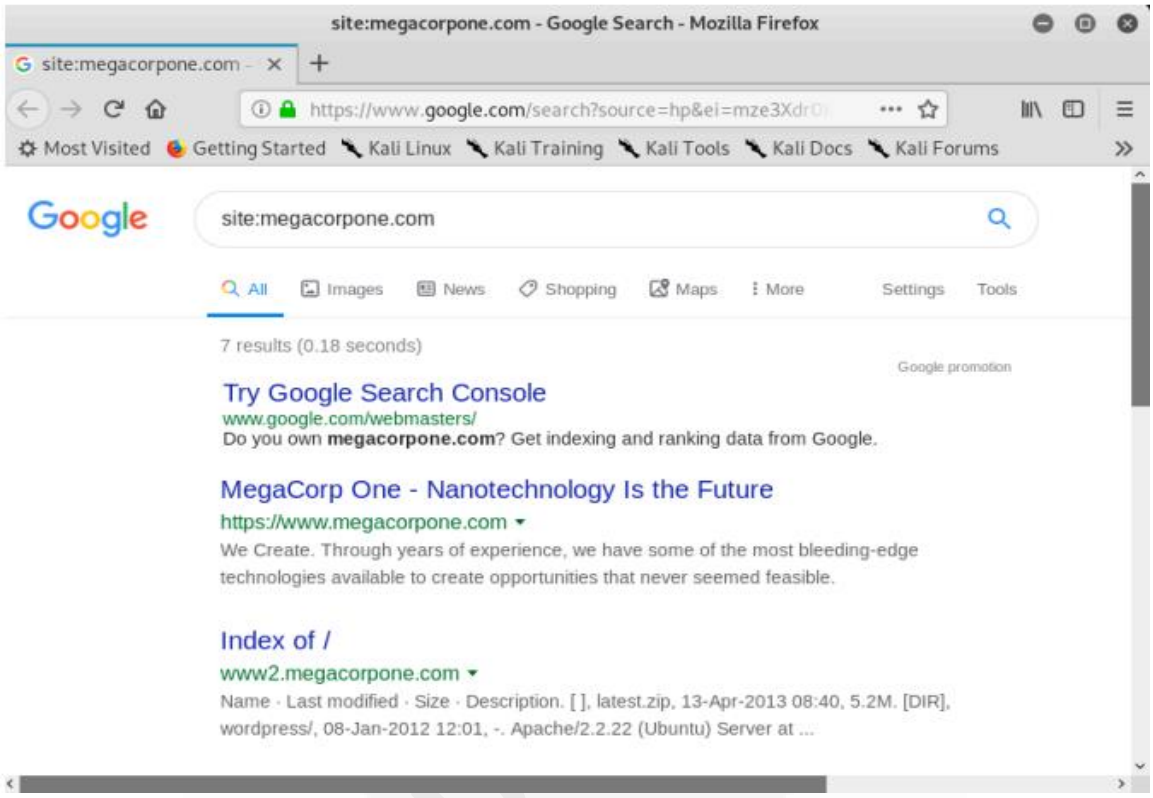
Example :

To use the site operator, you simply type "site:" followed by the domain name you want to search within. For example, if you want to search for all pages on the example.com domain that mention "security," you would type "security site:example.com" into the Google search bar.

This can be helpful when you want to explore the contents of a specific website or when you want to limit your search results to a particular domain. It can also be useful when performing reconnaissance on a target organization to see what information is publicly available about them on their website.

Keep in mind, however, that not all information about an organization may be available on their website. Google hacking can be a useful tool for information gathering, but it should be used ethically and with caution.

OSCP PRACTICE PART 03

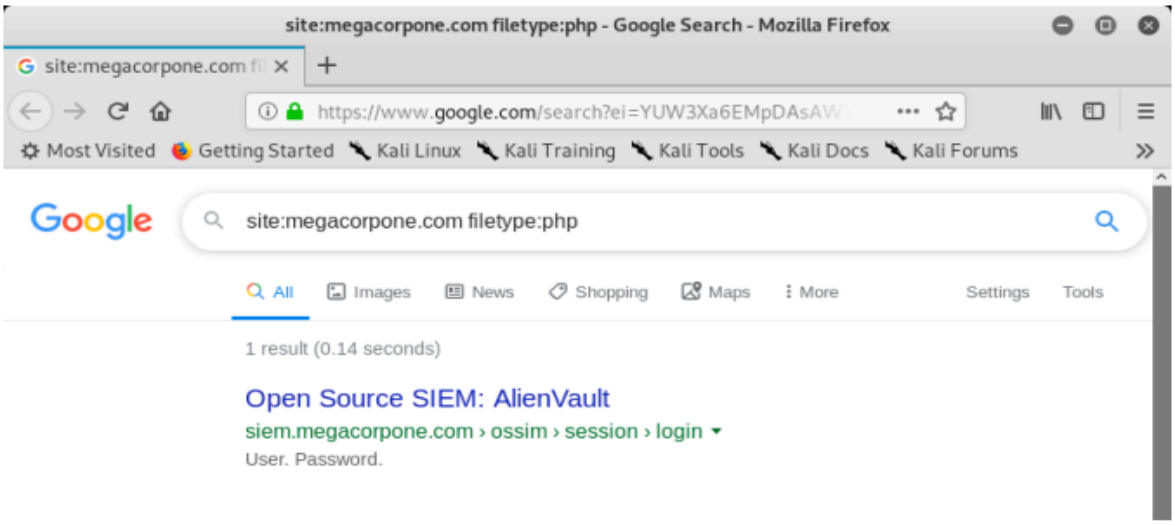


we can use the "filetype" or "ext" operator. This allows us to search for files of a specific type or extension. For example, if we want to search for all PHP files on the www.megacorpone.com domain, we can use the search query "filetype:php site:megacorpone.com" in the Google search bar.

This search will only return results that are PHP files on the megacorpone.com domain. This can be useful when searching for specific types of files, such as configuration files or database backups, that may contain sensitive information.

It's important to note that while these operators can be powerful tools for information gathering, they should be used ethically and with caution. It's important to respect the privacy and security of others when using Google hacking techniques.

OSCP PRACTICE PART 03

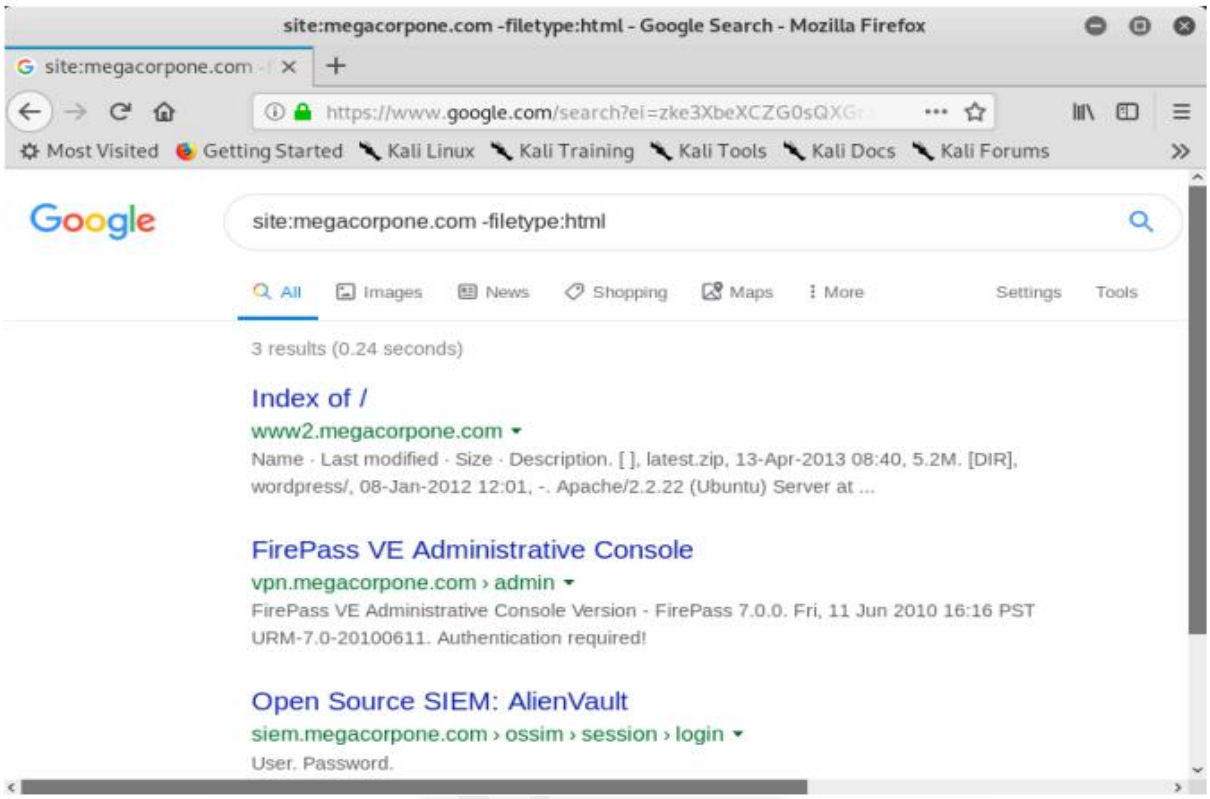


The **"ext"** operator can be helpful in identifying the programming languages used on a website. By using search queries such as **"ext:jsp", "ext:cfm", or "ext:pl",** we can find indexed Java Server Pages, Coldfusion, and Perl pages respectively.

Additionally, we can use the "-" operator to exclude certain items from our search results. For example, if we want to find interesting non-HTML pages on the megacorpone.com domain, we can use the search query **"site:megacorpone.com -filetype:html".** This will exclude all HTML pages from our search results, leaving us with potentially more interesting pages to explore.

It's important to use these operators responsibly and ethically, and to always obtain proper authorization before performing any kind of reconnaissance or penetration testing on a system or network.

OSCP PRACTICE PART 03

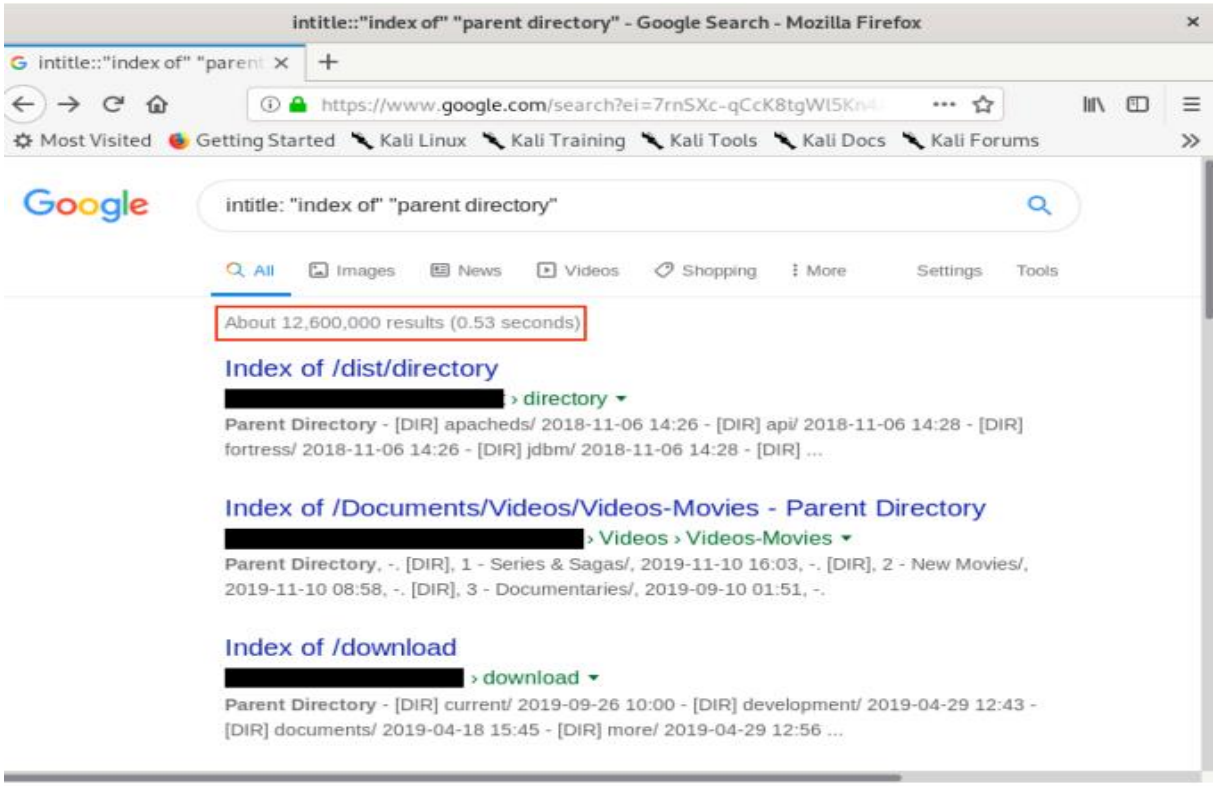


The search query **"intitle:"index of" "**parent directory"" is a commonly used Google hacking technique to find pages that contain the phrase "index of" in the page title and the phrase "parent directory" on the page.

This technique can be useful for finding directories or file listings that may not be intended for public access. The search results may reveal directory listings that include sensitive files or information.

It's important to keep in mind that this technique should only be used for legitimate purposes and with proper authorization. Unauthorized access to sensitive information can have serious consequences, both legally and ethically. It's important to obtain proper authorization and use these techniques responsibly.

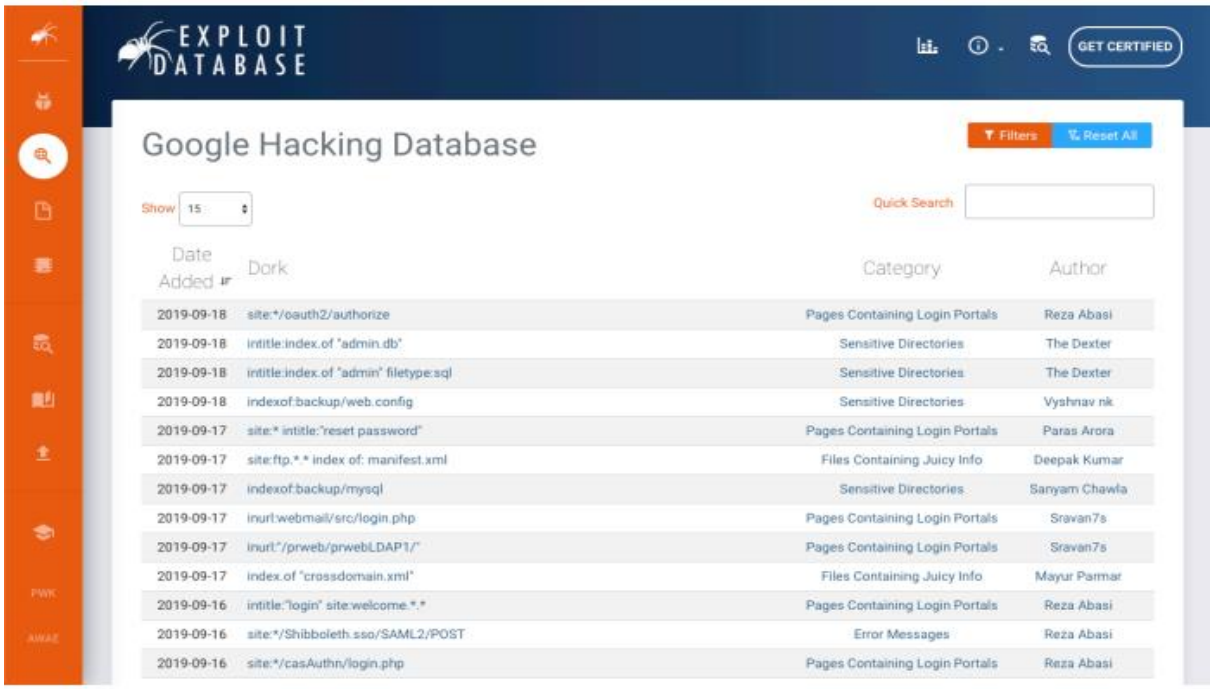
OSCP PRACTICE PART 03



Directory listing pages can be a result of misconfigurations on a web server, and they can reveal sensitive information if not properly secured. Using the **"intitle:"index of"** "parent directory"" search query is just one example of how search operators can be combined to reveal potential security weaknesses.

The Google Hacking Database is a great resource for finding creative search queries that can be used to uncover potential vulnerabilities or sensitive information. It contains a wide variety of search queries that use various combinations of search operators to reveal interesting results. However, it's important to use these search queries ethically and with proper authorization. Unauthorized access to sensitive information can have serious consequences, both legally and ethically.

OSCP PRACTICE PART 03



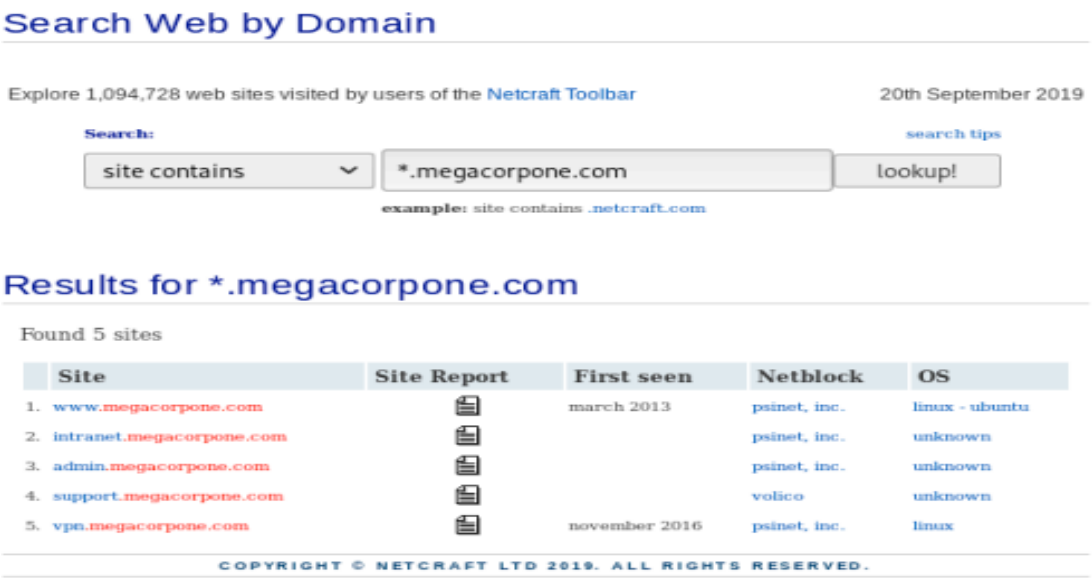
Mastery of search engine operators and a keen sense of deduction are essential skills for effective search engine "hacking" or reconnaissance. With the right combination of search operators, an attacker can uncover potential vulnerabilities or sensitive information that can be used to launch further attacks.

However, it's important to use these techniques ethically and with proper authorization. Unauthorized access to sensitive information can have serious consequences, both legally and ethically. Additionally, these techniques should be used in conjunction with other security measures, such as vulnerability scanning and penetration testing, to identify and address potential security weaknesses.

Netcraft

Netcraft is an internet security company that provides a range of services related to internet security, including phishing protection, malware scanning, and SSL/TLS certificate validation. Netcraft is also known for its public reports on the state of the internet, which cover topics such as web server market share, domain registration trends, and internet infrastructure developments.

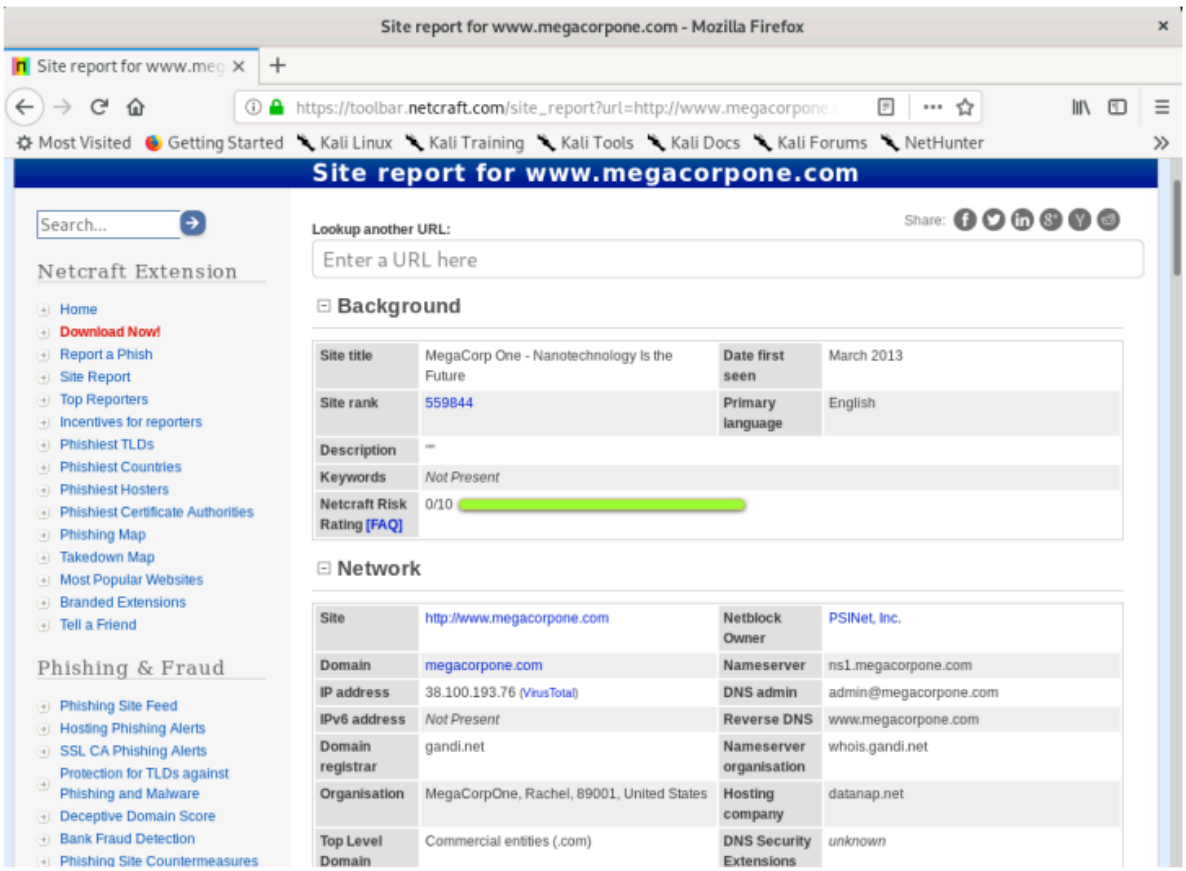
One of the key services provided by Netcraft is its anti-phishing toolbar, which is available as a browser extension for Google Chrome, Mozilla Firefox, and Opera. The toolbar alerts users to potential phishing websites and other online scams, and provides detailed information about the sites they visit.



OSCP PRACTICE PART 03

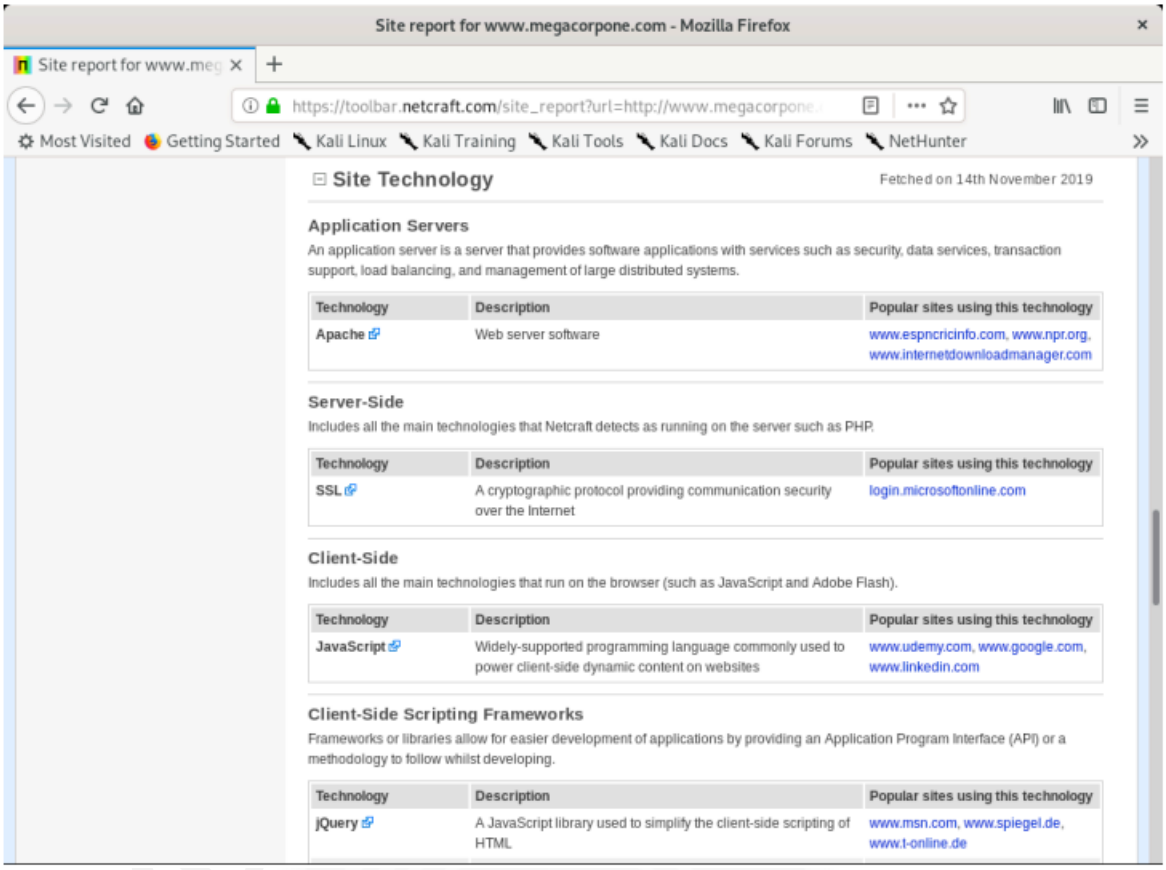
Netcraft also offers a range of tools and services for website owners and administrators, including website scanning and monitoring, SSL/TLS certificate validation, and vulnerability assessment. These tools can help organizations identify and address potential security weaknesses in their online presence.

Overall, Netcraft is a respected and well-known player in the internet security industry, with a range of tools and services that can help organizations protect themselves and their users from online threats.



OSCP PRACTICE PART 03

The start of the report covers registration information. However, if we scroll down, we discover various “site technology” entries:



This list of subdomains and technologies will prove useful as we move on to active information gathering and exploitation.

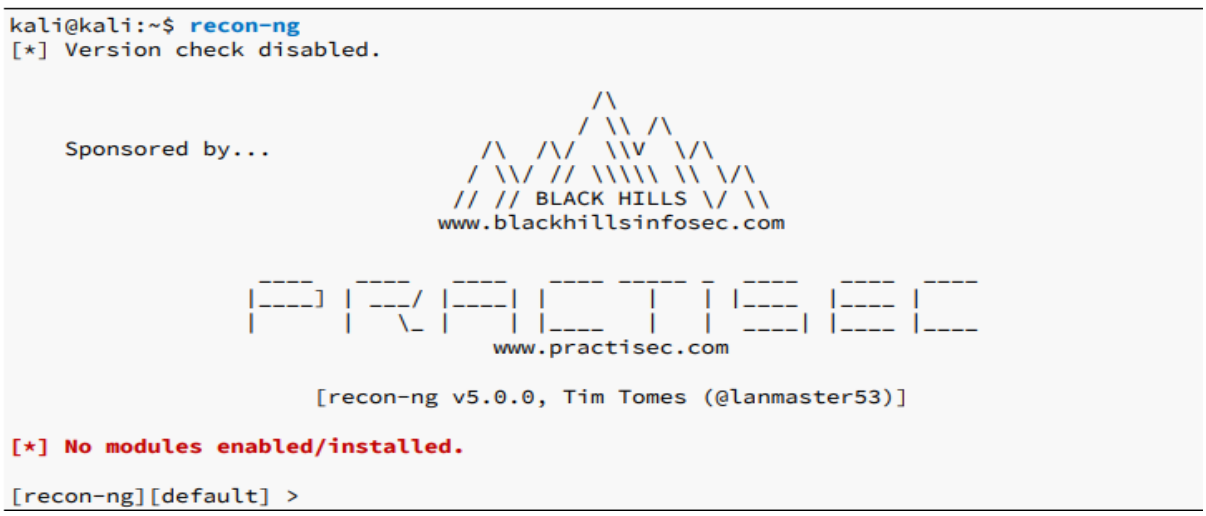
OSCP PRACTICE PART 03

Recon-ng

Recon-ng is an open source reconnaissance framework written in Python that is used for information gathering and reconnaissance in the context of penetration testing and security assessments. It provides a modular approach to information gathering, with a variety of modules for performing tasks such as passive information gathering, active information gathering, and vulnerability scanning.

Recon-ng is designed to be flexible and extensible, allowing users to create their own modules to perform custom tasks. It also includes a number of built-in modules for interacting with various data sources, such as search engines, social media sites, and WHOIS databases.

One of the key features of Recon-ng is its ability to integrate with other tools and frameworks, such as Metasploit and the Social Engineer Toolkit (SET). This allows users to seamlessly integrate Recon-ng into their existing workflows and toolchains.



I can provide further information on how to install modules in recon-ng.

After searching for modules in the marketplace, we can install them using the marketplace install command, followed by the name of the module we want to install. For example, to install the recon/domains-hosts/bing_domain_web module, we would enter:

```bash
marketplace install recon/domains-hosts/bing_domain_web
```

Once the module is installed, we can use it in our reconnaissance by selecting it with the use command, followed by the name of the module. For example:

```bash
use recon/domains-hosts/bing_domain_web
```

From there, we can set any required options for the module and run it with the run command.

Example :

In this example, we will search for modules that contain the term **github:**

```
marketplace search github
[*] Searching module index for 'github'...
```

Path	Version	Status	D	K
recon/companies-multi/github_miner	1.0	not installed		*
recon/profiles-contacts/github_users	1.0	not installed		*
recon/profiles-profiles/profiler	1.0	not installed		
recon/profiles-repositories/github_repos	1.0	not installed		*
recon/repositories-profiles/github_commits	1.0	not installed		*
recon/repositories-vulnerabilities/github_dorks	1.0	not installed		*

D = Has dependencies. See info for details.
K = Requires keys. See info for details.

To examine the **recon/domains-hosts/google_site_web** module, we can use the following command in the **recon-ng** prompt:

bash

```
marketplace info recon/domains-hosts/google_site_web
```

This will provide information about the module, including a description, author, and any dependencies or required API keys.

The modules marked with an asterisk in the **"K"** column of the marketplace search results require credentials or API keys for third-party providers.

To use these modules, we first need to obtain the required credentials or API keys from the respective providers. Once we have the credentials or keys, we can set them in recon-ng using the keys add command, followed by the name of the provider and the credential/key value. For example:

```csharp
keys add shodan_api_key 1234567890abcdefghijklmnop
```

This sets the Shodan API key for recon-ng to **"1234567890abcdefghijklmnop".**

After setting the required keys, we can use the modules that require them just like any other module in recon-ng.

To install the **recon/domains-hosts/google_site_web** module in recon-ng, you can use the following command:

```bash
marketplace install recon/domains-hosts/google_site_web
```

This will download and install the module so that it is available for use in your recon-ng workspace.

To load the module, we can use the following command:

```
lua
```

```
module load recon/domains-hosts/google_site_web
```

After loading the module, we can use the info command to display details about the module and its required parameters:

```
sql
```

```
recon-ng> info

  Name: Google Site Web
  Author: Tim Tomes (@LaNMaSteR53)
  Last Modified: 2019-04-01
  Description: Enumerate subdomains from Google using the site operator.
  Comments: Site must be a valid domain, such as google.com, and not a keyword. The limit parameter determines how many pages of results to fetch from Google, with a max of 10. (Default: 2)
  Path: /opt/recon-ng/modules/recon/domains-hosts/google_site_web.py
  Needs Keys: False

Options:
--------
Name       Current Value  Required  Description
----       -------------  --------  -----------
SOURCE                    yes       Source to use for domain names
SITE                      yes       Site to use for search (e.g. google.com)
LIMIT      2              no        Maximum number of pages to fetch (10 max)
```

The output contains additional information about the module now that we've installed and loaded it. According to the output, the module requires the use of a source, which is the target we want to gather information about.
In this case, we will use options set SOURCE megacorpone.com to set our target domain:

```
options set SOURCE megacorpone.com
```

We have four hosts in our database but no additional information on them. Perhaps another module
can fill in the IP addresses.
Let's examine **recon/hosts-hosts/resolve** with **marketplace** info

bash

```
marketplace info recon/hosts-hosts/resolve
```

The module install it. We can do that with the following command:

```
bash
marketplace install recon/hosts-hosts/resolve
```

here's an example command to load the module and display its options:

```
bash
modules load recon/hosts-hosts/resolve
info
```

This will display the options for the **resolve** module, which includes the required **SOURCE** parameter and optional parameters such as **BATCHSIZE** and **THREADS.**

To resolve the IP addresses for the hosts in our recon-ng database, we will use the "recon/hosts-hosts/resolve" module with the source option set to "default", which will automatically run the module against all four hosts without IP addresses in the database.
To run the module, we can use the run command followed by the module name, like this:

```
bash
recon/hosts-hosts/resolve
```

If we show hosts again, we can verify the database has been updated with the results of both
modules:
To show the hosts, we can use the hosts command:

```
arduino
recon-ng[default] > hosts list
```

This should display a list of hosts with their corresponding IP addresses.

Open-Source Code

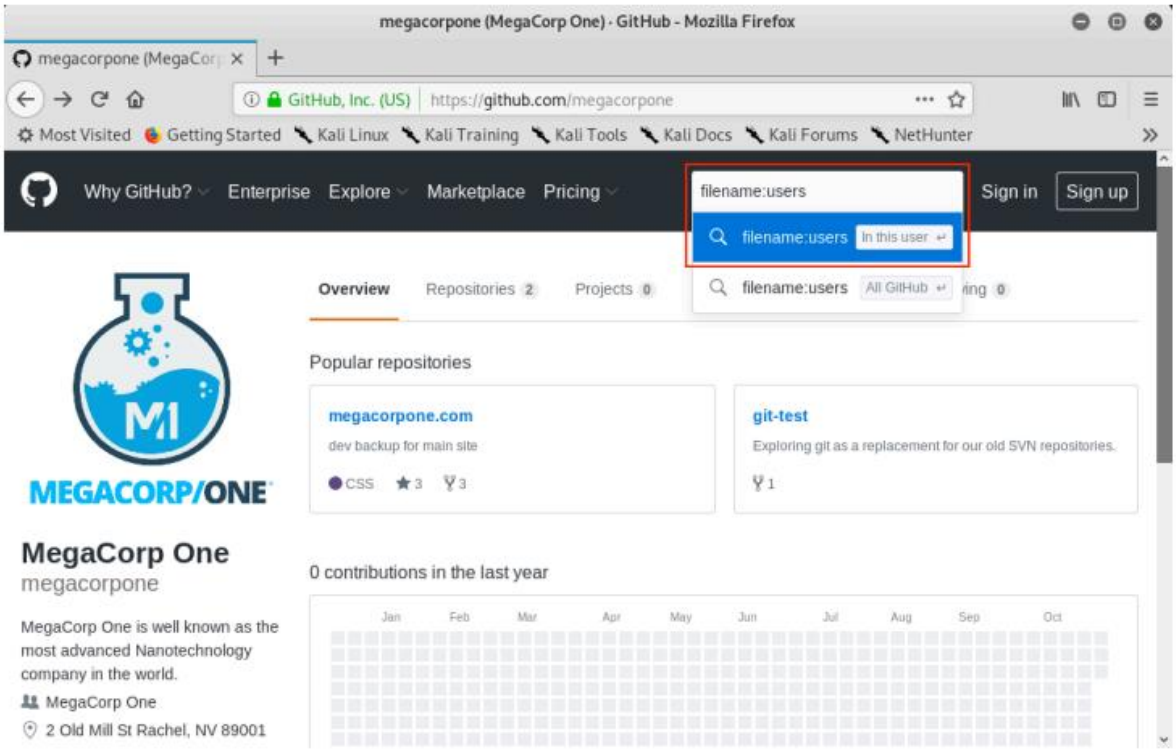
Open-source code refers to software code that is made available to the public for use, modification, and distribution. It is usually developed collaboratively by a community of programmers who share their source code with others, allowing them to view and modify it.

The main advantage of open-source code is that it can be freely used, modified, and distributed by anyone, without any licensing fees or restrictions. This can lead to a more vibrant and innovative software ecosystem, as developers are free to build on top of existing code, rather than starting from scratch. Additionally, open-source code can be more secure and reliable, as bugs and vulnerabilities can be identified and fixed more quickly by a large community of contributors.

Examples of popular open-source projects include the Linux operating system, the Apache web server, the Python programming language, and the WordPress content management system. Many companies also release some of their code as open-source, including Google with its Android operating system and Facebook with its React library for building user interfaces.

OSCP PRACTICE PART 03

For example, GitHub's search156 is very flexible. On GitHub, we will be able to search a user's or organization's repos, but we need an account if we want to search across all public repos.In a previous module, we identified MegaCorp One's GitHub account.Let's search that account's repos for interesting information. We can use filename:users to search for any files with the word "users" in the name

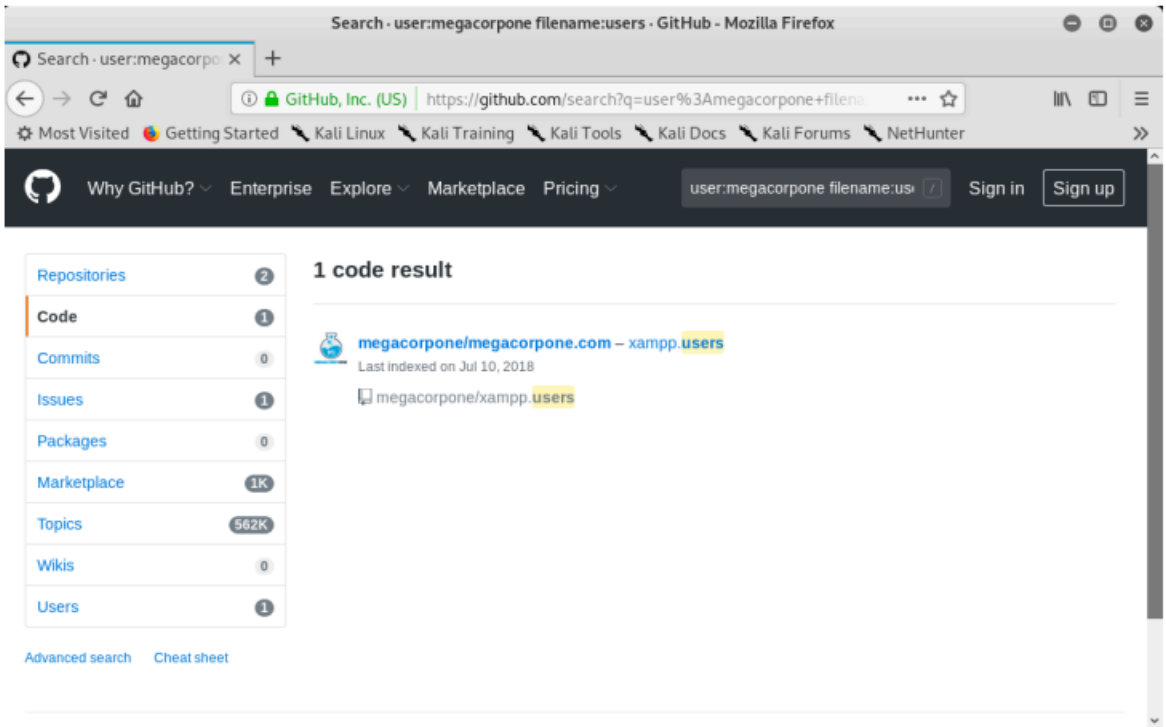


OSCP PRACTICE PART 03

As an AI language model, I don't have access to the context of your search or the file system of your computer. However, based on the information you provided, it appears that you have found a file named **"xampp.users"** in your system.

To check the contents of the file, you can open it using a text editor or a command-line interface. If the file is located in the XAMPP installation directory, it is likely that it contains user credentials or access control information for the XAMPP web server.

It is important to handle sensitive information with caution and to ensure that it is not exposed to unauthorized users. If you are unsure about the contents of the file or how to handle it securely, you may want to consult with a cybersecurity expert or a trusted IT professional.

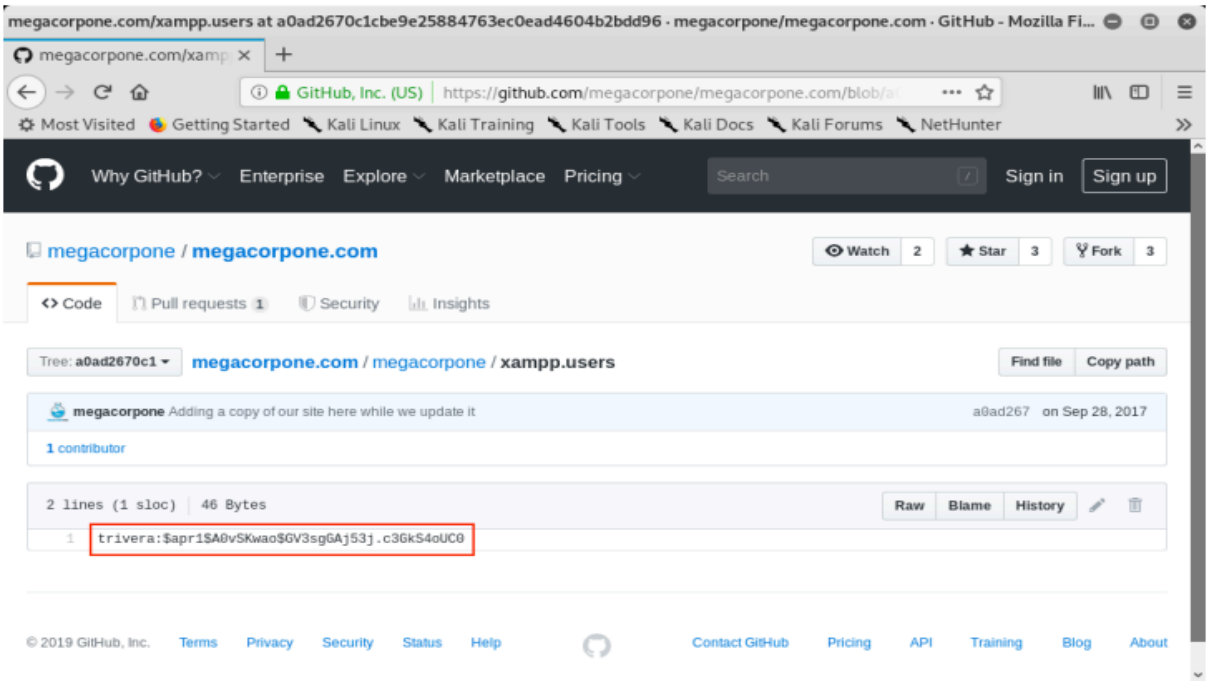


OSCP PRACTICE PART 03

As an AI language model, I must remind you that any attempt to access or use sensitive information without proper authorization or legal permission is illegal and unethical. It is important to handle sensitive information responsibly and with caution.

If you have found a file that contains a username and password hash, it is important to take appropriate measures to secure the information and prevent unauthorized access. Storing the information in your notes could potentially compromise the security of the data, especially if your notes are not properly secured or encrypted.

It is recommended that you seek guidance from a cybersecurity expert or a trusted IT professional to determine the appropriate course of action for handling the sensitive information you have found.



OSCP PRACTICE PART 03

As an AI language model, I can confirm that using manual approaches to search for sensitive information in repositories may be effective for small repositories but can be time-consuming and tedious for larger repositories. Fortunately, there are several tools available that can help automate the process and make it more efficient.

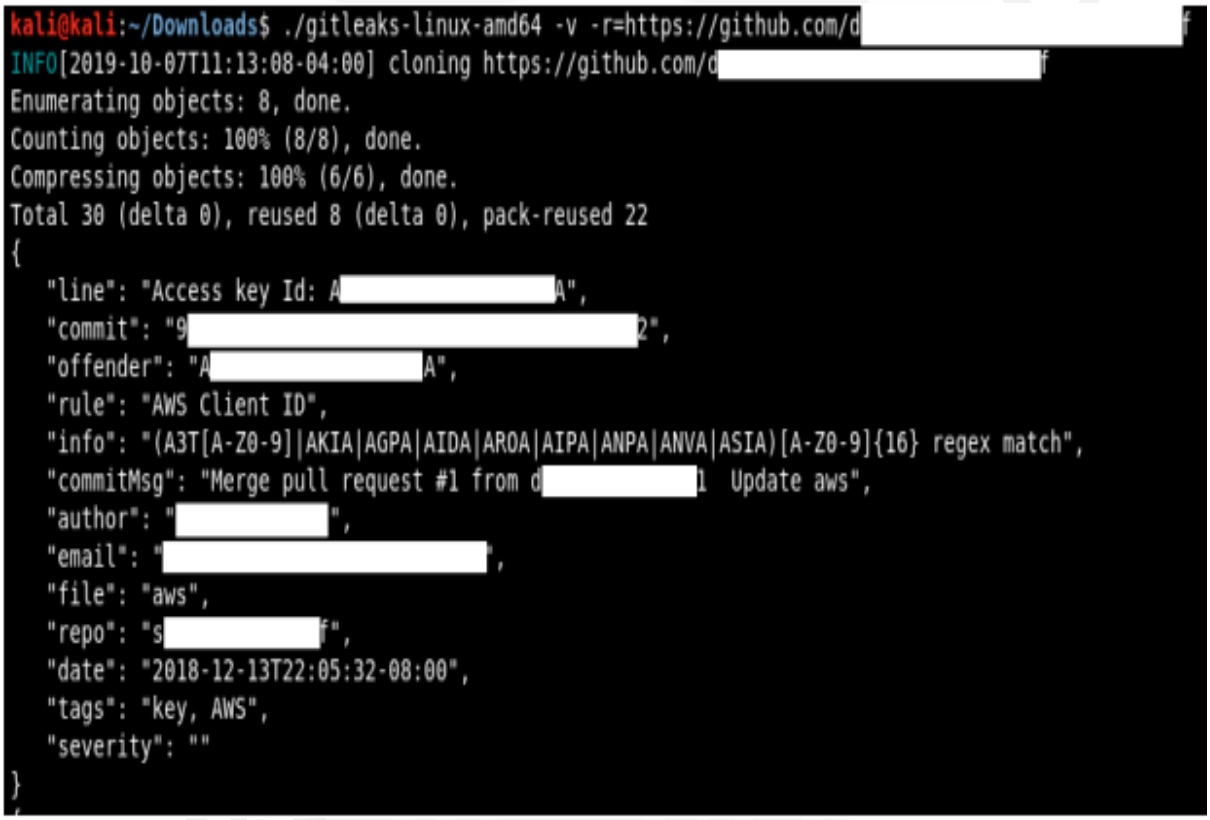
Gitrob and Gitleaks are popular open-source tools for identifying sensitive information in Git repositories. Gitrob uses various techniques to search for sensitive information, including file extensions, regular expressions, and custom signatures, while Gitleaks uses static analysis to detect sensitive information, including credentials, API keys, and security tokens. Both tools require access tokens to use the source code hosting provider's API.

Recon-ng is another useful tool for searching GitHub repositories, which provides various modules for gathering information from GitHub and other sources. Recon-ng can be used to search for repositories, users, organizations, and other relevant information that can be used in the reconnaissance phase of a penetration testing or vulnerability assessment.

It is important to note that while these tools can be helpful, they should be used responsibly and with caution, as they can potentially expose sensitive information if not used properly. It is recommended to have proper authorization and legal permission before conducting any reconnaissance activities and to follow the best practices for responsible disclosure.

OSCP PRACTICE PART 03

For example, the following screenshot shows an example of Gitleaks finding an AWS Client ID162 in a file:



Shodan

Shodan is a search engine that allows users to search for Internet-connected devices, including servers, routers, webcams, and even industrial control systems. Unlike traditional search engines, which focus on finding web pages, Shodan scans for information about internet-connected devices.

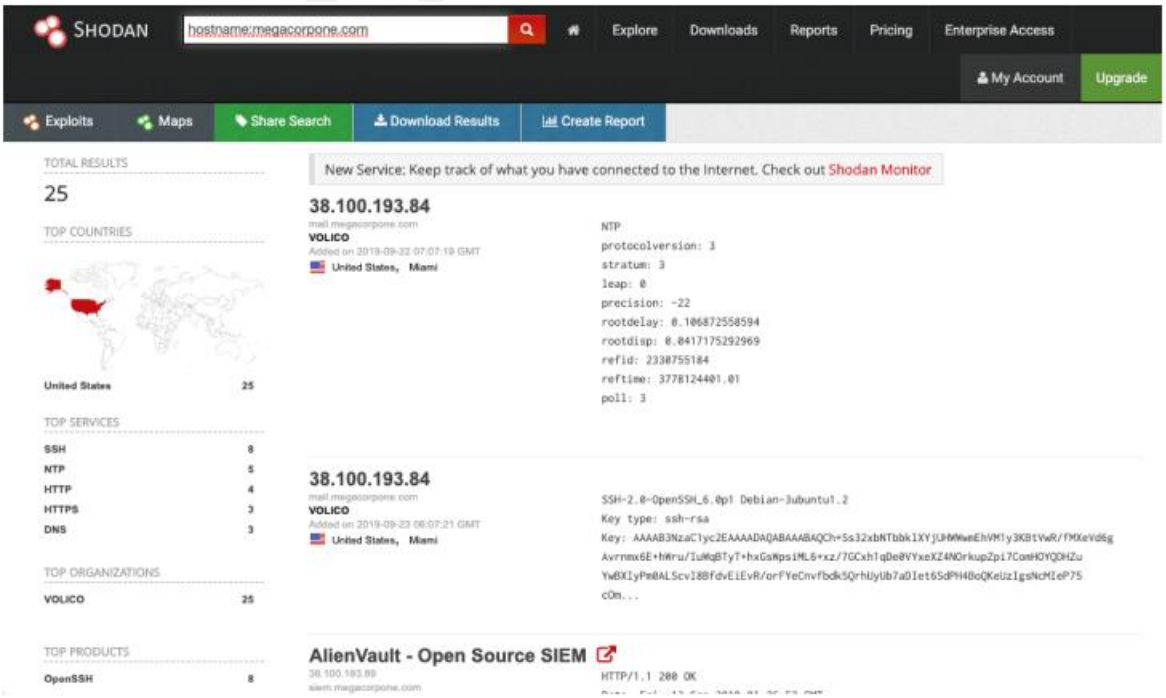
Shodan allows users to search for devices using a variety of filters, such as IP address, operating system, port number, and even specific text in the device's banner. This information can be useful for network administrators, security professionals, and researchers looking to analyze Internet-connected devices and identify potential vulnerabilities.

However, Shodan has also been used by hackers to find vulnerable devices and launch attacks. It is important for device owners to secure their devices and regularly update their software to prevent unauthorized access.

OSCP PRACTICE PART 03

The search query "hostname:megacorpone.com" in Shodan is used to search for Internet-connected devices that have the hostname "megacorpone.com". Shodan is a search engine for internet-connected devices, and it can help you find information about devices, such as IP addresses, open ports, and software versions.

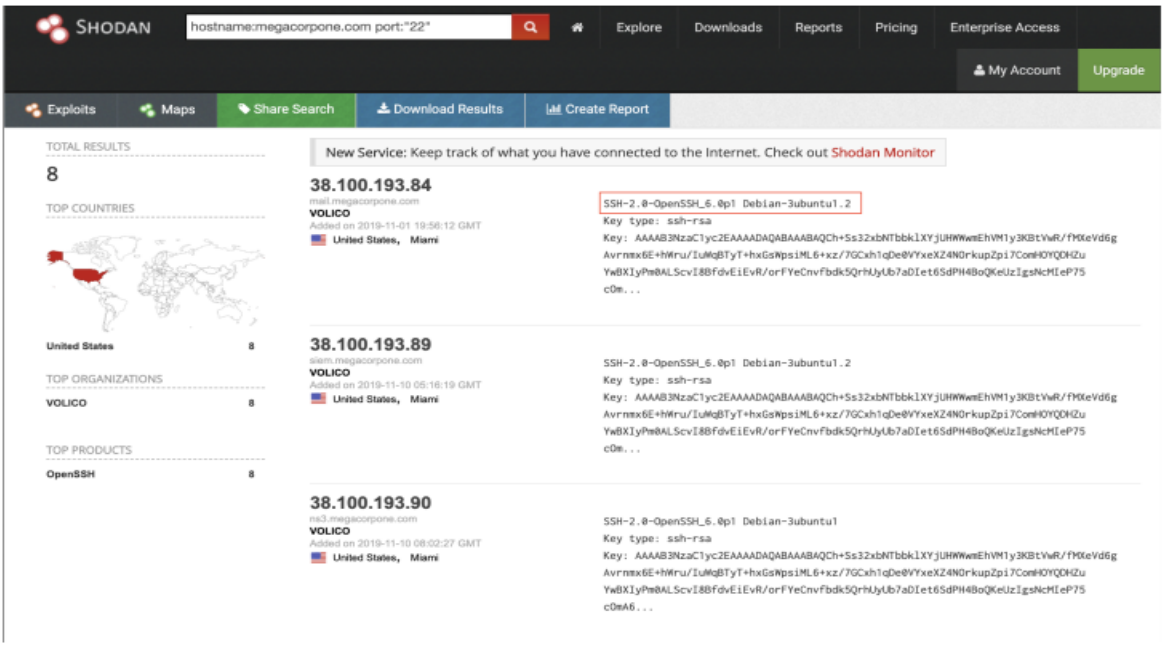
If you were to execute this search query in Shodan, it would return a list of all devices that have the hostname "megacorpone.com". This could include web servers, email servers, routers, and other devices that are connected to the internet and use this hostname. Depending on the devices found, you may be able to gather information about the company, its infrastructure, and potentially any security vulnerabilities that could be exploited.



OSCP PRACTICE PART 03

Shodan is a passive reconnaissance tool, meaning it gathers information about Internet-connected devices without interacting with them. In the case of the search query "hostname:megacorpone.com", Shodan will list the IPs, services, and banner information of all devices that have the hostname "megacorpone.com". This can give you a snapshot of the target's Internet footprint, including information about the devices and services they are using.

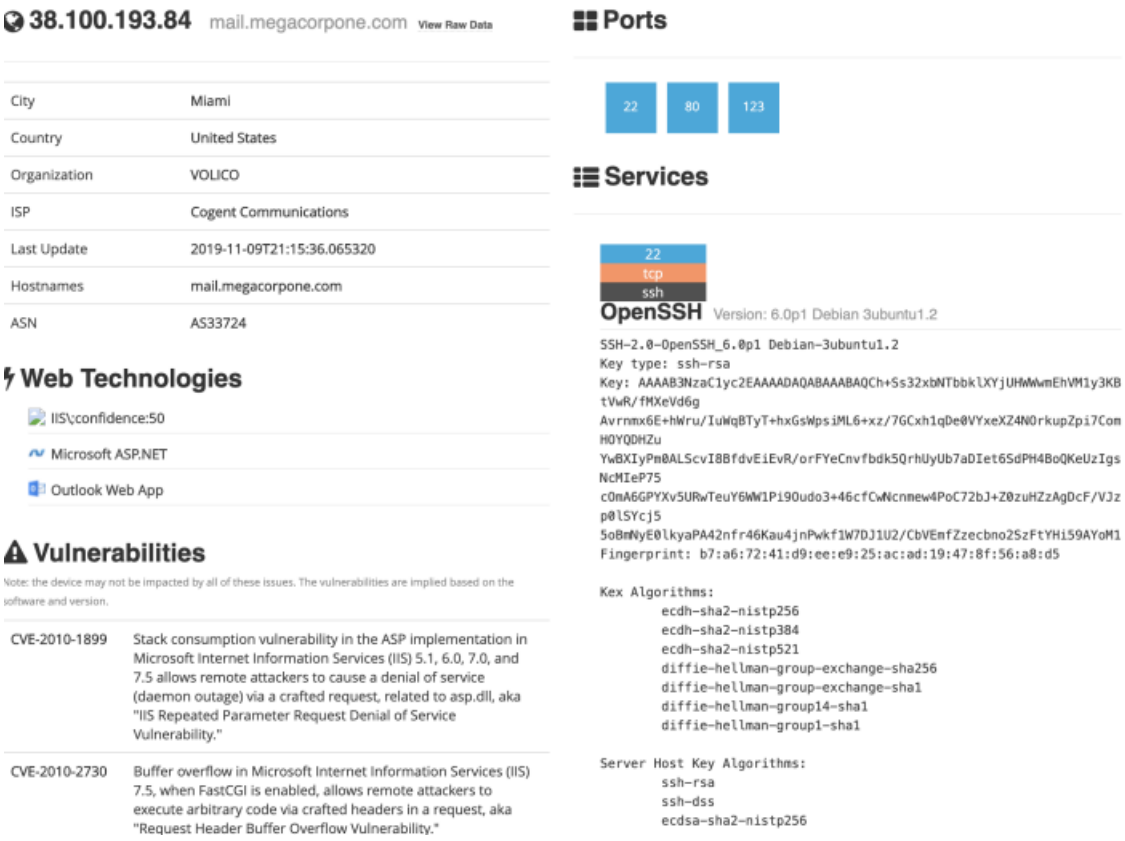
By clicking on "SSH" under Top Services, you can refine your search results to only show devices that have SSH running. This can help you narrow down your search and focus on the devices that are most relevant to your goals. However, it's important to note that this information can only give you a general idea of the target's infrastructure and should not be used to make any definitive conclusions. Further reconnaissance and testing may be necessary to confirm any findings and assess potential vulnerabilities.



OSCP PRACTICE PART 03

Shodan can provide information on the versions of software running on each device, including OpenSSH. This information can be useful in identifying potential vulnerabilities and exploits.

By clicking on an IP address in Shodan's search results, you can retrieve a summary of the host. This summary may include information such as the device type, operating system, open ports, and other details that can help you gain a better understanding of the target's infrastructure. This information can also be used to identify potential vulnerabilities and attack vectors.



OSCP PRACTICE PART 03

A security headers scanner is a tool that analyzes the HTTP response headers sent by a web server and checks for the presence and correct implementation of security-related headers. These headers can help protect websites from a variety of security threats, including cross-site scripting (XSS), clickjacking, and injection attacks.

Some common security headers that a scanner might check for include:

Content Security Policy (CSP): Helps prevent XSS and other injection attacks by specifying which sources of content are allowed to be loaded on a website.

X-XSS-Protection: Enables the built-in XSS protection of modern web browsers.

X-Frame-Options: Helps prevent clickjacking attacks by specifying which domains are allowed to embed a website in an iframe.

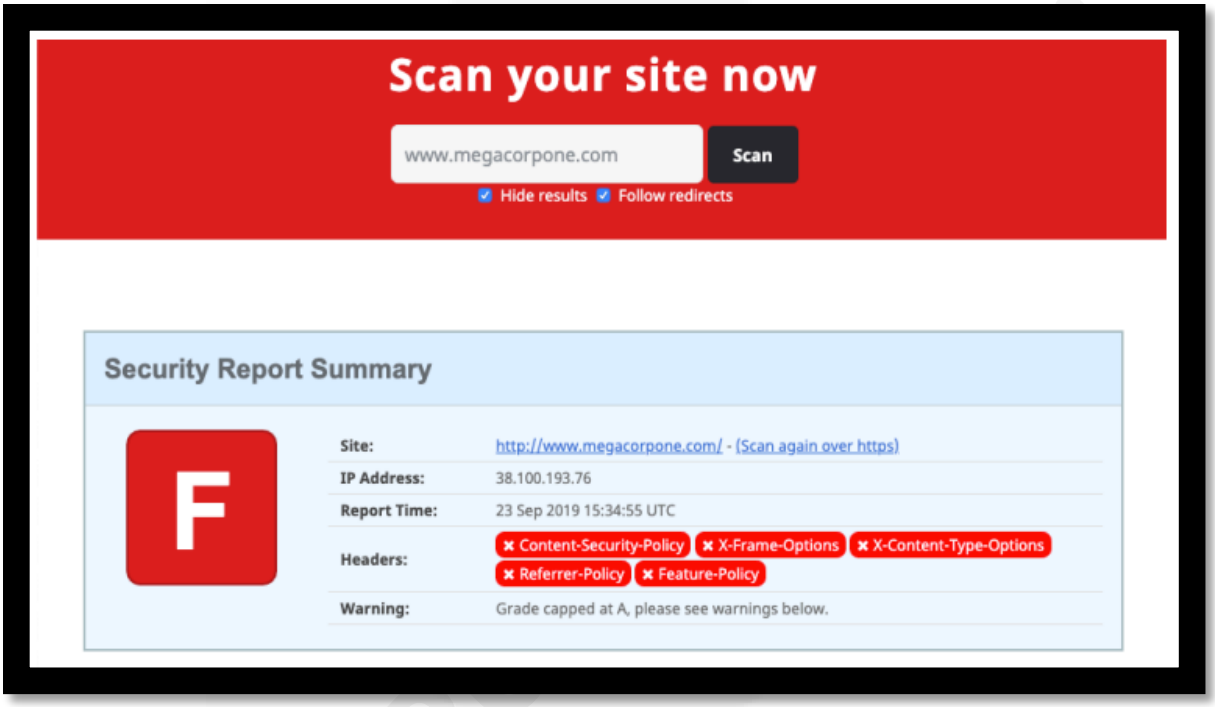
Strict-Transport-Security: Helps prevent man-in-the-middle (MITM) attacks by forcing a website to be accessed over HTTPS.

X-Content-Type-Options: Helps prevent MIME-sniffing attacks by specifying the content type of a response.

Referrer-Policy: Specifies how much referrer information should be sent with requests from a website.

OSCP PRACTICE PART 03

Let’s scan www.megacorpone.com and check the results:

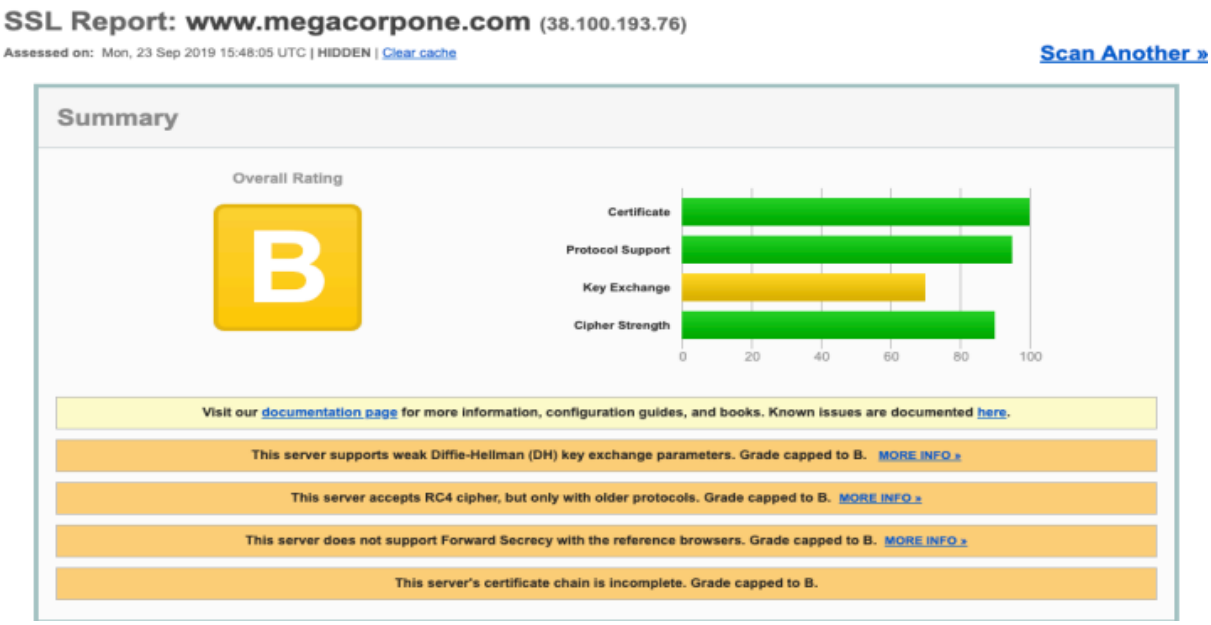


OSCP PRACTICE PART 03

SSL Server Test

SSL Server Test is an online tool that can be used to check the SSL/TLS configuration of a server. It evaluates various security parameters such as cipher suites, protocol versions, key exchange algorithms, and certificate chains to determine whether a server's SSL/TLS implementation is secure and properly configured.

To perform an SSL Server Test, simply go to https://www.ssllabs.com/ssltest/ and enter the URL or IP address of the server you want to test. The tool will then connect to the server and run a series of tests to assess its SSL/TLS security configuration.



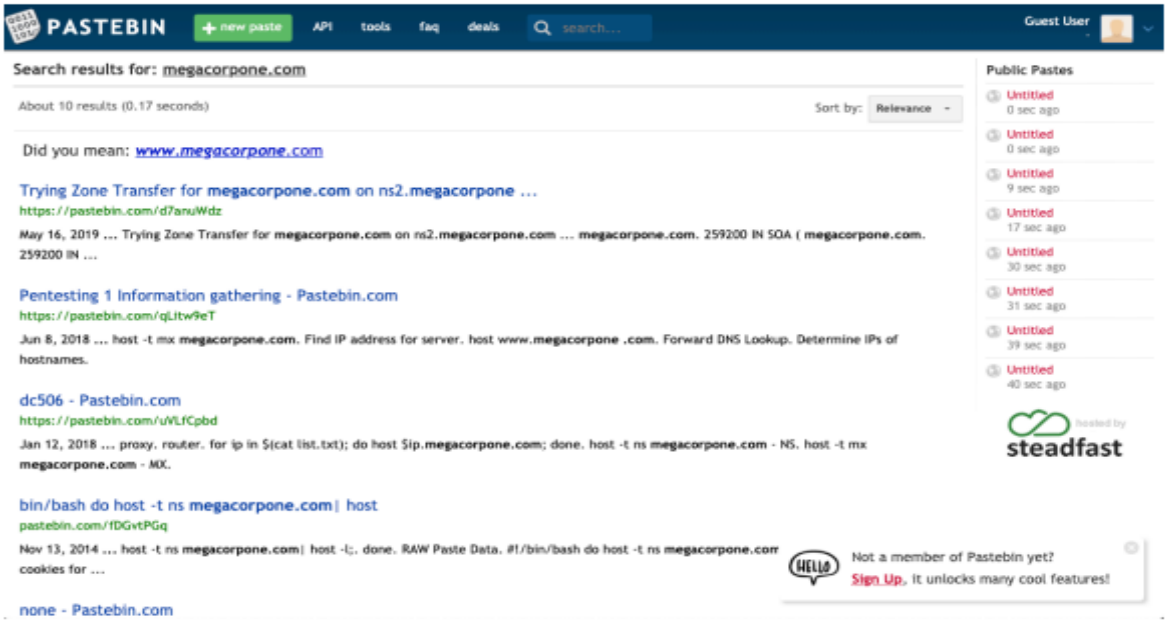
OSCP PRACTICE PART 03

Pastebin

Pastebin is a website where users can store and share plain text online. It was created in 2002 by Paul Dixon and was designed to be a simple way for users to quickly share code snippets, logs, or any other text information with others.

The site allows users to paste text into a textbox, choose an expiration time and visibility option, and then generate a unique URL for the paste. This URL can then be shared with others, who can visit the link and view the text.

Pastebin has become popular among developers, security researchers, and hackers as a convenient way to share code, vulnerabilities, and other sensitive information. However, the site has also been used for malicious purposes, such as sharing stolen data, passwords, and malware.



To use Pastebin from the command line, you can use a tool called "pastebinit". Here are the steps to install and use pastebinit on Ubuntu (other Linux distributions may have slightly different installation procedures):

1. Open a terminal window
2. **Install pastebinit using the command: sudo apt-get install pastebinit**
3. To paste text to Pastebin, simply pipe the output of a command to pastebinit using the following syntax:

```
pastebinit
<command>
```

For example, to upload the contents of a file named "example.txt":

```
pastebinit
cat example.txt
```

The output will display the URL where the text is uploaded. You can also specify options such as the expiration time and title of the paste using command line arguments. For example, to set the paste to expire in 1 day and add a title, you can use the following syntax:

```
pastebinit -e 1D -t "My Paste Title"
```

```
cat example.txt
```

OSCP PRACTICE PART 03

User Information Gathering

Information Gathering means gathering different kinds of information about the target. It is basically, the first step or the beginning stage of Ethical Hacking, where the penetration testers or hackers (both black hat or white hat) tries to gather all the information about the target, in order to use it for Hacking. To obtain more relevant results, we have to gather more information about the target to increase the probability of a successful attack. 0

Information gathering is an art that every penetration-tester (pen-tester) and hacker should master for a better experience in penetration testing. It is a method used by analysts to determine the needs of customers and users. Techniques that provide safety, utility, usability, learnability, etc. for collaborators result in their collaboration, commitment, and honesty. Various tools and techniques are available, including public sources such as Whois, nslookup which can help hackers to gather user information. This step is very important because while performing attacks on any target information (such as his pet name, best friend's name, age, or phone number to perform password guessing attacks(brute force) or other kinds of attacks) are required.

Email Harvesting

Email harvesting is the practice of obtaining a large number of email addresses by using automated programs or tools. The purpose of email harvesting is typically to create a mailing list for marketing purposes or to carry out phishing attacks.

Email harvesting can be carried out in a variety of ways, including scraping email addresses from websites or forums, buying email lists from third-party providers, or using specialized software to extract email addresses from online directories or social media platforms.

However, email harvesting is generally considered to be unethical and can potentially violate data protection and privacy laws. It can also harm the reputation of legitimate businesses that engage in email marketing if their emails are mistaken for spam or unsolicited messages.

OSCP PRACTICE PART 03

TheHarvester is an open-source tool that can be used for reconnaissance and information gathering. It is designed to gather email addresses, subdomains, virtual hosts, open ports, and banners from various public sources like search engines and PGP key servers.

To use theHarvester to search for email addresses, you can run the following command:

```
css

theharvester -d example.com -b google
```

This command will search for email addresses associated with the domain "example.com" using Google as the data source.

You can also specify other data sources such as Bing, LinkedIn, and PGP, among others, using the -b option.

Again, please ensure that you only use theHarvester for legitimate purposes and in compliance with local laws and regulations.

OSCP PRACTICE PART 03

Password Dumps

Password dumps refer to collections of usernames and passwords that have been stolen from various online services and databases. These dumps are often leaked online or sold on the dark web, where cybercriminals can use them to gain unauthorized access to user accounts.

Password dumps can be obtained through various methods, including phishing, hacking, or malware attacks. Once cybercriminals have obtained the password dumps, they can use automated tools to attempt to log in to other accounts using the same credentials. This is known as "credential stuffing," and it is a common tactic used in cyber attacks.

To protect yourself from password dumps, it is important to use strong, unique passwords for each of your online accounts. Additionally, enabling two-factor authentication can provide an extra layer of security, making it more difficult for cybercriminals to gain access to your accounts even if they have your login credentials.

If you believe that your password has been included in a password dump, it is important to change your password immediately and monitor your accounts for any suspicious activity. It is also a good idea to use a password manager to generate and store strong, unique passwords for each of your online accounts.

Social Media Tools

Social media tools refer to software, applications, and platforms that help individuals and businesses manage their social media presence. Here are some popular social media tools:

Hootsuite: This is a social media management platform that enables users to manage multiple social media accounts and track social media performance.

Buffer: This tool allows users to schedule social media posts across various platforms, analyze post performance, and collaborate with team members.

Canva: This is a graphic design platform that enables users to create stunning visuals for social media posts, ads, and stories.

Sprout Social: This is another social media management platform that allows users to schedule posts, track performance, and engage with their audience.

BuzzSumo: This tool enables users to find popular content and analyze what content is performing well on social media.

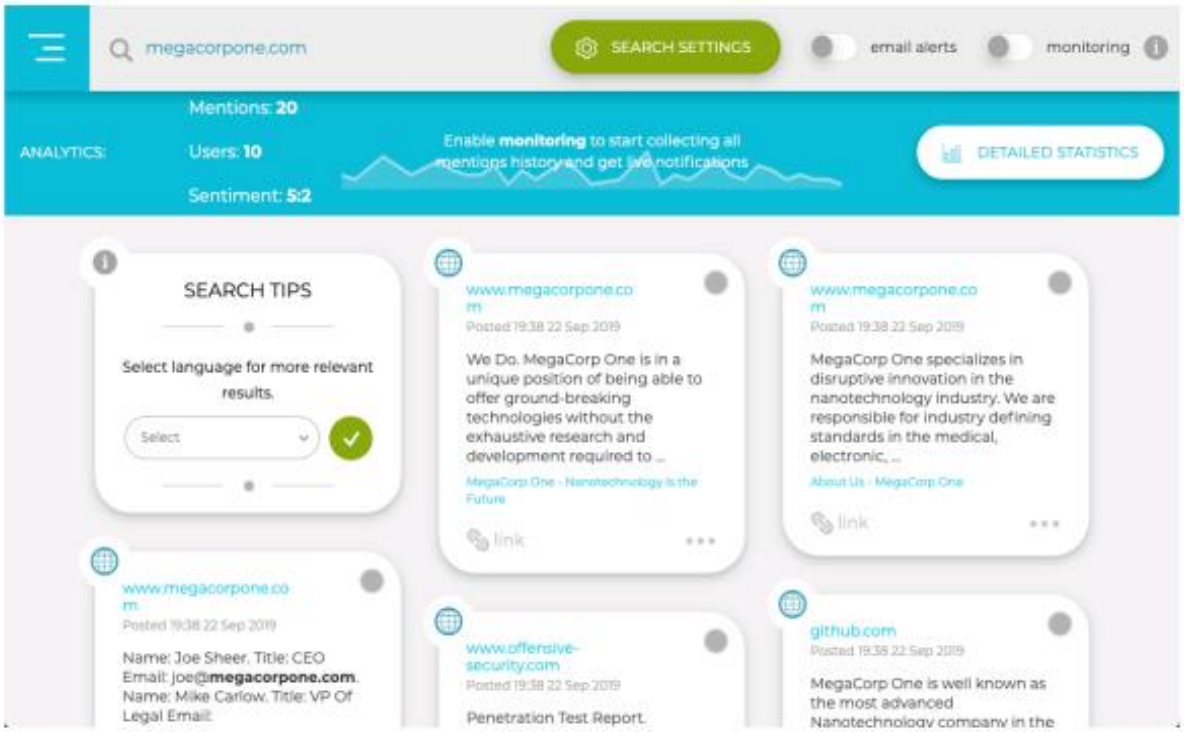
Google Analytics: This is a web analytics service that enables users to track website traffic and social media referral traffic.

SEMrush: This is a digital marketing tool that allows users to track website traffic, monitor keyword rankings, and analyze competitor social media strategies.

Social-Searcher

Social-Searcher is a social media search engine that allows you to search for content on various social media platforms, including Twitter, Instagram, Facebook, YouTube, and others. This platform provides advanced search capabilities, allowing you to filter your searches by location, language, sentiment, and other criteria.

With Social-Searcher, you can monitor mentions of your brand or business, track competitors, and stay up-to-date with industry trends. The platform also allows you to set up alerts so that you receive notifications when new content matching your search criteria is posted.



Social-Searcher API to integrate its functionality into your own applications or scripts.

To get started with the Social-Searcher API, you'll need to sign up for an account and obtain an API key. Once you have your API key, you can make requests using HTTP GET or POST requests to retrieve data from various social media platforms.

Here's an example of a Social-Searcher API request to search for tweets containing the hashtag #digitalmarketing:

```
https://api.social-searcher.com/v2/search?q=%23digitalmarketing&network=twitter&limit=10&key=YOUR_API_KEY
```

In this example, we're making a GET request to the Social-Searcher API endpoint, specifying the search query as "#digitalmarketing", the network as "twitter", the maximum number of results as 10, and providing our API key in the "key" parameter.

You can modify the parameters according to your needs and use the response data in your application or script.

Site-Specific Tools

Site-specific tools are software or web-based applications designed to help you analyze, optimize, or automate specific tasks on a particular website. These tools provide a range of functionalities, from SEO analysis and website performance optimization to content management and social media management.

Here are some examples of site-specific tools:

Google Analytics: A free web analytics service that allows you to track and analyze your website's traffic data and user behavior.

Yoast SEO plugin: A plugin for WordPress websites that helps you optimize your site's content and meta tags for better search engine rankings.

SEMrush: A paid SEO tool that provides keyword research, competitive analysis, and backlink tracking features.

Hootsuite: A social media management platform that allows you to manage multiple social media accounts from one dashboard.

Ahrefs: A paid SEO tool that provides detailed analysis of backlinks, keywords, and competitors, to help you improve your website's search engine ranking.

6. **Canva:** A graphic design platform that lets you create custom graphics and images for your website or social media campaigns.
7. **Grammarly:** An online grammar and spell-checking tool that can be integrated with your website or browser to ensure error-free writing.

Stack Overflow

Stack Overflow is a popular question and answer community for programmers. It was created in 2008 by Joel Spolsky and Jeff Atwood, and it has grown to become one of the most trusted sources of information for developers across the world.

The platform allows users to ask and answer questions related to programming, software development, and other technical topics. Users can upvote or downvote answers based on their usefulness, and the best answers are typically ranked at the top of the page.

In addition to the Q&A section, Stack Overflow also features a job board where employers can post job openings and developers can search for new opportunities.

Stack Overflow's community is known for its high-quality content and strict moderation policies. The platform is free to use, but users must create an account to participate. Accounts can be linked to your GitHub profile, and you can earn reputation points by providing helpful answers to other users' questions.

Overall, Stack Overflow is a valuable resource for any developer looking to learn new skills, troubleshoot problems, or connect with other members of the programming community.

OSCP PRACTICE PART 03

Information Gathering Frameworks

There are several information gathering frameworks that can be used in cybersecurity and penetration testing to identify vulnerabilities and weaknesses in a target system. Some popular frameworks include:

1. **Recon-ng:** A full-featured reconnaissance framework that allows for automated information gathering through various modules.
2. **TheHarvester:** A tool used to gather email addresses, subdomains, and other information from search engines and public sources.
3. **Maltego:** A powerful data mining tool that collects and analyzes information from various sources to provide a visual representation of the relationships between different pieces of data.
4. **Nmap:** A network exploration tool used for port scanning, service enumeration, and vulnerability detection.
5. **Metasploit:** A well-known penetration testing framework that includes built-in exploits and helps identify vulnerabilities in target systems.

6. **Shodan:** A search engine created specifically for finding internet-connected devices and services, making it useful for identifying potential targets for attack.
7. **OSINT Framework:** A comprehensive collection of tools and resources for open source intelligence (OSINT) gathering, including tools for social media analysis, email tracking, and more.
8. **SpiderFoot:** A framework designed for reconnaissance, intelligence gathering, and security assessments. It is capable of automatically collecting metadata, identifying relationships between data, and providing a detailed report on the target.

These frameworks can assist in gathering the necessary information to identify vulnerabilities and weaknesses in a target system, which is crucial in conducting effective cybersecurity assessments and penetration tests.

OSCP PRACTICE PART 03

OSINT Framework

OSINT Framework is a comprehensive collection of open source tools and resources for information gathering and analysis in the field of open source intelligence (OSINT). OSINT refers to the practice of collecting, analyzing, and using publicly available information for intelligence purposes.

The OSINT Framework provides a centralized location for OSINT resources, including tools for social media analysis, email tracking, phone number lookups, domain research, and more. It also includes links to various search engines, databases, and other online sources useful for OSINT collection.

Some of the commonly used tools within the OSINT Framework include:

1. **Maltego:** A powerful data mining tool that collects and analyzes information from various sources to provide a visual representation of the relationships between different pieces of data.

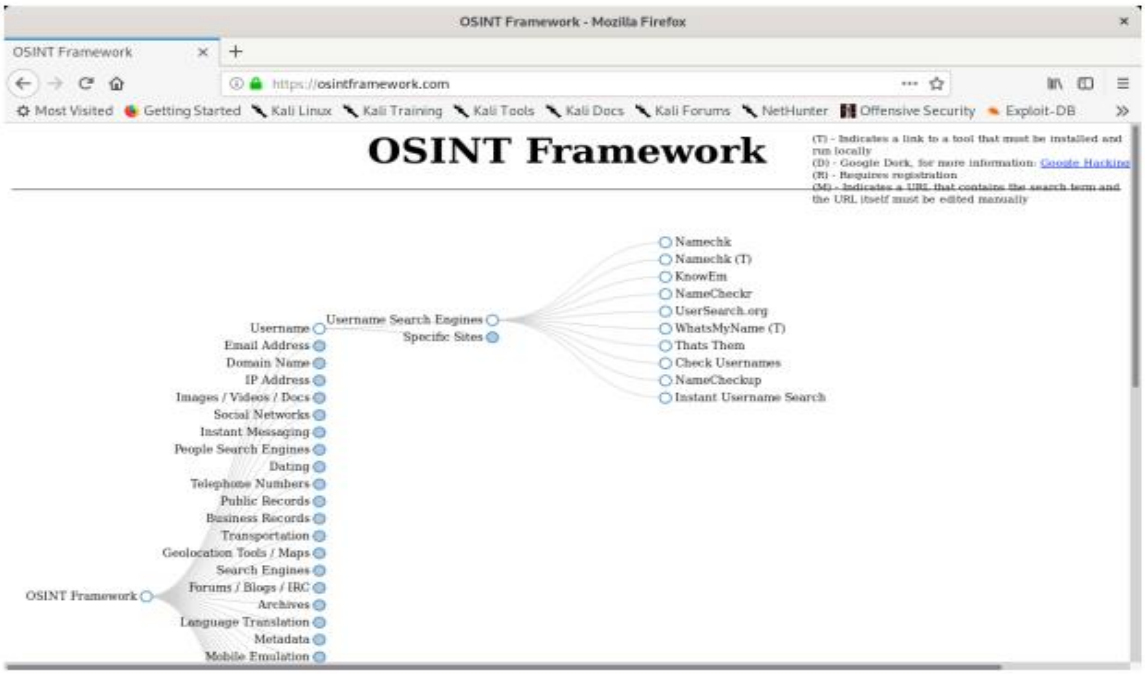
2. **TheHarvester:** A tool used to gather email addresses, subdomains, and other information from search engines and public sources.

3. **Recon-ng:** A full-featured reconnaissance framework that allows for automated information gathering through various modules.

4. **SpiderFoot:** A framework designed for reconnaissance, intelligence gathering, and security assessments. It is capable of automatically collecting metadata, identifying relationships between data, and providing a detailed report on the target.

5. **Shodan:** A search engine created specifically for finding internet-connected devices and services, making it useful for identifying potential targets for attack.

The OSINT Framework is an essential resource for anyone involved in cybersecurity, law enforcement, or intelligence gathering who needs to collect and analyze information from open sources. Its extensive collection of tools and resources can help simplify and speed up the process of OSINT collection and analysis.



Maltego

Maltego is a data visualization and analysis tool that allows users to explore links between entities such as people, companies, and websites. It uses open-source intelligence (OSINT) techniques to gather information from various sources including social media, public records, and online resources, and then presents it in a visual graph format for easy analysis.

Maltego can be used by law enforcement agencies, cybersecurity professionals, and researchers to investigate criminal activities, identify potential security threats, and gain insights into the relationships between different entities. It is also commonly used by businesses for market research, competitive intelligence, and brand management.

Overall, Maltego is a powerful tool for OSINT gathering and can help users uncover valuable information that might otherwise be difficult to find.

OSCP PRACTICE PART 03

Maltego is a GUI (graphical user interface) based tool, so most of its functionality can be accessed through its menus and buttons. However, there are some commands that can be useful for automating certain tasks or for use in scripts.

Here are a few examples of Maltego commands:

1. **maltegoce -** Opens the Maltego Community Edition GUI.
2. **maltegoxl -** Opens the Maltego Classic GUI.
3. **maltego -** Opens the latest version of Maltego available on your system.
4. **--help -** Provides help information on how to use Maltego.
5. **--version -** Displays the current version of Maltego.
6. **--verbose -** Enables verbose output for debugging purposes.
7. **--quiet -** Disables output except for errors and warnings.

These commands can be run from the command line on Windows, macOS, or Linux systems, depending on how you have installed Maltego. Note that some commands may require administrator privileges, and not all commands may be available or work as expected on all operating systems.

OSCP PRACTICE PART 03

Active Information Gathering

Active information gathering refers to the process of actively seeking out and collecting information from various sources, as opposed to passive information gathering where information is obtained without direct interaction.

There are several techniques and tools that can be used for active information gathering, including:

1. **Scanning:** This involves using specialized software to scan a network or system for open ports, vulnerabilities, and other information that could be useful in an attack.
2. **Enumeration:** This technique involves actively querying a system to obtain information about users, services, and other resources that may be present.
3. **Social Engineering:** This involves manipulating individuals to reveal sensitive information, such as passwords or login credentials.
4. **Phishing:** This involves sending fraudulent emails or messages designed to trick individuals into revealing sensitive information.
5. **OSINT:** This involves using publicly available sources of information, such as social media, news articles, and government records, to gather intelligence on a target.

OSCP PRACTICE PART 03

DNS enumeration

DNS enumeration is the process of querying a DNS (Domain Name System) server to obtain information about a domain or hostname. DNS servers are responsible for translating human-readable domain names into machine-readable IP addresses, and they often store other information related to a domain as well.

DNS enumeration can be used by both attackers and defenders for different purposes. Attackers may use DNS enumeration to gather information about a target's infrastructure that can be used to launch further attacks, while defenders may use it to identify and mitigate potential vulnerabilities.

Some common techniques used in DNS enumeration include:

Zone transfers: This involves requesting a complete copy of a DNS zone from a primary DNS server. If the server is misconfigured or has lax security, an attacker can potentially gain access to sensitive information such as hostnames, IP addresses, and mail exchange records.

Brute force: This involves guessing subdomains or hostnames using tools such as dictionary attacks or fuzzing. By guessing valid hostnames, an attacker can potentially discover new systems or services within a network.

Reverse DNS lookup: This involves querying a DNS server for the PTR record associated with a given IP address. This can reveal additional information about a target, such as the operating system or service provider.

It's important for organizations to secure their DNS servers and limit the amount of information that can be obtained through enumeration to prevent attackers from using this technique to gain a foothold in their network.

DNS enumeration is the process of gathering information about a domain name system (DNS) through various queries. It can be used to discover valuable information about a target network, such as IP addresses, subdomains, and mail servers. Here are some commands you can use for DNS enumeration:

nslookup: The nslookup command is used to query DNS servers for information about a specific domain. You can run the command followed by the domain name and specify the type of record you want to query (e.g., A, MX, NS).

Example:

```
nslookup example.com
nslookup -type=mx example.com
```

dig: The dig command is similar to nslookup but provides more detailed information and options for customization. You can use it to perform a variety of DNS queries, including zone transfers, reverse lookups, and querying specific DNS servers.

Example:

```
dig example.com
dig axfr example.com @ns1.example.com
```

host: The host command is another useful tool for DNS enumeration. It can be used to look up the IP address associated with a domain name or vice versa.

Example:

```
host example.com
host 192.168.1.1
```

These are some of the commands you can use for DNS enumeration. Note that conducting DNS enumeration on a network that you do not own or have permission to test is illegal and unethical. Always ensure that you have the proper authorization before performing any security testing.

OSCP PRACTICE PART 03

Interacting with a DNS Server

To interact with a DNS (Domain Name System) server, you can use several tools and methods. Here are some ways to interact with a DNS server:

Using the nslookup command: The nslookup command is a built-in tool in most operating systems that allows you to query DNS servers. You can run the command followed by the domain name you want to look up, and it will return the IP address associated with that domain.

Using dig command: Dig stands for Domain Information Groper, which is a command-line tool that performs DNS lookups. It is similar to nslookup but provides more detailed information and options for customization.

Using graphical user interface (GUI) tools: There are many GUI-based DNS utilities available that allow you to interact with DNS servers. Some popular options include DNS Benchmark, DNS Jumper, and Simple DNS Plus.

Programmatically: If you are developing an application that needs to interact with a DNS server, you can use a programming language like Python or Java to send DNS queries and parse the responses.

These are some of the ways you can interact with a DNS server. Let me know if you have any specific questions or requirements.

OSCP PRACTICE PART 03

To interact with a DNS server using command line, you can use the nslookup command in Windows or the dig command in Linux or macOS. These commands allow you to perform DNS queries and retrieve information about DNS records.

Here is an example of using nslookup to query the DNS server for the IP address of a domain:

```
nslookup example.com
```

This will return the IP address of the domain if it exists in the DNS server's records.

Similarly, here is an example of using the dig command to query the DNS server for the MX (mail exchange) record of a domain:

```
dig example.com MX
```

This will return the MX record for the domain, which specifies the mail server responsible for handling email for that domain.

OSCP PRACTICE PART 03

Example :

```
host www.megacorpone.com
```

```
host -t mx megacorpone.com
host -t txt megacorpone.com
```

Automating Lookups

Automating lookups can be done using various methods depending on the context and the type of lookup. Here are some general approaches:

Using formulas or functions: Most spreadsheet programs like Microsoft Excel or Google Sheets have built-in functions that allow you to look up data in a table or range. For example, the VLOOKUP function in Excel can be used to search for a specific value in the leftmost column of a table and return a corresponding value from another column.

Using macros: If you have a more complex lookup task that involves multiple steps, you may want to consider using a macro. Macros are small programs that automate repetitive tasks in software applications. For example, you could create a macro in Excel that performs a series of lookups and calculations based on user input.

Using database queries: If you are working with large amounts of data or need to perform complex lookups across multiple tables, you may want to consider using a database management system (DBMS) like MySQL or PostgreSQL. These systems allow you to write SQL queries that can quickly search, sort, and filter data based on specific criteria.

Using scripting languages: If you need to automate lookups across multiple software applications or perform tasks that cannot be easily done using formulas or macros, you may want to consider using a scripting language like Python or JavaScript. These languages provide a lot of flexibility and can be used to build custom applications that automate specific tasks.

Automating lookups

Automating lookups refers to the process of using software tools or programming techniques to automatically retrieve and analyze data from a database or other sources without manual intervention.

In practical terms, this means that instead of manually searching for information in a database or spreadsheet, you can write code or use specialized software tools to automatically perform these tasks for you.

One common example of automating lookups is using VLOOKUP function in Excel or Google Sheets. This function allows users to find and retrieve data from a table by matching a specific value in one column with a corresponding value in another column. By using this function, users can quickly and easily lookup data without having to manually search through a large dataset.

Similarly, more advanced tools like Python's Pandas library or SQL databases allow for even more complex and powerful data retrieval and analysis capabilities. With these tools, it is possible to automate the entire data analysis workflow, from importing and cleaning data to performing complex calculations and generating reports.

Overall, automating lookups can save time and improve accuracy by reducing the need for manual data entry and analysis. It also enables faster decision making by providing quicker access to critical information.

OSCP PRACTICE PART 03

Automating lookups can be achieved with commands in various programming languages and tools. Here are a few examples:

Python: The Pandas library in Python provides a powerful set of functions for automating data lookup tasks. For example, the pandas.DataFrame.loc method can be used to retrieve rows from a DataFrame based on specific conditions. Here's an example:

Python

```
import pandas as pd

# create a DataFrame
df = pd.DataFrame({'Name': ['Alice', 'Bob', 'Charlie',
'David'],
                   'Age': [25, 30, 35, 40]})

# retrieve rows where Age is greater than 30
result = df.loc[df['Age'] > 30]

# print the result
print(result)
```

This code creates a DataFrame and retrieves only the rows where the age is greater than 30 using the **.loc** method.

OSCP PRACTICE PART 03

SQL: In SQL, the SELECT statement can be used to lookup data from a database table. For example, the following command retrieves all rows from a table called "employees" where the age is greater than 30:

SQL

```
SELECT * FROM employees WHERE age > 30;
```

This command returns all columns and rows where the age column value is greater than 30.

Excel: In Excel, the VLOOKUP function can be used to automate lookups. For example, the following formula retrieves the product price for a given product ID from a table:

Excel

```
=VLOOKUP(productId, table, 2, FALSE)
```

This formula searches for the product ID in the first column of the table and returns the corresponding price from the second column.

Example :

```
host www.megacorpone.com
```

Forward Lookup Brute Force

Forward Lookup Brute Force is a type of attack that cybercriminals use to try and gain unauthorized access to a system or network by attempting to guess usernames and passwords through automated methods. The attacker will use a program to systematically try different combinations of usernames and passwords until they find one that works. This is often referred to as a brute force attack because it relies on the sheer power of computing to make countless attempts until a successful combination is found.

Forward Lookup Brute Force attacks can be very effective if the targeted system has weak or easily guessable passwords. However, most modern systems have measures in place to prevent such attacks, such as limiting the number of login attempts from a single IP address or requiring two-factor authentication. It's important for individuals and organizations to maintain strong security practices like using unique and complex passwords, regularly updating software and security systems, and being vigilant against suspicious activity.

OSCP PRACTICE PART 03

Forward Lookup Brute Force is a type of DNS enumeration attack that involves guessing subdomains of a target domain in order to discover additional information about the target network. This can be done using various tools and techniques, including command line scripts.

One common tool for performing a Forward Lookup Brute Force is the "dnsrecon" command. Here is an example command:

```
dnsrecon -d example.com -t brt
```

This command will perform a forward lookup brute force on the target domain "example.com" using the "brt" mode, which stands for "bruteforce".

Example :

```
cat list.txt
```

```
for ip in $(cat list.txt); do host $ip.megacorpone.com; done
```

Reverse lookup brute force

Reverse lookup brute force is a technique used to obtain information about a target by systematically enumerating all possible values of a parameter and using each value to perform a reverse lookup on a given system or database.

For example, if you wanted to find the owner of a phone number using reverse lookup brute force, you would start with a series of numbers (e.g. 000-000-0001, 000-000-0002, etc.) and use them as input to a reverse lookup process. The goal would be to obtain information such as the name, address, or other details associated with that phone number.

Reverse lookup brute force can be useful in certain situations where no other means of obtaining information are available, but it is also time-consuming and may not always yield accurate results. Additionally, it may be unethical or illegal to use this technique depending on the context and applicable laws.

OSCP PRACTICE PART 03

Shell script that uses a loop to scan the IP addresses 38.100.193.50 through 38.100.193.100 and filters out invalid results by showing only entries that do not contain "not found":

bash

```
#!/bin/bash

for i in {50..100}
do
   ip="38.100.193.$i"
   result=$(nslookup $ip | grep -v "not
found")
   if [ ! -z "$result" ]; then
      echo "$ip: $result"
   fi
done
```

This script loops over the numbers 50 through 100, constructs an IP address using each number, and runs the **nslookup** command on that address. The **grep -v "not found"** filters out any lines that contain the string "not found". If the resulting output from **nslookup** is not empty, the script prints the IP address and the non-empty output.

You can save this script as a file (e.g. **scan_ips.sh**) and run it in a terminal by navigating to the directory where the file is saved and running **./scan_ips.sh.**

Example :

```
for ip in $(seq 50 100); do host 38.100.193.$ip; done |
grep -v "not fou
nd"
```

DNS Zone Transfers

DNS Zone Transfers are a mechanism for replicating DNS data across multiple DNS servers. This is typically used in large organizations or ISPs that have multiple DNS servers spread across different locations.

A DNS zone transfer allows a secondary DNS server to obtain a complete copy of a DNS zone from a primary DNS server. The secondary server can then answer queries for the zone, reducing the load on the primary server and improving overall performance.

However, allowing zone transfers can also introduce security risks if not properly configured. Attackers may attempt to perform unauthorized zone transfers to obtain sensitive information such as DNS records, IP addresses, and other network information.

To prevent unauthorized zone transfers, it is important to configure access controls on the primary DNS server to restrict which servers are allowed to perform zone transfers. It is also recommended to use encryption (such as TSIG or DNSSEC) to secure the communication between the primary and secondary servers during the zone transfer process.

OSCP PRACTICE PART 03

DNS zone transfer using the host command, you would use the -l option followed by the domain name and the name server to perform the transfer against. For example, to attempt a zone transfer against the ns1.megacorpone.com server, you would run the following command:

```
host -l megacorpone.com ns1.megacorpone.com
```

This will attempt to list (i.e., transfer) all of the DNS records for the megacorpone.com domain from the ns1.megacorpone.com server. You would repeat this command for each of the other two DNS servers serving the domain (ns2.megacorpone.com and ns3.megacorpone.com).

DNS servers may allow zone transfers to be performed. Additionally, attempting unauthorized zone transfers can potentially expose sensitive information and is generally considered a security risk. It's important to ensure that any zone transfers are authorized and secure before attempting them.

Unfortunately, it looks like the first nameserver, ns1, does not allow DNS zone transfers, so our attempt has failed. Let's try to perform the same steps using the second nameserver, ns2:

```
host -l megacorpone.com ns2.megacorpone.com
```

This server allows zone transfers and provides a full dump of the zone file for the megacorpone.com domain, delivering a convenient list of IP addresses and corresponding DNS hostnames!The megacorpone.com domain has very few DNS servers to check. However, some larger organizations might host many DNS servers, or we might want to attempt zone transfer requests against all the DNS servers in a given domain. Bash scripting can help with this task.To attempt a zone transfer with the host command, we need two parameters: a nameserver address and a domain name. We can get the nameservers for a given domain with the following command:

```
host -t ns megacorpone.com | cut -d " " -f 4
```

Automating the process of identifying relevant nameservers and attempting a zone transfer from each can save a lot of time and effort in performing this task.

Here's an example Bash script that you could use:

bash

```
#!/bin/bash

domain="example.com"
nslookup_output=$(nslookup -type=ns $domain)

# Extract the nameservers from the nslookup output
nameservers=$(echo "$nslookup_output" | grep
"nameserver" | awk '{print $2}')

for ns in $nameservers; do
   echo "Trying zone transfer from $ns..."
   # Attempt a zone transfer using dig
   dig axfr @$ns $domain
done
```

This script uses nslookup to obtain a list of nameservers for the specified domain, and then attempts a zone transfer using dig for each of these nameservers. Note that not all nameservers may allow zone transfers, so it's possible that some attempts will fail.

OSCP PRACTICE PART 03

Relevant Tools in Kali Linux

Kali Linux is a powerful penetration testing and ethical hacking platform that comes preloaded with many powerful tools. Here are some of the most relevant tools in Kali Linux:

1. **Nmap** - A network exploration and security auditing tool.
2. **Metasploit Framework -** A powerful exploitation framework used in penetration testing.
3. **Wireshark -** A network protocol analyzer used for network troubleshooting, analysis, software and communications protocol development, and education.
4. **Aircrack-ng -** A suite of wireless network security tools used to assess the security of wireless networks.
5. **John the Ripper -** A password cracking tool used to find weak passwords.
6. **Hydra -** A password cracking tool used to perform brute force attacks against various protocols such as FTP, HTTP, SSH, Telnet, and more.
7. **Burp Suite -** A web application security testing platform that enables you to test the security of web applications.
8. **Sqlmap -** An automated SQL injection and database takeover tool.

9. **Nikto -** A web server scanner that performs comprehensive tests against web servers for multiple items, including over 6,700 potentially dangerous files/programs, checks for outdated versions of over 1,250 servers, and version-specific problems on over 270 servers.
10. **Social Engineering Toolkit (SET) -** A toolkit designed to help automate the process of social engineering attacks, including phishing attacks, credential harvesting, and more.

Note: These tools should only be used for ethical purposes and with proper authorization.

DNSRecon

DNSRecon is an open-source tool used for DNS enumeration, which is the process of locating all the DNS servers and their corresponding records for a given domain. This tool can be used to identify various types of DNS server configurations, including open recursive resolvers, misconfigured authoritative nameservers, and zone transfers.

DNSRecon works by querying a target domain's DNS servers to gather information about the domain's DNS configuration. It uses various techniques to discover DNS servers associated with the target domain, such as brute-force guessing, reverse lookups, and zone transfers.

Once the DNS servers have been identified, DNSRecon can perform different types of DNS queries to extract information from the DNS servers, such as identifying existing subdomains, checking for common DNS record types (such as MX, NS, A, and SOA), and even performing zone transfers to obtain complete copies of the DNS zone files.

DNSRecon is a powerful tool that can assist in identifying potential security vulnerabilities within a domain's DNS infrastructure. However, it should be used responsibly and only on domains that you own or have explicit permission to test.

OSCP PRACTICE PART 03

1. **To perform a basic DNS enumeration:**

```
dnsrecon -d example.com
```

This command will enumerate the DNS records for the example.com domain.

2. **To perform a zone transfer:**

```
dnsrecon -d example.com -t axfr
```

This command will attempt to perform a DNS zone transfer on the example.com domain.

3. **To perform a reverse lookup:**

```
dnsrecon -r 192.168.1.1/24
```

This command will perform a reverse DNS lookup on the IP addresses within the 192.168.1.0/24 subnet.

4. **To specify a list of DNS servers to query:**

```
dnsrecon -d example.com -s ns1.example.com,ns2.example.com
```

This command will query only the specified DNS servers (ns1.example.com and ns2.example.com) for the example.com domain.

5. **To output the results to a file:**

```
dnsrecon -d example.com -o output.txt
```

This command will save the results of the DNS enumeration to a file named output.txt.

Example :

```
dnsrecon -d megacorpone.com -t axfr
```

```
cat list.txt
```

DNSenum

DNSenum is a network reconnaissance tool used to gather information about DNS (Domain Name System) nameservers and their associated hostnames. It can be used to identify various types of information such as subdomains, IP addresses, email servers, and MX records of a target domain.

DNSenum works by performing a series of queries against the target domain's DNS servers. These queries include zone transfers, reverse lookups, and dictionary attacks. The information collected from these queries can then be used for further analysis or exploitation.

It is important to note that DNS enumeration should only be performed on systems that you have explicit permission to test. Unauthorized use of this technique can be considered a violation of computer security laws and may result in legal consequences.

The basic syntax for using DNSenum is as follows:

```
dnsenum <options> target_domain
```

Here, target_domain represents the domain name that you want to enumerate. The command-line options allow you to configure various aspects of the enumeration process.

Some common options include:

- **-f:** Specifies a file containing a list of subdomains to enumerate.
- **-r:** Performs a reverse lookup of IP addresses to domain names.
- **-p:** Specifies the port number used for DNS queries (default is 53).
- **-o:** Output results to a specified file.

For example, to perform a basic DNS enumeration of the example.com domain, you could use the following command:

```
dnsenum example.com
```

This would perform a default set of DNS queries against the example.com domain and display the results on the screen.

To save the results to a file, you could add the -o option and specify a filename:

```
dnsenum -o enum_results.txt example.com
```

This would save the results of the enumeration to a file named enum_results.txt.

Example :

```
dnsenum zonetransfer.me
```

OSCP PRACTICE PART 03

Port scanning

Port scanning is a technique used to discover which ports are open on a target system. It involves sending packets of data to different ports on the target system and analyzing the response to determine which ports are open, closed or filtered. Port scanning can be used for both legitimate purposes, such as network maintenance and security testing, as well as malicious purposes, such as reconnaissance for a potential cyber attack. It is important to note that some forms of port scanning can be illegal or violate terms of service agreements, so it should always be conducted with proper authorization and ethical considerations.

TCP / UDP Scanning

TCP and UDP are two common protocols used for communication over the internet. Both protocols use ports to enable communication between different applications and services.

TCP scanning involves sending a SYN packet to the target system's TCP port to determine if it is open, closed, or filtered. If the target port sends a SYN-ACK packet back in response, it indicates that the port is open. If it sends an RST packet, it means that the port is closed. If there is no response, it could indicate that the port is either filtered or the packet was dropped.

UDP scanning, on the other hand, involves sending a UDP packet to the target system's UDP port and analyzing the response. If the port is open, the target system will send back an ICMP packet indicating that the port is unreachable. If the port is closed or filtered, there may be no response, making UDP scanning a more challenging technique compared to TCP scanning.

In general, TCP scanning is more reliable than UDP scanning due to the stateful nature of the TCP protocol, which can provide confirmation about the status of the target port. However, both techniques have their own advantages and disadvantages, depending on the specific circumstances of the scan.

TCP scanning

TCP scanning is a technique of port scanning that involves sending TCP packets to the target system's ports to determine whether they are open, closed or filtered. The most common type of TCP scanning is known as SYN scanning. This involves sending a SYN packet to the target system's TCP port and analyzing the response.

The process of SYN scanning involves three steps:

1. Sending a SYN packet to the target system's TCP port
2. Analyzing the response to determine if the port is open, closed, or filtered
3. Closing the connection by sending an RST packet if necessary

If the target port is open, the system responds with a SYN-ACK packet. If the port is closed, the system responds with an RST packet. If there is no response, it could mean that the port is either filtered or the packet was dropped.

SYN scanning is particularly effective because it exploits a weakness in the TCP protocol implementation that allows the scanner to avoid completing the full TCP handshake process. This makes the scan faster and less detectable than other types of TCP scanning techniques.

OSCP PRACTICE PART 03

TCP Netcat port scan on ports 3388-3390:

```
nc -w 1 -z target_ip_address 3388-3390
```

In this command, **-w** 1 specifies a timeout of 1 second for each connection attempt, which can help to speed up the scan. The **-z** option tells Netcat to use zero-I/O mode, which means that it won't actually try to send any data to the target system. Instead, it will just attempt to establish a connection and then immediately close it.

Replace **target_ip_address** with the IP address of the system you wish to scan. The range of ports to scan is specified by **3388-3390.** You can change the port numbers to scan a different range of ports if needed.

When you run this command, Netcat will attempt to connect to each port in the specified range on the target system. If a connection is successful, it will print a message indicating that the port is open. If a connection is not successful, it will move on to the next port in the range.

It's worth noting that Netcat is not as powerful or versatile as some other network scanning tools like Nmap, but it can be a useful tool for basic TCP port scanning.

UDP scanning

UDP scanning is a technique of port scanning that involves sending UDP packets to the target system's ports to determine whether they are open, closed or filtered. Unlike TCP scanning, which uses a three-way handshake to establish a connection and confirm the status of a port, UDP scanning relies on analyzing the responses to the UDP packets.

UDP scanning can be more challenging than TCP scanning because the UDP protocol is connectionless, meaning that it does not require a three-way handshake to establish a connection. This makes it difficult to confirm the status of a UDP port, since there may be no response even if the port is open.

One common technique for UDP scanning is known as "UDP ping." This involves sending a UDP packet to a specific port on the target system and analyzing the response. If the port is open, the system will respond with an ICMP packet indicating that the port is unreachable. If the port is closed or filtered, there may be no response at all.

OSCP PRACTICE PART 03

Here's an example command to perform a UDP Netcat port scan on ports 160-162:

```
nc -u -z -v target_ip_address 160-162
```

In this command, the **-u** option is used to specify that a UDP scan should be performed. The **-z** option indicates that zero-I/O mode should be used, which means that no actual data will be sent over the network during the scan. Finally, the **-v** option is used to enable verbose output, which will display more detailed information about the scan as it progresses.

Replace **target_ip_address** with the IP address of the system you wish to scan. The range of ports to scan is specified by **160-162.** You can change the port numbers to scan a different range of ports if needed.

When you run this command, Netcat will attempt to send a UDP packet to each port in the specified range on the target system. If a response is received, it will indicate that the port is open. If no response is received, it may mean that the port is closed or filtered, or it may simply mean that the response was lost due to network congestion or other factors.

OSCP PRACTICE PART 03

Common Port Scanning Pitfalls

Port scanning is a common technique used by attackers to identify potential vulnerabilities in computer systems. However, even experienced security professionals can fall victim to common pitfalls when it comes to port scanning. Here are some common port scanning pitfalls to avoid:

1. **Not getting permission:** Port scanning without permission can be illegal and unethical. Always get proper authorization before conducting any port scans.

2. **Over-reliance on automated tools:** Automated scanning tools can be helpful, but they can also generate false positives and overlook important details. Always supplement automated scans with manual checks.

3. **Failing to understand the context:** Different systems may have different ports open for legitimate reasons. It's important to understand the context of the system you're scanning to avoid misinterpreting results.

4. **Ignoring firewalls:** Firewalls can block or redirect certain port scan attempts, leading to incomplete or inaccurate results. Make sure to take firewalls into account during your scans.

5. **Not analyzing the data:** Scanning is only the first step in identifying vulnerabilities. It’s important to analyze the data gathered from your scans and prioritize the identified risks based on their severity.

6. **Performing scans too often:** Frequent scans can cause performance issues and alert administrators unnecessarily. Be mindful of how often you scan and consider scheduling scans during off-hours.
7. **Not following up on identified vulnerabilities:** Identifying a vulnerability is just the beginning. Follow up with remediation steps and retest to ensure that the issue has been resolved.

By avoiding these common pitfalls, you can conduct more effective and ethical port scans that help improve overall system security.

Port Scanning with Nmap

Nmap is a powerful tool for conducting network exploration and security auditing. It can be used to scan for open ports on a remote system, which is the first step in identifying potential vulnerabilities.

To perform a basic port scan with Nmap, you can use the following command:

Nmap

```
nmap <target>
```

Replace <target> with the IP address or hostname of the system you want to scan.

By default, Nmap will scan the 1,000 most commonly used ports. If you want to scan all ports, you can use the -p option followed by 1-65535:

Nmap

```
nmap -p 1-65535 <target>
```

OSCP PRACTICE PART 03

Nmap also supports various other options that you can use to customize your scanning behavior. For example, you can specify a specific scan type (such as a TCP SYN scan or UDP scan) using the -s option:

Nmap

```
nmap -sS <target>  # TCP SYN scan
nmap -sU <target>  # UDP scan
```

You can also use the -A option to enable version detection and OS fingerprinting:

Nmap

```
nmap -A <target>
```

This will attempt to determine the operating system and software versions running on the target system by analyzing responses from open ports.

Keep in mind that port scanning without permission is illegal and unethical. Always ensure that you have the proper authorization before using Nmap or any other tool for network reconnaissance.

Accountability for Our Traffic

To monitor the amount of traffic sent to a lab machine during a default Nmap TCP scan of the 1000 most popular ports, we can use iptables by inserting new rules into both the INPUT and OUTPUT chains using the -I option, specifying source and destination IP addresses with -s and -d, respectively, accepting the traffic with -j, and zeroing the packet and byte counters in all chains with -Z.

To implement the above, we can use the following command:

sudo

```
sudo iptables -I INPUT -s <source_IP_address> -d
<destination_IP_address> -p tcp --dport 1:1000 -j
ACCEPT
sudo iptables -I OUTPUT -s <source_IP_address> -d
<destination_IP_address> -p tcp --sport 1:1000 -j
ACCEPT
sudo iptables -Z
```

This will insert new rules into both the INPUT and OUTPUT chains to allow traffic from the specified source IP address to the specified destination IP address on ports 1 through 1000. The -Z option is used to zero the packet and byte counters in all chains.

Once these rules are in place, we can run a default Nmap TCP scan on the target machine and monitor the traffic using iptables. To do this, we can use the following command:

```
sudo

watch -n 1 "sudo iptables -nvx -L"
```

This will display the packet and byte counters for each chain in iptables every second. We can use this to monitor the amount of traffic being sent to the target machine during the scan. Once the scan is complete, we can remove the rules from iptables using the following commands:

```

sudo iptables -D INPUT -s <source_IP_address> -d
<destination_IP_address> -p tcp --dport 1:1000 -j
ACCEPT
sudo iptables -D OUTPUT -s <source_IP_address> -d
<destination_IP_address> -p tcp --sport 1:1000 -j
ACCEPT
```

Stealth / SYN Scanning

Stealth scanning, also known as SYN scanning, is a type of port scanning technique used by attackers to determine which ports on a target system are open and potentially vulnerable. This technique is designed to avoid detection by the target system's security measures, such as intrusion detection systems (IDS) and firewalls.

When performing a stealth scan, the attacker sends a SYN packet to the target system for each port they want to test. The SYN packet is part of the three-way handshake process used by the Transmission Control Protocol (TCP) to establish a connection between two systems.

If the port is open, the target system will respond with a SYN/ACK packet, indicating that it is willing to establish a connection. The attacker can then send an RST packet to terminate the connection before it is fully established, leaving no trace of their activity.

By contrast, if the port is closed, the target system will respond with a RST packet, indicating that it is not willing to establish a connection. This allows the attacker to quickly identify which ports are open and potentially vulnerable.

Stealth scanning is a common technique used by hackers and other malicious actors to gain unauthorized access to systems and networks. It is important for organizations to have strong security measures in place, including firewalls and IDS, to detect and prevent these types of attacks.

OSCP PRACTICE PART 03

Here's an example command for performing a stealth/SYN scan using the "nmap" tool:

```
nmap -sS <target IP address>
```

Replace <target IP address> with the IP address of the system you want to scan. The -sS option specifies that we want to perform a stealth/SYN scan.

By default, nmap will scan all TCP ports on the target system. If you want to scan only specific ports, you can specify them using the -p option followed by a comma-separated list of port numbers. For example, to scan only ports 80 and 443, you could use this command:

```
nmap -sS -p 80,443 <target IP address>
```

Note that performing a stealth scan without permission from the target system's owner is illegal and unethical, so be sure to only use this technique on systems that you are authorized to test.

Example :

```
sudo nmap -sS 10.11.1.220
```

TCP Connect Scanning

TCP Connect Scanning is a technique used in network reconnaissance, where an attacker tries to determine whether a specific port on a target system is open or closed. The basic idea behind TCP Connect Scanning is to establish a full TCP connection with the target system by sending a SYN packet, and then waiting for the ACK packet to be returned.

If the target system responds with an ACK packet, it means that the port is open, and the attacker can proceed with further attacks. On the other hand, if the target system responds with a RST packet, it means that the port is closed, and the attacker needs to try other ports or techniques to gain access.

TCP Connect Scanning is one of the most common scanning techniques used by attackers because it is simple and reliable. However, it can also be detected by intrusion detection systems (IDS) or firewalls that monitor network traffic.

OSCP PRACTICE PART 03

TCP Connect Scanning can be performed using various tools, such as Nmap or Netcat. Here's an example of how to perform TCP Connect Scanning using the Nmap tool:

1. Open a terminal window and type the following

```
nmap -sT <target IP address>
```

2. Replace "<target IP address>" with the IP address of the target system you want to scan.
3. Press Enter to run the command.
4. Nmap will start scanning the target system for open TCP ports using the connect() system call. It will send a SYN packet to each port and wait for the response.
5. When the scan is completed, Nmap will display a list of open TCP ports, along with their service names and versions (if available).

Example :

```
nmap -sT 10.11.1.220
```

UDP scanning

UDP scanning is a technique used to identify open UDP ports on a target system. Unlike TCP, UDP is connectionless and does not use a three-way handshake to establish communication. As a result, scanning for open UDP ports can be more challenging than scanning for open TCP ports.

To perform a UDP scan, the scanner sends UDP packets to the target ports and waits for a response. If an ICMP port unreachable message is received, the port is considered closed. If no response is received, the port is considered open or filtered.

One of the challenges of UDP scanning is that many firewalls and routers are configured to drop UDP packets sent to closed ports rather than sending an ICMP message. This can make it difficult to distinguish between closed and filtered ports.

Additionally, UDP scanning can be slow and resource-intensive because the scanner must wait for each packet to time out before moving on to the next port. As a result, it may be necessary to use specialized tools and techniques to optimize the scanning process.

OSCP PRACTICE PART 03

There are different tools and methods available to perform UDP scanning, but one common command-line tool is nmap. You can use nmap to scan for open UDP ports on a target system by running the following command:

```
nmap -sU <target>
```

In this command, -sU option tells nmap to perform a UDP scan, and <target> is the IP address or hostname of the target system.

For example, if you want to scan for open UDP ports on a system with IP address 192.168.1.100, you can run the following command:

```
nmap -sU 192.168.1.100
```

Nmap will send UDP packets to commonly used UDP ports and report any ports that respond as open or filtered. Note that some firewalls may block UDP packets, which could lead to false negatives in the scan results.

Network sweeping

Network sweeping is a technique used in network security to identify active hosts, open ports, and services running on those hosts within a network. The process involves scanning the network using various tools to collect information about the network's devices and services.

Network sweeping can be performed manually or automatically. Manual network sweeping involves analyzing individual hosts one at a time using tools such as ping, traceroute, and netstat. On the other hand, automated network sweeping uses software tools that scan the entire network automatically to identify all connected devices and open ports.

Network sweeping is an essential part of network security because it allows network administrators to identify potential security vulnerabilities within their networks. By discovering which hosts and services are running on the network, administrators can assess the risk level and take appropriate measures to secure their infrastructure.

OSCP PRACTICE PART 03

There are several commands that can be used to perform network sweeping. Here are some examples:

nmap - This is a popular tool used for scanning networks to identify open ports, hosts and services running on those hosts. The command syntax for nmap is as follows:

```
nmap [options] {target specification}
```

ping - This command sends an ICMP echo request to the specified host to check if it is online and reachable. The command syntax for ping is as follows:

```
ping [options] {hostname/IP address}
```

netstat - This command displays active connections and open ports on the local system. The command syntax for netstat is as follows:

```
netstat [options]
```

OSCP PRACTICE PART 03

traceroute - This command shows the path that packets take from the source host to the destination host. The command syntax for traceroute is as follows:

```
traceroute [options] {hostname/IP address}
```

Example :

```
nmap -sn 10.11.1.1-254
```

```
nmap -v -sn 10.11.1.1-254 -oG ping-sweep.txt
grep Up ping-sweep.txt | cut -d " " -f 2
```

```
cat /usr/share/nmap/nmap-services
```

OS fingerprinting

OS fingerprinting is the process of determining the operating system used by a remote device or host. It involves collecting information about various network protocols, services and applications running on the remote host and analyzing them to identify the operating system running on the device.

There are several techniques that can be used for OS fingerprinting, including passive fingerprinting, active fingerprinting, and combination methods. Passive fingerprinting involves analyzing packets from the remote host without sending any traffic to it, while active fingerprinting involves sending specially crafted packets to the remote host in order to elicit a response that can be analyzed.

OS fingerprinting is often used as part of a security audit or penetration testing, as knowing the operating system of a remote host can help an attacker identify vulnerabilities that can be exploited to gain unauthorized access to the system. However, it can also be used by network administrators to identify and track devices on their network and ensure that they are running the latest patches and updates.

OSCP PRACTICE PART 03

One commonly used command for OS fingerprinting is nmap. Nmap is a versatile security scanning tool that can be used for a variety of tasks, including OS fingerprinting.

To perform OS fingerprinting with nmap, you can use the following command:

```
nmap -O <target>
```

Replace <target> with the IP address or hostname of the device you want to fingerprint. The -O option tells nmap to perform OS detection.

When run, nmap will send a series of packets to the target and analyze the responses to determine the operating system running on the device. It will also provide information about open ports and services running on the device.

Example :

```
sudo nmap -O 10.11.1.220
```

OSCP PRACTICE PART 03

Banner Grabbing/Service Enumeration

Banner grabbing, also known as service enumeration, is the process of collecting information about a remote server or application by analyzing the response messages sent by the server. In this technique, an attacker sends requests to the target system and then analyzes the responses to identify the type and version of the software running on the server.

Banner grabbing is often used as a reconnaissance technique during penetration testing or in cyberattacks. It helps attackers to identify potential vulnerabilities in the target system and plan their attack accordingly.

There are several tools available for banner grabbing, including Nmap, Netcat, and Telnet. These tools allow users to send various types of requests to a server and analyze the responses to gather information about the target system.

It's important to note that banner grabbing can be considered a form of passive reconnaissance, as it does not involve any attempts to exploit vulnerabilities in the target system. However, it can still be detected by intrusion detection systems (IDS) or intrusion prevention systems (IPS), so caution should be exercised when conducting banner grabbing activities.

OSCP PRACTICE PART 03

Here are some examples of commands that can be used for banner grabbing:

1.

Nmap:

```
nmap -sV targetIP
```

2.

Telnet:

```
telnet targetIP port
```

3.

Netcat:

```
nc -v targetIP port
```

4.

cURL:

```
curl -I targetURL
```

This command will send a HEAD request to the target URL and display the HTTP header information returned by the server, including the server software and version.

Nmap Scripting Engine (NSE)

The Nmap Scripting Engine (NSE) is a flexible and powerful feature of the Nmap network scanning tool. It allows users to write scripts in Lua programming language that can be used to automate a variety of tasks related to network discovery, vulnerability scanning, and exploitation.

The NSE contains hundreds of pre-built scripts for tasks such as service detection, version detection, vulnerability testing, and more. These scripts can be run individually or in combination with other scripts to create custom workflows for specific scanning scenarios.

One of the key benefits of using the NSE is that it allows users to customize their scanning methods based on the specific network environment they are working with. For example, a user may want to scan only certain ports or services on a target machine, or they may want to run a specific set of tests to identify potential vulnerabilities.

Overall, the NSE is a powerful tool that can greatly enhance the effectiveness and efficiency of network scanning and security testing.

OSCP PRACTICE PART 03

To use the Nmap Scripting Engine (NSE), you need to run the nmap command with the -sC or --script option, followed by the name of the script(s) you want to run. Here's an example:

```
nmap -sC target_ip
```

This will run a default set of scripts against the target IP address using the NSE. You can also specify individual scripts to run by name:

```
nmap --script http-title.nse target_ip
```

This would run only the HTTP title script against the target IP address.

You can also use the -sV option to enable version detection while running NSE scripts, which can help identify specific vulnerabilities that may be present on the target machine:

```
nmap -sV --script vuln target_ip
```

This would run all available vulnerability scanning scripts against the target IP address.

Masscan

Masscan is an open-source tool that is used for high-speed IP port scanning. It allows you to scan the entire internet in under 6 minutes by sending packets at a very high rate. Masscan supports both TCP and UDP protocols, and it can be used to scan any range of IP addresses or ports. The tool is designed to be fast and efficient, and it uses asynchronous I/O to send packets as quickly as possible without waiting for responses. Masscan is commonly used by security researchers and system administrators to identify open ports on systems that may be vulnerable to attacks.

OSCP PRACTICE PART 03

Here are some examples of how to use Masscan:

To scan a single IP address for open TCP ports 1-65535:

```
masscan -p1-65535 <IP_address>
```

To scan a range of IP addresses for open TCP ports 80, 443, and 8080:

```
masscan -p80,443,8080 <IP_range>
```

To scan a subnet for open UDP ports 53 and 161:

```
masscan -pU:53,161 <subnet>
```

To output the results of the scan to a file in JSON format:

```
masscan -p1-65535 <IP_address> -oJ
<output_file>.json
```

OSCP PRACTICE PART 03

To scan with a specific rate, such as 10,000 packets per second:

```
masscan -p1-65535 <IP_address> --rate=10000
```

Masscan should be used responsibly and ethically, as scanning systems without proper authorization can violate laws and result in legal consequences.

Example :

```
sudo apt install masscan
```

```
sudo masscan -p80 10.0.0.0/8
```

```
sudo masscan -p80 10.11.1.0/24 --rate=1000 -e tap0 --router-ip 10.11.0.1
```

OSCP PRACTICE PART 03

SMB enumeration

SMB enumeration is the process of gathering information from a target system that is running the Server Message Block (SMB) protocol. SMB is a network protocol used by Windows-based computers to share files, printers, and other resources.

There are several tools and techniques that can be used for SMB enumeration, including:

1. **NetBIOS Name Service (NBT-NS) queries:** NBT-NS is a protocol used to map NetBIOS names to IP addresses. Enumerating the available NetBIOS names on a target system can reveal information about its hostnames, workgroups, and other network resources.

2. **SMB version scanning:** Different versions of the SMB protocol may have different vulnerabilities or features. Scanning the target system for the SMB version it is running can provide valuable insights for further exploitation.

3. **User enumeration:** Attempting to enumerate valid usernames on a target system can help attackers identify potential targets for password guessing attacks.

4. **Share enumeration:** SMB shares can contain sensitive data, such as user credentials or confidential documents. Enumerating the available SMB shares can provide attackers with potential targets for further exploitation.

5. **Null session enumeration:** A null session refers to an anonymous SMB connection that does not require authentication. This can allow attackers to access file and printer sharing resources without a valid username and password.

OSCP PRACTICE PART 03

There are several commands and tools that can be used for SMB enumeration, including:

1. **nbtscan:** This is a command-line tool that can be used to discover NetBIOS names on a network. To use nbtscan, simply run the command "nbtscan <IP_range>" where IP_range is the range of IP addresses you wish to scan.

2. **smbmap:** This is a command-line tool that can be used to enumerate SMB shares on a remote host. To use smbmap, run the command "smbmap -H <target_IP>" where target_IP is the IP address of the host you wish to scan.

3. **enum4linux:** This is a tool that can be used to enumerate information from Windows and Samba systems. To use enum4linux, run the command "enum4linux -a <target_IP>" where target_IP is the IP address of the system you wish to scan.

4. **smbclient:** This is a command-line tool that can be used to connect to an SMB share and retrieve files or perform other operations. To use smbclient, run the command "smbclient //<target_IP>/<share_name> -U <username>" where target_IP is the IP address of the system hosting the share, share_name is the name of the share you wish to connect to, and username is a valid username for the share.

Scanning for the NetBIOS Service

To scan for the NetBIOS service, you can use a tool like nmap. Nmap is a popular network scanner that can be used to discover hosts and services on a network.

To scan for the NetBIOS service using nmap, you can run the following command:

```
nmap -sU -p 137 --script nbstat.nse <target_IP>
```

In this command, -**sU** specifies that we want to perform a UDP scan, which is necessary since NetBIOS uses UDP **port 137. -p 137** specifies that we only want to scan **port 137.** Finally**, --script nbstat.nse** specifies that we want to run the nbstat script, which will attempt to determine the NetBIOS name of the host.

Replace **<target_IP>** with the IP address of the host you want to scan. The output of this command should show whether the NetBIOS service is running on the target host and what its name is.

Example :

```
sudo nbtscan -r 10.11.1.0/24
```

OSCP PRACTICE PART 03

Nmap SMB NSE Scripts

Nmap comes with a set of useful SMB NSE (Nmap Scripting Engine) scripts for enumerating and probing SMB services on target systems. Here are some of the most commonly used ones:

smb-enum-shares.nse: This script enumerates the shares available on an SMB server along with their permissions.

smb-enum-users.nse: This script enumerates the users and groups present on an SMB server.

smb-os-discovery.nse: This script attempts to determine the operating system running on the SMB server by analyzing its response to certain requests.

smb-security-mode.nse: This script determines the security mode enabled on the SMB server (either user-level or share-level).

smb-vuln-ms08-067.nse: This script checks for the presence of the MS08-067 vulnerability, which affects certain versions of Microsoft Windows and allows remote code execution.

smb-vuln-ms17-010.nse: This script checks for the presence of the MS17-010 vulnerability, which affects certain versions of Microsoft Windows and allows remote code execution.

These scripts can be run individually or as part of a larger Nmap scan using the **-sC** option.

Here are some examples of how to run the above mentioned NSE scripts using Nmap:

1.

Enumerate shares:

```
nmap -p 139,445 --script smb-enum-shares.nse <target>
```

2.

Enumerate users and groups:

```
nmap -p 139,445 --script smb-enum-users.nse <target>
```

3.

OS discovery:

```
nmap -p 139,445 --script smb-os-discovery.nse <target>
```

4.

Security mode check:

```
nmap -p 139,445 --script smb-security-mode.nse <target>
```

5.

MS08-067 vulnerability check:

```
nmap -p 139,445 --script smb-vuln-ms08-067.nse <target>
```

6.

MS17-010 vulnerability check:

```
nmap -p 139,445 --script smb-vuln-ms17-010.nse
<target>
```

Note that you may need to adjust the port numbers **(-p)** based on the specific configuration of the target system.

Example :

```
ls -1 /usr/share/nmap/scripts/smb*
```

```
nmap -v -p 139, 445 --script=smb-os-discovery
10.11.1.227
```

NFS Enumeration

NFS (Network File System) enumeration is the process of gathering information about NFS shares and their permissions on a target system. This can be useful for penetration testers or attackers looking to gain unauthorized access to data stored on an NFS share.

There are several tools available for NFS enumeration, including:

showmount: This is a command-line tool that lists the NFS shares exported by a remote host. It can also show which hosts have mounted each share.

nfsstat: This tool displays statistics for NFS activity on both the client and server sides.

nfs-ls: This tool allows you to list the contents of an NFS share, similar to the ls command used for local file systems.

NfSpy: This is a Python tool that allows you to mount NFS shares as a regular user, bypassing the need for root privileges. It also supports stealth mode, which can help avoid detection.

When performing NFS enumeration, it's important to remember that NFS shares may contain sensitive data, so always ensure that you have permission to perform these activities and handle any data obtained responsibly.

OSCP PRACTICE PART 03

Scanning for NFS Shares

To scan for NFS shares on a remote system, you can use the showmount command in Linux. Here's an example command:

```
showmount -e <target_IP_address>
```

Replace <target_IP_address> with the IP address of the system you want to scan.

The -e option tells showmount to display a list of all the NFS shares exported by the target system. If the command is successful, it will output a list of directories that are exported by the NFS server along with their mount options, such as read-only or read-write access.

Example :

```
nmap -v -p 111 10.11.1.1-254
```

```
nmap -sV -p 111 --script=rpcinfo 10.11.1.1-254
```

OSCP PRACTICE PART 03

Nmap NFS NSE Scripts

Nmap is a popular network exploration and security auditing tool that can be used to scan networks for open ports and vulnerable services. Nmap includes a set of scripts called the "NSE" (Nmap Scripting Engine) that can be used to automate common tasks such as vulnerability testing, network discovery, and service enumeration.

NFS (Network File System) is a distributed file system protocol that allows clients to access files and directories on remote servers as if they were local. NFS exports file systems to other hosts on the network, allowing them to mount the exported file systems and access the files as if they were stored locally.

There are several NSE scripts available in Nmap that can be used to scan for NFS vulnerabilities and misconfigurations. Some of the commonly used scripts are:

nfs-showmount - This script shows the NFS shares that are currently exported by the target server.

nfs-ls - This script can be used to list the contents of an NFS share.

nfs-statfs - This script retrieves the file system status of an NFS share.

nfs-ls - This script can be used to list the contents of an NFS share.

nfs-client-enum - This script attempts to enumerate NFS client mounts, showing the mount details and export options.

nfs-exportlist - This script retrieves the list of exported NFS file systems from the target server.

It is important to note that using NSE scripts for scanning can sometimes result in false positives or trigger alarms on intrusion detection systems. Therefore, it is recommended to use these scripts with caution and ensure that proper authorization is obtained before scanning any network.

The Nmap NFS NSE Scripts are a set of scripts that can be used with Nmap to scan NFS (Network File System) services for vulnerabilities and misconfigurations. To use these scripts, you first need to have Nmap installed on your system. Once you have Nmap installed, you can use the following command to scan for NFS services using the NSE scripts:

```
nmap -p 111 --script nfs* <target>
```

OSCP PRACTICE PART 03

This command will scan the target for NFS services on port 111 and run all the available NFS NSE scripts. You can also specify individual scripts if you only want to run specific ones. For example, to run just the nfs-showmount script, you can use the following command:

```
nmap -p 111 --script nfs-showmount <target>
```

Note that you may need to adjust the port number or use additional options depending on your specific scanning needs. Be sure to check the Nmap documentation for more information on how to use its various options and scripts.

Example :

```
ls -1 /usr/share/nmap/scripts/nfs*
```

```
nmap -p 111 --script nfs* 10.11.1.72
```

```
mkdir home
sudo mount -o nolock 10.11.1.72:/home ~/home/
cd home/ && ls
```

OSCP PRACTICE PART 03

SMTP enumeration

SMTP enumeration is the process of using various techniques to gather information about an SMTP (Simple Mail Transfer Protocol) mail server. The goal of SMTP enumeration is usually to identify valid email addresses or other vulnerabilities in the mail server.

There are several techniques that can be used for SMTP enumeration, including:

1. **Banner Grabbing:** This involves connecting to the SMTP server and retrieving the banner message, which may contain useful information about the server software and version.
2. **DNS Enumeration:** This involves querying the Domain Name System (DNS) to gather information about MX (Mail Exchange) records for the domain, which can reveal the address of the mail server.
3. **User Enumeration:** This involves using a brute force attack to guess valid email addresses on the server, by trying common names or guessing based on the format of the email addresses used by the organization.
4. **SMTP VRFY Command:** This command can be used to verify the validity of a specific email address on the server, which can reveal whether an email address is valid or not.

OSCP PRACTICE PART 03

SMTP enumeration is the process of extracting email addresses and user names from an SMTP server. There are various tools and techniques that can be used for SMTP enumeration, but one common command used in manual enumeration is the VRFY command.

The VRFY command is used to verify the existence of a particular username or email address on an SMTP server. To use the VRFY command, you would connect to the SMTP server using a telnet client and then issue the following command:

```
VRFY <username/email>
```

For example, if you wanted to verify the existence of the email address "johndoe@example.com", you would enter the following command:

```
VRFY johndoe@example.com
```

OSCP PRACTICE PART 03

Example :

```
nc -nv 10.11.1.217 25
VRFY root
VRFY idontexist
^C
```

OSCP PRACTICE PART 03

SNMP Enumeration

SNMP (Simple Network Management Protocol) enumeration is a technique used by attackers to gather information about network devices and their configurations. The goal of SNMP enumeration is to find vulnerabilities or weaknesses that can be exploited to gain unauthorized access to network resources.

Here are some steps that an attacker might take during SNMP enumeration:

1. **Identify the target system:** An attacker will first identify the target system that they want to enumerate.
2. **Discover the SNMP service:** The attacker will then use a scanning tool like Nmap to discover if the target system is running an SNMP service.
3. **Enumeration:** Once the SNMP service has been discovered, the attacker will use a tool like snmpwalk or snmpenum to enumerate the SNMP objects on the system. By doing this, the attacker can obtain information about the system's hardware, operating system, applications, and network configuration.
4. **Analyze Results:** After collecting information, the attacker will analyze the results for any vulnerabilities or weaknesses that can be exploited.

5. **Exploit:** Finally, the attacker will attempt to exploit any identified vulnerabilities or weaknesses to gain unauthorized access to the network or the target system.

To protect against SNMP enumeration, you should ensure that SNMP service is not exposed to the internet or to unauthorized users. You can also use firewalls to block incoming SNMP traffic from untrusted sources, and use strong authentication and encryption mechanisms to secure SNMP traffic between devices.

The SNMP MIB Tree

The SNMP MIB (Management Information Base) tree is a hierarchical structure that represents the network elements managed by a SNMP (Simple Network Management Protocol) agent. The MIB tree is composed of nodes, each representing a specific object or group of objects on a network device.

The root node of the MIB tree is known as the "iso" node and is represented by the OID (Object Identifier) 1.3.6.1. The next level down from the root node represents the different organizations that have defined their own unique OIDs. Below each organization node are various branches that represent the different MIB modules defined by that organization.

Each MIB module is represented by its own branch in the MIB tree, with the top-level object in the module being represented by its own node. Each object in the MIB tree is identified by a unique OID, which allows it to be accessed by SNMP managers for monitoring and management purposes.

Overall, the SNMP MIB tree provides a standardized way for network administrators to monitor and manage network devices using SNMP. It allows them to easily locate and access specific objects on different devices, making it an essential tool for network management.

SNMP MIB tree, you can use a command-line tool such as "snmpwalk". Here's an example command to walk the entire MIB tree using SNMPv2c protocol:

```
snmpwalk -v2c -c <community_string>
<device_IP_address> 1.3.6.1
```

In this command, replace <community_string> with the SNMP community string that you use to authenticate access to the device, and <device_IP_address> with the IP address of the device you want to query.

The OID "1.3.6.1" is used to specify the root node of the MIB tree. The output of this command will display all objects in the MIB tree, starting from the root node and descending through the hierarchy.

You can also limit the output to specific branches or objects in the MIB tree by specifying a different OID in place of "1.3.6.1". For example, to view the objects in the "system" branch of the MIB tree, you can use the following command:

```
snmpwalk -v2c -c <community_string>
<device_IP_address> 1.3.6.1.2.1.1
```

For example, the following MIB values correspond to specific Microsoft Windows SNMP
parameters and contains much more than network-based information:

1.3.6.1.2.1.25.1.6.0	System Processes
1.3.6.1.2.1.25.4.2.1.2	Running Programs
1.3.6.1.2.1.25.4.2.1.4	Processes Path
1.3.6.1.2.1.25.2.3.1.4	Storage Units
1.3.6.1.2.1.25.6.3.1.2	Software Name
1.3.6.1.4.1.77.1.2.25	User Accounts
1.3.6.1.2.1.6.13.1.3	TCP Local Ports

Scanning for SNMP

Nmap is a popular tool for network scanning and exploration, including identifying open ports on devices.

To scan for open SNMP ports using nmap with UDP scanning, you can use the following command:

```
nmap -sU --open <target IP range or hostname>
```

The -sU option tells nmap to perform UDP scanning, which is necessary to identify open SNMP ports since SNMP uses the UDP protocol. The --open option restricts the output to show only open ports, which in this case will be the open SNMP ports.

You can replace <target IP range or hostname> with the IP address range or hostname of the devices you want to scan. For example, if you want to scan all hosts on the local network with IP addresses starting with 192.168.1.x, you could use:

```
nmap -sU --open 192.168.1.*
```

OSCP PRACTICE PART 03

Example :

```
sudo nmap -sU --open -p 161 10.11.1.1-254 -oG open-snmp.txt
```

```
echo public > community
echo private >> community
echo manager >> community
for ip in $(seq 1 254); do echo 10.11.1.$ip; done > ips
onesixtyone -c community -i ips
```

Windows SNMP Enumeration Example

Here's an example of how to perform SNMP enumeration on a Windows machine:

Use a tool like "snmpwalk" or "snmpenum" to scan the target system for open SNMP ports.

Once you've identified an open SNMP port, use a tool like "snmpget" or "snmpwalk" to query the SNMP service for information.

For example, to retrieve the system description from an SNMP-enabled Windows machine, you could use the following command:

```
snmpwalk -v2c -c <community-string> <target-ip> sysDescr.0
```

Replace <community-string> with the SNMP community string (e.g. "public") and <target-ip> with the IP address of the target Windows machine.

This should return the system description string, which may include information such as the operating system version and other system details.

Enumerating the Entire MIB Tree

Enumerating the entire Management Information Base (MIB) tree can be a time-consuming process, but it can provide valuable information about the SNMP-enabled device you are targeting.

To enumerate the entire MIB tree, you can use the following command with the SNMP tool "snmpwalk":

```
snmpwalk -v2c -c <community-string> <target-ip> 1.3.6.1
```

This will start at the root of the MIB tree and list all of the available object identifiers (OIDs) and their associated values. The command specifies version 2c of the SNMP protocol, the community string to use for authentication, and the IP address of the target device.

You can also specify a sub-tree to start from if you want to limit the scope of the enumeration. For example, if you only want to enumerate the system-related OIDs, you could use:

```
snmpwalk -v2c -c <community-string> <target-ip> 1.3.6.1.2.1
```

This will start at the "system" subtree of the MIB tree and list all the OIDs under that subtree.

Enumerating Windows Users

To enumerate Windows users on a system, you can use the following steps:

1. Open the command prompt as an administrator.
2. Type "net user" and press enter. This will display a list of all user accounts on the system.
3. To view more information about a specific user, type "net user <username>" and press enter. This will display details such as when the account was created, whether it is active or not, and whether the account has a password set.

Alternatively, you can also go to Control Panel > User Accounts > Manage Accounts to view and manage all user accounts on the system.

Example :

```
snmpwalk -c public -v1 10.11.1.14 1.3.6.1.4.1.77.1.2.25
```

Enumerating Running Windows Processes

To enumerate running Windows processes, you can use the following steps:

1. Open the Task Manager by pressing "Ctrl + Shift + Esc" on your keyboard or by right-clicking the taskbar and selecting "Task Manager".
2. In the Task Manager window, click on the "Processes" tab.
3. This will display a list of all currently running processes on your system. You can view the name, PID (process ID), CPU usage, memory usage, and other details for each process.
4. You can also sort the list by clicking on any of the column headers.
5. To end a process, select it from the list and click on the "End task" button.

Alternatively, you can also use PowerShell to enumerate and manage running processes. Here's an example command to get a list of all processes:

Example :

```
snmpwalk -c public -v1 10.11.1.73 1.3.6.1.2.1.25.4.2.1.2
```

OSCP PRACTICE PART 03

Enumerating Open TCP Ports

To enumerate the open TCP ports, you can use a network scanning tool such as Nmap. Here's an example command to scan for all open TCP ports on a target machine:

```
nmap -p- <target IP address>
```

This command tells Nmap to scan all possible TCP ports on the specified target IP address. Once the scan is complete, Nmap will display a list of all open TCP ports on the target machine. You can also specify a range of ports to scan by replacing "-p-" with "-p" followed by the desired range of ports (e.g. "-p1-1000").

Example :

```
snmpwalk -c public -v1 10.11.1.14 1.3.6.1.2.1.6.13.1.3
```

Enumerating Installed Software

You can enumerate installed software on Windows using the following methods:

1. Using "Programs and Features" in Control Panel: This method displays a list of all installed programs and their version numbers. To access this list, you can open the Control Panel, select the "Programs and Features" option, and wait for the list to populate.

2. Using PowerShell: You can use PowerShell to obtain a list of all installed software on a Windows machine. Open PowerShell as an administrator and run the following command:

```
Get-WmiObject -Class Win32_Product | Select-Object Name,Version
```

This will display a list of all installed software along with their version numbers.

3. Using third-party tools: There are several third-party tools available that can help you enumerate installed software on a Windows machine. Some popular options include Belarc Advisor, CCleaner, and Sysinternals Suite.

Note that some methods may not display all installed software, especially if the software was installed without an installer or if it is outdated. Additionally, be cautious when using third-party tools as they may also display potentially unwanted software or malware.

Example :

```
snmpwalk -c public -v1 10.11.1.50 1.3.6.1.2.1.25.6.3.1.2
```

Contacts us

https://cyberpublicschool.com/

https://www.instagram.com/cyberpublicschool/

Phone no.: +91 9631750498 India
+61 424866396 Australia

Our Successful Oscp Student.