Introduction

Ethical Hackers begin with **reconnaissance**, gathering information through scanning tools and public data. Penetration testers mirror these methods to assess system exposure. This module explains **passive** vs. **active** reconnaissance, introduces common tools, and explores **vulnerability scanning** - how scanners work, how to interpret results, and how findings support exploitation. It ends with key challenges to consider when running scans.

Performing Passive Reconnaissance

Overview

Reconnaissance is the first step in a cyber attack, where attackers collect information about a target to ensure success. Borrowed from military practice, it involves learning about an enemy's location, capabilities, and movements. In penetration testing, reconnaissance typically means scanning and enumeration. The key question is: how does this process appear from an attacker's perspective?

Active Reconnaissance vs. Passive Reconnaissance

Active reconnaissance involves sending probes to a target system or network to elicit responses and assess its posture. While effective, it can sometimes disrupt fragile devices, so scanner settings must be used carefully.

Passive reconnaissance gathers information without directly interacting with the target, often through third-party databases or by monitoring network traffic. This method is less invasive, unlikely to crash systems, and harder to detect—making it ideal for analyzing production networks that cannot be disrupted.

A strong penetration testing methodology helps determine when to use active or passive techniques and which tools best fit the engagement.

Common active reconnaissance tools and methods include the following:

- Host enumeration
- Network enumeration
- User enumeration
- Group enumeration
- Network share enumeration
- Web page enumeration
- Application enumeration
- Service enumeration
- Packet crafting

Common passive reconnaissance tools and methods include the following:

- Domain enumeration
- Packet inspection
- Open-source intelligence (OSINT)
- Recon-ng
- Eavesdropping

Using OSINT Tools

Passive reconnaissance relies on publicly accessible data to support active efforts and gather details about an organization and its employees. In **OSINT (Open Source Intelligence)**, the focus is on open data, though tools used may be free, require registration, or be paid. Since attackers can access the same sources, penetration testers report on sensitive information that could expose vulnerabilities.

OSINT objectives include:

- Identifying the organization’s digital footprint
- Revealing what data is available to cybercriminals

Instructions

Part 1: Examine OSINT Resources

Step 1: Access the OSINT Framework

- Go to https://osintframework.com/ and explore the tree of OSINT categories.
- Click **Username** at the top of the tree → two subcategories appear.
    - Open WhatsMyName Tool
    - Under **Username Search Engines**, click **WhatsMyName (T)**.
        - This links to the GitHub repository. In the **README.md**, you'll see sites that implement the tool.
    - Click the first link: https://whatsmyname.app/
        - The parent organization for the site is https://www.osintcombine.com/, which also offers other free resources.
    - Run Username Searches:
    - Enter one or more usernames (each on a new line).
    - Optionally, use a common username wordlist for test inputs.
    - Click the green magnifying glass to start the search.
    - Analyze Results:
        - Results show accounts across multiple platforms.
        - You can open links directly from the green rectangles or the results table.
        - Reports are flexible: sort by column, filter by username, or export as CSV/PDF.
        - Each entry includes a direct link to the profile page.

Step 2: Investigate SMART - Start Me Aggregated Resource Tool

- start.me is a popular bookmark manager and productivity tool. My OSINT Training (MOT) has built a search system on it that collects OSINT-related links shared by users. By entering OSINT keywords, you can quickly find and explore relevant resources.
- Go to https://smart.myosint.training/

    - In the search box, enter the term **usernames**
        - You will see a list of username-related OSINT tools that other people have found.
    - Open some of the links to review the resources. Be careful however, these websites come from public sources. Some may be malicious.
    - Choose some of the categories that you saw in the OSINT Framework and see what links appear.
    - Use this site to search for OSINT tools and resources to help you in your pentesting work.

Part 2: Use SpiderFoot

SpiderFoot is an automated OSINT scanner. It is included with Kali. SpiderFoot queries over 1000 open-information sources and presents the results in an easy-to-use GUI. SpiderFoot can also be run from a console. SpiderFoot seeds its scan with one of the following:

- Domain names
- IP addresses
- Subnet addresses
- Autonomous System Numbers (ASN)
- Email addresses
- Phone numbers
- Personal names

SpiderFoot offers the option of choosing scans based on use case, required data, and by SpiderFoot module.
The use cases are:

- All – Get every possible piece of information about the target. This use case can take a very long time to complete.
- Footprint – Understand the target's network perimeter, associated identities and other information that is yielded by extensive web crawling and search engine use.
- Investigate – This is or targets that you suspect of malicious behavior. Footprinting, blacklist lookups, and other sources that report on malicious sites will be returned.

- Passive – This type of scan is used if it is undesirable for the target to suspect that it is being scanned. This is a form of passive OSINT.

Step 1: Start and run SpiderFoot

In a terminal, enter the following command:

```
┌──(kali㉿Kali)-[~]
└─$ spiderfoot -l 127.0.0.1:5001
```

Open a browser and enter the IP address and port for the SpiderFoot GUI. You will see the SpiderFoot interface appear.

Hint: scanners with a lock next to them indicate an API key is necessary. Further information regarding the key requirements is provided in the details for the scanner. Click the "**?**" icon next to the API Settings option.

Hint: You can interact with SpiderFoot from the terminal too. You can display all the modules that are available in SpiderFoot and pipe the output to a text file. Enter spiderfoot –h to view the command line options.

The grep command can then be used to search the file for keywords. This will not provide information about API requirements, but it will help you to make sense of the list of available modules.

```
┌──(kali㉿Kali)-[~]
└─$ spiderfoot -M | grep [search term]
```

Step 2: Run a SpiderFoot Scan for a Domain

1. Click the **New Scan** tab in the GUI.
2. Enter a name for the scan and select a target. In this case, we will use **h4cker.org**
3. Select the scan use case as **Footprint**
4. Click the **Run Scan Now** button
5. Mouse over the bars for a summary of the findings for that data type.

Step 3: Investigate Scan Results

1. Go back to the scan results, by clicking the **Scans** tab
2. Click the name of the scan in the table to return to the scan view. You will be taken to the **Browse** tab. Each row in the table represents data found by the various modules. Some modules contribute to multiple types of data.

Step 4: Register API Keys (optional)

API keys will enhance the functionality of SpiderFoot.

1. Go to the **Settings** tab
2. Enter the API keys in the settings for each module. Be sure to save your changes.
3. Click **New Scan**. Go to the **By Module** tab. Select only the modules for which you have added API keys. All other modules should be unchecked.
4. Enter the target as **h4cker.org** and click **Start Scan**.

Part 3: Investigate Recon-ng

Recon-ng is an OSINT framework modeled after tools like Metasploit and SET. It uses modular workspaces where each module has its own configurable options, making command-line use simpler. Once configured, modules run searches based on those settings. Recon-ng supports a wide range of reconnaissance tasks, with some modules included in Kali Linux and others available for download from its marketplace.

Step 1: Create a workspace.

Recon-ng has auto complete. Press the tab button to complete commands and command options. Use the tab key twice to list the available commands and options at different places in the command line. This is very handy.

- To run Recon-ng, open a new terminal window and enter **recon-ng**
    - You can also start the program by going to the Kali tools menu, searching for the app, and clicking the icon.
- Enter **help** to get a sense of the commands that are available.

- Recon-ng uses workspaces to isolate investigations from one another. Workspaces can be created for different parts of a test or different customers for example. Type **workspaces help** to view options for the workspaces command.
- Create a workspace named **test** by entering **workspaces create** followed by the workspace name.
- Type **help** to see the commands that are available within workspaces.

Step 2: Investigate modules.

Recon-ng is a modular framework. Modules are Python programs with different functions. They are stored in an external marketplace that permits developers to create their own modules and contribute them for use by others.

Return to the Recon-ng prompt. Enter the **modules search** command. This will display the currently installed modules.

Step 3: Investigate the module marketplace.

Recon-ng will not function without modules. Search the web for **recon-ng-marketplace** to view the repository.

- In the terminal, view **help** for the **marketplace** command.
- Use the **search** option to list all the modules that are currently available.

[recon-ng][default] > **marketplace search**

- To learn more about individual modules, use the **marketplace info** command followed by the full name of the module.

Step 4: Install a new module.

Recon-ng accesses modules from the Github repository and downloads them to Kali when they are installed.

- Search the marketplace modules using bing as a search term.

- Enter the **marketplace install** command followed by the full name of the module.

  [recon-ng][default] > **marketplace install recon/domains-hosts/bing_domain_web**

- After installation, enter the **modules search** command to verify that the new module is now available.

Step 5: Run the new modules

- To start working with a module, it must be initialized. Enter **modules load bing_domain_web** to begin working with the module.
- Type **info** at the module prompt to view important details about the module.
- Use the **options set source** command to set the only option for this module. Complete the command by specifying the target as **hackxor.net**
- Verify the option setting with the **info** command.
- Type **run** to execute the module.
- Enter the **dashboard** command. This queries the Recon-ng database and provides a summary of the information that has been gathered. It is specific to this workspace.
- The **show** command displays the data for specific categories. Enter the **show hosts** command to display the list of hosts that were discovered.

Step 6: Investigate the web interface.

Recon-ng has a web interface that simplifies and improves viewing results that are stored in Recon-ng databases. It also allows easy export of the results tables for reporting purposes.

- Open a new terminal
- Enter the **recon-web** command to start the Recon-ng server process.
- In a new browser tab, access the webpage using the URL information provided in the output.

DNS Lookups

- DNSRecon Example:

  |--[omar@websploit]--[~]
  |--- $ **dnsrecon -d h4cker.org**

- You can use other basic DNS tools, such as the **nslookup**, **host**, and **dig** Linux commands, to perform name resolution and obtain additional information about a domain.

  |--[omar@websploit]--[~]
  |--- $**dig h4cker.org**

- Obtaining the MX Record of h4cker.org

  |--[omar@websploit]--[~]
  |--- $**dig h4cker.org mx**

- Whois Information for the Domain h4cker.org

  |--[omar@websploit]--[~]
  |--- $ **whois h4cker.org**

- Showing Technical and Administrative Email Contacts

```
|--[omar@websploit]--[~]
|--- $whois cisco.com | grep '@cisco.com'
Registrant Email: infosec@cisco.com
Admin Email: infosec@cisco.com
Tech Email: infosec@cisco.com
```

- Access the manual pages for nslookup using the man command:

  ┌──(kali㉿Kali)-[~]
  └─$ **man nslookup**

- To query for the mail server mx record within a domain

    set querytype=mx or set type=mx

Using the nslookup command

- Use the **nslookup** command with no options to enter interactive mode. To exit interactive mode at any time, type **exit** to return to the CLI prompt.
- The CLI prompt changes to **>** to indicate that you are now in interactive mode and can enter the various nslookup commands. Enter the domain name **cisco.com** to resolve the domain name to an IP address. By default, the **nslookup** command queries A and AAAA records for the target.

    ```
    ┌──(kali㉿Kali)-[~]
    └─$ nslookup
    > cisco.com
    ```

- To find the domain name servers configured for cisco.com, use the **set type** command to change the query type to "**ns**" to return the name server information.

    ```
    > set type=ns
    > cisco.com
    ```

- Enter **exit** to leave interactive mode and return to the CLI prompt.

Change the server used to perform lookups

- In CLI prompt:

    ```
    ┌──(kali㉿Kali)-[~]
    └─$ nslookup netacad.com 8.8.8.8
    ```

- In interactive mode:

```
┌──(kali㉿Kali)-[~]
└─$ nslookup
> server 8.8.8.8
> netacad.com
```

- The any query type can retrieve much, or all, of the information contained in the DNS record for a host name.

```
┌──(kali㉿Kali)-[~]
└─$ nslookup
> server 8.8.8.8
> set type=any
> netacad.com
```

Use the Whois function to obtain domain information

- ```
  ┌──(kali㉿Kali)-[~]
  └─$ whois cisco.com
  ```

Use whois to determine IP address registration information

- ```
  ┌──(kali㉿Kali)-[~]
  └─$ whois 72.163.4.185
  ```

Compare the Output of the Nslookup and Dig Functions

- ```
  ┌──(kali㉿Kali)-[~]
  └─$ dig cisco.com
  ```

- To obtain the **IPv6** address of cisco.com it is necessary to add a type to the command structure. The syntax to instruct Dig to query a specific record type is dig [hostname] [record type]

    - ┌──(kali㉿Kali)-[~]
  ```

└─$ **dig cisco.com AAAA**

Use Dig to Obtain Additional Information

- ┌──(kali㉿Kali)-[~]
  └─$ **dig cisco.com 8.8.8.8 ns**

- The any record type can also be queried using Dig.

  - ┌──(kali㉿Kali)-[~]
    └─$ **dig netacad.com any**

Perform Reverse DNS Lookups

- ┌──(kali㉿Kali)-[~]
  └─$ **dig -x 72.163.4.185**

- ┌──(kali㉿Kali)-[~]
  └─$ **dig -x 72.163.1.1**

- ┌──(kali㉿Kali)-[~]
  └─$ **host 72.163.10.1**

- ┌──(kali㉿Kali)-[~]
  └─$ **host hsrp-72-163-10-1.cisco.com**

- ┌──(kali㉿Kali)-[~]
  └─$ **host hsrp-72-163-10-1.cisco.com**

Use nslookup to Perform rDNS Lookups

- In CLI prompt:

  ┌──(kali㉿Kali)-[~]
  └─$ **nslookup 72.163.4.185**

- In interactive mode:

```
┌──(kali㉿Kali)-[~]
└─$ nslookup
> 72.163.4.185
```

Cloud vs. Self-Hosted Applications and Related Subdomains

DNS Name Resolution for netflix.com

- |--[omar@websploit]--[~]
  |--- $**host netflix.com**

Ownership of IP Addresses and Applications Hosted in the Cloud

- |--[omar@websploit]--[~]
  |--- $**whois <*ip address*> | grep OrgName**

Cryptographic Flaws

- Digital Certificate
- Certificate Transparency
  - https://certificate.transparency.dev/
  - https://crt.sh/
- Finding Information from SSL Certificates
  - View site certificates from a browser
    - Navigate to netacad.com
    - In most browsers, a **padlock** icon appears next to the URL
- View stored certificates in the operating system
  - In Windows:
    - Microsoft Windows has a security management application that is part of the Microsoft Management Console. Enter **certmgr.msc** in the search box.

                - Access information about trusted root and intermediate certificates in Windows by selecting the appropriate certificate folders in the management app.
        - In Kali
            - The stored certificates are in the **/usr/share/ca-certificates/mozilla** folder.
            - **Right-click** a certificate and select Open With "**ViewFile**" to access the information for a certificate.
                - Access the certificates folder and use **ls -l | grep root** to list root certificate files, or search for the word **root** in the file manager window.
- Use SSL Analysis Tools in Kali
    - Kali comes with several SSL-related tools. Click the Kali programs icon and search on the term **ssl**
- Use Kali Tools to Gather Certificate Information
    - We will use **sslscan** to gather information about certificates and use another utility, called **aha**, to output the results to an HTML file.
        - Install **aha**
            - **sudo apt update**
            - **sudo apt install -y aha**
        - Run sslscan and save the output to an HTML file
            - **sslscan netacad.com**
                - The meaning of the color coding is as follows:
                    - Red background text – NULL cipher. No encryption was used.
                    - Red – broken cipher (less than or equal to 40-bit), vulnerable or broken protocol such as SSLv2 or SSLv3 or broken certificate signing algorithm such as MD5.
                    - Yellow – weak cipher (less than or equal to 56-bit) or weak signing algorithm such as SHA-1.

        - Purple – anonymous cipher such as ADH or AECDH.
    - **sslscan netacad.com | aha > sfa_cert.html**
        - sslscan will save the file in the Kali Home directory as indicated by the prompt. You can add a path to the filename or run the terminal from a destination directory to save it elsewhere.
        - Locate the HTML file and open it with Firefox.

Company Reputation and Security Posture

- Security breaches can have a direct impact on a company's reputation. Attackers can leverage information from past security breaches that an organization might have experienced. They may, for example, leverage the following data while trying to gather information about their victims:
    - **Password dumps**
        - Attackers can leverage password dumps from previous breaches.
        - There are a number of ways that an attacker can get access to such password dumps, such as by using Pastebin, dark web websites, and even GitHub in some cases.
        - Several different tools and websites make this task very easy.
        - An example of a tool that allows you to find email addresses and passwords exposed in previous breaches is **h8mail**.
            - Install h8mail command
                - **pip3 install h8mail**
                - **h8mail -h**
        - The following are additional tools that allow you to search for breach data dumps:

        - **WhatBreach**: https://github.com/Ekultek/WhatBreach
        - **LeakLooker**: https://github.com/woj-ciech/LeakLooker
        - **Buster**: https://github.com/sham00n/buster
        - **Scavenger**: https://github.com/rndinfosecguy/Scavenger
        - **PwnDB**: https://github.com/davidtavarez/pwndb
    - Tools like **h8mail** and **WhatBreach** take advantage of breached data repositories of websites such as **haveibeenpwned.com** and **snusbase.com**
- **File metadata**
    - You can obtain a lot of information from metadata in files such as images, Microsoft Word documents, Excel files, PowerPoint files, and more.
    - For instance, Exchangeable Image File Format (**Exif**) is a specification that defines the formats for images, sound, and supplementary tags used by digital cameras, mobile phones, scanners, and other systems that process image and sound files.
    - Several tools can show Exif details. One of the most popular of them, **ExifTool**
    - Example:
        - |--[omar@websploit]--[~]
          |--- $**exiftool IMG_4730.jpg**
- **Strategic search engine analysis/enumeration**
    - By using basic search techniques combined with advanced operators, both you and attackers can use Google as a powerful vulnerability search tool. The following are some advanced operators:
        - **Filetype**
        - **Inurl**
        - **Link**
        - **Intitle**

        - To see how Google dorking works, enter the following phrase into Google:
            - Example 1
                - **intext:JSESSIONID OR intext:PHPSESSID inurl:access.log ext:log**
            - Example 2
                - **"public $user =" | "public $password = " | "public $secret =" | "public $db =" ext:txt | ext:log -git**
        - Now let's look at advanced Google hacking
            - Visit the Google Hacking Database (GHDB) repositories at https://www.exploit-db.com/google-hacking-database/
            - GHDB has the following search categories:
                - Footholds
                - Files containing usernames
                - Sensitive directories
                - Web server detection
                - Vulnerable files
                - Vulnerable servers
                - Error messages
                - Files containing juicy info
                - Files containing passwords
                - Sensitive online shopping info
                - Network or vulnerability data
                - Pages containing login portals
                - Various online devices
                - Advisories and vulnerabilities
    - **Website archiving/caching**
        - One of the most popular repositories is the "**Wayback Machine**" of Internet Archive https://archive.org/web
    - **Public source code repositories**

Find Information about Email Breaches

- Online services that provide that provide the ability to search on individual email addresses and entire domains to reveal breaches:
    - haveibeenpwned.com
    - f-secure.com
    - hacknotice.com
    - breachdirectory.com
    - keepersecurity.com
- Use a tool to find email addresses for a domain
    - You will use a tool called **EmailHarvester** to find information about a domain, including email addresses of personnel
        - Open a terminal
        - **emailharvester**
        - Enter **y**
        - **emailharvester -h**
    - The results can be output to a file that can be used by other tools as an input list.
    - Use the **-s** option to specify a file name.
    - Emailharvester creates both XML and text files. Supply a path if desired. Otherwise, the files will appear in the **/user/share/emailharvester** folder.
- Use Spiderfoot to research email addresses
    - Open the Spiderfoot GUI:

      ```
      ┌──(kali㉿Kali)-[~]
      └─$ spiderfoot -l 127.0.0.1:5001
      ```

    - Minimize (don’t close) the terminal.
    - Open the Spiderfoot GUI in your browser using the IP address and port assigned above.
    - Click the **Settings** menu item and investigate the available modules. Read the descriptions of the modules and investigate the API requirements if any. Optionally, register for some of the free API keys. Configure the modules with the keys.

- A few interesting modules are:
  - Ahmia
  - AccountFinder
  - Archive.org
  - Bing
  - Leak-Lookup
  - CommonCrawl
  - Dehashed
  - DuckDuckGo
  - EmailCrawlr
- Selecting the interesting modules
  - **New Scan** > **By Module** tab

View File Metadata

- **ExifTool**
  - It comes in a GUI version that is available for Windows, MacOS, and Linux
  - **Install ExifTool**
    - In Firefox, click the **Kali Tools** shortcut or navigate to https://www.kali.org/tools
    - Select **List all tools** as necessary.
    - Locate the entry for **libimage-exiftool-perl**
    - Follow the instructions to install ExifTool.
    - ExifTool refers to metafile attributes as tags. Use the **-list** option to view all the tags that ExifTool can process.
    - Issue the **exiftool -listf** command to review the file types that ExifTool can analyze
  - **Use ExifTool**
    - Go to the Google Hacker Database (GHDB).
    - Locate dorks that will help you to find a variety of files of various types or modify the dorks that you find to do so.

        - Download the files to your VM. Make note of the path to the folder.
    - ExifTool can be run for a **specific file** or **entire folder**, based on the path I provide.
    - You can save the metadata for each image in the folder, or for individual images by adding the **-csv** option. For example:

```
┌──(kali㉿Kali)-[~]
└─$ exiftool -csv > /path/to/out.csv <File Or Dir>
```

    - On the internet, research some of the values that you find in the file. For example, the tag **CREATOR: gd-jpeg v1.0** indicates that the image was generated by the PHP GD library version 1.0. Search the internet for **PHP GD vulnerability** to learn more.

Advanced Searches

- **Google Advanced Searches (Dorking)**
    - Explore Google dorking:
        - There are many Google advanced search operators. Lists of them are available on the internet on sites such as **SpyFu**.
        - Search the internet for "**advanced search operators**"
    - Conduct searches using the Google Advanced Search form
        - Type **advanced search** in the Google search window. This will return a link to the advanced search form.
    - Conduct passive reconnaissance with advanced search operators
        - inurl: operator
            - **site:examplecompany.com inurl:admin**
        - intitle: operator
            - **site:examplecompany.com intitle:login**
        - filetype: operator

            - **site:examplecompany.com filetype:pdf**
        - Use the intext: and filetype: operators
            - **site:examplecompany.com intext:<keyword> filetype:<file type>**
- **The Google Hacking Database**
    - Explore the Google Hacking Database main page
        - Do a Google search for **GHDB**. The first returned page should be The Google Hacking Database.
        - On the GHDB main page, click the **Filters** button in the top right of the window.
        - This allows you to filter the database results by Category or Author. There is also a **Quick Search**.
    - Select Categories to find interesting Dorks
        - Conduct a search for **tsweb**
        - Click the **allinurl:tsweb/default.htm** Dork
    - Combine Category filters with search terms
        - Select **Files Containing Passwords** in the **Categories** drop down.
        - In the **Quick Search** window, type **db_pass**. This will return dork searches for database passwords
- **The Wayback Machine**
    - Explore the Wayback Machine database
        - Navigate to https://web.archive.org to bring up the Wayback Machine home page.
        - Enter the URL of a target company in the Search box.
    - Explore the **Calendar** tab
    - Explore the **Collections** tab
    - Explore the **Changes** tab
        - Click the **Changes** tab
        - This shows how much the page has changed over time
            - **Grey** = has not changed much since the last crawl
            - **Blue** = significant changes.
        - You can also compare changes from two captures to see what has changed

            - Select two captures. They can be on the same day or on different days.
            - Click the **Compare** button.
                - Things that have changed will be highlighted.
    - Explore the Summary tab
        - Click the **Summary** tab
            - The summary applies to the entire domain, whereas calendar, collections, and changes are specific to the URL (single page) searched.
            - This page shows the **MIME** type of the content that was hosted by the domain in the given date range. This can be text, images, javascript, etc.
        - Click the dropdown arrow in the **MIME**-types drop box and review the file types available.
        - Change the Year Start and Year End to see how things have changed over a time of period.
        - Click each of the data type buttons: All, text, application, image, message, audio, video, and explore the information revealed.
    - Explore the Site Map tab
        - The Site Map also applies to the entire domain. The center circle is the "root" and all the rings that surround the center circle are the various pages or trees of the web site. The further out from the root, the more complex the page is.
    - Explore the URLs tab
        - Click the **URLs** tab.
            - This shows all the URLs containing the domain prefix.
        - Use the filter box on the right of the page to search for specific files such as anything that ends in “.bak” to see if they contain any interesting backup information.
        - Some filters to try:
            - .zip
            - .backup

- .config
- .csv
- .pdf
- /api/
- /admin/

Open-Source Intelligence (OSINT) Gathering

- **Recon-ng**
  - Step 1: Start Recon-ng
    - To start using Recon-ng, you simply run **recon-ng** from a new terminal window.
  - Step 2. View available commands
    - Simply type **help** and press Enter
  - Step 3. Search for available modules
    - Use the **marketplace search** command to search for all the available modules in Recon-ng
  - Step 4. Refresh the marketplace.
    - You can refresh the data about the available modules by using the **marketplace refresh** command
  - Step 5. Search the marketplace
    - You can perform a keyword search for any modules by using the command **marketplace search < keyword >**
    - Example: **marketplace search bing**
  - Step 6. Install a module
    - You can install the module by using the **marketplace install** command
    - Example: **marketplace install recon/domains-hosts/bing_domain_web**
  - Step 7. Show installed modules
    - You can use the **modules search** command to show all the modules that have been installed in Recon-ng
  - Step 8. Load a module
    - To load the module that you would like to use, use the **modules load** command

        - For example: the bing_domain_web module is loaded
        - Notice that the prompt changed to include the name of the loaded module
        - After the module is loaded, you can display the module options by using the **info** command
    - Step 9. Change the source
        - You can change the source (the domain to be used to find its subdomains) by using the command **options set SOURCE**
        - Example: **options set SOURCE h4cker.org**
        - After the source domain is set, you can type **run** to run the query
- **Shodan**
    - Shodan is an organization that scans the Internet 24 hours a day, 365 days a year
    - The results of those scans are stored in a database that can be queried at shodan.io or by using an API
    - You can use Shodan to query for vulnerable hosts, Internet of Things (IoT) devices, and many other systems that should not be exposed or connected to the public Internet
    - Create a Shodan Account and Register for an API Key
    - https://www.shodan.io/
    - **Use the Shodan search bar to discover IoT devices**
        - On the Shodan home page, enter **webcam** in the search bar near the top of the screen and press enter
    - **Use Shodan filters to refine the results**
    - Use Shodan to search for a specific product or service
        - Example: **Apache port:80 city:"*your-city*"**
    - Use Shodan from the CLI to Perform a Search
        - **shodan init <paste your API key here>**
        - **shodan -h**
        - **shodan search webcam**
        - **shodan info**
        - **shodan myip**
        - **shodan stats webcam**

Performing Active Reconnaissance

Nmap

- Port Scans
  - Example: **nmap -sS 192.168.88.251**
- Nmap Scan Types
  - TCP Connect Scan (**-sT**)
  - UDP Scan (**-sU**)
  - TCP FIN Scan (**-sF**)
  - Host Discovery Scan (**-sn**)
  - Timing Options (**-T 0-5**)
    - **-T0** (Paranoid) : Very slow, used for IDS evasion
    - **-T1** (Sneaky) : Quite slow, used for IDS evasion
    - **-T2** (Polite) : Slows down to consume less bandwidth, runs about 10 times slower than the default
    - **-T3** (Normal) : Default, a dynamic timing model based on target responsiveness
    - **-T4** (Aggressive) : Assumes a fast and reliable network and may overwhelm targets
    - **-T5** (Insane) : Very aggressive; will likely overwhelm targets or miss open ports
- Nmap UDP scan on port 53 of the target 192.168.88.251
  - => **nmap -sU -p 53 192.168.88.251**
- Nmap TCP FIN Scan
  - => **nmap -sF -p 80 192.168.88.251**
- Nmap Host Discovery Scan
  - => **nmap -sn 192.168.88.0/24**

Types of Enumeration

- Host Enumeration
- User Enumeration

    - Example: **nmap --script smb-enum-users.nse 192.168.88.251**
- Group Enumeration
    - Example: **nmap --script smb-enum-groups.nse --script-args smbusername=vagrant,smbpass=vagrant 192.168.56.3**
- Network Share Enumeration
    - Example: **nmap --script smb-enum-shares.nse -p 445 192.168.88.251**
- Additional SMB Enumeration Examples
    - Example: **nmap -sC 192.168.88.251**
    - Confirming Scan Results in the Target System
        - **sudo samba -V**
    - Enumerating Additional Information Using enum4linux
        - **enum4linux 192.168.88.251**
    - There is a Python-based enum4linux implementation called enum4linux-ng that can be downloaded from https://github.com/cddmp/enum4linux-ng
        - Enumeration Using enum4linux-ng
            - **./enum4linux-ng.py -As 192.168.88.251**
    - Enumeration Using smbclient
        - **smbclient -L \\\192.168.88.251**
- Web Page Enumeration/Web Application Enumeration
    - **nmap -sV --script=http-enum -p 80 192.168.88.251**
    - Nikto Scan
        - **nikto -h 192.168.88.251**
- Service Enumeration
    - **nmap --script smb-enum-processes.nse --script-args smbusername=<username>, smbpass=<password> -p445 <host>**
- Exploring Enumeration via Packet Crafting
    - **Scape**
        - When it comes to enumeration via packet crafting and generation, Scapy is one of pentesters' favorite tools and frameworks. Scapy is a very comprehensive

Python-based framework or ecosystem for packet generation

- Scapy must be run with root permissions to be able to modify packets
- Starting scapy in terminal
  - **sudo scapy**
- Crafting a Simple ICMP Packet Using Scapy
  - **send(IP(dst="192.168.88.251")/ICMP()/"malicious_payload")**
- Collecting a Crafted Packet by Using tshark
  - **sudo tshark host 192.168.78.142**
- The Scapy ls() Function
  - **ls()**
- Listing the TCP Layer 4 Fields in Scapy
  - **ls(TCP)**
- Listing the Available DNS Packet Fields in Scapy
  - **ls(DNS)**
- Using the explore() Function in Scapy
  - **explore()**
- Using the explore("dns") Function to Display the Packet Types in scapy. Layers.dns
  - **explore("dns")**
- Sending a TCP SYN Packet Using Scapy
  - **ans, unans = sr(IP(dst='192.168.88.251')/TCP(dport=445,flags='S'))**
- Packet Capture of the TCP Packets on the Target Host
  - **sudo tshark host 192.168.78.142**

Packet Inspection and Eavesdropping

- As a penetration tester, you can use tools like **Wireshark**, **tshark**, and **tcpdump** to collect packet captures for packet inspection and eavesdropping

Packet Crafting with Scapy

- Scapy
    - Root privilege
        - **sudo su**
    - Run scapy
        - **scapy**
    - List all of the available default formats and protocols included with the tool
        - **ls()**
    - The syntax to use a function in Scapy is
        - **function_name(arguments)**
            - So to list the available fields in an IP packet header
                - => **ls(IP)**
    - Use the **sniff()** function
        - **sniff()**
        - Open a second terminal window and ping an internet address, such as www.cisco.com. Remember to specify the count using the -c argument.
            - **ping -c 5 www.cisco.com**
        - Return to the terminal window that is running the Scapy tool. Press **CTRL-C** to stop the capture. You should receive output similar to what is shown here:
            - **^C<Sniffed: TCP:75 UDP:42 ICMP:32 Other:2>**
        - View the captured traffic using the **summary()** function.
            - The **a=_** assigns the variable a to hold the output of the **sniff()** function. The underscore (**_**) in Python is used to temporarily hold the output of the last function executed.
            - >>> **a=_**
            - >>> **a.summary()**
- Capture and save traffic on a specific interface
    - Open a new terminal window.
    - Use the **ifconfig** command to determine the name of the interface that is assigned the IP address 10.6.6.1

    - This is the default gateway address for one of the virtual networks running inside Kali
    - Note the name of the interface
    - Return to the terminal window that is running the Scapy tool
    - Use the syntax **sniff(iface="***interface name***")** to begin the capture on the **br-internal** virtual interface
        - **sniff(iface="br-internal")**
    - Open Firefox and navigate to the URL http://10.6.6.23/
    - When the Gravemind home page opens, return to the terminal window that is running the Scapy tool
    - Press **CTRL-C**
    - You should receive output similar to:
        - **^C<Sniffed: TCP:112 UDP:0 ICMP:0 Other:2>**
    - View the captured traffic
        - >>> **a=_**
        - >>> **a.summary()**
- Examine the collected packets
    - **sniff(iface="br-internal",filter = "icmp",count = 10)**
    - Open a second terminal window and ping the host at 10.6.6.23
    - **ping –c 10 10.6.6.23**
    - Return to the terminal window running the Scapy tool
    - >>> **a=_**
    - >>> **a.nsummary()**
    - To view details about a specific packet in the series, refer to the blue line number of the packet. Do not include the leading zeros.
    - >>> **a[2]**
    - Use the wrpcap() function to save the captured data to a pcap file that can be opened by Wireshark and other applications
    - >>> **wrpcap("capture1.pcap", a)**
    - The **.pcap** file will be written to the default user directory
    - Use a different terminal window to verify the location of the capture1.pcap file using the Linux **ls** command
    - Open the capture in Wireshark to view the file contents
- Create and Send an ICMP Packet

    - In a Scapy terminal window, enter the command to sniff traffic from the interface connected to the 10.6.6.0/24 network
    - >>> **sniff(iface="br-internal")**
    - Open another terminal window, enter **sudo su** to perform packet crafting as root.
    - **sudo su**
    - Start a second instance of Scapy.
    - **scapy**
    - Enter the send() function to send a packet to 10.6.6.23 with a modified ICMP payload
    - >>> **send(IP(dst="10.6.6.23")/ICMP()/"This is a test")**
    - Return to the first terminal window and press **CTRL-C**
    - You should receive a response similar to this:
        - **^C<Sniffed: TCP:0 UDP:0 ICMP:2 Other:0>**
    - Enter the summary command to display the summary with packet numbers
    - >>> **a=_**
    - >>> **a.nsummary()**
- View and compare the ICMP packet contents
    - >>> **a[***packet number***]**
- Create and Send a TCP SYN Packet
    - Start the packet capture on the internal interface.
        - In the original Scapy terminal window, begin a packet capture on the internal interface attached to the 10.6.6.0/24 network
        - Use the interface name that you obtained previously
        - Navigate to the second terminal window
        - Create and send a TCP SYN packet using the command shown
        - >>> **send(IP(dst="10.6.6.23")/TCP(dport=445, flags="S"))**
        - This command sent an IP packet to the host with IP address 10.6.6.23. The packet is addressed to TCP port 445 and has the S (SYN) flag set
        - Close the terminal window

    - Review the captured packets
        - In the original Scapy terminal window, stop the packet capture by pressing **CTRL-C**
        - **^C<Sniffed: TCP:3 UDP:0 ICMP:0 Other:0>**
        - View the captured TCP packets using the nsummary() function
        - Display the detail of the TCP packet that was returned from the target computer at 10.6.6.23
        - >>> **a[***packet number***]**

Network Sniffing with Wireshark

- Prepare the Host to Capture Network Traffic
    - Start the virtual machine and log in
    - Verify the environment
        - **pwd**
    - **ifconfig**
    - Determine the default gateway assigned to the Kali host using the ip route command
        - **ip route**
    - Determine the address of the configured default DNS server by displaying the contents of the **/etc/resolv.conf** file
    - You can view the file using:
        - **cat /etc/resolv.conf**
- Capture and Save Network Traffic
    - Open a terminal and start tcpdump
    - **ifconfig**
    - **sudo tcpdump -i eth0 -s 0 -w packetdump.pcap**
- Generate network traffic using a web browser
    - To capture an HTTP request and reply, open a web browser in Kali desktop. Navigate to Google.com
    - Do not login or search
    - Open a second tab in the browser, enter netacad.com on the launch bar

    - Once the page appears, click the user icon at the top right of the page
    - Log in with your Skills for All login information.
    - Return to the terminal window that is running the tcpdump utility and enter **CTRL-C** to complete the packet capture
    - The tcpdump utility saved the output to a file named **packetdump.pcap**
    - This file should be saved in the default home directory
    - Verify that the file exists in the directory using the ls command
    - **ls packetdump.pcap**
- View and Analyze the Packet capture
    - **wireshark**
    - Use the **File -> Open** menu option and enter **packetdump.pcap** in the File Name field
    - Click Open
    - A screen should open displaying the contents of the packetdump.pcap file
- Analyze DNS traffic
    - Filter the captured traffic to only display DNS queries and responses
    - Enter **dns** in the Filter Field on the Wireshark main screen
    - You will notice that in addition to the **Skills for All** website that you requested, other DNS lookups are shown
    - These correspond to links contained within the Skills For All and Google homepages
- Analyze an HTTP Session
    - **ifconfig**
    - Open a browser window and enter the IP address 10.6.6.13 on the launch bar
    - A login screen for the DVWA web server appears
    - Username: **admin**
    - Password: **password**
    - Click the **Instructions** button
    - When the instructions page appears close the browser window

    - Return to the Wireshark window. Stop the capture using the **red square icon** on the menu bar
    - The web server DVWA is using HTTP, not HTTPs
    - Use the **search icon** to find the string **POST** in the captured packets
    - POST messages transfer form data from the client to the server, in this case the login information.
    - Double click the first POST packet to view the detail in a separate window
    - Expand the section titled HTML Form URL Encoded
    - Cookies are used for various purposes
    - Most frequently, they are used to save information about a user's session
    - Cookies can be hijacked and used in session hijacking attacks
    - The initial cookie for a session is sent from the web server to the client with the **Set-Cookie** value in a HTTP response
    - Use the search icon to find the string **302 Found** in the packet pane
    - Double click the first packet that was found and expand the **Hypertext Transport Protocol** section
    - Examine the next **GET** packet being sent from the Kali client browser after receiving the cookie information
    - Expand the **Hypertext Transfer Protocol** section
    - Look for the Cookie values being sent in the packet

Understanding the Art of Performing Vulnerability Scans

- Challenges to Consider When Running a Vulnerability Scan
    - Considering the best time to run a Scan
    - Determining what Protocols are in use
    - Network Topology
    - Bandwidth Limitations
    - Query Throttling
    - Fragile Systems/Nontraditional Assets

Understanding How to Analyze Vulnerability Scan Results

- Sources for Further Investigation of Vulnerabilities
  - https://www.us-cert.gov/
  - https://cert.org/
  - https://www.nist.gov/cyberframework
  - https://www.jpcert.or.jp/english/
  - Common Attack Pattern Enumeration and Classification (**CAPEC**)
  - https://cve.mitre.org/
  - https://cwe.mitre.org/
  - https://www.first.org/cvss/