UTSA®

The University of Texas at San Antonio™

IS 3513-001 Information Assurance and Security

Lab #01
The SimSpace Cyber Range

Student:

Dillen Dela Cruz, odv464

Prepared for Information Assurance and Security
8/29/2023
Professor: Cody Cunov

Contents

Introduction:

This lab aims to equip students with the tools and knowledge of several cyber defense tools hosted on SimSpace Cyber Range. Students will also be able to experiment with the Hunt Tools on a Kali-Hunt virtual machine (VM) and carry out various network scans. The end goal is to provide firsthand experience in reconnaissance (Geeks for Geeks) and into fields such as network auditing to scan traffic (Gautam) and network administration to discover active hosts (Buckbee).

Nmap:

Nmap or Network Mapper is a free and open-source utility that is designed to capture useful information on the architecture of a network, monitor hosts and open ports, as well as to identify any vulnerabilities in your network. Nmap was created by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich). At the heart of Nmap, its scripting engine consists of both a Lua interpreter and an NSE Library. Lua is a lightweight language designed for "extensibility" (Nmap.org) and possess extended libraries for interfacing with Nmap itself. NSE is what brings Lua and Nmap together. At this layer, NSE handles a wide variety of tasks such as allowing users to retrieve, write, and share scripts which can then be executed with the utmost efficiency and speed. With Nmap's diverse and wide range of utilities, it can be both useful to attackers and defenders

Attackers:

When put in the wrong hands Nmap can be very useful to Attackers. Intruders such as elite hackers, who make up 1-2% of intrusive activity, can use Nmap to utilize or even discover new vulnerabilities in a company's infrastructure. According to Emma Crockett, Nmap vulnerability scanners gather data about a target host, system, and networks, as well as enabling the testing and reporting of vulnerabilities (Crockett). If attackers were to discover flaws in the system, they could potentially deploy zero-day exploits. Although, before launching an attack, attackers can conduct various form of reconnaissance such as passive (Attackers process step 1), active (Attackers process step 2), and networking (step 4 of footprinting) reconnaissance, often using tools like Nmap to conduct these tasks (UTSA). The reason for reconnaissance is to gather extensive information about the computer network. Within this phase, port scanning will be done to determine what ports are open and if they are receiving and sending data. Nmap divides ports into 6 states: open, closed, filtered, unfiltered, open|filtered, closed|filtered. These states aren't inherent properties of the port but a description of how Nmap sees them (Nmap). Port scanning plays a crucial role in vulnerability testing and provides information such as "services that are running, users who own services, whether anonymous logins are allowed, and which network services require authentication(Fortinet)". As mentioned earlier, Nmap can collect a wide range of information about the system, making it useful to an attacker's reconnaissance or scanning efforts (step 2 of an anatomy of a hack) (UTSA). One vital piece of information that can also be determined by Nmap is the target's operating system. This important detail would give the attackers an advantage, as they can exploit different vulnerabilities that are specific to each OS, potentially gaining control over the entire system and or data on it. Since companies run many devices with the same OS a successful attack could have a widespread consequence.

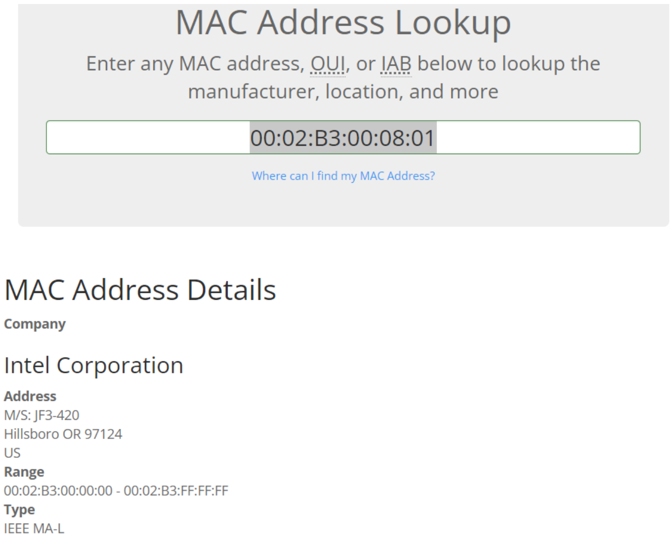
Defenders:

When put in the right hands, Nmap can be useful to defenders. In cybersecurity honeypot is a security tool used to lure intruders to “detect, deflect, and study hacking attempts to gain unauthorized access to information systems” (TechTarget). It is common to use Nmap to test a honeypot by imitating the behaviors of an attacker. Any attempts to access the honeypot will cause a trigger thus performing vulnerability and host discovery scans will test the performance of the virtual trap. As previously mentioned in the section above, Nmap can gather useful data that attackers can use to penetrate a system or execute there kill chain. However, defenders can use the same gathered data to bolster their network security. With Nmap’s vulnerability scanner, IT teams can assess how ill-protected their networks are. By simulating attacker activities, not only can the effectiveness of a honeypot be evaluated, but any limitations/weaknesses in the network can also be identified. Additionally, you’ll be able to find shadow ITs by examining the location of devices connected to the network. Shadow IT refers to the use of applications, services, software, and other technology systems without the explicit approval of the IT department (Kaseya). Detecting unauthorized devices early will allow network admins to respond by implementing different strategies such as restricting usage of third-party applications, maintaining physical inventory of devices and or monitoring the IT environment. The most important of these is the inventory of authorized and unauthorized devices as it aligns with the 20 Critical Security Controls, guiding actions against potential attacks (UTSA). Furthermore, Nmap’s port scanning capabilities allow for the evaluation of firewalls that are between the sender and the target of open and listening ports. This technique is better known as fingerprinting. A well-designed firewall can help prevent unauthorized access to your network as well as to control ports visibility and exposure. Reviewing Nmap's tools reveals their alignment with the NIST Cybersecurity Framework's key functions: Identify (vulnerability and port scanners), Protect (honeypot creation), Detect (honeypot usage, shadow IT, firewalls), Respond (tracking authorized and unauthorized devices), and Recover (firewall evaluation and vulnerability patching) (UTSA).

My Experience

Here, I will describe what I did for this section of work. I provide relevant information to the lab and discuss what I did and how. In lab 1, steps 5 through 15 require the usage of Nmap on windows.

Figure 1: MAC Address



MAC Address Lookup

Enter any MAC address, OUI, or IAB below to lookup the manufacturer, location, and more

00:02:B3:00:08:01

Where can I find my MAC Address?

MAC Address Details

Company

Intel Corporation

Address

M/S: JF3-420
Hillsboro OR 97124
US

Range

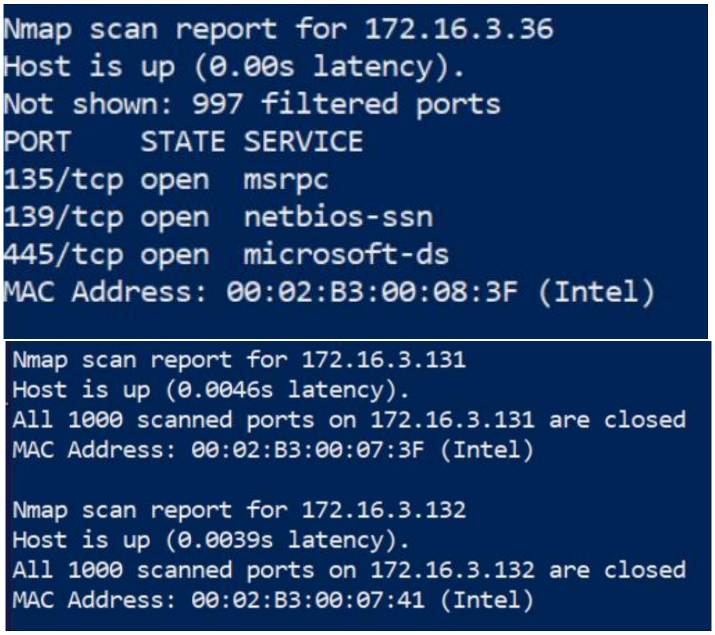
00:02:B3:00:00:00 - 00:02:B3:FF:FF:FF

Type

IEEE MA-L

In my first look into SimSpace each type of virtual machine seems to be numbered in order starting from one and has the same name convention; win-hunt-##. It appears that each machine has the same usernames and passwords and operates on Windows (x86-64). The x86-64 is a reference to the 64-bit version of the x86 processor architecture that Windows uses. This specification determines the way a processor functions when conducting various instructions from the OS and applications, with bits signifying how much information a CPU can process per cycle. On top of that, the virtualizations have 2 interfaces that have both private IP addresses that are both private and Mac addresses that lead to the vendor Intel Corporation. For the first half of the lab, I will be using win-hunt-01

Figure 2: Nmap scan -v

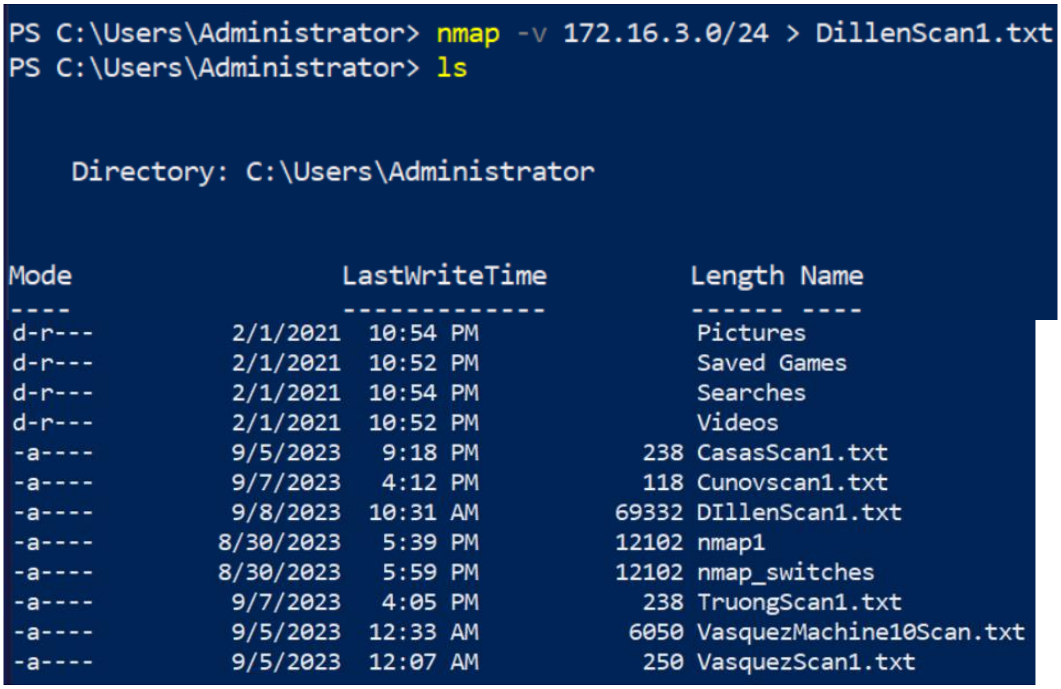


```
Nmap scan report for 172.16.3.36
Host is up (0.00s latency).
Not shown: 997 filtered ports
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:02:B3:00:08:3F (Intel)
```

```
Nmap scan report for 172.16.3.131
Host is up (0.0046s latency).
All 1000 scanned ports on 172.16.3.131 are closed
MAC Address: 00:02:B3:00:07:3F (Intel)

Nmap scan report for 172.16.3.132
Host is up (0.0039s latency).
All 1000 scanned ports on 172.16.3.132 are closed
MAC Address: 00:02:B3:00:07:41 (Intel)
```

nmap -v 172.16.3.0/24

The -v flag, which stands for verbose mode, is commonly used for troubleshooting and debugging in Nmap. To use this command the format would be ‘nmap -v [target]’ (Tutorials Point). When included in the command, it enables verbose output, providing additional information about the command's being run. Without the -v flag, Nmap would return only critical information. In this case, we are specifically interested in gathering information about the subnet 172.16.3.0/24. When I executed the code, it provided information on the various hosts and their associated IP addresses within the specified subnet. It also did a scan report for each address as well as a SYN Stealth Scan (a scan that helps determine the state of a port without establishing a full connection).



```
PS C:\Users\Administrator> nmap -v 172.16.3.0/24 > DillenScan1.txt
PS C:\Users\Administrator> ls


    Directory: C:\Users\Administrator


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---          2/1/2021  10:54 PM                Pictures
d-r---          2/1/2021  10:52 PM                Saved Games
d-r---          2/1/2021  10:54 PM                Searches
d-r---          2/1/2021  10:52 PM                Videos
-a----          9/5/2023   9:18 PM            238 CasasScan1.txt
-a----          9/7/2023   4:12 PM            118 Cunovscan1.txt
-a----          9/8/2023  10:31 AM          69332 DIllenScan1.txt
-a----         8/30/2023   5:39 PM          12102 nmap1
-a----         8/30/2023   5:59 PM          12102 nmap_switches
-a----          9/7/2023   4:05 PM            238 TruongScan1.txt
-a----          9/5/2023  12:33 AM           6050 VasquezMachine10Scan.txt
-a----          9/5/2023  12:07 AM            250 VasquezScan1.txt
```

Figure 3: Nmap scan > and ls

nmap –v 172.16.3.0/24 > YournameScan1.txt

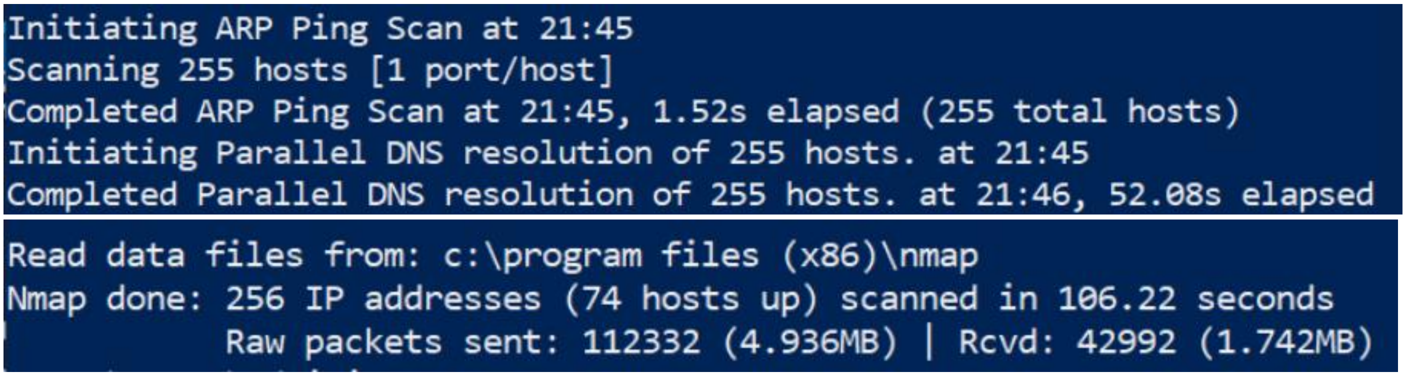
The > command allows you to ‘pipe’ the results of the previous command into a text document. In this case I saved the output on a file called DillenScan1. Storing the output allows you to resume aborted scans and compare results over time (NMAP). Comparing scans helps identify trends and changes in network security. These stored files can also serve as records for administrators and IT personnel.

ls

The ls command will show you the list files in a directory as well as ‘Mode’ and ‘LastWriteTime’. The mode represents the different file attributes that each file has. For example, '-a----' indicates that the file has been changed since the last backup (archive). 'd-r----' signifies a Read-only Directory, meaning users can only read the file and cannot modify it. This would be important during after an attack as you can see which files have been compromised and when those changes occurred

cat YournameScan1.txt

The command ‘**cat**’ will display the contents of the file. In this case the contents of the file will be the exact same output as the output for command ‘**nmap -v 172.16.3.0/24**’. This is crucial from a security perspective when dealing with suspicious files. By using the 'cat' command, you can inspect a file without executing it. This allows you to examine the file's contents and check for any potentially malicious scripts.



```
Initiating ARP Ping Scan at 21:45
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 21:45, 1.52s elapsed (255 total hosts)
Initiating Parallel DNS resolution of 255 hosts. at 21:45
Completed Parallel DNS resolution of 255 hosts. at 21:46, 52.08s elapsed
```

```
Read data files from: c:\program files (x86)\nmap
Nmap done: 256 IP addresses (74 hosts up) scanned in 106.22 seconds
           Raw packets sent: 112332 (4.936MB) | Rcvd: 42992 (1.742MB)
```

Figure 4:hosts and IP of Nmap scan -v

Looking deeper into the command ‘**nmap -v 172.16.3.0/24**’ we see that there were 256 IP addresses scanned and 74 hosts up. Nmap determined the number of hosts that were up by utilizing an ARP Ping scan, specifically designed for efficiently discovering hosts. Earlier I had mentioned that during the command execution, ‘**nmap -v 172.16.3.0/24**’ , it had run a stealth scan. Here is how it established the state of the ports:

Stealth Scan protocol (Pentest Tools):

1. The scanner sends an SYN packet.

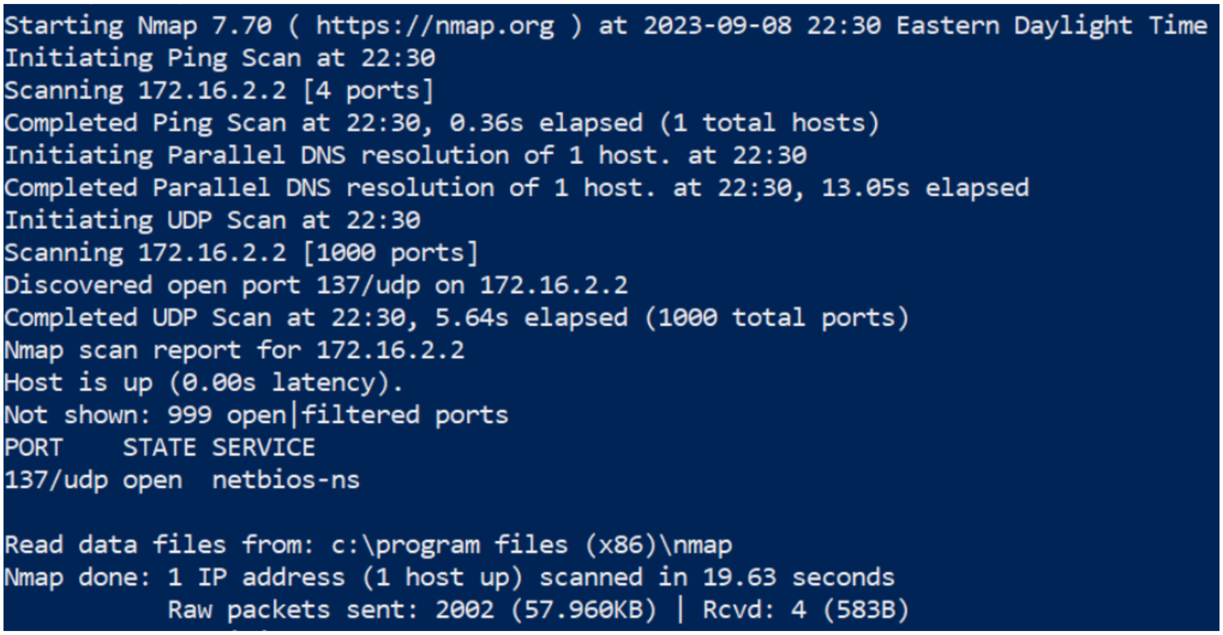
2. If the port is open, the machine replies with SYN/ACK.

3. If the port is closed the machine sends RST.

    a. TCP reset packet or an RST are sent to terminate a connection. It is an indicator that the port is closed and there is no listening service.

4. If no response is received after several retries, the port is marked as filtered.

5. Once the scanner receives SYN/ACK from the machine, it sends the RST packet and marks it as an open port.

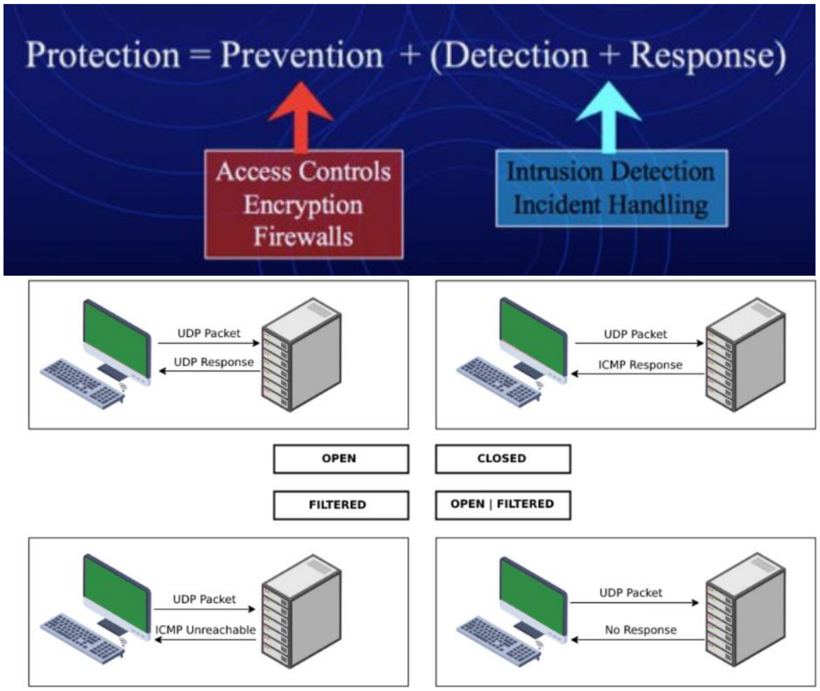


```
Starting Nmap 7.70 ( https://nmap.org ) at 2023-09-08 22:30 Eastern Daylight Time
Initiating Ping Scan at 22:30
Scanning 172.16.2.2 [4 ports]
Completed Ping Scan at 22:30, 0.36s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 22:30
Completed Parallel DNS resolution of 1 host. at 22:30, 13.05s elapsed
Initiating UDP Scan at 22:30
Scanning 172.16.2.2 [1000 ports]
Discovered open port 137/udp on 172.16.2.2
Completed UDP Scan at 22:30, 5.64s elapsed (1000 total ports)
Nmap scan report for 172.16.2.2
Host is up (0.00s latency).
Not shown: 999 open|filtered ports
PORT    STATE SERVICE
137/udp open  netbios-ns

Read data files from: c:\program files (x86)\nmap
Nmap done: 1 IP address (1 host up) scanned in 19.63 seconds
           Raw packets sent: 2002 (57.960KB) | Rcvd: 4 (583B)
```

Figure 5: Nmap scan -v -sU

nmap –v –sU 172.16.2.2 > YournameMachine10Scan.txt

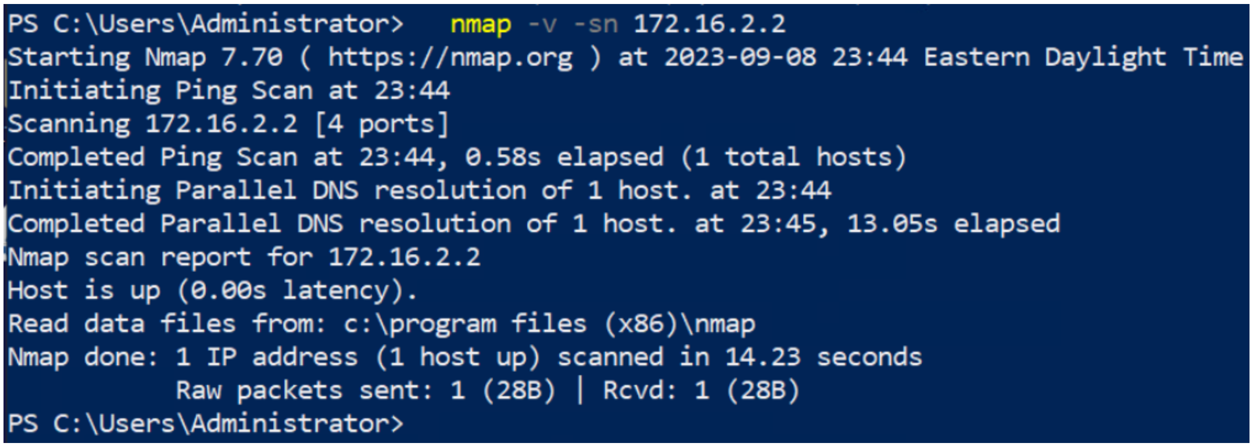
In this command, we applied what I've explained earlier and included the '-sU' option, commonly used for advanced scanning, specifically for UDP (User Datagram Protocol) scanning. To use it, you can follow the structure **'nmap -sU [target]'** (Tutorials Point). UDP scanning is the process in which we scan for the status of a UDP PORT on a target system. This scanning method is slower than regular TCP scans and is occasionally overlooked for this reason. However, it's a mistake to disregard it because UDP attacks are common over the internet. Attackers often use UDP in internet based DrDoS (Distributed Reflection Denial of Service) attacks, as these types of attacks are more effective over UDP (Plixer). UDP scanning is an essential tool to implement in network security as it can help “ identify potential targets for malicious activities quickly” (Conran). This enhances network security since prevention, in addition to detection and response, is essential for safeguarding your systems (The Computer Security Operational Model)



In this scan, one port was found to be open, and it was using a UDP service.

Figure 6 and 7 : The Computer Security Operational Model and status of ports

Here is how a UDP scan determines the status of a port (Jain).



```
PS C:\Users\Administrator>   nmap -v -sn 172.16.2.2
Starting Nmap 7.70 ( https://nmap.org ) at 2023-09-08 23:44 Eastern Daylight Time
Initiating Ping Scan at 23:44
Scanning 172.16.2.2 [4 ports]
Completed Ping Scan at 23:44, 0.58s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:44
Completed Parallel DNS resolution of 1 host. at 23:45, 13.05s elapsed
Nmap scan report for 172.16.2.2
Host is up (0.00s latency).
Read data files from: c:\program files (x86)\nmap
Nmap done: 1 IP address (1 host up) scanned in 14.23 seconds
           Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
PS C:\Users\Administrator>
```

Figure 8: Nmap scan -v -sn

nmap -v -sn 172.16.2.2

This command uses ‘**-sn**’, which means “no port scan”. This can be used to print out the available hosts that “responded to the host discovery probes” (Nmap), which in this case is 1. This is also known as a ping scan and is useful for quick target reconnaissance when used by itself. It helps attackers gather information without drawing much attention. The '-sn' option is more valuable than a list of IP addresses and host names because it identifies which hosts are active and responsive on the network. System administrators can perform ping sweep which allows them to “count available machines on a network or monitor server availability” (Nmap).

This aligns with Critical Security Control 1, which emphasizes maintaining an accurate inventory of all devices, including virtual ones, to ensure network security.

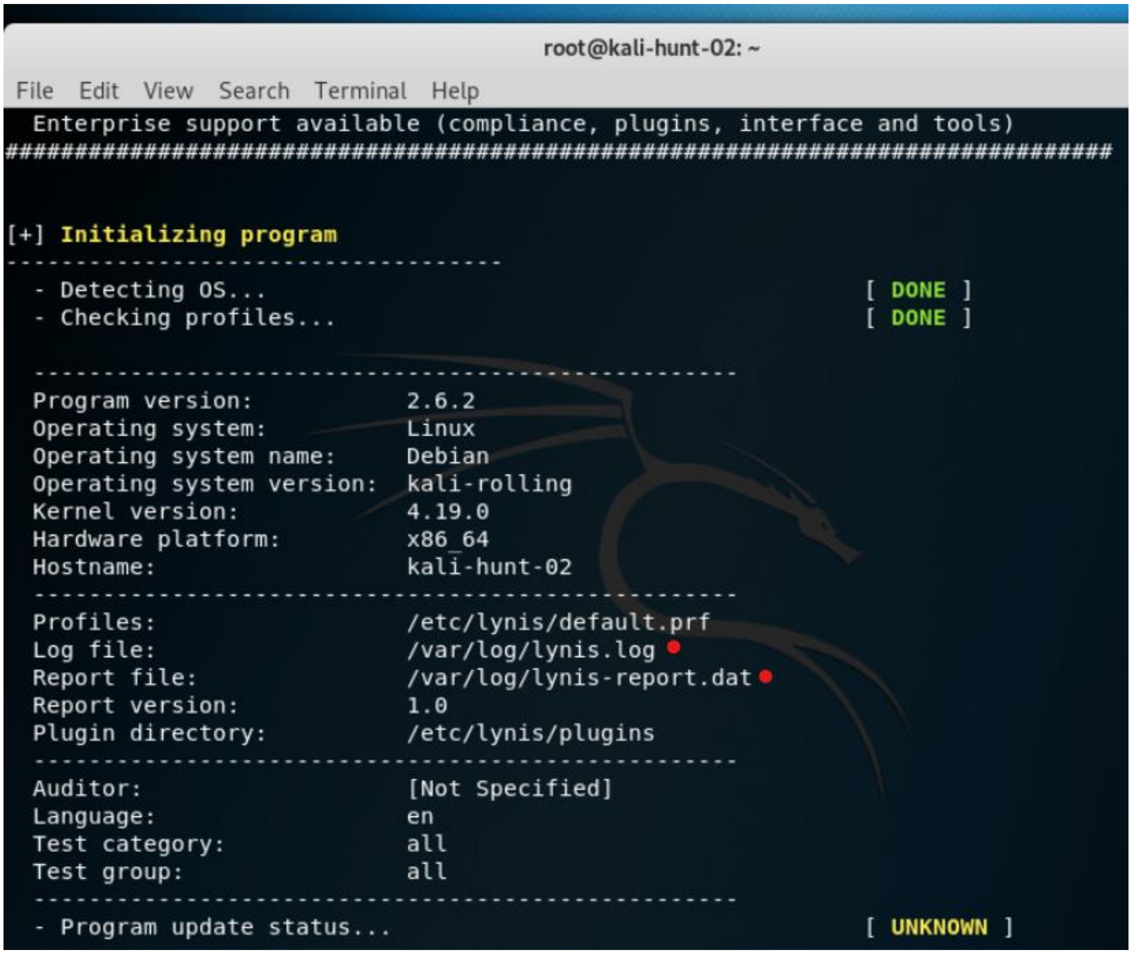
1. Vulnerability Analysis - Lynis

Lynis is a security auditing tool for any system running Linux, macOS, or Unix-Based operating system (CISOFY). This battle-tested security tool is widely utilized for security audits, penetration testing, vulnerability detecting, system hardening. Lynis is valuable for penetration testers or system administrators seeking to uncover security weaknesses that could become future threats. This tool aligns with Critical Security Controls 5 (maintenance, monitoring, and analysis of audit logs), 11 (secure configuration of network devices), and 20 (penetration tests and red team exercises). It achieves this alignment through its primary goal of system hardening, scanning for configuration issues, system information, and software vulnerabilities (GitHub). Furthermore, this open-source software is regularly updated and easy to run from a configuration folder if do not want to install it. Thus, you can run the software daily using a scheduled cronjob (creates jobs on a repeating schedule) ensuring continuous network testing and up-to-date information.

This is what happens during a typical scan with Lynis (CISOFY):

1. Initialization
    1. Perform basic checks, such as file ownership
    2. Determine operating system and tools
2. Search for available software components
3. Check latest Lynis version
4. Run enabled plugins
5. Run security tests per category
6. Perform execution of your custom tests (optional)
7. Report status of security scan

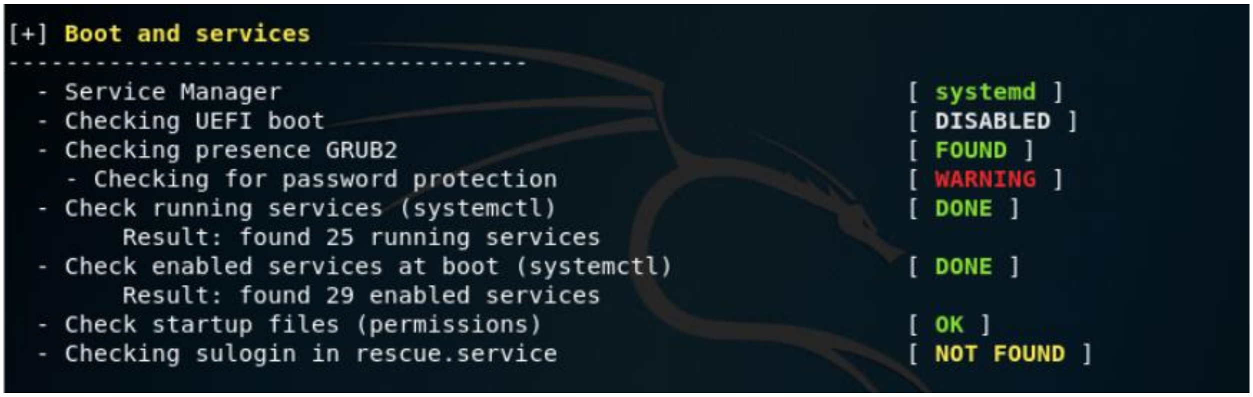
Figure 9: Lynis logo



```
  Enterprise support available (compliance, plugins, interface and tools)
################################################################################


[+] Initializing program
------------------------------------
  - Detecting OS...                                             [ DONE ]
  - Checking profiles...                                        [ DONE ]

  ---------------------------------------------------
  Program version:           2.6.2
  Operating system:          Linux
  Operating system name:     Debian
  Operating system version:  kali-rolling
  Kernel version:            4.19.0
  Hardware platform:         x86_64
  Hostname:                  kali-hunt-02
  ---------------------------------------------------
  Profiles:                  /etc/lynis/default.prf
  Log file:                  /var/log/lynis.log ●
  Report file:               /var/log/lynis-report.dat●
  Report version:            1.0
  Plugin directory:          /etc/lynis/plugins
  ---------------------------------------------------
  Auditor:                   [Not Specified]
  Language:                  en
  Test category:             all
  Test group:                all
  ---------------------------------------------------
  - Program update status...                                    [ UNKNOWN ]
```

Figure 10: Lynis audit

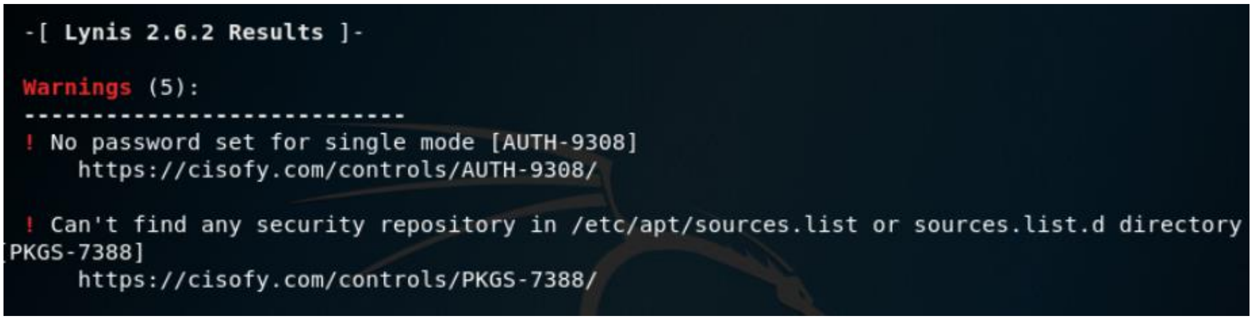
Here I initiated an audit scan using the code ***"sudo lynis audit system"***. When Lynis performs an audit it will go through several tests, divided into categories, which will display an output of all the results, debug information, and suggestions for hardening the system (Hogan). With every audit 2 files will be generated: the log file and the report file. The log file will have more thorough information and the report file will have the report data (general system info about server and application). Note that files are overwritten with each new audit, so results from will not be saved.



```
[+] Boot and services
------------------------------------
  - Service Manager                                       [ systemd ]
  - Checking UEFI boot                                    [ DISABLED ]
  - Checking presence GRUB2                               [ FOUND ]
    - Checking for password protection                    [ WARNING ]
  - Check running services (systemctl)                    [ DONE ]
        Result: found 25 running services
  - Check enabled services at boot (systemctl)            [ DONE ]
        Result: found 29 enabled services
  - Check startup files (permissions)                     [ OK ]
  - Checking sulogin in rescue.service                    [ NOT FOUND ]
```

Figure 11: Lynis Results

The first part of the output is the results of all the different tests by category. The information takes the form of different keywords such as the ones above but can also be **NONE**, **WEAK**, **NOT_FOUND, NOT ENCRYPTED, ENCRYPTED, etc.**



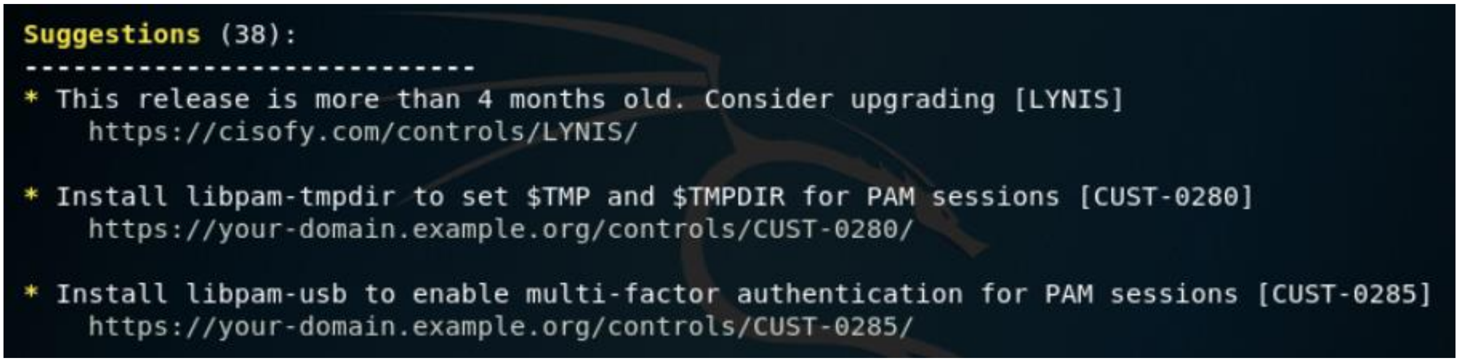
```
 -[ Lynis 2.6.2 Results ]-

 Warnings (5):
 ----------------------------
 ! No password set for single mode [AUTH-9308]
     https://cisofy.com/controls/AUTH-9308/

 ! Can't find any security repository in /etc/apt/sources.list or sources.list.d directory
[PKGS-7388]
     https://cisofy.com/controls/PKGS-7388/
```

Figure 12: Lynis Warning

The second part of the output consists of warnings, which may not appear in every audit. Each warning includes two lines of text:

1. The first line presents the warning text itself, enclosed in brackets to indicate the test that generated the warning.
2. The second line offers guidance on resolving the issue, whenever a suggested solution is available.

When you are having a problem with understanding the results from output one or even the warnings you can always inquire about Lynis using the test id. The command to do so is ***"sudo lynis show details test-id"*** where test-id is the information in brackets. If we wanted to know more information about the warning you would input: ***"sudo lynis show details AUTH-9308"***. The output for each specific test will follow, as Lynis walks through each test.



```
Suggestions (38):
----------------------------
* This release is more than 4 months old. Consider upgrading [LYNIS]
    https://cisofy.com/controls/LYNIS/

* Install libpam-tmpdir to set $TMP and $TMPDIR for PAM sessions [CUST-0280]
    https://your-domain.example.org/controls/CUST-0280/

* Install libpam-usb to enable multi-factor authentication for PAM sessions [CUST-0285]
    https://your-domain.example.org/controls/CUST-0285/
```

Figure 13: Lynis Suggestions

The final output consists of suggestions provided by Lynis to enhance your server's security. This output format is similar to the warning output, with two lines of text for each suggestion. The first line presents the suggestion along with the test ID, while the second line provides a “Security Control URL” where you can find more detailed information about the suggestion (Hogan).

2. Information Gathering - Sparta

Sparta is a python GUI (Graphical User Interface) application that is capable of building network penetration testing, scanning, information gathering, and vulnerability assessments. This software is designed to accelerate the scanning and enumeration phases before penetrating a system, producing faster results (GBHackers). An advantage of Sparta is its easy access toolkit that you can use with just one-click. You can use Sparta for reconnaissance to obtain information on live hosts in the network, services running on the host, or generate reports by scanning IP addresses and website domain names. It also offers easy save functionality, enabling you to reopen saved scans at any time from within the platform. In addition, Sparta supports brute-force attacks, using the brute option tab, to find default credentials for the common services running on the host. To perform basic functions such as brute-force attacks, taking screenshots of hosts, and or adding hosts to the tool, Sparta requires the use of Nmap, Hydra, and Cutycapt (Hacking Loops).

Critical Security Controls that Sparta aligns with:

CIS 3 – Continuous vulnerability Assessment and Remediation

- Sparta's vulnerability assessments can help identify and manage vulnerabilities in the network

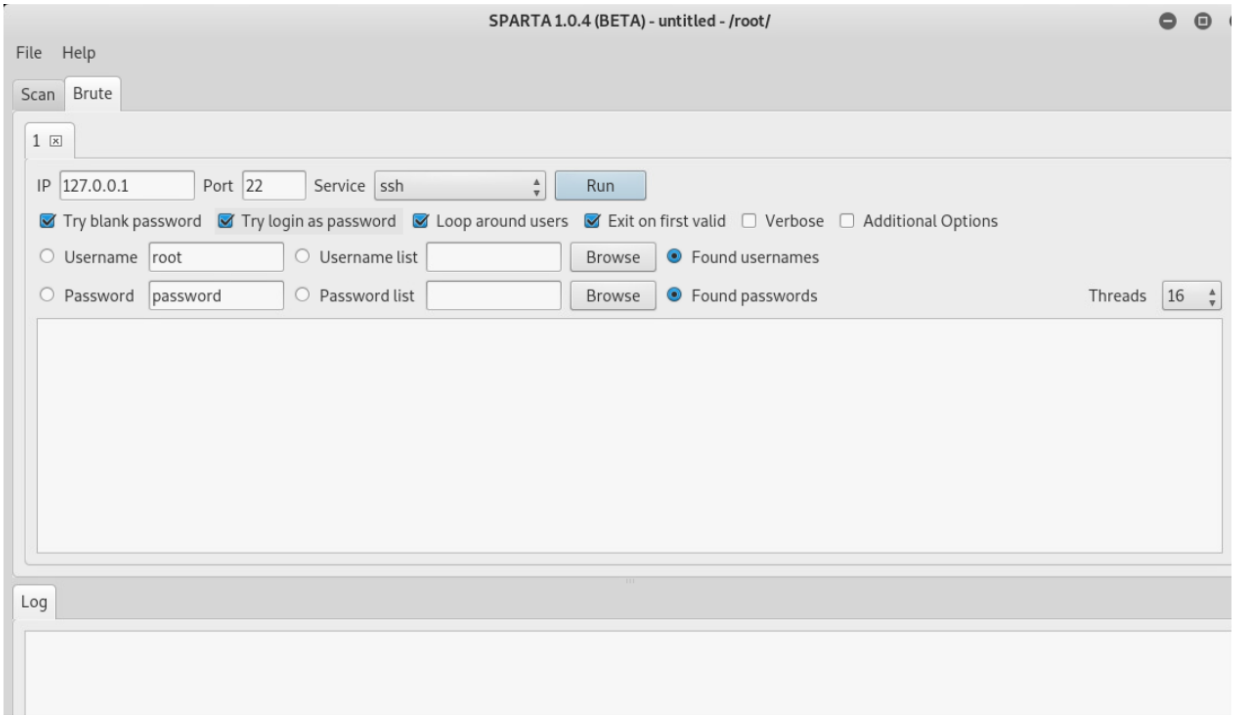
CIS 19 – Incident Response and Management

- Using CIS 3 can help administrators plan accordingly and enforce regulations in an effective and efficient manner during and after an attack

CIS 20 – Penetration Tests and Red Team Exercises

- Sparta's penetration testing can mitigate vulnerabilities by identifying them

Figure 14: Sparta logo


SPARTA 1.0.4 (BETA) - untitled - /root/

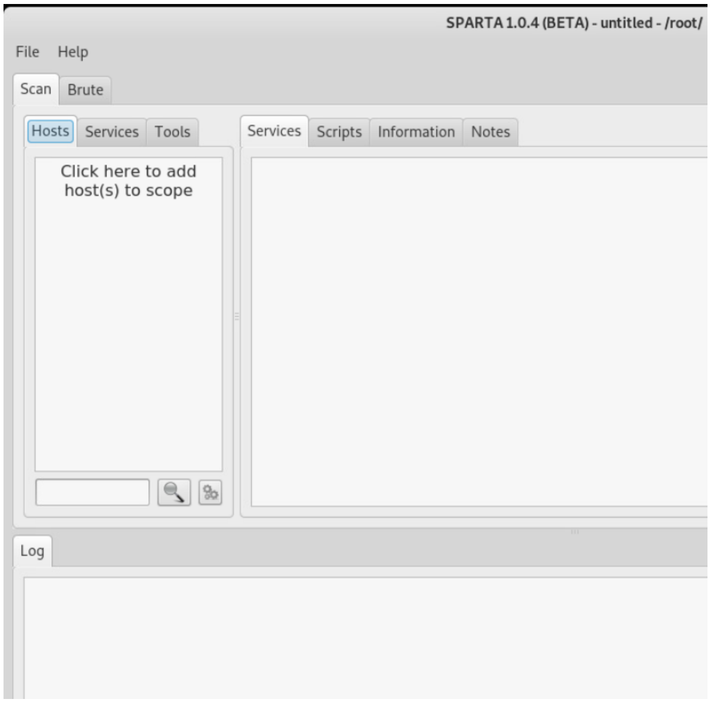
IP 127.0.0.1 Port 22 Service ssh

☑ Try blank password ☑ Try login as password ☑ Loop around users ☑ Exit on first valid □ Verbose □ Additional Options

○ Username root ○ Username list ● Found usernames

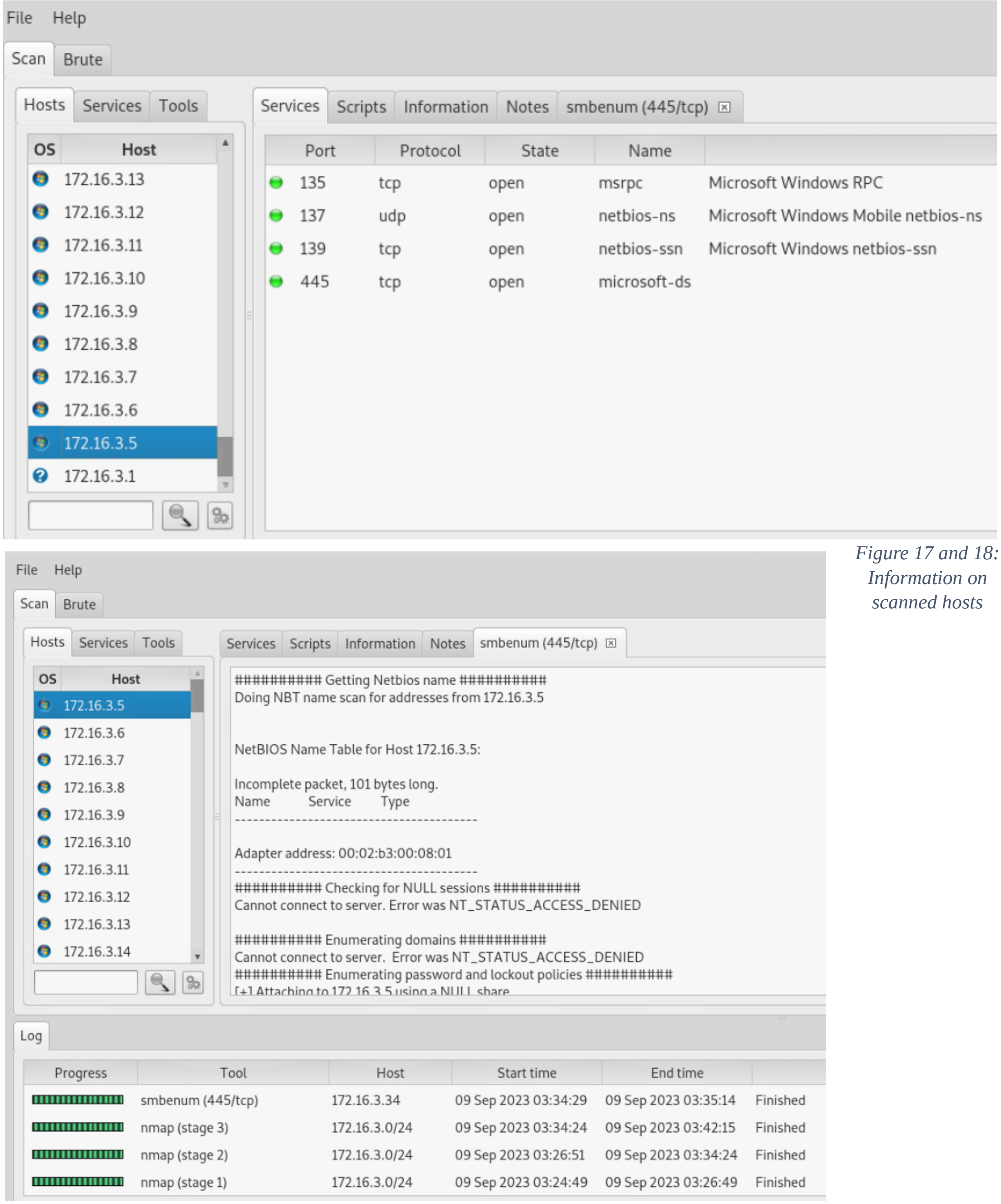
○ Password password ○ Password list ● Found passwords Threads 16

Figure 15 and 16: Sparta's Scan and Brute


SPARTA 1.0.4 (BETA) - untitled - /root/

Click here to add host(s) to scope

In these two images, we can see Sparta's two different graphical interfaces: one for 'Brute' and the other for 'Scan.' The 'Scan' option is divided into three sections. The first section on the left allows you to add host(s) to the scanning process. The top-right section displays information gathered during the scan, "including host information, network services, tool findings, and other useful details obtained by scanning open ports and running services (Hacking Loops)." The bottom box logs the scan's processes and allows you to abort the scan if needed. The 'Brute' (top image) option allows you to run brute-force attacks on the specified hosts and services listed in the 'Scan' option. The toolbar is user-friendly, with all options conveniently displayed. The graphical interface simplifies navigation through the tools required to run Sparta.



OS	Host
	172.16.3.13
	172.16.3.12
	172.16.3.11
	172.16.3.10
	172.16.3.9
	172.16.3.8
	172.16.3.7
	172.16.3.6
	172.16.3.5
?	172.16.3.1

	Port	Protocol	State	Name	
●	135	tcp	open	msrpc	Microsoft Windows RPC
●	137	udp	open	netbios-ns	Microsoft Windows Mobile netbios-ns
●	139	tcp	open	netbios-ssn	Microsoft Windows netbios-ssn
●	445	tcp	open	microsoft-ds	

OS	Host
	172.16.3.5
	172.16.3.6
	172.16.3.7
	172.16.3.8
	172.16.3.9
	172.16.3.10
	172.16.3.11
	172.16.3.12
	172.16.3.13
	172.16.3.14

```
########## Getting Netbios name ##########
Doing NBT name scan for addresses from 172.16.3.5


NetBIOS Name Table for Host 172.16.3.5:

Incomplete packet, 101 bytes long.
Name          Service        Type
----------------------------------------

Adapter address: 00:02:b3:00:08:01
----------------------------------------
########## Checking for NULL sessions ##########
Cannot connect to server. Error was NT_STATUS_ACCESS_DENIED

########## Enumerating domains ##########
Cannot connect to server.  Error was NT_STATUS_ACCESS_DENIED
########## Enumerating password and lockout policies ##########
[+] Attaching to 172.16.3.5 using a NULL share
```

Progress	Tool	Host	Start time	End time	
▮▮▮▮▮▮▮▮	smbenum (445/tcp)	172.16.3.34	09 Sep 2023 03:34:29	09 Sep 2023 03:35:14	Finished
▮▮▮▮▮▮▮▮	nmap (stage 3)	172.16.3.0/24	09 Sep 2023 03:34:24	09 Sep 2023 03:42:15	Finished
▮▮▮▮▮▮▮▮	nmap (stage 2)	172.16.3.0/24	09 Sep 2023 03:26:51	09 Sep 2023 03:34:24	Finished
▮▮▮▮▮▮▮▮	nmap (stage 1)	172.16.3.0/24	09 Sep 2023 03:24:49	09 Sep 2023 03:26:49	Finished

Figure 17 and 18: Information on scanned hosts

In this example, I used the IP range 172.16.3.0/24 as directed. This provided a list of all the IP addresses within the subnet. For each host, the scan revealed the open ports and the services running on them (top image). The 'smbenum (445/tcp)' tab was particularly useful for gathering detailed information on the host (bottom image). If you look at the logs you can see that by default Nmap scans all ports on a host for vulnerabilities and, if it detects suspicious files, takes action to eliminate them (Brisk Infosec). I attempted brute-force testing on an IP address, but the results were inconclusive. Interestingly, the tool switched from Nmap to Hydra. This is probably due to the fact that hydra is used to crack passwords of network services.

3. Password Attacks – John the Ripper

John the Ripper is a tool that administrators can use to find weak passwords and automatically mail users a warning about it. The platform is “typically used for password security auditing and password recovery (Rajalingham).” Key features about JTR are that it offers multiple modes such as single crack, wordlist, and incremental modes that makes password cracking faster. In addition, JTR can automatically detect hashing algorithm used by encrypted passwords and the easy configuration and set up of tools makes it popular amongst professionals. JTR also supports a plethora of encryptions by default but can further extend its capabilities through extensions (Mehnert). Furthermore, the software is already backed with a wordlist of common passwords making it very effective against weak passwords.

Critical Security Controls that John the Ripper aligns with:

CSC 3 – Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers

- John the Ripper can indirectly contribute to network device security by ensuring that strong passwords are used for device access.

CSC 13 – Data Protection

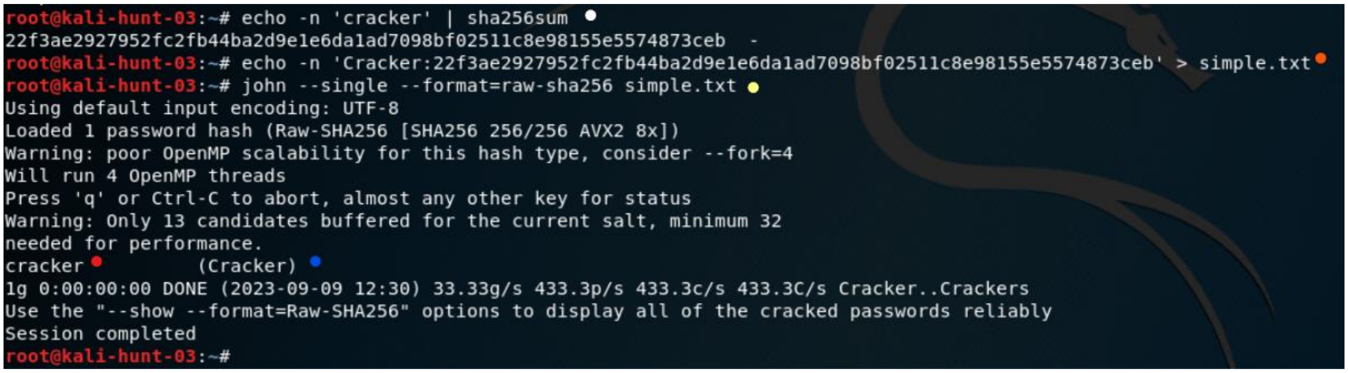
- John the Ripper helps protect data by identifying weak passwords. When weak passwords are discovered, administrators can take action to enhance data protection.

Figure 19: Jack the Ripper logo

Single crack mode – In single crack mode JTR will use generate variations of one string in order to generate a set of passwords. For example, if the username is “UTSA” and the password is “UtSa”, you would the single crack mode since John would be able to produce password variations of the word “UTSA” (“utSA”,”UTsa”,”Utsa”) until a match is found.

$ john --single --format=raw-sha1 [filename].txt

In this format the ‘single’ flag will let JTR know that we are using the single crack mode and the ‘format’ flag specifies the hash being used. You should use ‘--format’ if the hash is not specified. The [filename].txt contains the hash value of the password and the username.



```
root@kali-hunt-03:~# echo -n 'cracker' | sha256sum ●
22f3ae2927952fc2fb44ba2d9e1e6da1ad7098bf02511c8e98155e5574873ceb  -
root@kali-hunt-03:~# echo -n 'Cracker:22f3ae2927952fc2fb44ba2d9e1e6da1ad7098bf02511c8e98155e5574873ceb' > simple.txt●
root@kali-hunt-03:~# john --single --format=raw-sha256 simple.txt ●
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 13 candidates buffered for the current salt, minimum 32
needed for performance.
cracker ●        (Cracker) ●
1g 0:00:00:00 DONE (2023-09-09 12:30) 33.33g/s 433.3p/s 433.3c/s 433.3C/s Cracker..Crackers
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed
root@kali-hunt-03:~#
```

Figure 20: Single Crack Mode

In this example, I performed a password crack using single mode. In the first line of code (white dot), I changed the string ‘cracker’ into a SHA-256-hashed password. From there I saved the ‘username:password’ in a file called simple.txt (orange dot). Lastly, I ran the code, **$ john --single --format=raw-sha1 [filename].txt**, to generate variations of the string Cracker (username, blue dot) to get me the correct password (cracker, red dot). I used the format flag to specify the hash type I am trying to find (yellow dot).

Conclusion:

In conclusion, the exploration of various cybersecurity tools and techniques play a critical role in enhancing an organization's security posture. The focus has been on tools such as Nmap, Lynis, Sparta, and John the Ripper, each serving a unique purpose in bolstering network security and vulnerability management.

Nmap, a versatile network scanner, aids both attackers and defenders in understanding network architecture and identifying potential vulnerabilities. It aligns with security controls that emphasize inventory management and network monitoring.

Lynis, a battle-tested security auditing tool, is essential for continuous vulnerability assessment and system hardening. Its alignment with key security controls ensures that administrators can continuously address weaknesses and maintain a robust security framework.

Sparta, a powerful GUI application, simplifies network penetration testing, reconnaissance, and vulnerability assessments. Its user-friendly interface and alignment with penetration testing controls make it a valuable asset for identifying and mitigating vulnerabilities.

John the Ripper (JTR), a password cracking tool, serves as a crucial component of password security auditing and recovery. Its ability to detect weak passwords aligns with security controls related to secure configurations and data protection, emphasizing the importance of strong password policies.

By leveraging these tools and aligning them with critical security controls and different security protocols, organizations can assess vulnerabilities, respond to incidents effectively, and mitigate threats. As cybersecurity continues to be a top priority for organizations, the insights and techniques presented in this paper provide valuable resources on how to protect assets, detect and respond to threats, and maintain a strong security posture in an ever-evolving threat landscape.

Citations:

Attackers

Buckbee, Michael. "How to Use Nmap: Commands and Tutorial Guide." *Varonis,* https://www.varonis.com/blog/nmap-commands. Accessed 29 August 2023.

Crockett, Emma. "How to Easily Run a Vulnerability Scan Using Nmap." *Datamation*, 23 March 2023, https://www.datamation.com/security/how-to-easily-run-a-vulnerability-scan-using-nmap/. Accessed 4 September 2023.

Fortinet. "What Is A Port Scan? How To Prevent Port Scan Attacks?" *Fortinet,* https://www.fortinet.com/resources/cyberglossary/what-is-port-scan. Accessed 4 September 2023.

Gautam, Shubham. "What is Nmap (Network Mapper) & How Does It Work?" *KnowledgeHut,* 16 August 2023, https://www.knowledgehut.com/blog/security/network-mapper. Accessed 29 August 2023.

Geeks for Geeks. "Nmap Scans for Cyber Security and Penetration Testing." *GeeksforGeeks,* 8 September 2022, https://www.geeksforgeeks.org/nmap-scans-for-cyber-security-and-penetration-testing/. Accessed 29 August 2023.

Nmap. "Chapter 4. Port Scanning Overview." *Nmap,* https://nmap.org/book/port-scanning.html#port-scanning-what-is-it. Accessed 4 September 2023.

Nmap.org. "Chapter 9. Nmap Scripting Engine." *Nmap*, https://nmap.org/book/nse.html. Accessed 29 August 2023.

Nmap.org. "Script Language." *Nmap,* https://nmap.org/book/nse-language.html. Accessed 29 August 2023.

Nmap.org. “Script Language.” *Nmap*, https://nmap.org/book/nse-language.html. Accessed 29 August 2023.

UTSA. “2.1 Information Security and Cyber War Techniques.” 2 October 2022, https://utsa.instructure.com/courses/12478/pages/2-dot-1-%7C-read-and-explore?module_item_id=979193. Accessed 4 September 2023.

Defenders

Kaseya. “Shadow IT: Why It Exists and How to Deal With It.” *Kaseya*, 13 August 2021, https://www.kaseya.com/blog/shadow-it/. Accessed 5 September 2023.

Mesevage, Tobias Geisler. “What Is Port Scanning?” *Datto*, https://www.datto.com/blog/what-is-port-scanning. Accessed 5 September 2023.

Palo Alto Networks. “What is a Port Scan?” *Palo Alto Networks*, https://www.paloaltonetworks.com/cyberpedia/what-is-a-port-scan. Accessed 5 September 2023.

TechTarget. “honeypot (computing).” https://www.techtarget.com/searchsecurity/definition/honey-pot?Offer=abMeterCharCount_var2. Accessed 5 September 2023.

UTSA. “1.2 National Security and the Government's Role.” *UTSA*, 2 October 2022, https://utsa.instructure.com/courses/12478/pages/1-dot-1-%7C-read-and-explore?module_item_id=979175. Accessed 5 September 2023.

Ahlawat, Abhishek. “x86 vs x64 : What is the difference between x86 and x64 Architecture.” *Studytonight*, 17 June 2023, https://www.studytonight.com/post/x86-vs-x64-what-is-the-difference-between-x86-and-x64-architecture#google_vignette. Accessed 6 September 2023.

Phoenix Nap. “x64 vs. x86: Key Differences {Features, Limitations, and Use Cases}.” *phoenixNAP*, 20 July 2022, https://phoenixnap.com/kb/x64-vs-x86. Accessed 6 September 2023.

Tech Terms. “x86-64 Definition.” *TechTerms.com*, 6 March 2021, https://techterms.com/definition/x86-64. Accessed 6 September 2023.

(NMAP)

Buckbee, Michael. “How to Use Nmap: Commands and Tutorial Guide.” *Varonis*, https://www.varonis.com/blog/nmap-commands. Accessed 8 September 2023.

Conran, Matt. “UDP Scan.” *Network Insight*, 10 October 2014, https://network-insight.net/2014/10/10/udp-scan/. Accessed 8 September 2023.

Dell. “Avamar: Windows File System backup does not clear up the archive attribute after a successful backup.” *Dell*, 20 November 2020, https://www.dell.com/support/kbdoc/en-us/000063781/windows-file-system-backup-does-not-clear-up-the-archive-attribute-after-a-successful-backup. Accessed 8 September 2023.

Jain, Sandeep. “What is UDP Scanning?” *GeeksforGeeks*, 16 September 2022, https://www.geeksforgeeks.org/what-is-udp-scanning/. Accessed 8 September 2023.

Lackey, Jason. “What is Port Scanning?” *Keysight*, 1 May 2020, https://www.keysight.com/blogs/tech/nwvs/2020/06/17/what-is-port-scanning. Accessed 8 September 2023.

Nmap. “Host Discovery Controls.” *Nmap*, https://nmap.org/book/host-discovery-controls.html. Accessed 8 September 2023.

NMAP. “Output.” *Nmap*, https://nmap.org/book/man-output.html. Accessed 8 September 2023.

Nmap. “Port Scanning Techniques.” *Nmap*, https://nmap.org/book/man-port-scanning-techniques.html. Accessed 8 September 2023.

O*Reilly. “Discovering hosts with ARP ping scans - Nmap 6: Network Exploration and Security Auditing Cookbook [Book].” *O'Reilly*, https://www.oreilly.com/library/view/nmap-6-network/9781849517485/ch02s07.html. Accessed 8 September 2023.

Pentest Tools. “Inside Nmap, the world's most famous port scanner.” *Pentest-Tools.com*, https://pentest-tools.com/blog/nmap-port-scanner. Accessed 8 September 2023.

Phoenix NAP. “Nmap Commands - 17 Basic Commands for Linux Network.” *phoenixNAP*, 14 May 2019, https://phoenixnap.com/kb/nmap-commands. Accessed 8 September 2023.

Plixer. “Internet Threats: UDP Scans – Plixer.” *Plixer*, https://www.plixer.com/blog/udp-scan-work/. Accessed 8 September 2023.

RSI. “What are the 20 CIS Critical Security Controls?” *RSI Security*, 24 June 2020, https://blog.rsisecurity.com/what-are-the-20-cis-critical-security-controls/. Accessed 8 September 2023.

Springer Link. “TCP Reset Injection.” https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_119. Accessed 8 September 2023.

stack overflow. “What are the possible 'Mode' values returned by PowerShell's Get-ChildItem cmdlet?” *Stack Overflow*, 8 February 2011, https://stackoverflow.com/questions/4939802/what-are-the-possible-mode-values-returned-by-powershells-get-childitem-cmdle. Accessed 8 September 2023.

Tutorials Point. “NMAP Cheat Sheet.” *Tutorialspoint*, 18 March 2020, https://www.tutorialspoint.com/nmap-cheat-sheet. Accessed 7 September 2023.

LYNIS

CISOFY. “How often should I run Lynis on my system?” *CISOfy*, https://cisofy.com/faq/how-often-should-i-run-lynis-on-my-system/. Accessed 7 September 2023.

CISOFY. “Lynis - Security auditing and hardening tool for Linux/Unix.” *CISOfy*, https://cisofy.com/lynis/#introduction. Accessed 6 September 2023.

GitHub. “CISOfy/lynis: Lynis - Security auditing tool for Linux, macOS, and UNIX-based systems. Assists with compliance testing (HIPAA/ISO27001/PCI DSS) and system hardening. Agentless, and installation optional.” *GitHub*, https://github.com/CISOfy/lynis. Accessed 6 September 2023.

Hogan, Brian. “How to Perform Security Audits With Lynis on Ubuntu 16.04.” *DigitalOcean*, 28 April 2017, https://www.digitalocean.com/community/tutorials/how-to-perform-security-audits-with-lynis-on-ubuntu-16-04. Accessed 7 September 2023.

Kamathe, Gaurav. “Scan your Linux security with Lynis.” *Opensource.com*, 12 May 2020, https://opensource.com/article/20/5/linux-security-lynis. Accessed 7 September 2023.

Kumwenda, Mwiza. “How to Perform Security Audits on Linux With Lynis.” *MakeUseOf*, 9 January 2022, https://www.makeuseof.com/perform-linux-security-audits-lynis/. Accessed 7 September 2023.

Sparta

Brisk Infosec. “Sparta.” *Briskinfosec*, 9 October 2018, https://www.briskinfosec.com/blogs/blogsdetail/Sparta. Accessed 9 September 2023.

Codec Networks. “How to use Sparta for Reconnaissance. < Blogs.” *Codec Networks*, 14 October 2017, https://www.codecnetworks.com/blog/use-sparta-reconnaissance/. Accessed 9 September 2023.

GBHackers. “SPARTA - Network Penetration Testing GUI Toolkit.” *GBHackers*, https://gbhackers.com/sparta-network-penetration-testing-gui-toolkit/. Accessed 9 September 2023.

GitHub. “SECFORCE/sparta: Network Infrastructure Penetration Testing Tool.” *GitHub*, https://github.com/SECFORCE/sparta. Accessed 9 September 2023.

Hacking Loops. “Sparta –Network Scanning and Enumeration Tool.” *HackingLoops*, https://www.hackingloops.com/sparta/. Accessed 9 September 2023.

Jain, Sandeep. “Sparta Tool in Kali Linux.” *GeeksforGeeks*, 6 January 2021, https://www.geeksforgeeks.org/sparta-tool-in-kali-linux/. Accessed 9 September 2023.

Medium. “Scan, Crack passwords using Sparta wich contains many features.” *Medium*, https://medium.com/@danielwebimprints/scan-crack-passwords-using-sparta-wich-contains-many-features-2ebfcdc0bda3. Accessed 9 September 2023.

Quina, Antonio, and Leonidas Stavliotis. “How to Discover & Attack Services on Web Apps or Networks with Sparta.” *Null Byte*, 12 June 2019, https://null-byte.wonderhowto.com/how-to/discover-attack-services-web-apps-networks-with-sparta-0167255/. Accessed 9 September 2023.

Shivanandhan, Manish. “How to Use Hydra to Hack Passwords – Penetration Testing Tutorial.” *freeCodeCamp*, 18 November 2022, https://www.freecodecamp.org/news/how-to-use-hydra-pentesting-tutorial/. Accessed 9 September 2023.

Jack the Ripper

guide, step. “Getting Started With John The Ripper On Kali Linux – InfosecScout.” *InfosecScout*, https://infosecscout.com/john-the-ripper-on-kali-linux/#google_vignette. Accessed 9 September 2023.

Ivanovs, Alex. “Bash Create File: A Comprehensive Guide.” *Stack Diary*, 17 April 2023, https://stackdiary.com/tutorials/bash-create-file/. Accessed 9 September 2023.

Kali. “john.” *Kali Linux*, https://www.kali.org/tools/john/. Accessed 9 September 2023.

Lee, Cassandra. “How to Use John the Ripper: A Quick and Easy Guide.” *StationX*, 10 August 2023, https://www.stationx.net/how-to-use-john-the-ripper/. Accessed 9 September 2023.

Mehnert, Jacob. “How to use John the Ripper in Kali Linux.” *iFixit*, 21 June 2023, https://www.ifixit.com/Guide/How+to+use+John+the+Ripper+in+Kali+Linux/150615. Accessed 9 September 2023.

Open Wall. “John the Ripper - frequently asked questions (FAQ).” *Openwall*, https://www.openwall.com/john/doc/FAQ.shtml. Accessed 9 September 2023.

“A Practical Guide To Linux Echo Command.” *Earthly.dev*, 9 February 2023, https://earthly.dev/blog/practical-guide-to-linux-echo-cmd/. Accessed 9 September 2023.

Rajalingham, Kalyani. “How to use John, the ripper in Kali Linux.” *Linux Hint*, https://linuxhint.com/john-ripper-kali-linux/. Accessed 9 September 2023.

Sharma, Ax. “John the Ripper explained: An essential password cracker for your hacker toolkit.” *CSO Online*, 1 July 2020, https://www.csoonline.com/article/569533/john-the-ripper-explained-an-essential-password-cracker-for-your-hacker-toolkit.html. Accessed 9 September 2023.

Shivanandhan, Manish. “How to Crack Passwords using John The Ripper – Pentesting Tutorial.” *freeCodeCamp*, 17 November 2022, https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/. Accessed 9 September 2023.

Terra, John. “The Top Eight Kali Linux Tools For 2023.” *Simplilearn.com*, 24 March 2023, https://www.simplilearn.com/top-kali-linux-tools-article. Accessed 9 September 2023.