Module 3: Information Gathering and Vulnerability Scanning

Ethical Hacker

Module Objectives

Module Title: Information Gathering and Vulnerability Scanning

Module Objective: Perform information gathering and vulnerability scanning activities.

Topic Title	Topic Objective
Performing Passive Reconnaissance	Perform passive reconnaissance activities.
Performing Active Reconnaissance	Perform active reconnaissance activities.
Understanding the Art of Performing Vulnerability Scans	Perform vulnerability scans.
Understanding How to Analyze Vulnerability Scan Results	Analyze the results of reconnaissance exercises.

3.1 Performing Passive Reconnaissance

Overview

- Reconnaissance is always the initial step in a cyber attack.
- An attacker must first gather information about the target in order to be successful.
- The term **reconnaissance** is widely used in the military world to describe the gathering of information about the enemy, such as information about the enemy's location, capabilities, and movements.
- This type of information is needed to successfully perform an attack.
- Reconnaissance in a penetration testing engagement typically consists of scanning and enumeration.
- But what does reconnaissance look like from an attacker's perspective?

Active Reconnaissance vs. Passive Reconnaissance

- **Active reconnaissance** is a method of information gathering in which the tools used send out probes to the target network or systems in order to elicit responses that are then used to determine the posture of the network or system.
  - These probes can use various protocols and multiple levels of aggressiveness, typically based on what is being scanned and when.

- **Passive reconnaissance** is a method of information gathering in which the tools do not interact directly with the target device or network.
  - There are multiple methods of passive reconnaissance.
  - Some involve using third-party databases to gather information.
  - Others also use tools in such a way that they will not be detected by the target.
  - These tools work by simply listening to the traffic on the network and using intelligence to deduce information about the device communication on the network.

- One of the most important aspects of learning about penetration testing is developing a good methodology that will help you select the appropriate tools and technologies to use during the engagement.

Active Reconnaissance vs. Passive Reconnaissance (Cont.)

- 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

Lab - Using OSINT Tools

- In this lab, you will explore several OSINT tools that are commonly used by pentesters.
    - Examine OSINT resources.
    - Use SpiderFoot
    - Investigate Recon-ng
    - Find interesting files with Recon-ng

DNS Lookups

- Suppose, for example, that an attacker has a target, h4cker.org, in its sights.
- h4cker.org has an Internet presence, as most companies do.
- This presence is a website hosted at www.h4cker.org.
- Just as a home burglar would need to determine which entry and exit points exist in a home before he could commit a robbery, a cyber attacker needs to determine which of the target's ports and protocols are exposed to the Internet.
- A burglar might take a walk around the outside of the house, looking for doors and windows, and then possibly take a look at the locks on the doors to determine their weaknesses.
- Similarly, a cyber attacker would perform tasks like scanning and enumeration.
- Typically, an attacker would start with a small amount of information and gather more information while scanning, eventually moving on to performing different types of scans and gathering additional information.

DNS Lookups (Cont.)

- For instance, the attacker targeting h4cker.org might start by using ***DNS lookups*** to determine the IP address or addresses used by h4cker.org and any other subdomains that might be in use.
- Let's say that those queries reveal that h4cker.org is using the IP addresses 185.199.108.153 for www.h4cker.org, 185.199.110.153 for mail.h4cker.org, and 185.199.110.153 for portal.h4cker.org.
- The example shows an example of the DNSRecon tool in Kali Linux being used to query the DNS records for h4cker.org (output omitted for brevity).

```
|--[omar@websploit]--[~]
|--- $ dnsrecon -d h4cker.org
[*] Performing General Enumeration of Domain: h4cker.org
/usr/share/dnsrecon/./dnsrecon.py:816: DeprecationWarning: please use
dns.resolver.Resolver.resolve() instead
  answer = res._res.query(domain, 'DNSKEY')
[*] DNSSEC is configured for h4cker.org
[*] DNSKEYs:
[*] NSEC3 ZSK RSASHA256 030100019ed0af43a7dc09d07e1646d2 b4036075e9187c4c563519155f888b60
8fdffe9c6d8a0a01522f78d25d257772 0a8e97d1350e694b272ec63af9708609 b3721e6b53a2d7aa8839585714800319
dd98f97b39d8768f7e975a449c001ce9 55189ea83f30a4fe6b4dff7b3dd15f89 1cef3a8d84968a980bde65c0b1309d5b 825a0f23
[*] NSEC3 KSk RSASHA256 03010001840 3e0971df0dc1770f3b96a ca57eb68d03a84b4a712cadda60567fe
a264f0e5d7ec4c8e0187300f0933f419 d22a17548c3a046636666300c06711f0 761200245149a220b79918b3f38a9a6e
8228425cb39b6466adba9f6f7fe28d76 c1bcf44e19f035f658eef65cb630638f 7aa15d7706cc572c863d65619bd48f77
425ea0844716709b9923117ade41d414 c94f8e581db9274cf1c8bb41fbbd7838 24978c0f9b7125b9ce3e8abe442a6bc7
4bf519790a18a27916c946f503c02b08 0a8550bc5b9b147d581a3f5f763df377 9e1d655c51c2e06aa2062d1f08f34abc
37947ac48403dc0da9af846c7a4caeae 7567bb8fdf625b1a179e6fd6faf35be9 09488cb9
[*]   SOA ns-cloud-c1.googledomains.com 216.239.32.108
[*]   NS ns-cloud-c1.googledomains.com 216.239.32.108
[*]   NS ns-cloud-c1.googledomains.com 2001:4860:4802:32::6c
[*]   NS ns-cloud-c2.googledomains.com 216.239.34.108
[*]   NS ns-cloud-c2.googledomains.com 2001:4860:4802:34::6c
[*]   NS ns-cloud-c3.googledomains.com 216.239.36.108
[*]   NS ns-cloud-c3.googledomains.com 2001:4860:4802:36::6c
[*]   NS ns-cloud-c4.googledomains.com 216.239.38.108
[*]   NS ns-cloud-c4.googledomains.com 2001:4860:4802:38::6c
[*]   MX aspmx.l.google.com 173.194.206.27
[*]   MX alt1.aspmx.l.google.com 64.233.186.27
```

DNS Lookups (Cont.)

- From there, an attacker can begin to dig deeper by scanning the identified hosts.
- Once the attacker knows which hosts are alive on the target site, he or she then needs to determine what kind of services the hosts are running.
- To do this, the attacker might use the tried-and-true Nmap tool.
- Before we discuss this tool and others in depth, we need to look at the types of scans and enumerations you should perform and why.

- 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.

- The example shows how the Dig tool is used to show the DNS resolution details for h4cker.org.

- The highlighted lines show the IP addresses associated with h4cker.org.

```
|--[omar@websploit]--[~]
|--- $dig h4cker.org
; <<>> DiG 9.16.6-Debian <<>> h4cker.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6517
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;h4cker.org.                IN      A
;; ANSWER SECTION:
h4cker.org.             172     IN      A       185.199.110.153
h4cker.org.             172     IN      A       185.199.111.153
h4cker.org.             172     IN      A       185.199.108.153
h4cker.org.             172     IN      A       185.199.109.153
;; Query time: 72 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Fri Apr 30 20:45:42 EDT 2021
;; MSG SIZE  rcvd: 103

|--[omar@websploit]--[~]
|--- $
```

DNS Lookups (Cont.)

- Similarly, you can use the **dig** < domain > **mx** command to obtain the email servers used by h4cker.org (mail exchanger [MX] record), as demonstrated in the example.

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

; <<>> DiG 9.16.6-Debian <<>> h4cker.org mx
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62903
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;h4cker.org.                IN      MX

;; ANSWER SECTION:
h4cker.org.    77      IN      MX      1  aspmx.l.google.com.
h4cker.org.    77      IN      MX      5  alt1.aspmx.l.google.com.
h4cker.org.    77      IN      MX      5  alt2.aspmx.l.google.com.
h4cker.org.    77      IN      MX      10 alt3.aspmx.l.google.com.
h4cker.org.    77      IN      MX      10 alt4.aspmx.l.google.com.

;; Query time: 48 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Fri Apr 30 20:47:01 EDT 2021
;; MSG SIZE  rcvd: 157
```

Identification of Technical and Administrative Contacts

- You can easily identify domain technical and administrative contacts by using the Whois tool.
- Many organizations keep their registration details private and instead use the domain registrar organization contacts.
- Let's look at the technical and administrative contacts of h4cker.org shown in the example.
  - Omar Santos owns the h4cker.org domain; however, the technical and administrative details are private.
  - Only the abuse contact email and phone number from Google (the domain registrar) are displayed.

```
|--[omar@websploit]--[~]
|--- $ whois h4cker.org
Domain Name: H4CKER.ORG
Registry Domain ID: D402200000006011258-LROR
Registrar WHOIS Server: whois.google.com
Registrar URL: https://domains.google.com
Updated Date: 2018-07-03T03:48:35Z
Creation Date: 2018-05-04T03:43:52Z
Registry Expiry Date: 2028-05-04T03:43:52Z
Registrar Registration Expiration Date:
Registrar: Google LLC
Registrar IANA ID: 895
Registrar Abuse Contact Email: registrar-abuse@google.com
Registrar Abuse Contact Phone: +1.8772376466
Reseller:
Domain Status: ok https://icann.org/epp#ok
Registrant Organization: Contact Privacy Inc. Customer 1242605855
Registrant State/Province: ON
Registrant Country: CA
Name Server: NS-CLOUD-C1.GOOGLEDOMAINS.COM
Name Server: NS-CLOUD-C2.GOOGLEDOMAINS.COM
Name Server: NS-CLOUD-C4.GOOGLEDOMAINS.COM
Name Server: NS-CLOUD-C3.GOOGLEDOMAINS.COM
DNSSEC: signedDelegation
URL of the ICANN Whois Inaccuracy Complaint Form https://www.icann.org/wicf/)
```

Identification of Technical and Administrative Contacts (Cont.)

- Now let's look at the Whois details for tesla.com, shown in the example (output omitted for brevity).
- The highlighted lines in show the technical and administrative contacts for the domain (which are also the ones for the domain registrar [MarkMonitor]).

```
|--[omar@websploit]--[~]
|--- $whois tesla.com
   Domain Name: TESLA.COM
   Registry Domain ID: 187902_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: serverUpdateProhibited
https://icann.org/epp#serverUpdateProhibited
   Name Server: A1-12.AKAM.NET
   Name Server: A10-67.AKAM.NET
   Name Server: A12-64.AKAM.NET
   Name Server: A28-65.AKAM.NET
   Name Server: A7-66.AKAM.NET
   Name Server: A9-67.AKAM.NET
   Name Server: EDNS69.ULTRADNS.BIZ
   Name Server: EDNS69.ULTRADNS.COM
   Name Server: EDNS69.ULTRADNS.NET
   Name Server: EDNS69.ULTRADNS.ORG
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form:
https://www.icann.org/wicf/
<output omitted for brevity>
```

Identification of Technical and Administrative Contacts (Cont.)

- This example shows the technical and administrative email contacts for the domain cisco.com.
- The technical and administrative contacts are pointing to the InfoSec team at Cisco (infosec@cisco.com) instead of to the registrant.

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

Lab - DNS Lookups

- In this lab, you will explore common tools used to gather information about a target through the Domain Name System (DNS).
    - Use nslookup to obtain domain and IP address information.
    - Use the whois command to find additional registration information.
    - Compare the Output of the Nslookup and Dig tools.
    - Perform Reverse DNS Lookups.

Cloud vs. Self-Hosted Applications and Related Subdomains

- A company can own a domain and related subdomain, but its applications might be hosted in the cloud.
- For example, Netflix (at the time of writing) owns the domain netflix.com, which resolves to IPv4 addresses 3.230.129.93, 52.3.144.142, and 54.237.226.164 (as demonstrated with the Linux **host** command).(output omitted for brevity).
- However, the IPv4 addresses 3.230.129.93, 52.3.144.142, and 54.237.226.164 are owned by Amazon Web Services (AWS), which hosts Netflix.com
- In this example, the **whois** command is used to retrieve the organization name (OrgName) of the owner for each of the IP addresses 3.230.129.93, 52.3.144.142, and 54.237.226.164.

```
|--[omar@websploit]--[~]
|--- $host netflix.com
netflix.com has address 3.230.129.93
netflix.com has address 52.3.144.142
netflix.com has address 54.237.226.164
```

```
|--[omar@websploit]--[~]
|--- $whois 3.230.129.93 | grep OrgName
OrgName:        Amazon Technologies Inc.
OrgName:        Amazon Data Services NoVa
|--[omar@websploit]--[~]
|--- $whois 52.3.144.142 | grep OrgName
OrgName:        Amazon Technologies Inc.
|--[omar@websploit]--[~]
|--- $whois 54.237.226.164 | grep OrgName
OrgName:        Amazon Technologies Inc.
OrgName:        Amazon.com, Inc.
```

Social Media Scraping

- Attackers can easily gather valuable information about victims by scraping social media sites such as Twitter, LinkedIn, Facebook, and Instagram.
- Often attackers use other information such as key contacts (company stakeholders) and their job responsibilities.
- Attackers can use all that information to perform different types of social engineering attacks (including spear phishing and whaling).
- Attackers often leverage job listings in websites like Indeed, LinkedIn, CareerBuilder, and individual company websites to obtain information about the technologies these companies use.
- Attackers have also created job posts to attract people to apply for those positions.
- Then they interview their victims to try to get them to talk about what they do at work and the technologies used by their employer.

Lab - Employee Intelligence Gathering

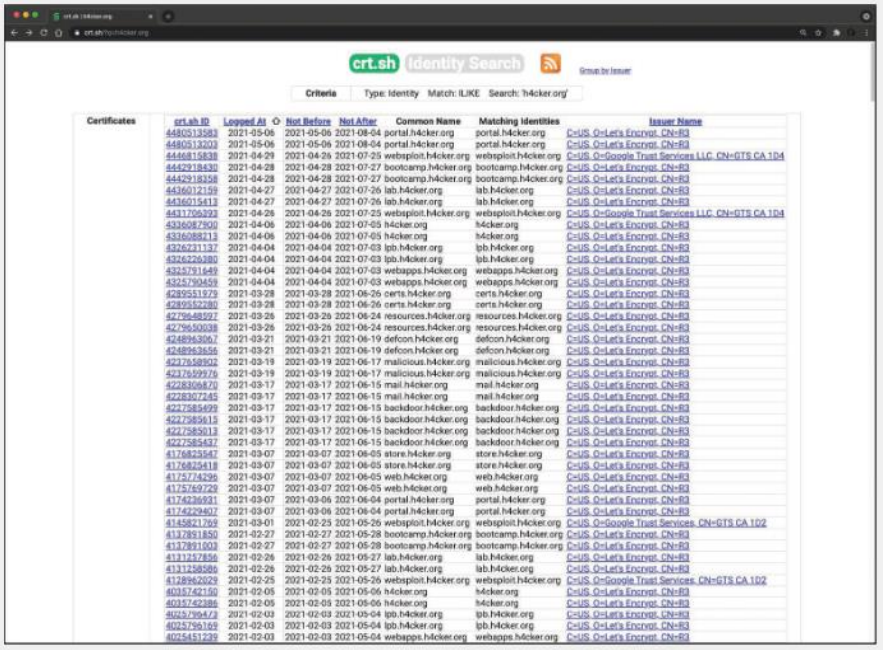
In this lab, you will use social media to collect personal identifiable information (PII).

Cryptographic Flaws

- During the reconnaissance phase, attackers often can inspect Secure Sockets Layer (SSL) certificates to obtain information about the organization, potential cryptographic flaws, and weak implementations.
- You can find a lot inside digital certificates: the certificate serial number, the subject common name, the uniform resource identifier (URI) of the server it was assigned to, the organization name, Online Certificate Status Protocol (OCSP) information, the certificate revocation list (CRL) URI, and so on.
- Attackers can also leverage certificate transparency to reveal additional information and enumerate subdomains.
- The goal of certificate transparency is for any organization or individual to be able to “transparently” verify the issuance of a digital certificate.
- Certificate transparency allows certificate authorities (CAs) to provide details about all certificates that have been issued for a given domain and organization.
- Attackers can also use this information to reveal what other subdomains and systems an organization may own!

Cryptographic Flaws (Cont.)

- Tools such as crt.sh enable you to obtain detailed certificate transparency information about any given domain.
- The figure shows the result of the query https://crt.sh/?q=h4cker.org in crt.sh for the domain h4cker.org.
- You can see in the search results multiple subdomains that were not known to the attacker before.



Lab - Finding Information from SSL Certificates

- In this lab, you will complete the following objectives: :
  - View Certificate Information on Hosts
  - Access Detailed Certificate Information
  - Use SSL Analysis Tools in Kali
  - Use Kali Tools to Gather Certificate Information

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
  - File metadata
  - Strategic search engine analysis/enumeration
  - Website archiving/caching
  - Public source code repositories

Company Reputation and Security Posture (Cont.)

Password Dumps

- Attackers can leverage password dumps from previous breaches.
- There are several 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**.
- You can install h8mail by using the **pip3 install h8mail** command, as demonstrated in the example.
- It also shows the h8mail command-line usage.

```
root@websploit# pip3 install h8mail
Collecting h8mail
  Downloading h8mail-2.5.5-py3-none-any.whl (33 kB)
Requirement already satisfied: requests in /usr/lib/python3/
dist-packages (from h8mail) (2.23.0)
Installing collected packages: h8mail
Successfully installed h8mail-2.5.5
root@websploit# h8mail -h
usage: h8mail [-h] [-t USER_TARGETS [USER_TARGETS ...]] [-u USER_URLS [USER_URLS ...]] [-q USER_QUERY] [--
loose] [-c CONFIG_FILE [CONFIG_FILE ...]] [-o OUTPUT_FILE] [-j OUTPUT_JSON] [-bc BC_PATH] [-sk] [-k
CLI_APIKEYS [CLI_APIKEYS ...]]
            [-lb LOCAL_BREACH_SRC [LOCAL_BREACH_SRC ...]] [-gz LOCAL_GZIP_SRC [LOCAL_GZIP_SRC ...]] [-sf]
[-ch [CHASE_LIMIT]] [--power-chase] [--hide] [--debug] [--gen-config]

Email information and password lookup tool

options:
  -h, --help            show this help message and exit
  -t USER_TARGETS [USER_TARGETS ...], --targets USER_TARGETS [USER_TARGETS ...]
                        Either string inputs or files. Supports email pattern matching from input or file,
filepath globing and multiple arguments
  -u USER_URLS [USER_URLS ...], --url USER_URLS [USER_URLS ...]
                        Either string inputs or files. Supports URL pattern matching from input or file,
filepath globing and multiple arguments. Parse URLs page for emails. Requires http:// or https:// in URL.
  -q USER_QUERY, --custom-query USER_QUERY
                        Perform a custom query. Supports username, password, ip, hash, domain. Performs an
implicit "loose" search when searching locally
```

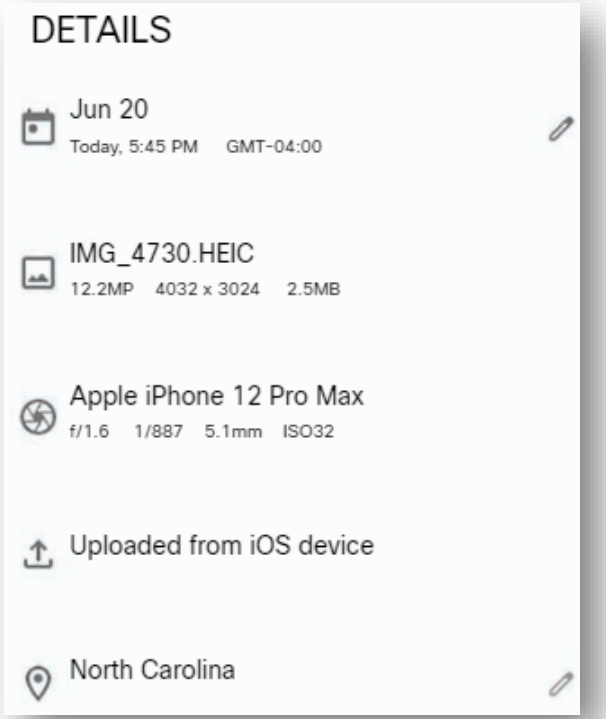
Company Reputation and Security Posture (Cont.)

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.

- The figure shows the Exif data of a digital image (picture) captured by an iPhone.



Company Reputation and Security Posture (Cont.)

- Several tools can show Exif details.
- One of the most popular of them, ExifTool, is demonstrated in this example (output omitted for brevity).
- It shows the Exif metadata details of the same image whose details are shown in the previous figure.

```
|--[omar@websploit]--[~]
|--- $exiftool IMG_4730.jpg
ExifTool Version Number                 : 12.06
File Name                               : IMG_4730.jpg
Directory                               : .
File Size                               : 2.4 MB
File Modification Date/Time             : 2021:06:20 21:33:36-04:00
File Access Date/Time                   : 2021:06:20 21:33:36-04:00
File Inode Change Date/Time             : 2021:06:20 21:33:36-04:00
File Permissions                        : rw-r--r--
File Type                               : JPEG
File Type Extension                     : jpg
MIME Type                               : image/jpeg
JFIF Version                            : 1.01
Exif Byte Order                         : Big-endian (Motorola, MM)
Make                                    : Apple
Camera Model Name                       : iPhone 12 Pro Max
Orientation                             : Horizontal (normal)
X Resolution                            : 72
Y Resolution                            : 72
Resolution Unit                         : inches
Software                                : 14.6
Modify Date                             : 2021:06:20 17:45:44
Host Computer                           : iPhone 12 Pro Max
Tile Width                              : 512
Tile Length                             : 512
```

Company Reputation and Security Posture (Cont.)

Strategic Search Engine Analysis/Enumeration

- Most of us use search engines such as DuckDuckGo, Bing, and Google to locate information.
- What you might not know is that search engines, such as Google, can perform much more powerful searches than most people ever dream of.
- Google can translate documents, perform news searches, and do image searches.
- In addition, hackers and attackers can use it to do something that has been termed *Google hacking*.
- 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:** Directs Google to search only within the text of a particular type of file (for example, filetype:xls)
  - **Inurl:** Directs Google to search only within the specified URL of a document (for example, inurl:search-text)
  - **Link:** Directs Google to search within hyperlinks for a specific term (for example, link:www.domain.com)
  - **Intitle:** Directs Google to search for a term within the title of a document (for example, intitle: “Index of /etc”)

Company Reputation and Security Posture (Cont.)

- By using these advanced operators in combination with key terms, both you and attackers can get Google to uncover many pieces of sensitive information that shouldn't be revealed.
- These search strings are often called *Google dorks*.

- To see how Google dorking works, enter the following phrase into Google:
  intext:JSESSIONID OR intext:PHPSESSID inurl:access.log ext:log

- This query searches in a URL for the session IDs that could be used to potentially impersonate users.
- It's not unusual for this search to find more than 100 sites that store sensitive session IDs in logs that were publicly accessible.
- If these IDs have not timed out, they could be used to gain access to restricted resources.
- You can use advanced operators to search for many types of data.

- Another example of a Google search string (or Google dork) that can reveal passwords of web applications:
  "public $user =" | "public $password = " | "public $secret =" | "public $db =" ext:txt | ext:log -git

Company Reputation and Security Posture (Cont.)

- To take a look at advanced Google hacking visit the Google Hacking Database (GHDB) repositories.
- Some of the following GHDB search categories are:
  - Footholds
  - Files containing usernames
  - Sensitive directories
  - Web server detection
  - Vulnerable files
  - Vulnerable servers
  - Error message

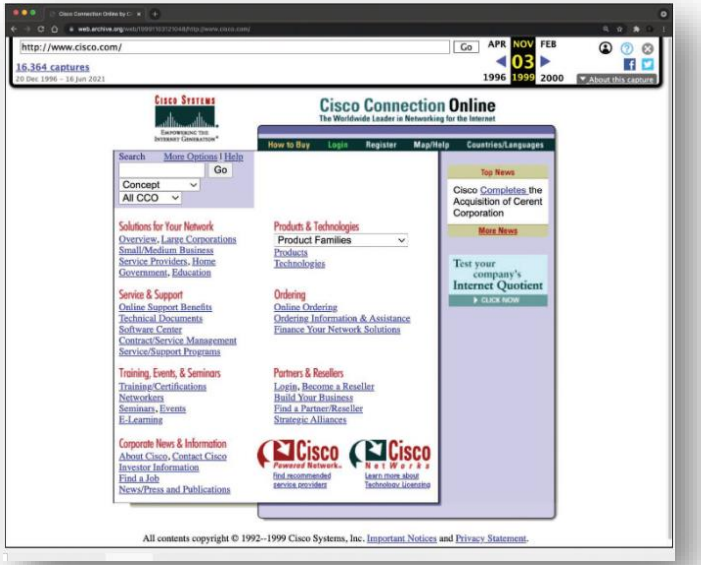
- GHDB is a community effort.
- Anyone can upload a new Google dork to perform these types of searches.
- Once you start playing with the dorks in GHDB, you will be surprised by the unbelievable things found through Google hacking.
- GHDB has made using Google dorks very easy, and there are other options as well.

Company Reputation and Security Posture (Cont.)

Website Archiving/Caching

- Several organizations archive and cache website data on the Internet.
- One of the most popular repositories is the “Wayback Machine” of Internet Archive.
- The Wayback Machine allows you to go back in time on the Internet.
- For example, the figure shows what Cisco’s website looked like on November 3, 1999.
- You can access the archive of the site shown by copying the following link into your browser address bar:

https://web.archive.org/web/19991103121048/http://www.cisco.com/



Company Reputation and Security Posture (Cont.)

Public Source Code Repositories

- An attacker can obtain extremely valuable information from public source code repositories such as GitHub and GitLab.
- Most of the applications and products we consume today use open-source software that is freely available in these public repositories.
- Attackers can find vulnerabilities in those software packages and use them to their advantage.
- Similarly, as a penetration tester, you can obtain valuable information from these public repositories.
- Even if you do not immediately find security vulnerabilities in the code, these repositories can give you insights into the architecture and underlying code used in the organization's applications and infrastructure.

Lab - Finding Out About the Organization

- In this lab, you will complete the following objectives:
  - Find information about email breaches.
  - View file metadata.

3.1.19 Lab - Advanced Searches

- In this lab, you will use Google Advanced Search to perform passive reconnaissance.
    - Part 1: Google Advanced Searches (Dorking)
    - Part 2: The Google Hacking Database
    - Part 3: The Wayback Machine

Open-Source Intelligence (OSINT) Gathering

- ***Open-source intelligence (OSINT) gathering*** is a method of gathering publicly available intelligence sources to collect and analyze information about a target.
- OSINT is “open source” because collecting the information does not require any type of covert methods.
- Typically, the information can be found on the Internet.
- The larger the online presence of the target, the more information that will be available.
- This type of collection can often start with a simple Google search, which can reveal a significant amount of information about a target.
- It will at least give you enough information to know what direction to go with your information-gathering process.
- Let's cover two tools that can be used for OSINT gathering: Recong-ng and Shodan.

Open-Source Intelligence (OSINT) Gathering (Cont.)

Recon-ng

- This module covers several of individual sources and tools used for information gathering.
- Wouldn't it be great if there were a tool that could pull together all these different functions?
- This is where Recon-ng comes in.
- This tool was developed in Python with Metasploit **msfconsole** in mind.
- If you have used the Metasploit console before, Recon-ng should be familiar and easy to understand.
- Recon-ng is a modular framework, which makes it easy to develop and integrate new functionality.
- It is highly effective in social networking site enumeration because of its use of APIs to gather information.
- It also includes a reporting feature that allows you to export data in different report formats.
- Because you will always need to provide some kind of deliverable in any testing you do, Recon-ng is especially valuable.
- Recon-ng is incredibly powerful because it uses the APIs of various OSINT resources to gather information.
- Its modules can query sites such as Facebook, Indeed, Flickr, Instagram, Shodan, LinkedIn, and YouTube.

Open-Source Intelligence (OSINT) Gathering (Cont.)

The following steps are an example of running Recon-ng in Kali Linux.

- **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**
  - To get an idea of what commands are available in the Recon-ng command-line tool, you can simply type **help** and press Enter.
- **Step 3. Search for available modules.**
  - Before you can start gathering information using the Recon-ng tool, you need to understand what modules are available.
  - Recon-ng comes with a "marketplace," where you can search for available modules to be installed.
  - You can 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.**
  - A quick search to find different subdomains of a domain can be performed.
  - You can perform a keyword search for any modules by using the command **marketplace search < *keyword* >**.

Open-Source Intelligence (OSINT) Gathering (Cont.)

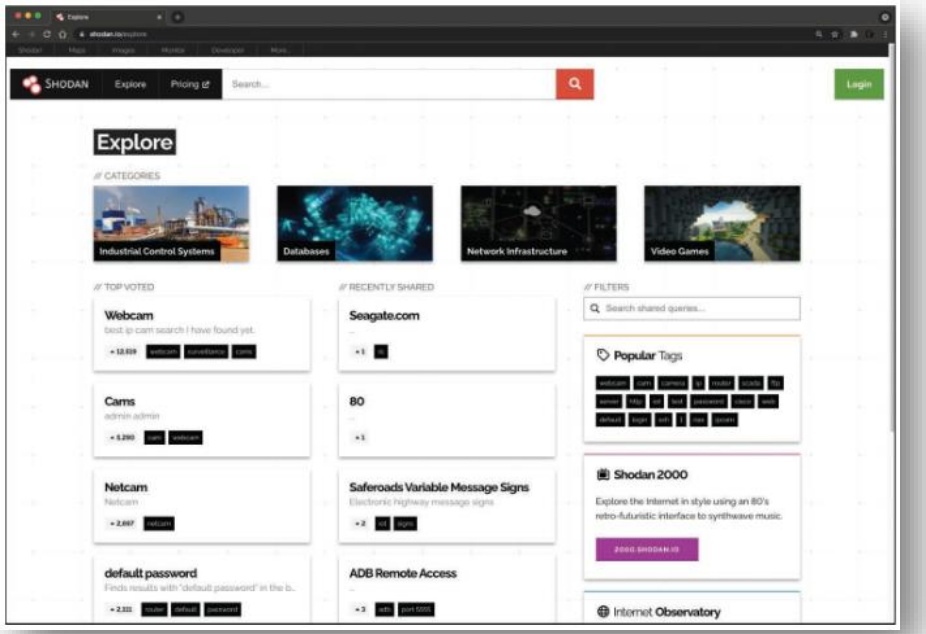
The following steps are an example of running Recon-ng in Kali Linux.

- **Step 6: Install a module**
  - Several results matched the keyword.
  - However, there is one that you are interested.
  - You can install the module by using the **marketplace install** command.
- **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.
  - The prompt changes 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 **options set SOURCE** command.
  - After the source domain is set, you can type **run** to run the query.
  - The highlighted lines shows the subdomains that were found using a specific module.

Open-Source Intelligence (OSINT) Gathering (Cont.)

Shodan

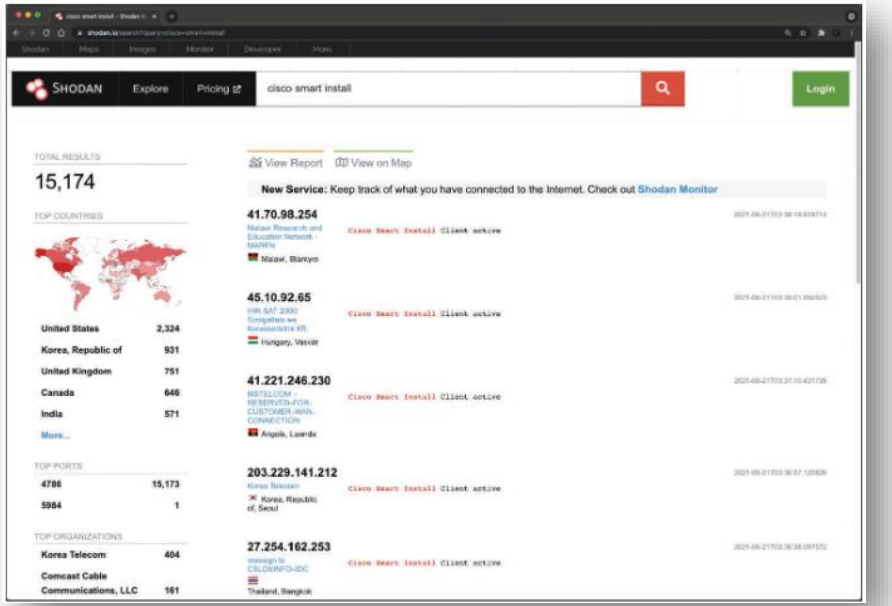
- It is an organization that scans the Internet 24 hours a day, 365 days a year.
- The results of those scans are stored in a database that can be queried at shodan.io or by using an API.
- It can be used to query for vulnerable hosts, IoT devices, and many other systems that should not be exposed or connected to the public Internet.
- The figure shows different categories of systems found by Shodan scans, including industrial control systems (ICS), databases, network infrastructure devices, and video games.



Open-Source Intelligence (OSINT) Gathering (Cont.)

Shodan

- The figure shows a query performed to find network infrastructure devices that are running a broken protocol called Cisco Smart Install.
- Attackers have leveraged this protocol for years to compromise different infrastructures.
- Cisco removed this protocol from its systems many years ago.
- However, many people are still using it in devices connected to the public Internet.



Lab - Shodan Searches

In this lab, you will complete the following objectives:

- Create a Shodan user account and register for an API key
- Use the Shodan website to search for vulnerable IoT devices
- Use Shodan from the CLI to perform a search

3.2 Performing Active Reconnaissance

Overview

- With each step of the information gathering phase, the goal is to gather additional information about the target. The process of gathering this information is called enumeration.
- External enumeration of hosts is usually one of the first things you do in a penetration test.
- Determining the Internet-facing hosts of a target network can help you identify the systems that are most exposed.
- After you identify those systems, you then need to identify which services are accessible.
- To determine if a network is running any services, you can run a port scan to enumerate the services that are running on the exposed hosts.

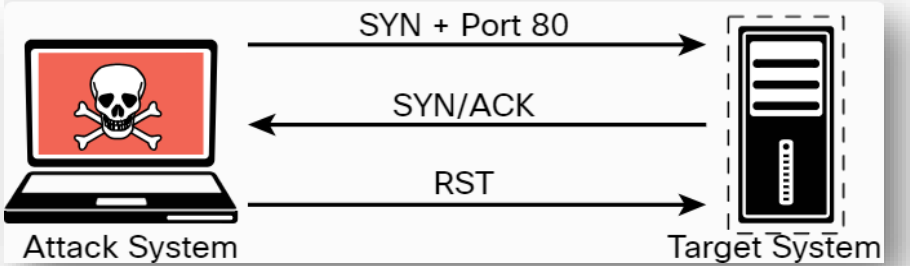
Overview (Cont.)

- A ***port scan*** is an active scan in which the scanning tool sends various types of probes to the target IP address and then examines the responses to determine whether the service is listening.
- For instance, with an Nmap SYN scan, the tool sends a TCP SYN packet to the TCP port it is probing.
- This process is also referred to as half-open scanning because it does not open a full TCP connection.
- If the response is a SYN/ACK, this would indicate that the port is in a listening state.
- If the response to the SYN packet is an RST (reset), this would indicate that the port is closed or is not in a listening state.
- If the SYN probe does not receive any response, Nmap marks it as filtered because it cannot determine if the port is open or closed.
- The table defines the SYN scan responses when using Nmap.

Nmap Port Status Reported	Response from Target	Nmap Analysis
Open	TCP SYN-ACK	The service is listening on the port.
Closed	TCP RST	The service is not listening on the port.
Filtered	No response from target or ICMP destination unreachable	The port is firewalled.

Overview (Cont.)

- The figure illustrates how a SYN scan works.



- The example shows how to run a TCP SYN scan using Nmap by specifying the **-sS** option against a host with the IP address 192.168.88.251.
- As you can see, this system has several ports open.
- In some situations, you will want to use the many different Nmap options in your scans to get the results you are looking for.

```
|--[root@websploit]--[~]
|---- #nmap -sS 192.168.88.251
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for 192.168.88.251
Host is up (0.00011s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
8888/tcp open  sun-answerbook
9000/tcp open  cslistener
9090/tcp open  zeus-admin
MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

Nmap Scan Types

- Some of the most common Nmap scanning options used for specific scenarios, including the following:
  - TCP Connect Scan (**-sT**)
  - UDP Scan (**-sU**)
  - TCP FIN Scan (**-sF**)
  - Host Discovery Scan (**-sn**)
  - Timing Options (**-T 0-5**)

TCP Connect Scan (-sT)

- A TCP connect scan makes use of the underlying operating system's networking mechanism to establish a full TCP connection with the target device being scanned.
- Because it creates a full connection, it creates more traffic (and thus takes more time to run).
- This is the default scan type that is used if no scan type is specified with the **nmap** command.
- However, it should typically be used only when a SYN scan is not an option, such as when a user who is running the **nmap** command does not have raw packet privileges on the operating system because many of the Nmap scan types rely on writing raw packets.

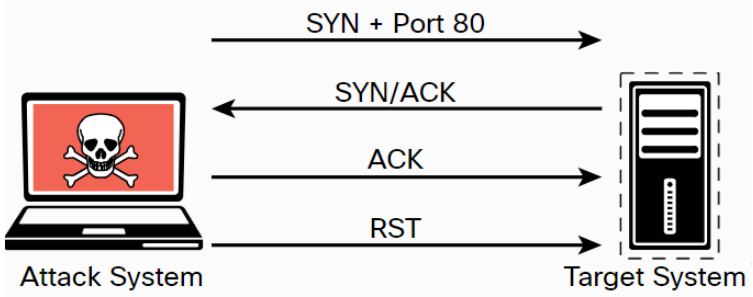
Nmap Scan Types (Cont.)

TCP Connect Scan (-sT) (Cont.)

- The table defines the TCP connect scan responses.

Nmap Port Status Reported	Response from Target	Nmap Analysis
Open	TCP SYN-ACK	The service is listening on the port.
Closed	TCP RST	The service is not listening on the port.
Filtered	No response from target	The port is firewalled.

- The figure illustrates how a TCP connect scan works.



Nmap Scan Types (Cont.)

TCP Connect Scan (-sT) (Cont.)

- The output in the figure shows the results of an Nmap TCP connect scan.
- A full TCP connect scan requires the scanner to send an additional packet per scan, which increases the amount of noise on the network and may trigger alarms that a half-open scan wouldn't trigger.
- Security tools and the underlying targeted system are more likely to log a full TCP connection, and intrusion detection systems (IDSs) are similarly more likely to trigger alarms on several TCP connections from the same host.

```
|--[root@websploit]--[~]
|--- #nmap -sT 192.168.88.251
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 12:48 EDT
Nmap scan report for 192.168.88.251
Host is up (0.00024s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
8888/tcp open  sun-answerbook
9000/tcp open  cslistener
9090/tcp open  zeus-admin
MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

Nmap Scan Types (Cont.)

UDP Scan (-sU)

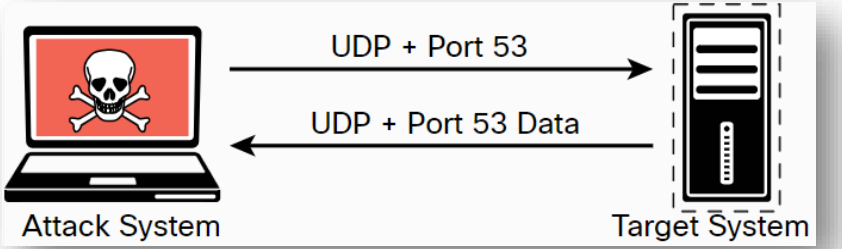
- You might encounter some instances in which you need to scan for UDP ports – for example, if you are trying to enumerate a DNS, SNMP, or DHCP server.
- These services all use UDP for communication between client and server.
- To scan UDP ports, Nmap sends a UDP packet to all ports specified in the command-line configuration.
- It waits to hear back from the target.
- If it receives an ICMP port unreachable message back from a target, that port is marked as closed.
- If it receives no response from the target UDP port, Nmap marks the port as open/filtered.
- The table shows the UDP scan responses.

Nmap Port Status Reported	Response from Target	Nmap Analysis
Open	Data returned from port	The service is listening on the port.
Closed	ICMP error message received	The service is not listening on the port.
Open/filtered	No ICMP response from target	The port is firewalled or timed out.

Nmap Scan Types (Cont.)

UDP Scan (-sU) (Cont.)

- The figure illustrates how a UDP scan works.



- The output in the example shows the results of an Nmap UDP scan on port 53 of the target 192.168.88.251.
- As you can see, the results indicate that this port is open.

```
|--[root@websploit]--[~]
|--- # nmap -sU -p 53 192.168.88.251
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 13:12 EDT
Nmap scan report for 192.168.88.251
Host is up (0.00057s latency).
PORT   STATE  SERVICE
53/udp open domain
MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Nmap Scan Types (Cont.)

TCP FIN Scan (-sF)

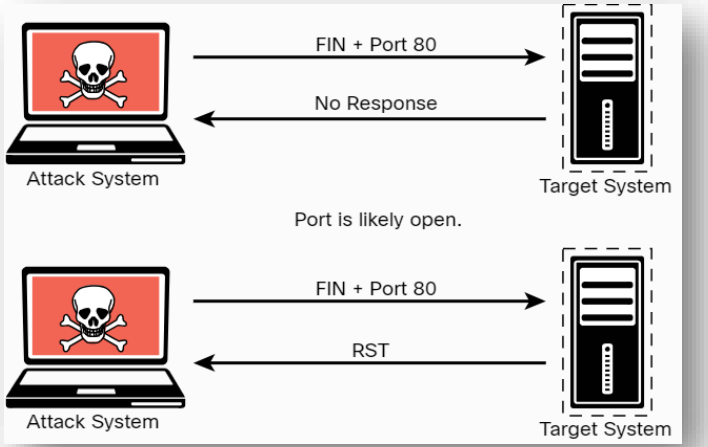
- There are times when a SYN scan might be picked up by a network filter or firewall.
- In such a case, you need to employ a different type of packet in a port scan.
- With the TCP FIN scan, a FIN packet is sent to a target port.
- If the port is closed, the target system sends back an RST packet.
- If nothing is received from the target port, you can consider the port open because the normal behavior would be to ignore the FIN packet.
- The Table shows the *TCP FIN Scan Responses*

Nmap Port Status Reported	Response from Target	Nmap Analysis
Open	ICMP unreachable error received	Closed port should respond with RST.
Closed	RST packet received	Closed port should respond with RST.
Open/filtered	No response received	Open port should drop FIN.

Nmap Scan Types (Cont.)

TCP FIN Scan (-sF) (Cont.)

- The figure illustrates how a TCP FIN scan works.



- The example output shows the results of an Nmap TCP FIN scan, specifying port 80 on the target.
- The response from the target indicates that the port is open/filtered.

```
|--[root@websploit]--[~]
|--- # nmap -sF -p 80 192.168.88.251
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 13:15 EDT
Nmap scan report for 192.168.88.251
Host is up (0.00045s latency).
PORT    STATE           SERVICE
80/tcp open|filtered http
MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
|--[root@websploit]--[~]
|--- #
```

Nmap Scan Types (Cont.)

Host Discovery Scan (-sn)

- A host discovery scan is one of the most common types of scans used to enumerate hosts on a network because it can use different types of ICMP messages to determine whether a host is online and responding on a network.
- The example shows a ping scan of the 192.168.88.0/24 subnet.
- This is a very basic host discovery scan that can be performed to determine what devices on a network are live.
- Such a scan for host discovery of an entire subnet is sometimes referred to as a *ping sweep*.

```
|--[root@websploit]--[~]
|---- #nmap -sn 192.168.88.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 14:32 EDT
Nmap scan report for 192.168.88.1
Host is up (0.00045s latency).
MAC Address: E0:55:3D:E9:61:74 (Cisco Meraki)
Nmap scan report for 192.168.88.12
Host is up (0.00094s latency).
MAC Address: 0E:64:AF:27:9C:44 (Unknown)
Nmap scan report for 192.168.88.14
Host is up (0.0092s latency).
MAC Address: 00:B8:B3:FD:BF:C2 (Cisco Systems)
Nmap scan report for 192.168.88.24
Host is up (0.0033s latency).
MAC Address: 00:E1:6D:E5:43:C2 (Cisco Systems)
Nmap scan report for 192.168.88.32
Host is up (0.00046s latency).
MAC Address: BE:38:F5:2D:6C:C0 (Unknown)
Nmap scan report for 192.168.88.231
Host is up (0.00061s latency).
MAC Address: FE:82:8C:A3:D2:3C (Unknown)
Nmap scan report for 192.168.88.251
Host is up (0.00040s latency).
MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)
Nmap scan report for 192.168.88.71
Host is up.
Nmap scan report for 192.168.88.225
Host is up.
Nmap done: 256 IP addresses (11 hosts up) scanned in 2.45 seconds
|--[root@websploit]--[~]
|--- #
```

Nmap Scan Types (Cont.)

Timing Options (-T 0-5)

- The Nmap scanner provides six timing templates that can be specified with the -T option and the template number (0 through 5) or name.
- Nmap timing templates enable you to dictate how aggressive a scan will be, while leaving Nmap to pick the exact timing values.
- These are the timing options:
  - **-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

Types of Enumeration

- This section covers enumeration techniques that should be performed in the information-gathering phase of a penetration test.
- You will learn how and when these enumeration techniques should be used.
- This section also includes examples of performing these types of enumeration by using Nmap as well as a deep dive into packet crafting with Scapy.
  - Host Enumeration
  - User Enumeration
  - Group Enumeration
  - Network Share Enumeration
  - Additional SMB Enumeration Examples
  - Web Page Enumeration/Web Application Enumeration
  - Service Enumeration
  - Exploring Enumeration via Packet Crafting

Types of Enumeration (Cont.)

Host Enumeration

- The enumeration of hosts is one of the first tasks you need to perform in the information-gathering phase of a penetration test.
- ***Host enumeration*** is performed internally and externally.
- When performed externally, you typically want to limit the IP addresses you are scanning to just the ones that are part of the scope of the test.
- This reduces the chance of inadvertently scanning an IP address that you are not authorized to test.
- When performing an internal host enumeration, you typically scan the full subnet or subnets of IP addresses being used by the target.
- Host enumeration is usually performed using a tool such as Nmap or Masscan; however, vulnerability scanners also perform this task as part of their automated testing.

Types of Enumeration (Cont.)

User Enumeration

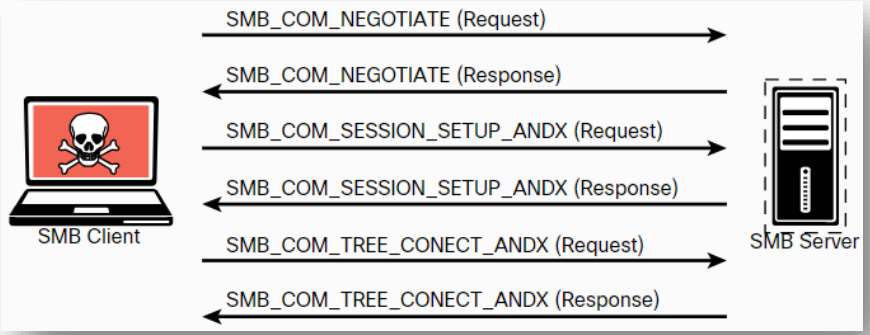
- Gathering a valid list of users is the first step in cracking a set of credentials.
- When you have the username, you can then begin brute-force attempts to get the account password.
- You perform ***user enumeration*** when you have gained access to the internal network.
- On a Windows network, you can do this by manipulating the Server Message Block (SMB) protocol, which uses TCP port 445.

Types of Enumeration (Cont.)

User Enumeration

- The figure illustrates how a typical SMB implementation works.
- The information contained in the responses to these messages enables you to reveal information about the server:
- **SMB_COM_NEGOTIATE:** It allows the client to tell the server what protocols, flags, and options it would like to use. The response from the server is also this message. This response is relayed to the client about which protocols, flags, and options it prefers. The response from the server also provides additional information, such as the time and time zone the server is using. This is necessary information for many penetration testing tasks.

- **SMB_COM_SESSION_SETUP_ANDX** : After the client and server have negotiated the protocols, flags, and options they will use for communication, the authentication process begins. Authentication is the primary function of this message. Information sent in this message includes the client username, password, and domain.



Types of Enumeration (Cont.)

User Enumeration (Cont.)

- The example shows the results of the Nmap **smb-enum-users** script run against the target 192.168.88.251.

- As you can see, the results indicate that the script was able to enumerate the users who are configured on this Windows target.

- The highlighted line reveals the user who was enumerated by Nmap (derek).

```
|--[root@websploit]--[~]
|--- #nmap  --script smb-enum-users.nse 192.168.88.251
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-22 11:14 EDT
Nmap scan report for 192.168.88.251
Host is up (0.012s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
8888/tcp open  sun-answerbook
9000/tcp open  cslistener
9090/tcp open  zeus-admin
Host script results:
| smb-enum-users:
|    VULNHOST-1\derek (RID: 1000)
|      Full name:
|      Description:
|_     Flags:          Normal user account
Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds
```

Types of Enumeration (Cont.)

Group Enumeration

- For a penetration tester, ***group enumeration*** is helpful in determining the authorization roles that are being used in the target environment.
- The Nmap NSE script for enumerating SMB groups is **smb-enum-groups**.
- This script attempts to pull a list of groups from a remote Windows machine.
- You can also reveal the list of users who are members of those groups. The syntax of the command is as follows: **nmap --script smb-enum-groups.nse -p445** *<host>*

- The example shows sample output of this command run against the Windows server at 192.168.56.3 (output omitted for brevity).
- This example uses known credentials to gather information.
- The highlighted output shows the enumerated groups and users in the target host.

```
|--[root@websploit]--[~]
|--- # nmap --script smb-enum-groups.nse --script-args smbusername=vagrant,smbpass=vagrant 192.168.56.3
Starting Nmap 7.91 ( https://nmap.org )
Nmap scan report for 192.168.56.3
Host is up (0.0062s latency).
Not shown: 979 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
3389/tcp open  ms-wbt-server
MAC Address: 08:00:27:1B:A4:60 (Oracle VirtualBox virtual NIC)
Host script results:
| smb-enum-groups:
|   Builtin\Administrators (RID: 544): Administrator, vagrant, sshd_server
|   Builtin\Users (RID: 545): vagrant, sshd, sshd_server, leia_organa,
luke_skywalker, han_solo, artoo_detoo, c_three_pio, ben_kenobi, darth_
vader, anakin_skywalker, jarjar_binks, lando_calrissian, boba_fett,
jabba_hutt, greedo, chewbacca, kylo_ren
|   Builtin\Guests (RID: 546): Guest, ben_kenobi
|   Builtin\Power Users (RID: 547): boba_fett
|   Builtin\Print Operators (RID: 550): jabba_hutt
```

Types of Enumeration (Cont.)

Network Share Enumeration

- Identifying systems on a network that are sharing files, folders, and printers is helpful in building out an attack surface of an internal network.

- The Nmap **smb-enum-shares** NSE script uses Microsoft Remote Procedure Call (MSRPC) for ***network share enumeration***.

- The syntax of the Nmap **smb-enum-shares.nse** script is as follows:

  nmap --script smb-enum-shares.nse -p 445 <*host*>

- The example demonstrates the enumeration of SMB shares (output omitted for brevity).

```
|--[root@websploit]--[~]
|--- # nmap --script smb-enum-shares.nse -p 445 192.168.88.251
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-22 11:27 EDT
Nmap scan report for 192.168.88.251
Host is up (0.0011s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\192.168.88.251\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (Samba 4.9.5-Debian)
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\192.168.88.251\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|     Current user access: <none>
|   \\192.168.88.251\secret_folder:
```

Types of Enumeration (Cont.)

Additional SMB Enumeration Examples

- The system used in earlier examples (with the IP address 192.168.88.251) is running Linux and Samba.
- However, it is not easy to determine that it is a Linux system from the results of previous scans.
- An easy way to perform additional enumeration and fingerprinting of the applications and operating system running on a host is by using the **nmap -sC** command.
- The - **sC** option runs the most common NSE scripts based on the ports found to be open on the target system.
- The example shows the output of the **nmap -sC** command launched against the Linux system at 192.168.88.251, which is running Samba (output omitted for brevity).

```
|--[root@websploit]--[~]
|--- # nmap -sC 192.168.88.251
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 17:38 EDT
Nmap scan report for 192.168.88.251
Host is up (0.00011s latency).
Not shown: 992 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
| ssh-hostkey:
|   2048 d0:0c:83:4d:7f:84:2c:60:96:9f:df:26:da:d2:11:9a (RSA)
|   256 e2:aa:69:ab:a3:e6:0f:13:c5:5a:65:f2:d5:16:8c:3e (ECDSA)
|_  256 21:4b:27:7b:6e:a6:d4:33:86:60:cb:39:3b:48:9c:0b (ED25519)
80/tcp   open  http
|_http-title: WebSploit Mayhem
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
| mysql-info:
|   Protocol: 10
|   Version: 5.5.47-0ubuntu0.14.04.1
|   Thread ID: 3
|   Capabilities flags: 63487
|   Some Capabilities: InteractiveClient,
```

Types of Enumeration (Cont.)

Additional SMB Enumeration Examples

- You can also use tools such as enum4linux to enumerate Samba shares, including user accounts, shares, and other configurations.

- There is a Python-based enum4linux implementation called enum4linux-ng that can be used for SMB enumeration.

- You can also use simple tools such as smbclient to enumerate shares and other information from a system running SMB.

Types of Enumeration (Cont.)

Web Page Enumeration/Web Application Enumeration

- After identifying that a web server is running on a target host, the next step is to take a look at the web application and begin to map out the attack surface performing ***web page enumeration*** or often referred to as ***web application enumeration***.
- The Nmap tool has an NSE script available for brute forcing the directory and file paths of web applications.
- Armed with a list of known files and directories used by common web applications, it probes the server for each of the items on the list.
- Based on the response from the server, it can determine whether those paths exist.
- The syntax of the http-enum NSE script is as follows:

nmap -sV --script=http-enum *<target>*

- The example displays the results of running this script against the host with the IP address 192.168.88.251.
- The highlighted output shows the version of the web server being used (Nginx 1.17.2) and several enumerated directories/folders.

```
|--[root@websploit]--[~]
|--- #nmap -sV --script=http-enum -p 80 192.168.88.251
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-22 11:53 EDT
Nmap scan report for 192.168.88.251
Host is up (0.0011s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.17.2
| http-enum:
|   /admin/: Possible admin folder
|   /admin/index.html: Possible admin folder
|_  /s/: Potentially interesting folder
|_http-server-header: nginx/1.17.2
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.54 seconds
|--[root@websploit]--[~]
|--- #
```

Types of Enumeration (Cont.)

Web Page Enumeration/Web Application Enumeration (Cont.)

- Nikto is another web server enumeration tool.
- It is an open-source web vulnerability scanner that has been around for many years.
- It's not as robust as the commercial web vulnerability scanners; however, it is very handy for running a quick script to enumerate information about a web server and the applications it is hosting.
- Because of the speed at which Nikto works to scan a web server, it is very noisy.
- It provides several options for scanning, including the capability to authenticate to a web application that requires a username and password.
- The example shows the output of a Nikto scan being run against the host 192.168.88.251.

```
|--[root@websploit]--[~]
|--- #nikto -h 192.168.88.251
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.88.251
+ Target Hostname:    192.168.88.251
+ Target Port:        80
---------------------------------------------------------------------------
+ Server: nginx/1.17.2
+ The anti-clickjacking X-Frame-Options header is not present.
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to
the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the
user agent to render the content of the site in a different fashion
to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible
dirs)
+ OSVDB-3092: /admin/: This might be interesting...
+ /admin/index.html: Admin login page/section found.
+ /wp-admin/: Admin login page/section found.
+ /wp-login/: Admin login page/section found.
+ 7916 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2021-06-22 11:57:59 (GMT-4) (15 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
|--[root@websploit]--[~]
|--- #
```

Types of Enumeration (Cont.)

Service Enumeration

- ***Service enumeration*** is the process of identifying the services running on a remote system, and it is a primary focus of what Nmap does as a port scanner.
- When you are connected to a system that is on a directly connected network segment, you can run some additional scripts to enumerate further.
- A port scan takes the perspective of a credentialed remote user.
- The Nmap **smb-enum-processes** NSE script enumerates services on a Windows system, and it does so by using credentials of a user who has access to read the status of services that are running.
- This is a handy tool for remotely querying a Windows system to determine the exact list of services running.
- The syntax of the command is as follows:

nmap --script smb-enum-processes.nse --script-args smbusername=<*username*>**, smbpass=**<*password*> **-p445** <*host*>

Types of Enumeration (Cont.)

Exploring Enumeration via Packet Crafting

- 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.
- This section looks at some of the simple ways you can use this tool to perform basic network reconnaissance.
- Launching the Scapy interactive shell is as easy as typing **sudo scapy** from a terminal window.
- You can use Scapy as a scanner in several ways.

Lab - Enumeration with Nmap

- In this lab, you will complete the following objectives:
  - Investigate Nmap
  - Perform Basic Nmap Scans

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.
- For a penetration tester, such tools can be convenient for performing passive reconnaissance.
- Of course, this type of reconnaissance requires either a physical or a wireless connection to the target.
- If you are concerned about being detected, you are probably better off attempting a wireless connection because it would not require you to be inside the building.
- Many times, a company's wireless footprint bleeds outside its physical walls.
- This gives a penetration tester an opportunity to potentially collect information about the target and possibly gain access to the network to sniff traffic.

Lab - Packet Crafting with Scapy

- In this lab, you will use Scapy, a Python-based packet manipulation tool, to craft custom packets. These custom packets will be used to perform reconnaissance on a target system.

  - Part 1: Investigate the Scapy Tool.
  - Part 2: Use Scapy to Sniff Network Traffic.
  - Part 3: Create and Send an ICMP Packet.
  - Part 4: Create and Send TCP SYN Packets.

Lab - Network Sniffing with Wireshark

- In this lab, you will use the Linux utility **tcpdump** to capture and save network traffic.
- You will then use Wireshark to investigate the traffic capture.

    - Prepare the host to capture network traffic.
    - Capture and save network traffic.
    - View and Analyze the Packet capture.

3.3 Understanding the Art of Performing Vulnerability Scans

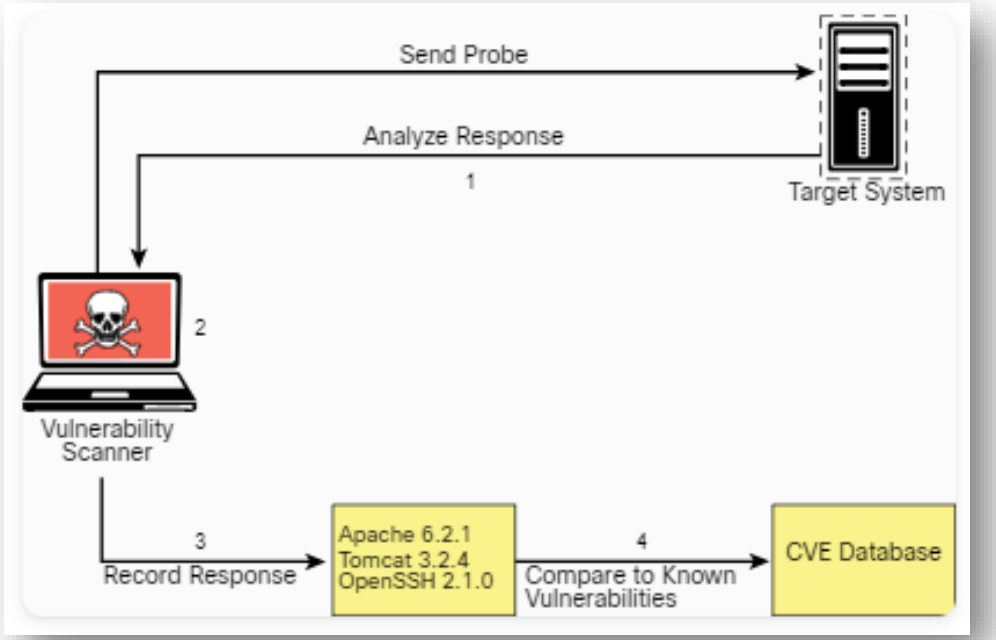
Overview

- Once you have identified the target hosts that are available and the services that are listening on those hosts, you can begin to probe those services to determine if there are any weaknesses; this is what vulnerability scanners do.
- Vulnerability scanners use several different methods to determine whether a service is vulnerable.
- The primary method is to identify the version of the software that is running on the open service and try to match it with an already known vulnerability.
- The main concern with automated vulnerability scanners is false positives; the output from a vulnerability scan may be useless if no validation is done on the findings.
- Turning over a report full of false positives to a developer or an administrator who is then responsible for fixing the issues can really cause conflicts.
- You don’t want someone chasing down findings in your report just to find out that they are false positives.

How a Typical Automated Vulnerability Scanner Works

This is how a typical vulnerability scanner works:

- Step 1. In the discovery phase, the scanner uses a tool such as Nmap to perform host and port enumeration.
- Step 2. When the scanner has enough information about the open port to determine what software and version are running on that port, it records that information in a database for further analysis.
- Step 3. The scanner tries to determine if the software that is listening on the target system is susceptible to any known vulnerabilities.
- Step 4. The scanner produces a report on what it suspects could be vulnerable.



Types of Vulnerability Scans

- The type of vulnerability scan to use is usually driven by scan policy that is created in the automated vulnerability scanning tool.
- Each tool has many options available for scanning.
- You can often just choose to do a full scan that will operate all scanning options, although you might not be able to use every option (for instance, if you are scanning a production environment or a device that is prone to crashing when scanning occurs).
- In such situations, you must be careful to select only the scan options that are less likely to cause issues.
- Let's take a closer look at the following typical scan types:
  - Unauthenticated Scans
  - Authenticated Scans
  - Discovery Scans
  - Full Scans
  - Stealth Scans
  - Compliance Scans

Types of Vulnerability Scans (Cont.)

Unauthenticated Scans

- By default, vulnerability scanners do not use credentials to scan a target.
- If you provide only the IP address of the target and click Scan, the tool will begin enumerating the host from the perspective of an unauthenticated remote attacker.
- An ***unauthenticated scan*** shows only the network services that are exposed to the network.
- The scanner attempts to enumerate the ports open on the target host.
- If the service is not listening on the network segment that the scanner is connected to, or if it is firewalled, the scanner will report the port as closed and move on.
- However, this does not mean that there is not a vulnerability.
- Sometimes it is possible to access ports that are not exposed to the network via SSH port forwarding and other tricks.
- It is still important to run a credentialed (or authenticated) scan when possible.

Types of Vulnerability Scans (Cont.)

Authenticated Scans

- In some cases, it is best to run an authenticated scan against a target to get a full picture of the attack surface.
- An ***authenticated scan*** requires you to provide the scanner with a set of credentials that have root-level access to the system.
- The scanner logs in to the target via SSH or some other mechanism.
- It then runs commands like **netstat** to gather information from inside the host.
- Many of the commands that the scanner runs require root-level access to be able to gather the correct information from the system.

Types of Vulnerability Scans (Cont.)

Authenticated Scans (Cont.)

- The first example shows the **netstat** command run by a non-privileged user (omar); the second example shows the **netstat** command run by a root user.
- You can see that the output is different for the different user-level permissions.
- Specifically, notice that when running as the user omar, the PID/program name is not available, and when running as the user root, that information is displayed.

```
omar@vulnhost-1 ~ % netstat -tunap
(Not all processes could be identified, non-
owned process info will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address       Foreign Address
State       PID/Program name
tcp        0      0 0.0.0.0:9000        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:9001        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:9002        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:3306        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:139         0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:80          0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:8881        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:8882        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:8883        0.0.0.0:*     LISTEN      -
tcp        0      0 0.0.0.0:8884        0.0.0.0:*     LISTEN
<output omitted for brevity>
```

```
omar@vulnhost-1 ~ % sudo netstat -tunap
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address       Foreign Address     State       PID/Program name
tcp        0      0 0.0.0.0:9000        0.0.0.0:*           LISTEN      1076/docker-proxy
tcp        0      0 0.0.0.0:9001        0.0.0.0:*           LISTEN      762/docker-proxy
tcp        0      0 0.0.0.0:9002        0.0.0.0:*           LISTEN      1287/docker-proxy
tcp        0      0 0.0.0.0:3306        0.0.0.0:*           LISTEN      1095/docker-proxy
tcp        0      0 0.0.0.0:139         0.0.0.0:*           LISTEN      247/smbd
tcp        0      0 0.0.0.0:80          0.0.0.0:*           LISTEN      506/docker-proxy
tcp        0      0 0.0.0.0:8881        0.0.0.0:*           LISTEN      831/docker-proxy
tcp        0      0 0.0.0.0:8882        0.0.0.0:*           LISTEN      530/docker-proxy
tcp        0      0 0.0.0.0:8883        0.0.0.0:*           LISTEN      702/docker-proxy
tcp        0      0 0.0.0.0:8884        0.0.0.0:*           LISTEN      1133/docker-proxy
tcp        0      0 0.0.0.0:8885        0.0.0.0:*           LISTEN      989/docker-proxy
tcp        0      0 0.0.0.0:8886        0.0.0.0:*           LISTEN      604/docker-proxy
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      161/sshd
tcp        0      0 0.0.0.0:8887        0.0.0.0:*           LISTEN      1163/docker-proxy
tcp        0      0 0.0.0.0:8888        0.0.0.0:*           LISTEN      494/docker-proxy
tcp        0      0 0.0.0.0:88          0.0.0.0:*           LISTEN      482/docker-proxy
tcp        0      0 0.0.0.0:8889        0.0.0.0:*           LISTEN      470/docker-proxy
tcp        0      0 127.0.0.1:25        0.0.0.0:*           LISTEN      338/master
tcp        0      0 0.0.0.0:445         0.0.0.0:*           LISTEN      247/smbd
tcp        0      0 0.0.0.0:9090        0.0.0.0:*           LISTEN      819/docker-proxy
<output omitted for brevity>
```

Types of Vulnerability Scans (Cont.)

Discovery Scans

- A ***discovery scan*** is primarily meant to identify the attack surface of a target.
- A port scan is a major part of what a discovery scan performs.
- A scanner may use a tool like Nmap to perform the port scan process.
- It then pulls the results of the port scan into its database to use that information for further discovery.
- For instance, the result of the port scan might come back showing that ports 80, 22, and 443 are open and listening.
- From there, the scanning tool probes those ports to identify exactly what service is running on each port.
- For example, say that it identifies that an Apache Tomcat 8.5.22 web server is running on ports 80 and 443.
- Knowing that a web server is running on the ports, the scanner can then perform further discovery tasks that are specific to web servers and applications.
- Now say that, at the same time, the scanner identifies that OpenSSH is listening on port 22.
- From there, the scanner can probe the SSH service to identify information about its configuration and capabilities, such as preferred and supported cryptographic algorithms.
- This type of information is useful for identifying vulnerabilities in later phases of testing.

Types of Vulnerability Scans (Cont.)

Full Scans

- A ***full scan*** typically involves enabling every scanning option in the scan policy.
- The options vary based on the scanner, but most vulnerability scanners have their categories of options defined similarly.
- For instance, they are typically organized by operating system, device manufacturer, device type, protocol, compliance, and type of attack, and the rest of the options might fall into a miscellaneous category.

- The example shows a sample list of the plugin categories from the Nessus vulnerability scanner (output omitted for brevity).
- There are a lot of plugins available for the scanner to run.
- Based on the names of the plugin categories, there will never be a single device that all these plugins apply to.
- For instance, plugins for a macOS device would not be applicable to a Windows device.
- So you normally need to customize your plugin selection to reflect the environment that you are scanning.

Family	Count
AIX Local Security Checks	11416
Amazon Linux Local Security Checks	1048
Backdoors	114
Brute force attack	26
CGI abuses	3841
CGI abuses : XS	666
CISCO	918
CentOS Local Security Checks	2585
DNS	172
Databases	577
Debian Local Security Checks	5532
Default Unix Accounts	168
Denial of Service	109
F5 Networks Local Security Checks	607
FTP	255

Types of Vulnerability Scans (Cont.)

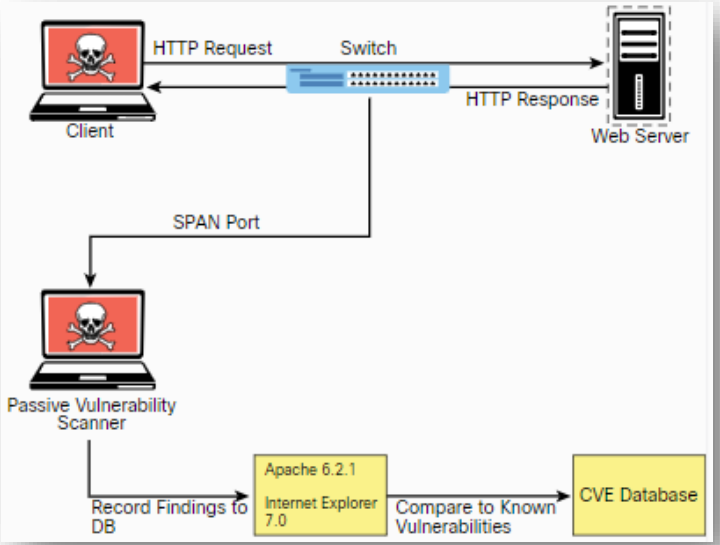
Stealth Scans

- There are sometimes situations in which you must scan an environment that is in a production state.
- In such situations, there is typically a requirement for running a scan without alerting the defensive position of the environment; such a scan is called a ***stealth scan***.
- In this case, you will want to implement a vulnerability scanner in a manner that makes the target less likely to detect the activity.
- Vulnerability scanners are noisy; however, there are some options you can configure to make it quieter.
- There are different types of Nmap scans, and they can be detected by network intrusion prevention systems (IPSs) or host firewalls.
- A SYN scan is a fairly stealthy type of scan to run.
- This same concept applies to vulnerability scanners because they all use some kind of port scanner to enumerate the target.
- These same options are available in the vulnerability scanner's configuration.
- You can also disable any plugins/attacks that might be especially likely to generate noisy traffic, such as any that perform denial-of-service attacks, which would definitely arouse some concerns on the target network.

Types of Vulnerability Scans (Cont.)

Stealth Scans (Cont.)

- There is also the concept of a passive vulnerability scanner, that monitors and analyzes the network traffic.
- Based on the traffic it sees, it can determine what the topology of the network consists of and what service the hosts on the network are listening on.
- From the detailed information about the traffic at the packet layer, it can determine if any of those services or even clients have vulnerabilities.
- For instance, if a Windows client with an outdated version of Internet Explorer is connecting to an Apache web server that is also outdated, the scanner will identify the versions of the client and server from the monitored traffic.
- It can then compare those versions to its database of known vulnerabilities and report the findings based on only the passive monitoring it performed.
- The figure illustrates how this type of scanner typically works.



Types of Vulnerability Scans (Cont.)

Compliance Scans

- They are network and application tests (scans) typically driven by the market or governance that the environment serves and regulatory compliance.
- An example of this would be the information security environment for a healthcare entity, which must adhere to the requirements sent forth by the HIPAA.
- This is where a vulnerability scanner comes into play.
- It is possible to use a vulnerability scanner to address the specific requirements that a policy requires.
- Vulnerability scanners often have the capability to import a compliance policy file.
- This policy file can typically map to specific plugins/attacks that the scanner is able to perform.
- Once the policy is imported, the specific set of compliance checks can be run against a target system.
- The challenge with compliance requirements is that there are many different types for different industries and government agencies, and they can all be interpreted in various ways.
- Some of the checks might be straightforward.
- If a requirement check is looking for a specific command to be run and that the output be a 1 instead of a 0, that is very simple for a vulnerability scanner to determine; however, many requirements leave more to be interpreted.

Lab - Vulnerability Scanning with Kali Tools

- In this lab, you will explore network vulnerability scanning tools and use them to perform a vulnerability scan on a target host.

    - Perform network scans with Nmap
    - Use Greenbone Vulnerability Management to perform a vulnerability scan.

Challenges to Consider When Running a Vulnerability Scan

Some of the specific things you should consider when building a scanning policy and performing scans:

- **Considering the best time to run a scan**
  - The timing of when to run a scan is typically of most concern when you are scanning a production network.
  - If you are scanning a device in a lab environment, there is normally not much concern because a lab environment is not being used by critical applications.
  - There are a few reasons running a scan on a production network should be done carefully.
  - First, the network traffic that is being generated by a vulnerability scan can and will cause a lot of noise on the network.
  - It can also cause significant congestion, especially when your scans are traversing multiple network hops.
  - Another consideration is the fact that many of the options or plugins that are performed in a vulnerability scan can and will crash the target device as well as the network infrastructure.
  - For this reason, you should be sure that when scanning on a production network, you are scanning at times that will have the least possible impact on end users and servers.
  - Most of the time, scanning in the early hours of the day, when no one is using a network for critical purposes, is best.

Challenges to Consider When Running a Vulnerability Scan (Cont.)

- **Determining What Protocols Are in Use**
  - This is one of the first things you need to know about a network or target device before you begin running vulnerability scans.
  - If a target device is using both TCP and UDP protocols for services that are running, and you only run a vulnerability scan against TCP ports, then you are going to miss any vulnerabilities that might be found on the UDP services.

- **Network Topology**
  - The network topology should always be considered when it comes to vulnerability scanning.
  - Of course, scanning across a WAN connection is never recommended because it would significantly impact any of the devices along the path.
  - The rule of thumb when determining where in the network topology to run a vulnerability scan is that it should always be performed as close to the target as possible.
  - Aside from the impact on the network infrastructure, another concern is that any device that you traverse could also affect the results of your scanner.
  - This is mostly a concern when traversing a firewall device; in addition, other network infrastructure devices could possibly impact the results as well.

Challenges to Consider When Running a Vulnerability Scan (Cont.)

- **Bandwidth Limitations**
  - Any time you flood a network with a bunch of traffic, it is going to cause an issue with the amount of bandwidth that is available.
  - As a penetration testing professional, you need to be cognizant of how you are affecting the bandwidth of the networks or systems you are scanning.
  - Specifically, depending on the amount of bandwidth you have between the scanner and the target, you might need to adjust your scanner settings to accommodate lower-bandwidth situations.
  - If you are scanning across a VPN or WAN link that most likely has limited bandwidth, you will want to adjust your scanning options so that you are not causing bandwidth consumption issues.
  - The settings that need to be adjusted are typically those related to flooding and denial-of-service (DoS) type attacks.

Challenges to Consider When Running a Vulnerability Scan (Cont.)

- **Query Throttling**
  - To work around the issue of bandwidth limitations and vulnerability scanning, slowing down the traffic created by your scanner can often help.
  - This is often referred to as ***query throttling***, and it can typically be achieved by modifying the options of the scanning policy.
  - One way to do this is to reduce the number of attack threads that are being sent to the target at the same time.
  - There isn't a specific rule of thumb for the number of threads.
  - It really depends on the robustness of the target, and some targets are more fragile than others.
  - Another way to accomplish this is to reduce the scope of the plugins/attacks that the scanner is checking for.
  - If you know that the target device is a Linux server, you can disable the attacks for other operating systems, such as Windows.
  - Even though the attacks won't work against the Linux server, it still needs to receive and respond to the traffic.
  - This additional traffic can cause a bottleneck in processing and network traffic consumption.
  - Limiting the number of requests that the target would need to respond to would reduce the risk of causing issues such as crashing on the target and result in a more successful scan.

Challenges to Consider When Running a Vulnerability Scan (Cont.)

- **Fragile Systems/Nontraditional Assets**
  - When using a vulnerability scanner against your internal network, you must take into consideration the devices on the network that might not be able to stand up to the traffic that is hurled at them by a vulnerability scanner.
  - For these systems, you might need to either adjust the scanning options to reduce the risk of crashing the devices or completely exempt the specific devices from being scanned.
  - Unfortunately, by exempting the specific devices, you reduce the overall security of the environment.
  - Printers are often considered “fragile systems.”
  - Historically, they have been devices that have not been able to withstand vulnerability scanning attempts.
  - With the surge in IoT devices, today there are many more devices that may be considered fragile, and you need to consider them when planning for vulnerability scanning.
  - The typical way to address fragile devices is to exempt them from a scan; however, these devices can pose a risk to the environment and do need to be scanned.
  - To address this issue, you can “throttle” the scan frequency as well as the options used in the scan policy to reduce the likelihood of crashing the device.

3.4 Understanding How to Analyze Vulnerability Scan Results

Overview

- Running a vulnerability scan is really the easy part of the information gathering and vulnerability identification process.
- Most of the work goes into analyzing the results you obtain from the tools you use for vulnerability scanning.
- When you are providing a report as a deliverable of a paid penetration testing assignment, it is especially important that the report be accurate.
- So how do you go about eliminating false positives?
- The process involves a detailed and thorough look into the results that your vulnerability scanning tool has provided.
- The results of a credentialed scan are more likely than remote analysis to be valid.
- Each vulnerability will typically map to one or many items in the Common Vulnerabilities and Exposures (CVE) list.
- When you dig into the CVE details, you might find that for the vulnerability to be exploitable.
- The number-one method of validating a finding from a vulnerability scan is to exploit the vulnerability.

Sources for Further Investigation of Vulnerabilities

The following table describes some helpful sources for further investigation of vulnerabilities that you might find during your scans.

US-CERT (U.S. Computer Emergency Readliness Team)	It was established to protect the Internet infrastructure of the United States. The main goal is to work with public- and private-sector agencies to increase the efficiency of vulnerability data sharing. The work done by US-CERT is meant to improve the nation's cybersecurity posture.
The CERT Division of Carnegie Mellon University	It is a cybersecurity center whose experts help coordinate vulnerability disclosures across the industry. It researches security vulnerabilities and contributes to many different cybersecurity efforts in the industry. It also develops and delivers training to many organizations to help them improve their cybersecurity practices and programs.
NIST (National Institute of Standards and Technology)	It is an agency of the U.S. Department of Commerce. Its core focus is to promote innovation and industrial competitiveness. NIST is responsible for the creation of the NIST Cybersecurity Framework. This framework includes a policy on computer security guidance. In general, the framework outlines the standards and industry best practices that can be used to improve organizations' cybersecurity posture. Anyone who is responsible for making decisions related to cybersecurity in an organization should consult this framework for guidance on standards and best practices.

Sources for Further Investigation of Vulnerabilities (Cont.)

JPCERT (Japan Computer Emergency Response Team)	Similar to the US-CERT, the JPCERT is an organization that works with service providers, security vendors, and private-sector and government agencies to provide incident response capabilities, increase cybersecurity awareness, conduct research and analysis of security incidents, and work with other international CERT teams. It is responsible for Computer Security Incident Response Team (CSIRT) activities in the Japanese and Asia Pacific region.
CAPEC (Common Attach Pattern Enumeration and Classification)	It is a community-driven effort to catalog the attack patterns seen in the wild so that they can be used to more efficiently identify active threats. It is maintained by MITRE, acting as a dictionary of known attacks that have been seen in the real world.
CVE (Common Vulnerabilities and Exposures)	It is an effort that reaches across international cybersecurity communities. It was created in 1999 with the idea of consolidating cybersecurity tools and databases. A CVE ID is composed of the letters CVE followed by the year of publication and four or more digits in the sequence number portion of the ID (for example, CVE-YYYY-NNNN with four digits in the sequence number, CVE-YYYY-NNNNN with five digits in the sequence number, CVE-YYYY-NNNNNNN with seven digits in the sequence number, and so on).
CWE (Common Weakness Enumeration)	It is at a high level, is a list of software weaknesses. Its purpose iss to create a common language to describe software security weaknesses that are the root causes of given vulnerabilities. CWE provides a common baseline for weakness identification to aid the mitigation process.

Sources for Further Investigation of Vulnerabilities (Cont.)

CVSS (Common Vulnerability Scoring System)	Each vulnerability represents a potential risk that threat actors can use to compromise your systems and your network. Each vulnerability carries an associated amount of risk. One of the most widely adopted standards for calculating the severity of a given vulnerability is the CVSS, which has three components: base, temporal, and environmental scores. Each component is presented as a score on a scale from 0 to 10. In CVSS, a vulnerability is evaluated according to three aspects, with a score assigned to each of them: the base group, the temporal group, and the environmental group. The score for the base group is between 0 and 10, where 0 is the least severe and 10 is assigned to highly critical vulnerabilities. In addition, the score comes in the form of a vector string that identifies each of the components used to make up the score. The formula used to obtain the score takes into account various characteristics of the vulnerability and how the attacker is able to leverage these characteristics. CVSS defines several characteristics for the base, temporal, and environmental groups. The base group defines Exploitability metrics that measure how the vulnerability can be exploited, as well as Impact metrics that measure the impact on confidentiality, integrity, and availability. In addition to these two metrics, a metric called Scope Change (S) is used to convey the impact on other systems that may be impacted by the vulnerability but do not contain the vulnerable code.

Lab - Investigate Vulnerability Information Sources

- In this lab, you will use multiple helpful sources to further investigate vulnerabilities.
  - Part 1: Investigate Common Vulnerabilities and Exposures (CVEs)
  - Part 2: Explore Common Weakness Enumerations (CWEs)
  - Part 3: Investigate National Institute of Standards and Technology (NIST) Vulnerability Resources
  - Part 4: Research Vulnerabilities in the Common Vulnerability Scoring System (CVSS)

How to Deal with a Vulnerability

- After you identify a vulnerability, you need to verify it.
- The ultimate validation is exploitation.
- To determine if a vulnerability is exploitable, you need to first identify an exploit for a vulnerability.
- As a general rule, if a vulnerability has a matching module in Metasploit, it should almost always be considered high severity.
- To determine the priority, you need to answer a few questions:
  - What is the severity of the vulnerability?
  - How many systems does the vulnerability apply to?
  - How was the vulnerability detected?
  - Was the vulnerability found with an automated scanner or manually?
  - What is the value of the device on which the vulnerability was found?
  - Is this device critical to your business or infrastructure?
  - What is the attack vector, and does it apply to your environment?
  - Is there a possible workaround or mitigation available?

How to Deal with a Vulnerability (Cont.)

- Answering these questions can help you determine the priority you should assign to the vulnerabilities found.
- Standard protocol would have you start with the highest-severity vulnerabilities that have the greatest likelihood of being exploited.
- If at any time during a penetration test, you find that a system is being actively exploited, you should report it right away to the system owner.
- Next, you should address any vulnerabilities that are on critical systems, regardless of the severity level.
- You need to protect critical systems first.
- Next, you might want to prioritize based on how many systems are affected by the finding.
- If a large number of systems are affected, then this would raise the priority because many exploits on this vulnerability would have a higher impact on your environment.
- These are suggested guidelines, but when it comes to prioritization of vulnerability management and mitigation, it really depends on the specific environment.

3.5 Information Gathering and Vulnerability Scanning Summary

What Did I Learn in this Module?

- Reconnaissance is the initial step in a cyber attack where an attacker gathers information about the target.
- There are two types of reconnaissance: active and passive.
- Active reconnaissance involves sending probes to the target network or systems, while passive reconnaissance does not interact directly with the target, using third-party databases and eavesdropping on network traffic instead.
- Common active reconnaissance methods include host, network, user, group, network share, web page, application, and service enumeration, as well as packet crafting.
- Passive reconnaissance methods include domain enumeration, packet inspection, open-source intelligence (OSINT), Recon-ng, and eavesdropping.
- Performing active reconnaissance typically starts with a small amount of information, and then gather more while scanning, eventually moving on to different types of scans and gathering additional information.
- Some techniques used by attackers include DNS lookups, identification of technical and administrative contacts, social media scraping, and inspecting cryptographic flaws in SSL certificates.
- Certificate transparency is another tool attackers can use to gather information about an organization's subdomains and systems.
- Security breaches can directly impact a company's reputation.

What Did I Learn in this Module? (Cont.)

- Attackers use various methods to gather information, including password dumps, file metadata, strategic search engine analysis, website archiving, and public source code repositories.
- Tools like h8mail and WhatBreach exploit breached data repositories, while ExifTool reveals metadata in files.
- Advanced search engine operators can uncover sensitive information, and website archiving allows for a historical view of websites.
- Open-source intelligence (OSINT) gathering involves collecting and analyzing publicly available information, with Recon-ng being a powerful framework for this purpose.
- Shodan scans the internet for vulnerable hosts and other exposed systems.
- Performing active reconnaissance involves enumeration, which is the process of gathering information about a target during a penetration test.
- The first step is to identify the target's internet-facing hosts, followed by a port scan to enumerate the services running on those hosts.
- Nmap is a popular tool for such scans, including SYN scans, TCP connect scans, UDP scans, and TCP FIN scans.
- A SYN scan sends a TCP SYN packet to the target port and analyzes the response to determine if the service is listening.

What Did I Learn in this Module? (Cont.)

- TCP connect scans use the operating system's networking mechanism to establish a full TCP connection, which may trigger alarms on intrusion detection systems.
- UDP scans are useful for enumerating services like DNS, SNMP, or DHCP, which use UDP for communication.
- TCP FIN scans send a FIN packet to the target port, and if no response is received, the port is considered open.
- Host discovery scans help determine if a host is online and responding on a network.
- Nmap also provides six timing templates (-T 0-5) to dictate the aggressiveness of a scan, ranging from very slow for IDS evasion to very aggressive, which may overwhelm targets or miss open ports.
- Enumeration techniques used in the information-gathering include: host enumeration, user enumeration, group enumeration, network share enumeration, web page enumeration/web application enumeration, service enumeration, and enumeration via packet crafting.
- Additionally, packet inspection and eavesdropping can be performed using tools like Wireshark, tshark, and tcpdump, aiding in passive reconnaissance during penetration testing.
- Vulnerability scanning is the process of identifying weaknesses in a system by probing services to determine if they are vulnerable.

What Did I Learn in this Module? (Cont.)

- Vulnerability scanners use different methods, but typically follow a four-step process: discovery, software/version identification, vulnerability correlation, and report generation.
- However, these reports may contain false positives, so validation is crucial.
- There are various types of vulnerability scans including: unauthenticated (scanner operates without credentials), authenticated (scanner uses root-level access credentials), discovery (scanner identifies the attack surface of a target), full (scanner enables all scanning options), stealth (scanner minimizes noise to avoid detection), passive (scanner monitors and analyzes network traffic), and compliance (scanner checks for adherence to industry regulations).
- Each type of scan has its own strengths and limitations.
- For example, unauthenticated scans only show exposed network services, while authenticated scans provide more comprehensive information.
- Stealth scans are useful for production environments but may not detect all vulnerabilities.
- Compliance scans address specific industry requirements but can be challenging due to varying interpretations of regulations.
- Challenges to consider when running a vulnerability scan on a network or device include: **Best Time to Run a Scan, Determining Protocols in Use**, **Network Topology**, **Bandwidth Limitations**, **Query Throttling**, and **Fragile Systems/Nontraditional Assets**.

What Did I Learn in this Module? (Cont.)

- Running a vulnerability scan is the easy part of identifying potential threats; the main challenge lies in analyzing the results.
- Vulnerability scanning tools can produce false positives, which need to be eliminated to accurately identify actual vulnerabilities.
- Reducing false positives is particularly important when providing a report for a paid penetration testing assignment.
- Eliminating false positives involves validating version information and investigating the details of the vulnerability.
- Each vulnerability maps to items in the Common Vulnerabilities and Exposures (CVE) list, which should be examined to better understand the criteria.
- Various organizations and resources, such as US-CERT, NIST, JPCERT, CAPEC, CVE, CWE, and CVSS, provide helpful information for further investigation of vulnerabilities.
- When dealing with a vulnerability, it is important to determine its priority by assessing its severity, the number of affected systems, and other factors.
- Overall, properly analyzing vulnerability scan results involves a detailed examination of the tool's findings and prioritizing vulnerabilities for mitigation based on their severity and potential impact.

Reflection Questions

- Once the contract is signed and the penetration testing engagement is underway, it is time to learn as much as possible about the client's network, applications, and systems.
- Passive reconnaissance is a good way to start.
- What kinds of valuable information can be found from passive reconnaissance?
- Active reconnaissance is more intrusive than passive reconnaissance.
- Because of this, why does a tester need to be careful of when doing active reconnaissance?
- Many vulnerability scanners are automated.
- While vulnerability scanners are generally accurate, it is important to manually verify some results.
- Why is this true?
- When analyzing vulnerability scanner results, why is it important to be familiar with CVE, CWE, and CVSS?