Starting Point - Tier 0 Notes

1 Meow

- `ping {IP address}` is used to check if a machine can be reached. Wait for 4 - 5 pings to ensure everything is alright.
- Enumeration is the process of learning as much as possible about a machine.
- **VPN** = Virtual Private Network.
- By being on the same VPN as the target network, we gain the ability to access resources on the target network as if we were physically present on that network, having access to systems, files, and services that are only available to that network (if we have the necessary credentials, of course).

Ports and Protocols

- Every server uses **ports**, which are virtual endpoints on a server that allow different applications to communicate over a network.
- Ports have port numbers divided into:
  - **Well-known ports** (0 - 1023): reserved.
  - **Registered ports** (1024 - 49151): used for applications not universally recognized but registered with IANA.
  - **Dynamic or Private Ports** (49152 - 65535): can be used by applications for temporary connections.
- Each port is associated with a specific protocol. The two most common are **TCP** (Transmission Control Protocol) and **UDP** (User Datagram Protocol, which is faster but less secure).
- When a client wants to communicate with a server, it specifies the port number along with the server's IP address.
- Specific ports may be blocked for security reasons, such as **Telnet** (Port 23).

Enumeration and Scanning

- The first step in enumeration is to scan for open ports to find any vulnerabilities.
- **Nmap** (network mapper) is used to quickly scan for open ports.
- **Attack vectors** are the methods through which an attacker can gain unauthorized access to a computer system.
- `nmap` sends requests to target ports, hoping for replies that indicate open ports.
- Using the `-sV` flag in `nmap` provides detailed output, including the specific name and description of the service on the port. This is useful for services on unusual ports.
- Running `nmap` with `sudo` grants root access, allowing necessary permissions for actions that a standard account cannot perform.

Telnet Service

- After running `nmap`, we identified that Port 23 is open, running the **Telnet** service.
- **telnetd** is the daemon process that listens for incoming Telnet connections.
- When a client initiates a Telnet session, **telnetd** accepts the connection, creating a new session for the user and handling authentication and communication between the client and server.
- `telnet` {`IP address`} connects to a remote device via Telnet.
- Some network devices or hosts are left open with blank passwords, making them vulnerable to brute-force attacks.
- Common usernames include `admin`, `administrator`, and `root`.

Useful Commands

- `ls`: Displays directory contents.
- `cat` {`file name`}: Displays the contents of a file.
- **VM**: Virtual Machine.

Network Proximity

- Being on the same network as a machine significantly increases hacking potential, though it is not always necessary.

2 Fawn

Introduction to FTP

- **FTP** (File Transfer Protocol) is a communication protocol used to transfer files from a server to a client over a network. Although its usage has declined, more secure alternatives like **SFTP** (SSH File Transfer Protocol) and **HTTPS** are preferred today.
- FTP can also transfer log files between network devices and log collection servers.
- Logs contain information about system activities, user actions, and security events. These logs can assist hackers by:
  - Mapping out the network structure and device connections.
  - Enumerating usernames, potentially identifying user accounts and access levels.
  - Identifying services (software applications or protocols) running on different hosts.

FTP Client-Server Model

- The client-server model is fundamental to how FTP works.
- **Client**: The device that initiates requests to download or upload files.
- **Server**: The centralized device storing the data being transmitted.
- Clients can also browse available files on an FTP server.
- FTP services often come with a **GUI** (Graphical User Interface).

FTP and Ports

- Ports allow hosts to perform multiple tasks, like playing music and browsing the web simultaneously.
- **SSH** stands for Secure Shell Protocol.
- FTP is vulnerable to **MITM** (Man-in-the-Middle) attacks as data is not encrypted.
- Wrapping FTP connections with **SSL/TLS** or tunneling them through SSH adds encryption that only the sender and receiver can decrypt.

Using FTP with nmap

- `ping` may not work due to firewall rules between hosts, even within the same subnet, to prevent insider threats.
- Using `-sV` with `nmap` gives the actual version of the service running on a port, which helps identify potential vulnerabilities in outdated versions.
- `sudo apt install ftp -y` installs or updates the FTP service to the latest version.
- `ftp -?` displays FTP service capabilities.
- `ftp` {`IP address`} establishes an FTP connection with the target.

FTP Authentication and Commands

- A common misconfiguration allows an `anonymous` account to access the service like any authenticated user. Any password can be entered after typing `anonymous` as the username.
- `help` shows available commands, and `man` {`Command Name`} provides details about specific commands.

FTP Status Codes

- **Typical FTP status codes and meanings**:
  - **200**: Port command successful.
  - **150**: Directory listing starting.
  - **226**: Directory sent successfully.
- After using the `help` command, it becomes clear that the `get` command is used to download files. The file is saved in the current directory.
- `bye` stops the FTP connection.
- `ls` lists downloaded files, and `cat` {`file name`} accesses the file contents.

Additional FTP Information

- FTP usually listens on **port 21**.
- `ftp -h` displays the FTP client help menu.
- The response code for a successful FTP connection is **230**.

3 Dancing

SMB (Server Message Block)

- **SMB** (Server Message Block) is used to transfer files between two computers on the same network.
- SMB provides shared access to files, printers, and serial ports between endpoints on a network.
- **Port 445** is reserved for the SMB protocol, and it typically runs on Windows.
- SMB usually operates on the Application or Presentation layers of the **OSI (Open Systems Interconnection) model**.
  - **Presentation layer**: Translates, encrypts, and compresses data for the application layer.
  - **Application layer**: Interacts directly with the user and application software.
- The **OSI model** consists of seven layers in the following order: Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, and Application Layer.

Ports and Protocols

- **Filtered ports** are blocked by a firewall or network filter, preventing responses to connection attempts.
- The SMB protocol relies on lower-level protocols for transport since it operates on the application or presentation layer.
  - **Low-level protocols** provide basic data transport with speed and simplicity.
  - **High-level protocols** offer reliability, data flow control, and user-friendly functionality.
- The transport layer protocol most often used with Microsoft SMB Protocol is **NetBIOS over TCP/IP (NBT)**.

SMB Client Access

- Using the SMB protocol, a user can access files on a remote server along with other resources such as printers, allowing read, create, and update capabilities.
- The storage device or resource (share) is accessible to multiple users or devices on the network.

Smbclient Command

- `smbclient` allows users to interact with and manage SMB shares from the command line.
- Use `sudo apt-get install smbclient` to install `smbclient`.
- `smbclient` will attempt to connect to the remote host and check for any required authentication. If no username is specified, it uses the local machine's username.
- `smbclient -L {IP address}` lists available SMB shares on a specific server.

Administrative Shares

- **ADMIN$**: Hidden network share allowing administrators remote access to all disk volumes on a network-connected system. It cannot be permanently deleted but can be disabled.
- **C$**: An administrative share hosting the OS (operating system).
- **IPC$** (Inter-process Communication share): Used for communication via named pipes, not part of the file system, and contains no browseable files.
- The `$` symbol indicates a hidden share, meaning it is not visible when users browse for shared folders on the network.
- `smbclient \\\\{IP address}\\ADMIN$` attempts to connect to the ADMIN$ share.

Useful Commands

- `get`: Downloads files (e.g., `flag.txt`).
- `cd`: Navigates into directories (e.g., `Amy.J` directory).
- `cd ..`: used to go out of the Amy.J directory to its parent directory that you were in before digging deeper into the child directory Amy.J.

4 Redeemer

Introduction to Databases

- **Databases** are collections of organized information that can be easily read and manipulated.
- Databases often contain critical information, such as sales transactions, customer profiles, and marketing activities, requiring high security.

In-Memory Databases and Redis

- **Redis** is an example of an in-memory database.
- **In-memory databases** are managed by the system's RAM, offering faster data retrieval than secondary storage.

Types of Primary Memory

- Types of primary memory include **RAM**, **ROM**, **Cache**, and **Registers**.
  - **RAM** (Random Access Memory): The most common type of primary memory, temporarily storing data and instructions currently in use by the CPU.
  - **ROM** (Read-Only Memory): Non-volatile memory that retains data even when the computer is powered off.
  - **Cache Memory**: High-speed, volatile memory that stores frequently accessed data and instructions.
  - **Registers**: The smallest and fastest type of memory, holding data currently being processed by the CPU.

Types of Secondary Memory

- Types of secondary memory include **HDDs** (Hard Disk Drives), **SSDs** (Solid State Drives), and **Cloud Storage**.
  - **HDDs**: Mechanical drives that store data on spinning magnetic disks, offering large storage capacities.
  - **SSDs**: Use flash memory to store data, making them faster and more efficient than HDDs due to the lack of moving parts.
  - **Cloud Storage**: Data stored on remote servers accessed via the internet.
- In-memory databases like **Redis** are typically used to cache frequently requested data for quick retrieval.
- **Redis** (Remote Dictionary Server) is an open-source, advanced NoSQL key-value data store used as a database, cache, and message broker.
  - **Open-source** means the software's source code is freely available.
  - **NoSQL** databases do not use traditional relational structures (e.g., tables and rows); instead, they use models like key-value structures.
  - Example: A key could be `"username"`, and the value would be `"Zaid"`.
- Redis typically runs on **port 6379** by default.

Assignment Instructions: Nmap Usage

- In this assignment, we are instructed to use `nmap -p- -sV {IP address}`.
- By using the `-p-` option, nmap scans all 65,535 TCP ports on the target, not just the default ports.
- **Sudo** is not used, as it primarily aids in scanning well-known ports (0-1023).

Redis Usage and Commands

- Redis is typically used for short-term storage of data that needs fast retrieval.
- Redis does backup data to hard drives as a safety measure.
- Operating as a server-side application, Redis runs on a server and provides services to clients.
- Redis stores its database in **RAM**, allowing for extremely fast data access and manipulation.
- Use `sudo apt install redis-tools` to interact remotely with the Redis server. Alternatively, the `netact` utility can also be used.

Redis CLI Commands

- Access Redis help page with `redis-cli --help`.
- Use `redis-cli -h {IP address}` to specify the host for connection.
- After connecting, use the `info` command to retrieve information and statistics about the Redis server.
- In the **Keyspace section**, note:
  - `db0`: Indicates that we are in Redis's default database.
  - `keys=4`: Shows there are 4 keys currently stored in `db0`.
  - `expires=0`: Indicates that none of the keys have expiration times set.
  - `avg_ttl=0`: Shows the average time-to-live (TTL) for keys in `db0` is 0, meaning no TTL is set.
- To select a different database, use `select {database number}`.
- Use `keys *` to list all keys in the database.
- In Redis, retrieve the content or check the value of a specific key using `get {key}`. (Note: This differs from prior use of `get` for downloading files.)

5 Explosion

- **Remote access software** allows users to connect to and control a computer or network from a different location over the internet.
- Remote access software enables users to access files, applications, and settings as if they were physically present.
- Interactions using remote access software can be conducted via:
  - **CLI** (Command Line Interface)
  - **GUI** (Graphical User Interface)
- Remote access tools typically use **RDP** (Remote Desktop Protocol) to communicate with other hosts.
  - **RDP** operates on ports 3389 TCP and 3389 UDP.

CLI-Based Remote Access: Telnet and SSH

- A type of CLI-based remote access tool is **Telnet**, which was discussed in the Meow machine.
- Telnet is considered insecure because data transmitted through it is not encrypted.
- **Telnet** (running on port 23) has largely been replaced by **SSH** (Secure Shell Protocol), which provides enhanced security.
- **SSH** adds authentication and encryption layers to the communication model and is used for tasks like patch delivery, file transfers, log transfer, and remote management.

SSH and Public Key Cryptography

- SSH uses **public key cryptography** to verify the remote host's identity.
- **Public key cryptography** verifies a host's identity using digital certificates and digital signatures, relying on a pair of cryptographic keys: the public key and the private key.
- Key components:
  - **Digital Certificate**: A trusted authority, known as a Certificate Authority (CA), issues a digital certificate to the host. This certificate includes the host's public key and is digitally signed by the CA to verify the host's identity.
  - **Public and Private Key Pair**: The host has a public key (available to anyone) and a private key (kept secret). These keys are mathematically linked.

- **Authentication Process**:
  - When a client connects to the host (e.g., a secure website), the host provides its digital certificate.
  - The client verifies the certificate's authenticity by checking the CA's digital signature. If valid, it confirms the host's identity and that the public key genuinely belongs to it.
- **Establishing Trust**: Once the client trusts the host's public key, the host can use its private key to sign data. The client can verify this signature with the public key, confirming that the data sent by the host is authentic and untampered, thereby verifying the host's identity.

Symmetric Encryption in SSH Tunnels

- Once the tunnel is established using public key cryptography, **symmetric encryption methods** and **hashing algorithms** are used to ensure the confidentiality and integrity of data transmitted over the tunnel.

How Symmetric Encryption Ensures Data Confidentiality and Integrity

- **Data Encryption**: The sender uses a secret key to transform plaintext data into unreadable ciphertext.
  - This encryption process ensures that data remains unintelligible to anyone without the same secret key, preserving confidentiality.
- **Data Transmission**: The encrypted data (ciphertext) is sent over a potentially insecure network.
  - Even if intercepted, the data remains unreadable without the secret key, ensuring security.
- **Data Decryption**: The receiver, who also has the shared secret key, decrypts the ciphertext back into readable plaintext.
  - Since only those with the correct key can decrypt the data, confidentiality is maintained.
- **Single Shared Key**: Both parties must securely share and protect the secret key, as access to this key would allow anyone to decrypt the data.
- **Speed and Efficiency**: Symmetric encryption algorithms (e.g., AES, DES) are fast and efficient, making them ideal for encrypting large amounts of data.

Using xfreerdp for RDP Sessions

- `xfreerdp` enables users to establish an RDP session to a Windows machine from a Linux or Unix system.
- Start with `sudo nmap -sV {IP address}`, which will reveal multiple open ports.
- It is good practice to use **SpeedGuide** as a resource to research open ports to understand the broader context. Including notes on each open port is helpful.

Common Open Ports and Their Functions

- **Port 135** is used for **RPC** (Remote Procedure Call), which facilitates client/server communications in applications like Windows services.
  - RPC is essential for some remote access but has security risks.
  - RPC has a **DoS (Denial of Service) exploit** that can crash systems through malformed requests.
- **Port 139** is used for **NetBIOS** (Network Basic Input/Output System), which also operates on ports 137 and 138.
  - NetBIOS is used in Windows for file and print sharing.
  - When enabled, it may expose shared resources over the internet if not properly configured.

xfreerdp Installation and Usage

- Install `xfreerdp` using `sudo apt-get install freerdp2-x11`.
- If no username and password are required, connect using `xfreerdp /v:{IP address}`, which defaults to the local username.
- `xfreerdp /v:{IP address} /cert:ignore /u:Administrator` can also be used, where:
  - `/cert:ignore` bypasses SSL certificate validation errors, useful for servers with untrusted or self-signed certificates.
  - `/u:Administrator` specifies the username (in this case, `Administrator`) for authentication.

6 Preignition

- **WordPress** is a popular open-source CMS (Content Management System) used to create websites and blogs, making it easier to manage content.

Dirbusting (Directory Busting)

- Dirbusting is a web security technique used to discover hidden directories, files, and resources on a web server.
- It uses a wordlist of potential directory and file names, systematically attempting to access each one on the target server.
- Tools for dirbusting, such as Gobuster, send HTTP requests for each directory and file in the wordlist. If the server responds with a valid status code (like 200 OK), that path is potentially available.
- **Port 80** is the default port for HTTP, enabling web traffic.
- Typing the IP address as a URL in a browser allows you to view accessible web content.

Gobuster and Directory Busting

- The web page may appear new, suggesting the possibility of incomplete configuration or default credentials.
- Instead of guessing random URLs, dirbusting can efficiently discover hidden pages.
- We use **Gobuster**, a tool written in Go, for this purpose.
  - Install Go with `sudo apt install golang-go`.
  - Install Gobuster with `sudo apt install gobuster`.
  - Access Gobuster's options with `gobuster --help`.

Gobuster Command Options

- `dir` : Specifies directory busting mode.
- `-w` : Specifies a wordlist, containing common directory names for web discovery.
- `-u` : Specifies the target's IP address.

Preparing and Using Wordlists for Directory Busting

- Before using a specific wordlist, make sure to download it.
- If you encounter a "Permission Denied" error when downloading, add `sudo` at the beginning of the command to gain elevated privileges.
- If the intended wordlist is unavailable, use this alternative:
  - `sudo wget https://github.com/danielmiessler/SecLists/raw/master/Discovery/Web-Conte`
    `-O /usr/share/wordlists/common.txt`

- **Explanation of Command Components**:
  - `wget` is a Linux utility used for downloading files from the internet via HTTP, HTTPS, and FTP protocols.
  - `-O` specifies the output filename and location for the downloaded file.
  - `common.txt` is the filename, and `/usr/share/wordlists/` is the directory where the file will be saved.
- For more resources on web application attacks, especially with **Ffuf**, see the Hack The Box Academy Ffuf Module.

Using Gobuster for Directory Busting

- When using Gobuster to perform directory busting, the `-x php` option can be added to include .php pages in the search.

7 Mongod

- **Databases** are collections of organized information that can be easily accessed and manipulated.
- **Mongod** is a type of **NoSQL** database.

Data Hierarchy in NoSQL Databases

- In NoSQL databases, data is organized in a hierarchical structure:
  - **Databases**: The top-level storage unit. For a product, there may be separate databases for product information, customers, orders, and inventory.
  - **Collection**: Collections serve as containers for documents. Multiple documents are stored in a single collection.
  - **Documents**: Individual records within collections, similar to files like `.pdf` or `.doc`.

Connecting to MongoDB Servers with `mongosh`

- MongoDB servers can be accessed using the `mongosh` command-line utility.
- To connect, start by scanning the server to identify open ports and services.

Using Nmap to Scan for MongoDB Services

- Command: `nmap -p- --min-rate=1000 -sV {target_IP}`
- Explanation:
  - `-p-` scans all ports.
  - `--min-rate=1000` specifies a minimum rate of 1000 packets, enabling faster scanning and maintaining a constant speed without automatic rate adjustment, which could slow down the process.
  - `-sV` detects the version of the service running on the target.
- **Port 27017** is typically used by MongoDB servers.

Installing and Using the MongoDB Shell

- Download the MongoDB shell utility using `curl`:
  - `curl -O https://downloads.mongodb.com/compass/mongosh-2.3.2-linux-x64.tgz`
- Extract the contents with `tar`:
  - `tar xvf mongosh-2.3.2-linux-x64.tgz`

Basic Commands in MongoDB

- To display all databases on the server, use: `show dbs;`
- To access a specific database, use: `use {database name};`
- To list collections within a database, use: `show collections;`
- To view contents of a specific collection, use: `db.{collection name}.find();`

8 Synced

- **Rsync** (remote synchronization) is used when efficient data transfer is needed, such as transferring only changes (deltas) rather than entire files.

Rsync Process

1. Rsync establishes a connection to the remote host and starts an rsync receiver process.
2. The sender and receiver processes compare files to identify changes.
3. The changed files are updated on the remote host.

Rsync Details

- **Port**: Rsync typically runs on port 873.
- **Installation**: Rsync is pre-installed on most Linux distributions.

Rsync Command Syntax

- General syntax: `rsync [OPTION] ... [USER@]HOST::SRC [DEST]`
  - `SRC`: The source file or directory to copy from.
  - `DEST`: The destination file or directory to copy to.
  - `OPTION`: The rsync option used in the command.
  - `USER@` (optional): Used for authenticated access to a remote machine.

Rsync Usage Examples

- To list all available files: `rsync --list-only {IP address}::`
- To list documents in a public share: `rsync --list-only {IP address}::{document name}`
- To copy a file from the public share:
  - Command: `rsync {target_IP}::public/flag.txt flag.txt`
  - Here, `SRC` is `public/flag.txt` and `DEST` is `flag.txt`.
- Use `cat {file name}` to display the contents of the copied file.