TASK 1

Cyber Security Internship Report

Presented to:

Digital Empowerment Network

Presented by:

Ali Aitazaz [LinkedIn]

+92 332 5890142

www.digitalempowermentnetwork.org

Contents

Conducting Security Audits

1.Objective:

- Perform comprehensive security audits for a network.

2.Description:

- Evaluate the security posture of a network by identifying vulnerabilities and weaknesses provide recommendations to enhance security measures

3.Key Steps:

- Conducting a risk assessment and identifying potential threats.
- Using tools to scan for vulnerabilities.
- Reviewing security policies and procedures.
- Compiling a report with findings and recommendations.
- Presenting the audit results to stakeholders.

Phase 1:

Asset Discovery and Inventory Report: Tech Enthusiast Setup

Assets:

- Laptop
- Mobile phone
- Router

Laptop Specification:

- **Manufacturer:** Dell
- **Computer Name:** Ali Aitazaz
- **OS:** Windows 10 Home 64-bit
- **Processor:** 10th Gen Intel(R) Core (TM) i7-10750H @ 2.60GHz
- **RAM:** 16 GB
- **SSD:** 512GB
- **System Model:** Dell XPS 15 9500
- **Graphics Card:** NVIDIA GeForce GTX 1650 Ti

Mobile phone specification:

- **Manufacturer:** Samsung
- **Model:** Galaxy S21
- **Storage:** 128GB
- **Battery health:** 93%

Router specification:

- **Manufacturer:** TP-Link
- **Device type:** Archer C7
- **Hardware version:** 2.0
- **Software version:** 3.0.0

Inventory: Device Inventory

Physical Devices:

- Personal laptop: Dell XPS 15
- Router: TP-Link Archer C7

Virtual Devices:

- Ubuntu 20.04
- CentOS 7
- Windows Server 2019

Software Inventory:

- **Operating Systems:** Windows 10 Home, Ubuntu 20.04
- **Applications:** Antivirus, Nessus Professional, Wireshark, VPN, Minecraft, Instagram, Signal, Slack
- **Databases:** Dropbox, Microsoft OneDrive

Application Inventory:

- **Web Applications:** Chrome, Firefox, Docker
- **Cloud Services:** Dropbox, Microsoft OneDrive
- **API Endpoints:** NULL

Phase 2

Security hardening

Importance of security hardening in mitigating cybersecurity risks:

The practice of fortifying digital systems' and networks' defenses against cyberattacks is known as security hardening. It entails putting in place a variety of security measures, including software upgrades, access limits, encryption, and strong authentication techniques, to lessen the possibility of hostile activity, unauthorized access, and data breaches. Enhancing overall security posture and making it more difficult for hackers to exploit system vulnerabilities are the two main objectives of security hardening.

Picture yourself as the person in charge of protecting a museum full of rare and precious items. To keep everything safe, you've got security guards patrolling, cameras watching every corner, and strict rules about who can go where. This is a lot like what we mean by "security hardening" in the digital world.

Chosen Controls:

1. **Ensure 'Maximum password age' is set to '365 or fewer days, but not 0'**

This policy setting defines how long a user can use their password before it expires. Values for this policy setting range from 0 to 999 days. If you set the value to 0, the password will never expire.

2. **Ensure 'Minimum password age' is set to '1 or more day(s)' (Automated)**

This policy setting determines the number of days that you must use a password before you can change it. The range of values for this policy setting is between 1 and 999 days. (You may also set the value to 0 to allow immediate password changes.) The default value for this setting is 0 days.

3. **Ensure 'Account lockout threshold' is set to '5 or fewer invalid logon attempt(s), but not 0' (Automated)**

This policy setting determines the number of failed logon attempts before the account is locked. Setting this policy to 0 does not conform to the benchmark as doing so disables the account lockout threshold.

4. **Ensure 'Require a password when a computer wakes (on battery)' is set to 'Enabled' (Automated)**

Specifies whether or not the user is prompted for a password when the system resumes from sleep.

5. **Ensure 'Require a password when a computer wakes (plugged in)' is set to 'Enabled' (Automated)**

Specifies whether or not the user is prompted for a password when the system resumes from sleep.

6. **Ensure 'Allow a Windows app to share application data between users' is set to 'Disabled' (Automated).**

Manages a Windows app's ability to share data between users who have installed the app. Data is shared through the Shared Local folder. This folder is available through the Windows Storage API.

7. **Ensure device firmware is up to date (Manual)**

Ensure that the device is kept up to date with security patch levels. The recommended state for this setting is: Apply updates.

8. **Ensure 'Screen Lock' is set to 'Enabled' (Manual)**

Enabling the Screen lock setting helps secure Android devices. This setting helps prevent unauthorized access to your device and protects data from being compromised.

9. **Do not connect to untrusted Wi-Fi networks (Manual)**

The Do not connect to untrusted Wi-Fi networks setting protects users from potential security threats when connecting to Wi-Fi networks. When this setting is enabled, the device will automatically avoid connecting to any Wi-Fi networks that it deems untrusted, based on a set of predefined criteria.

10. **Ensure 'Install unknown apps' is set to 'Disabled' (Manual)**

The Install unknown apps setting in Android allows users to install apps from sources other than the Google Play Store.

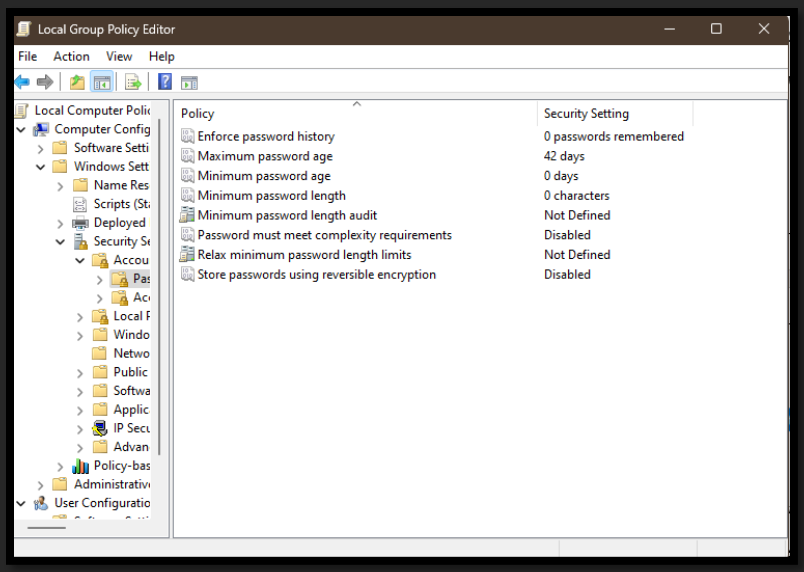
11. **Ensure 'Lock Screen Message' is configured (Manual)**

Set a message to be displayed on the locked screen. The recommended state for this setting is: Configure Lock Screen Message**.**

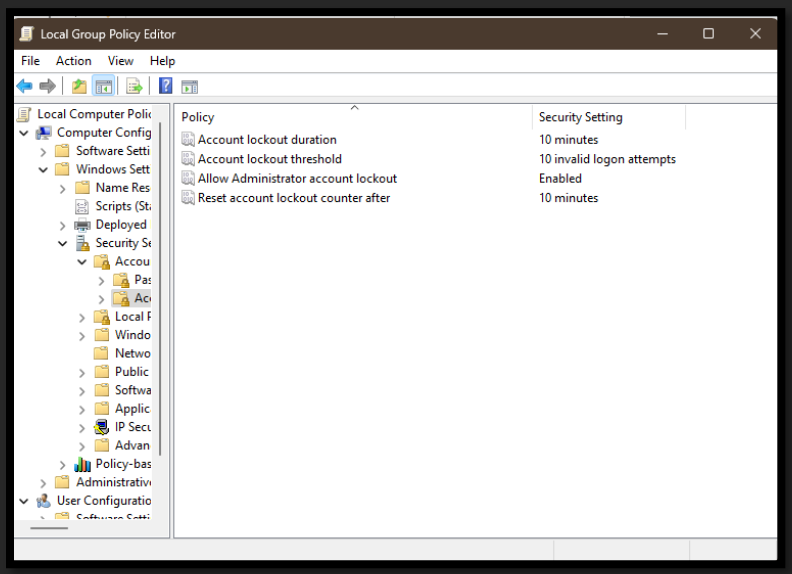
Implementation Steps:

A. Workstation

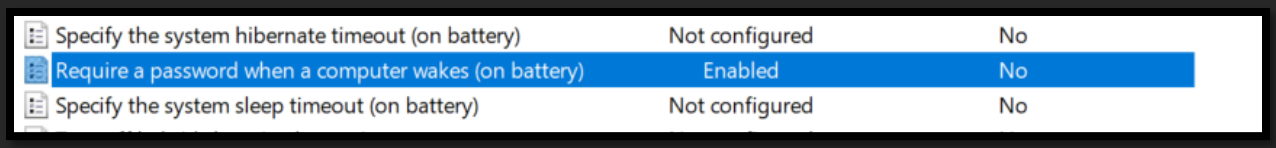
1.


Local Group Policy Editor
File Action View Help
Policy
Security Setting
Enforce password history
0 passwords remembered
Maximum password age
42 days
Minimum password age
0 days
Minimum password length
0 characters
Minimum password length audit
Not Defined
Password must meet complexity requirements
Disabled
Relax minimum password length limits
Not Defined
Store passwords using reversible encryption
Disabled

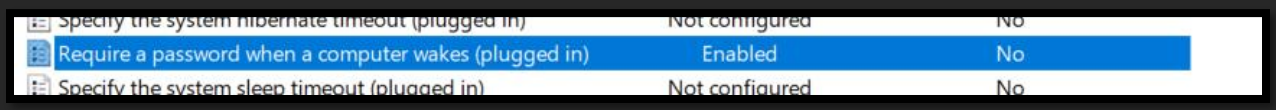
2.


Local Group Policy Editor
File Action View Help
Policy
Security Setting
Account lockout duration
10 minutes
Account lockout threshold
10 invalid logon attempts
Allow Administrator account lockout
Enabled
Reset account lockout counter after
10 minutes

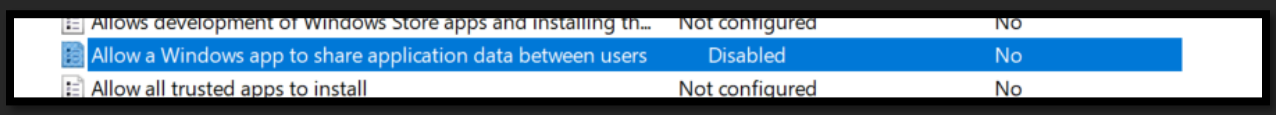
3.



4.

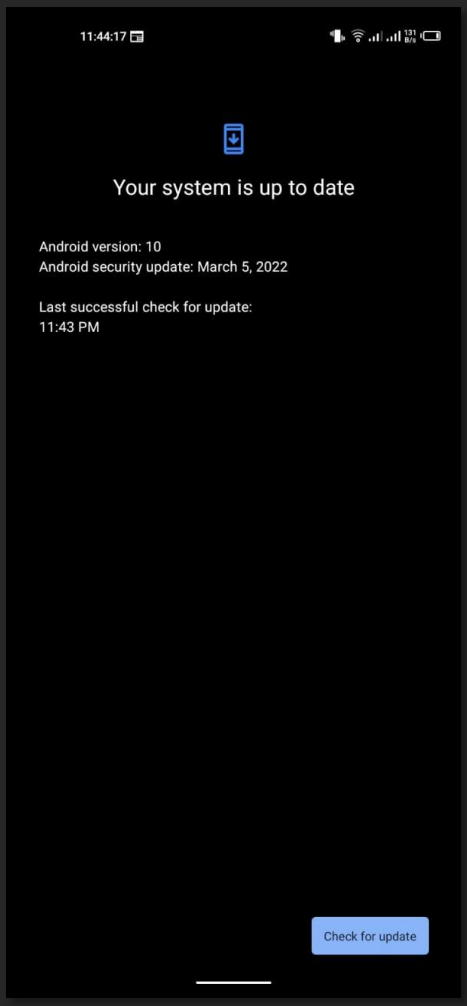


5.

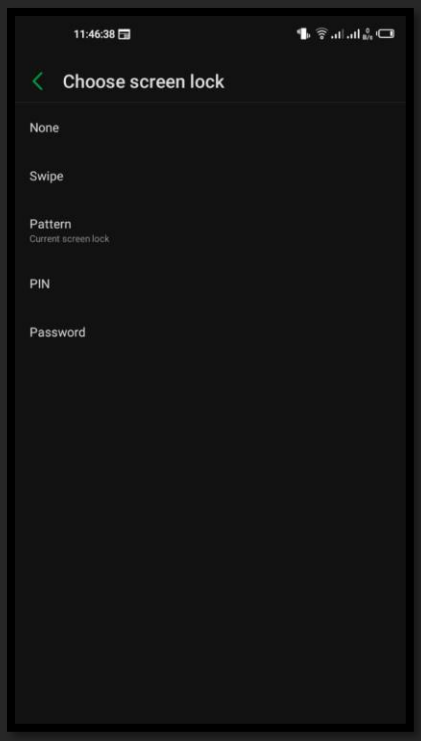


B. Mobile Device:

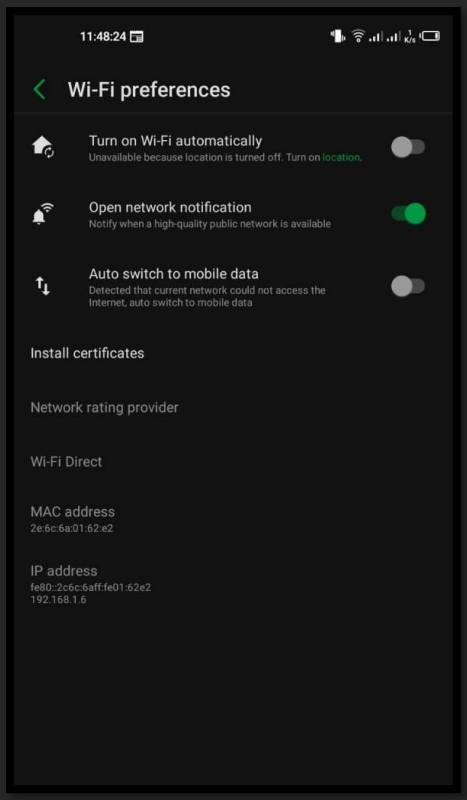
1. **Ensure device firmware is up to date (Manual):**



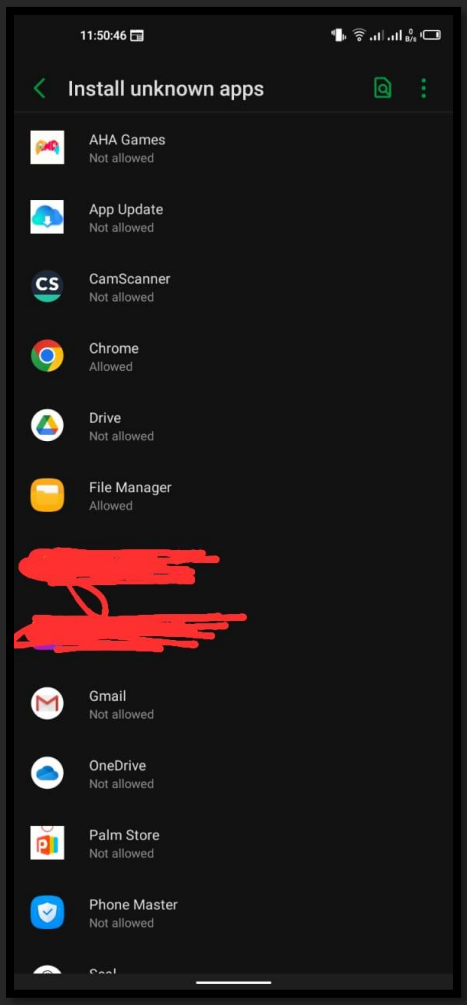
2. **Ensure 'Screen Lock' is set to 'Enabled' (Manual):**



3. **Do not connect to untrusted Wi-Fi networks (Manual):**



4. **Ensure 'Install unknown apps' is set to 'Disabled' (Manual):**



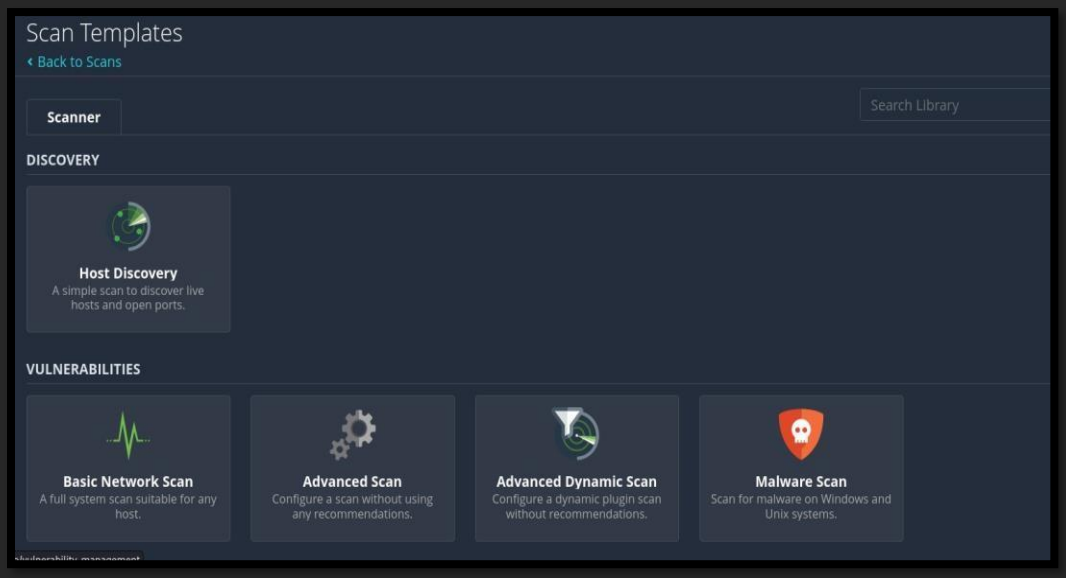
Conclusion:

You've significantly reduced the risk of internet threats to your laptop and phone by implementing these security measures on them. You may prevent potential hackers by using strong passwords, lockout thresholds, and other settings like screen locks and Wi-Fi choices. Software upgrades for your device also aid in closing security gaps. By taking these precautions, you demonstrate that you take online security seriously, which safeguards your personal data and fosters confidence in your digital endeavors.

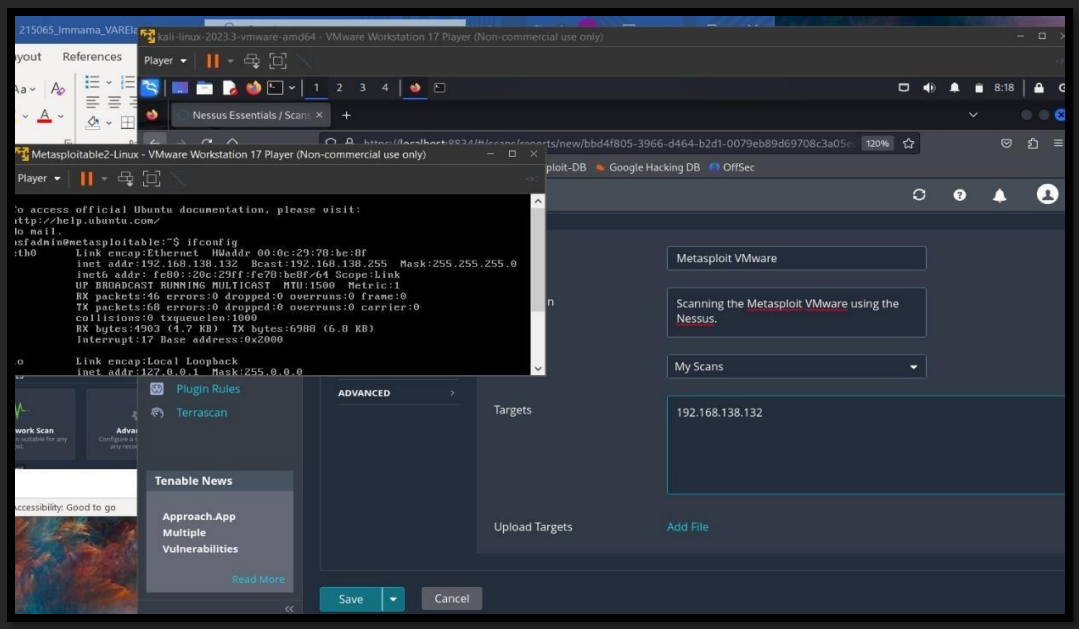
Phase 3

Normal Scan:

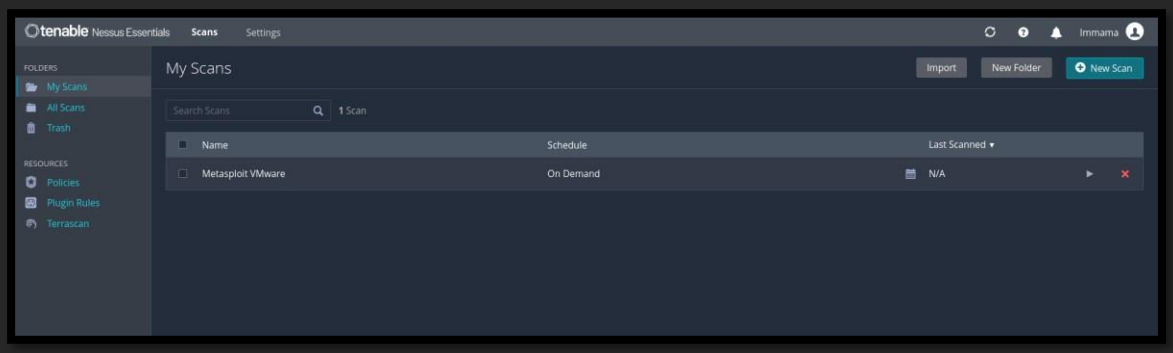
- Go to the “New Scan” and select the “Host Discovery”.
- Nessus Give us different and various scan options.



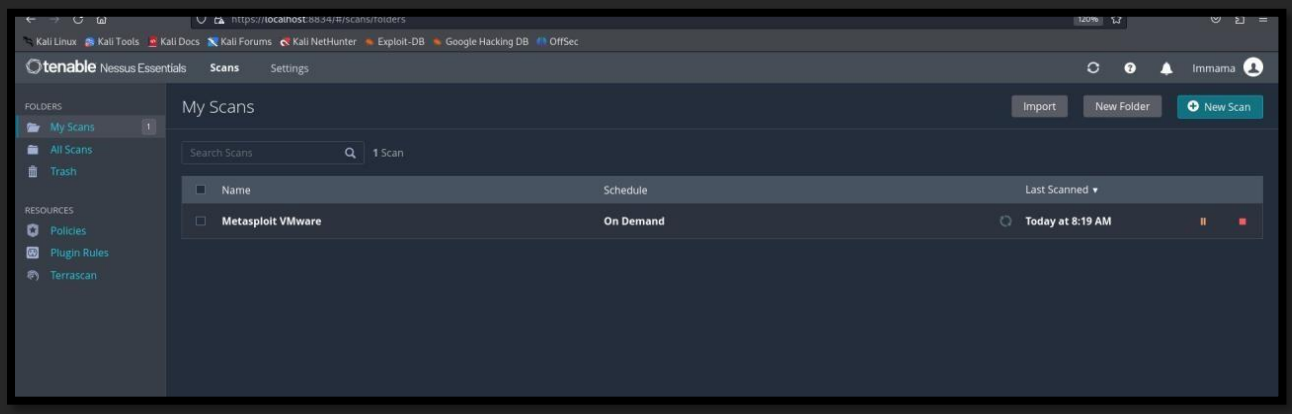
- Add the Details of Scan.
- Add the IP address of Metasploitable Machine.



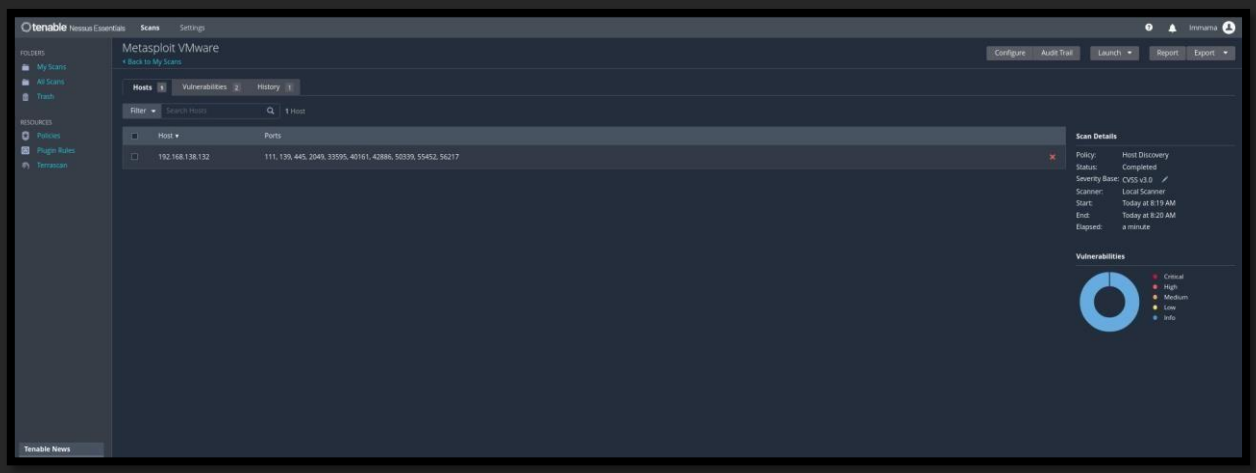
- Save the details.



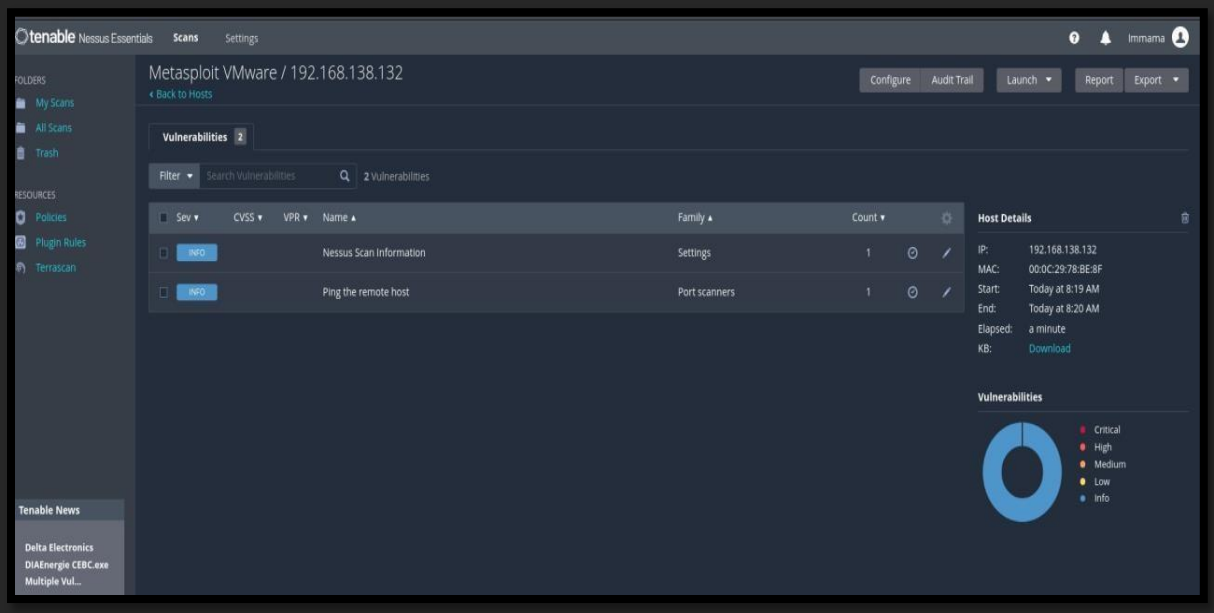
- And start the scan.



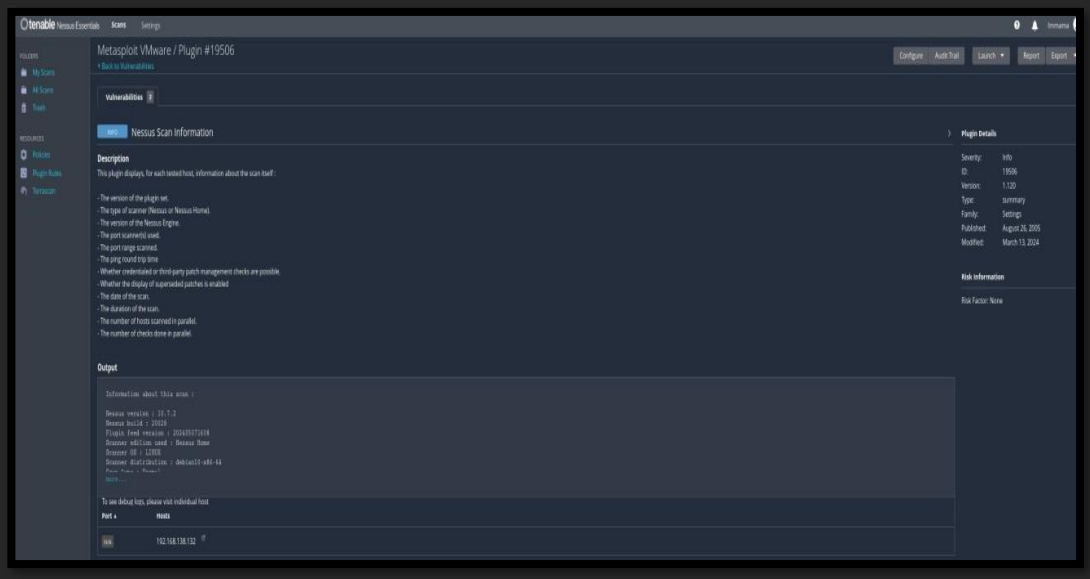
- After completion, following screen pops up:

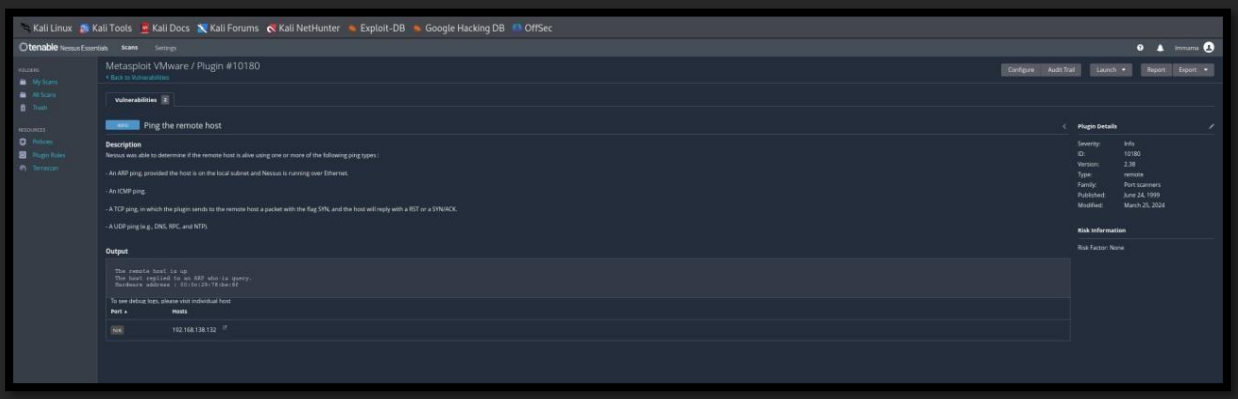


- Click on the IP tab, it will show details about the scan.
- The details are shown in below image:

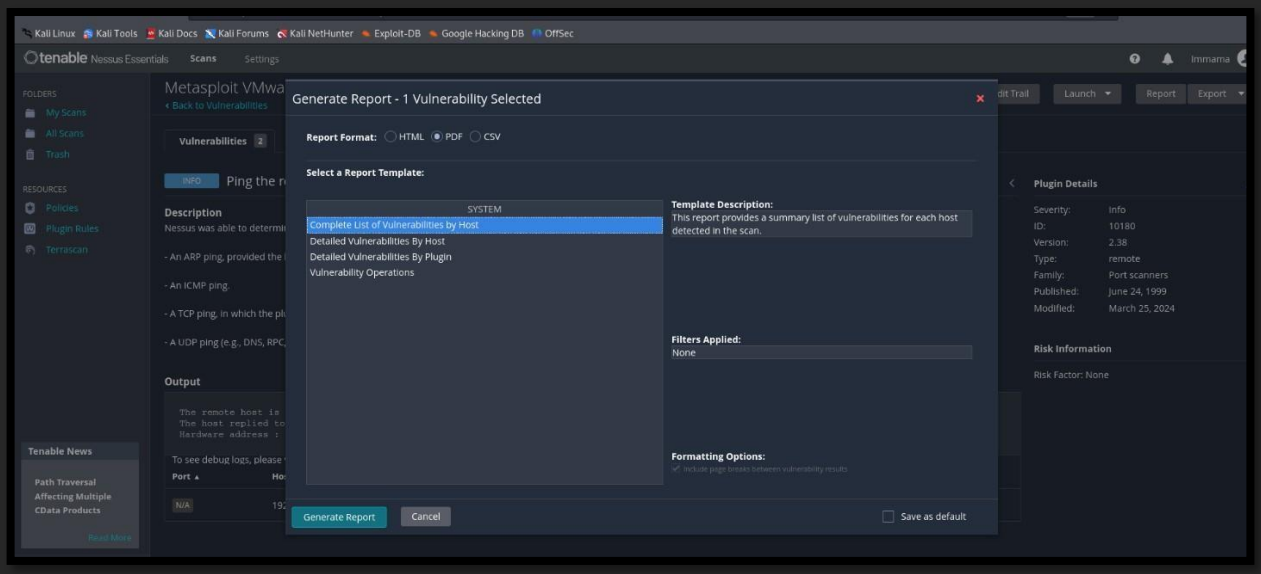


- Now, we further look for the details of each section that occur after the scan.

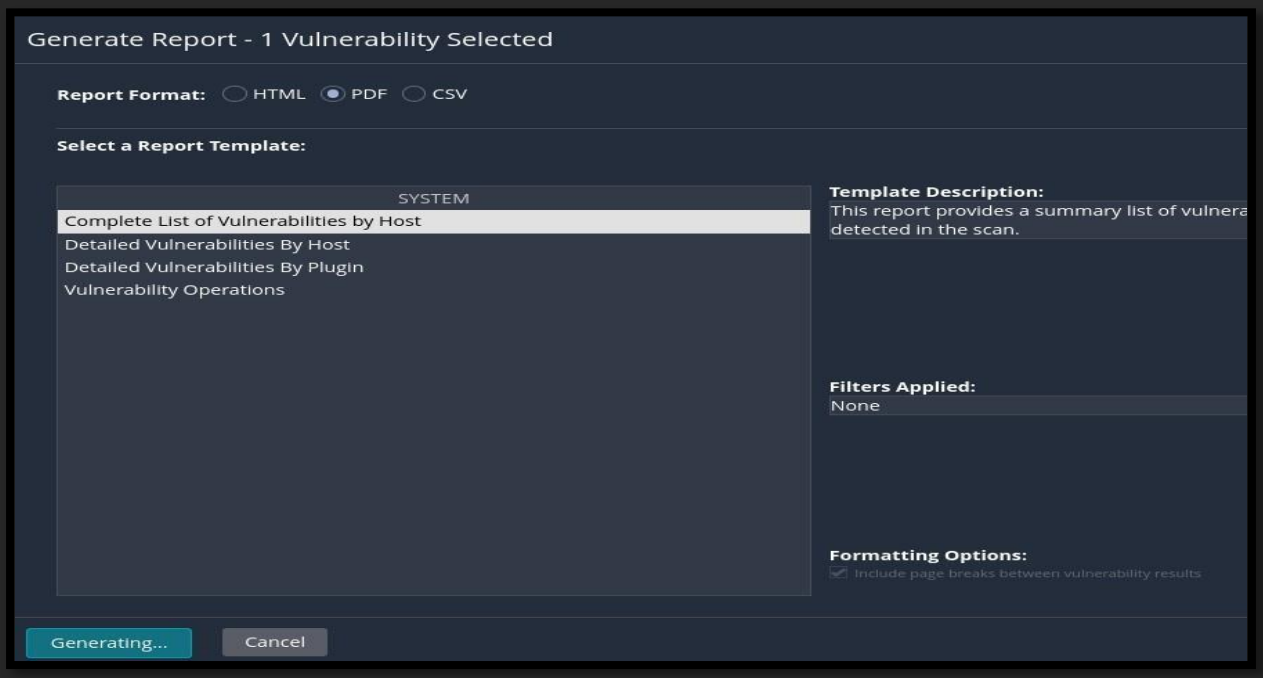




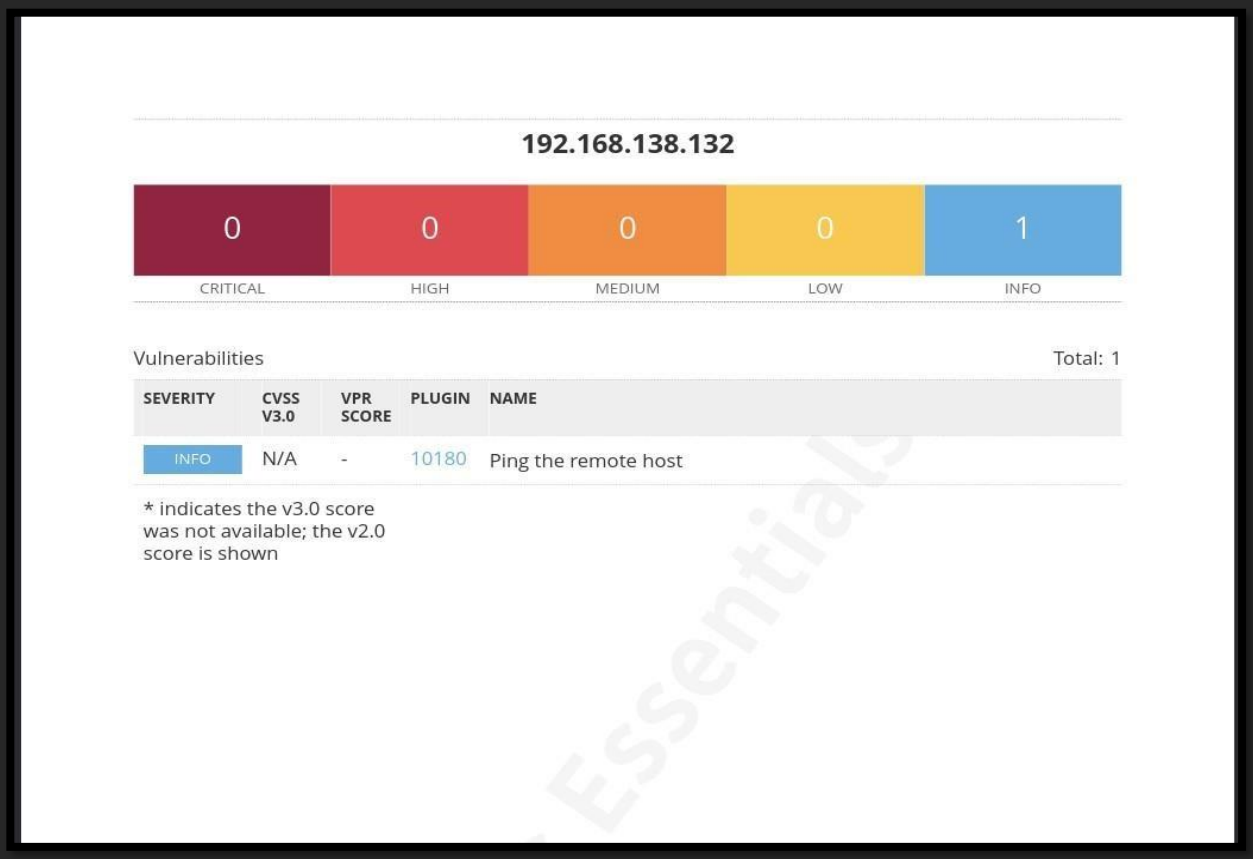
- And Generate the report, according to your specification.
- Nessus give various format and Templates.


Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
tenable Nessus Essentials
Scans
Settings
Immama
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Terrascan
Tenable News
Path Traversal Affecting Multiple CData Products
Read More
Metaspoit VMwa
Back to Vulnerabilities
Vulnerabilities 2
INFO
Ping the r
Description
Nessus was able to determi
- An ARP ping, provided the
- An ICMP ping.
- A TCP ping, in which the pl
- A UDP ping (e.g., DNS, RPC
Output
The remote host is
The host replied to
Hardware address :
To see debug logs, please
Port
Ho
N/A
19
Launch
Report
Export
Plugin Details
Severity: Info
ID: 10180
Version: 2.38
Type: remote
Family: Port scanners
Published: June 24, 1999
Modified: March 25, 2024
Risk Information
Risk Factor: None
Generate Report - 1 Vulnerability Selected
Report Format: HTML PDF CSV
Select a Report Template:
SYSTEM
Complete List of Vulnerabilities by Host
Detailed Vulnerabilities By Host
Detailed Vulnerabilities By Plugin
Vulnerability Operations
Template Description:
This report provides a summary list of vulnerabilities for each host detected in the scan.
Filters Applied:
None
Formatting Options:
Include page breaks between vulnerability results
Generate Report
Cancel
Save as default

- Generate the complete report.



The report shows the following details:



192.168.138.132

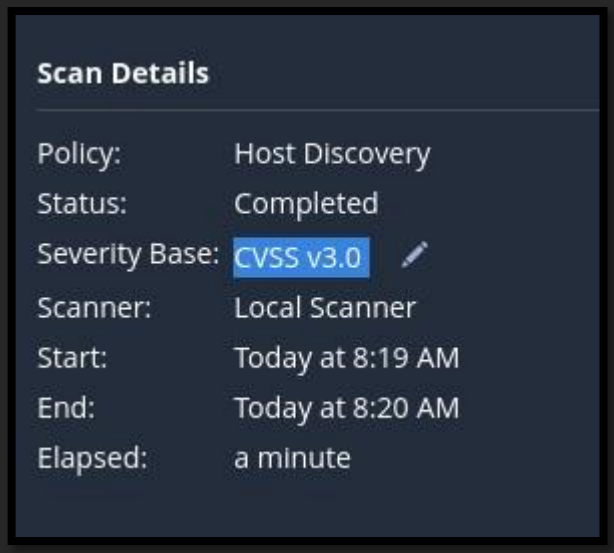
0	0	0	0	1
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 1

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
INFO	N/A	-	10180	Ping the remote host

* indicates the v3.0 score was not available; the v2.0 score is shown

- Explore the scanned details as well.



Scan Details

Policy:	Host Discovery
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 8:19 AM
End:	Today at 8:20 AM
Elapsed:	a minute

Common Vulnerability Scoring System v3.0: Specification Document

Also available in PDF format (316KiB) .

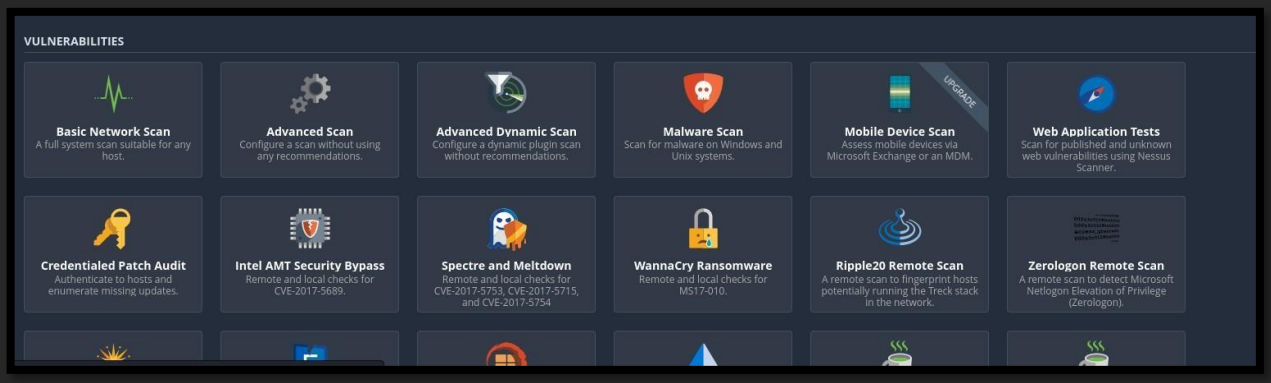
Resources & Links

Below are useful references to additional CVSS v3.0 documents.

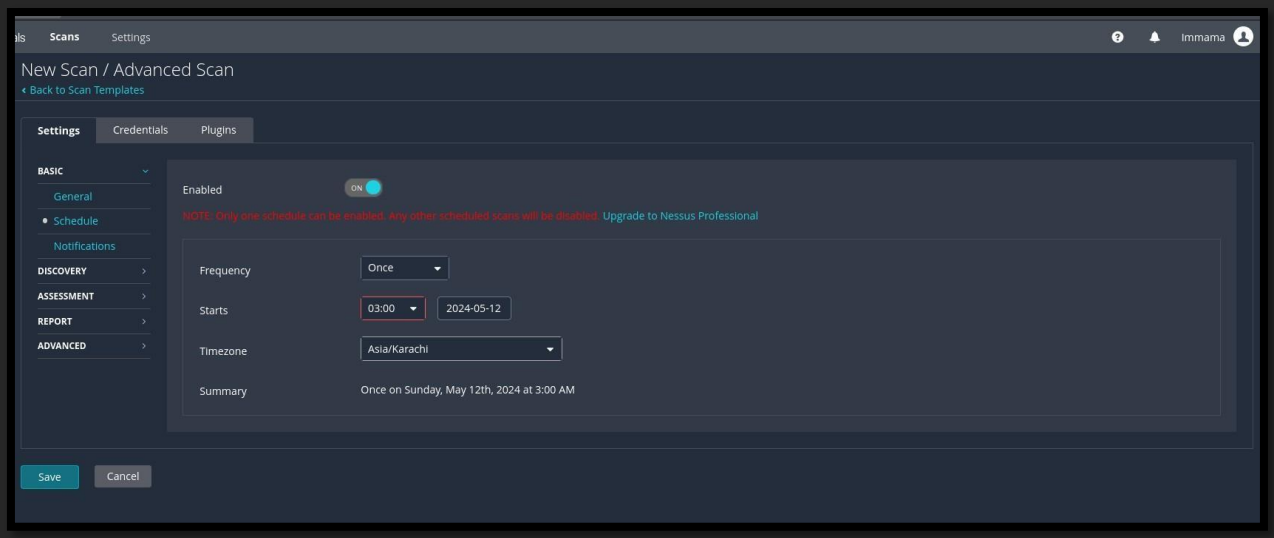
Resource	Location
Specification Document	Includes metric descriptions, formulas, and vector string. Available at, http://www.first.org/cvss/specification-document
User guide	Includes further discussion of CVSS v3.0, a scoring rubric, and a glossary. Available at http://www.first.org/cvss/user-guide
Example document	Includes examples of CVSS v3.0 scoring in practice. https://www.first.org/cvss/examples
CVSS v3.0 Calculator Use & Design	This guide covers the following aspects of the CVSS Calculator: Calculator Use, Changelog, Technical Design and XML Schema Definition. Available at http://www.first.org/cvss/use-design
CVSS v3.0 logo	Low and hi-res images available at http://www.first.org/cvss/identity
CVSS v3.0 calculator	Reference implementation of the CVSS v3.0 equations, available at http://www.first.org/cvss/calculator/3.0
JSON and XML schemas	JSON and XML schema definitions available at https://www.first.org/cvss/data-representations

Advance Scan:

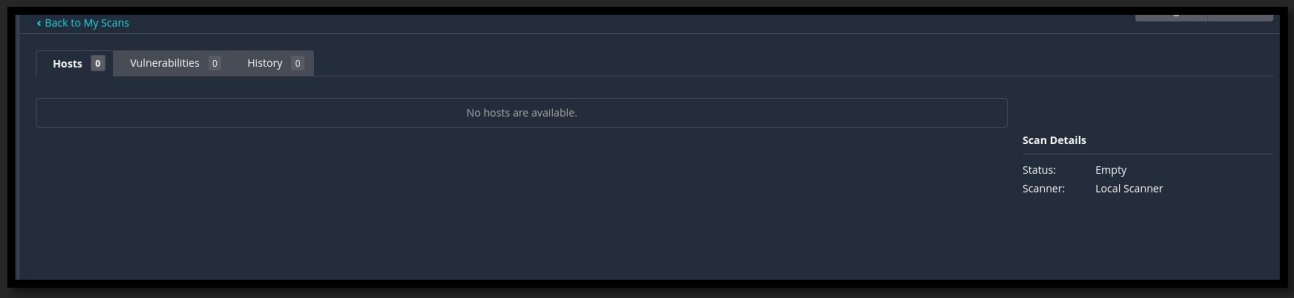
- Select the “Advance Scan” option from the Vulnerabilities portion.



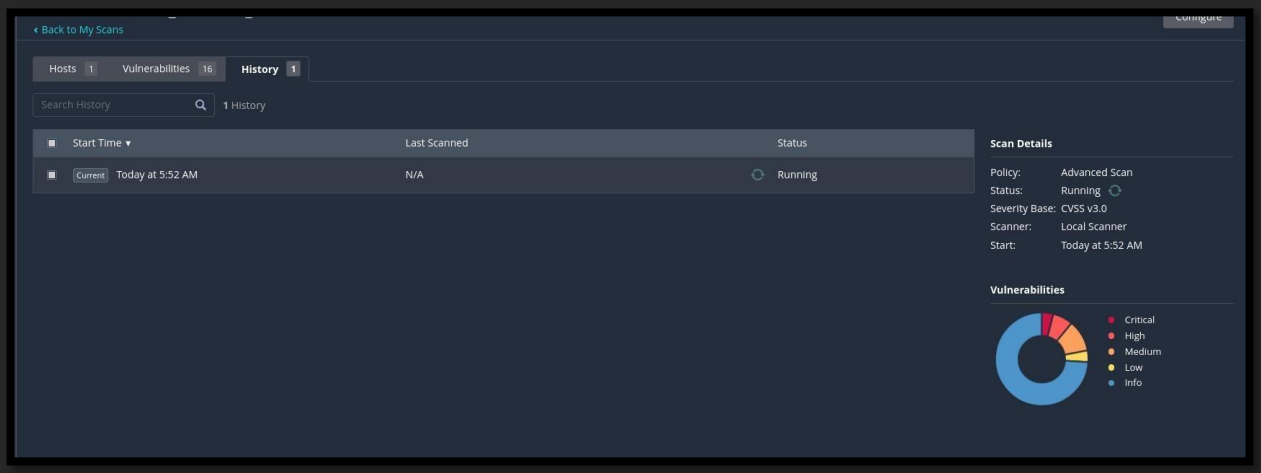
- Add details of the Scan such as IP of target machine
- Add Time Scheduling
- explore each option given in Advance scan.



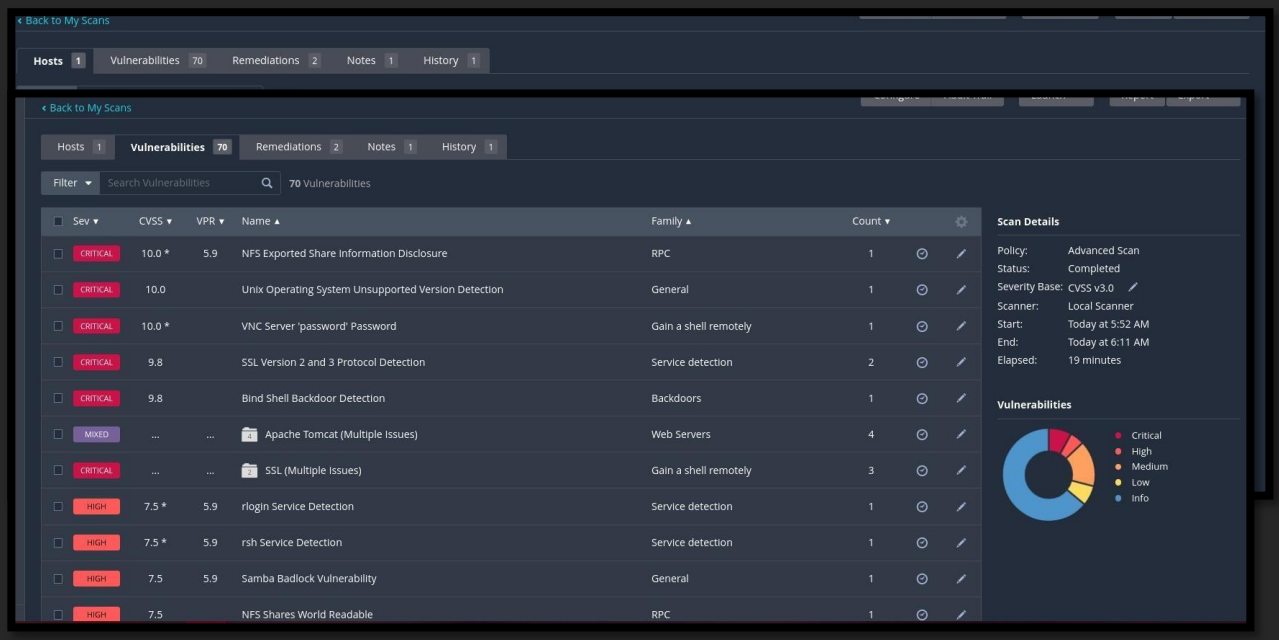
- Launch the scan.



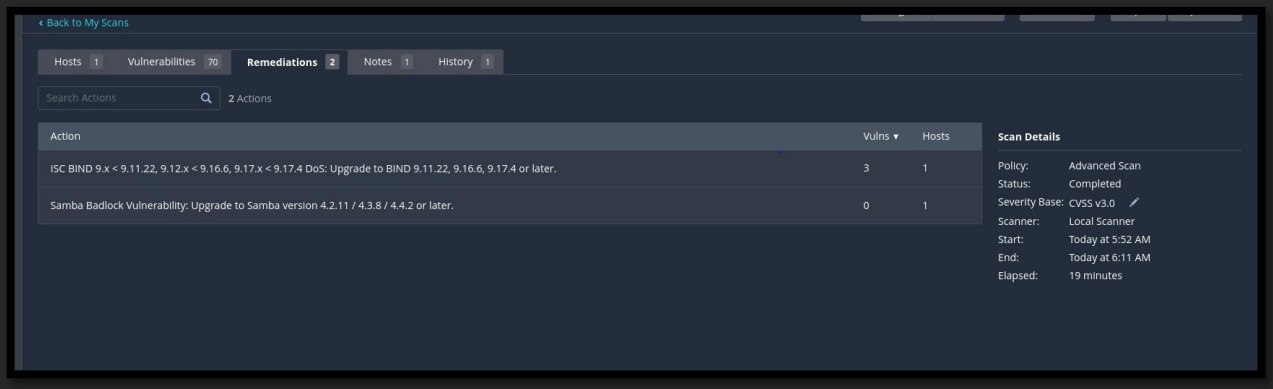
- It starts Scanning the target machine.



- The scan shows the following:

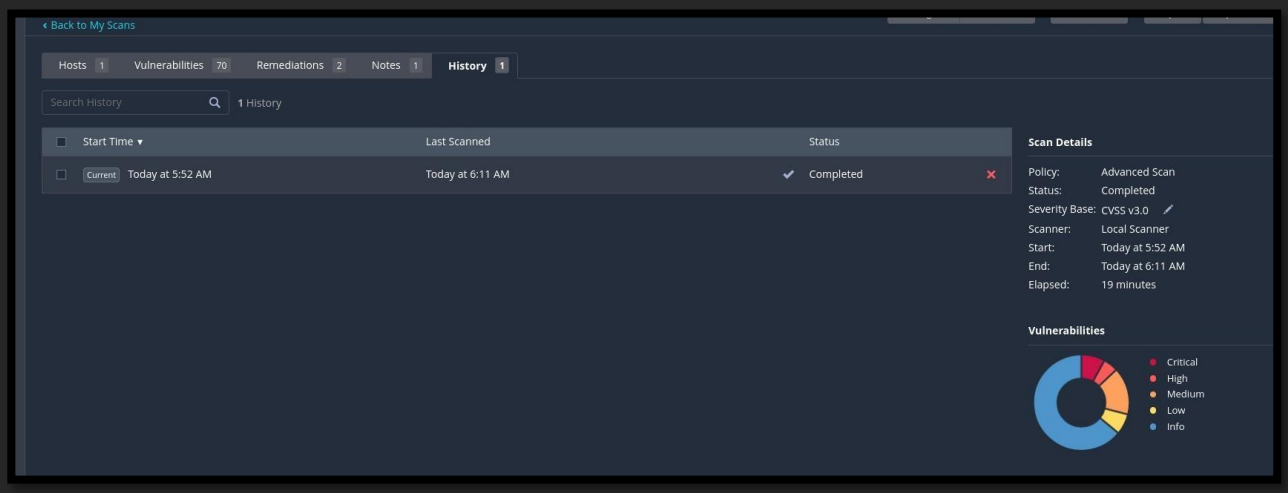


- Vulnerability tab gives the detail of each Vulnerability.
- Click on the remediation tab.
- Shows remediation for each Vulnerability.

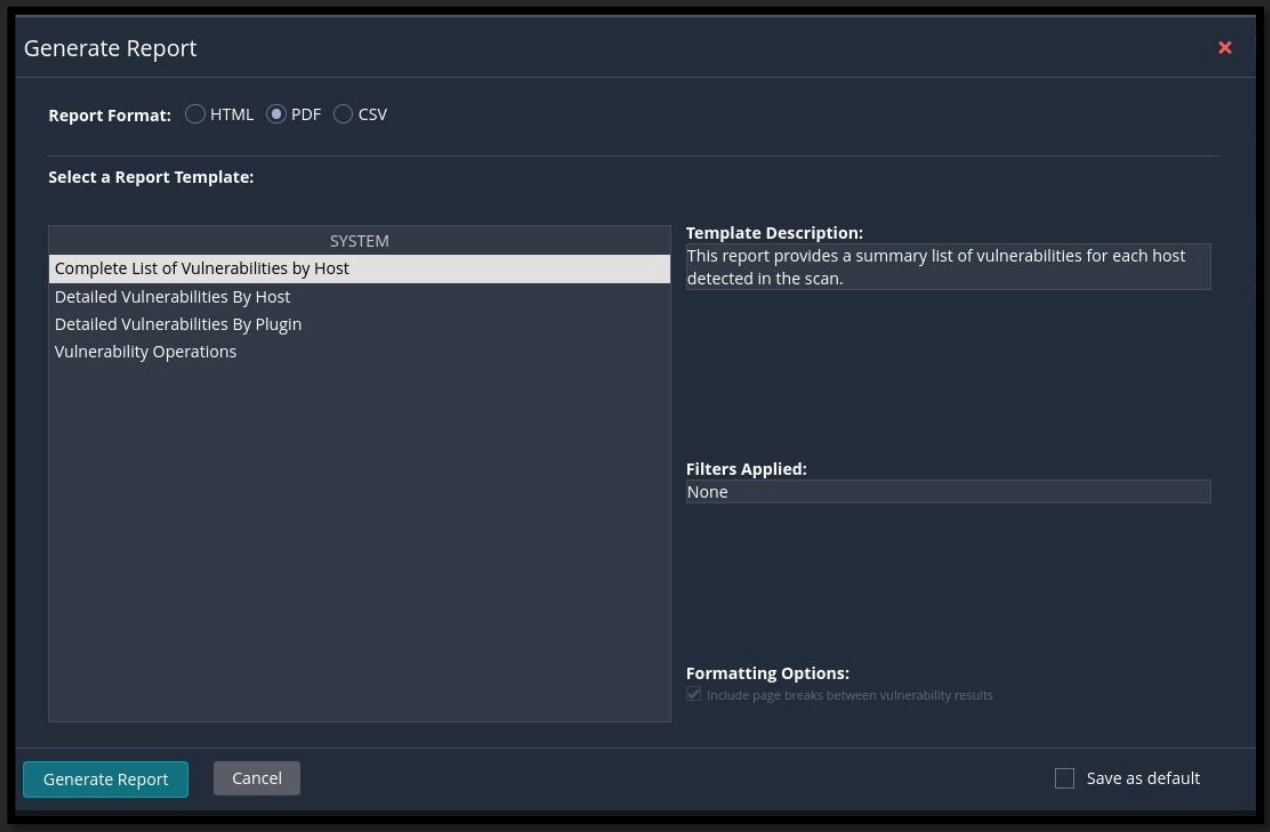


- Note tab:

- History Tab:



- Generate the Report.



- Report Details:

192.168.138.132

9	6	19	8	77
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 119

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	-	171340	Apache Tomcat SEoL (<= 5.5.x)
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0*	5.9	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0*	-	61708	VNC Server 'password' Password
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	42256	NFS Shares World Readable
HIGH	7.5	5.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	5.9	90509	Samba Badlock Vulnerability
HIGH	7.5*	5.9	10205	rlogin Service Detection
HIGH	7.5*	5.9	10245	rsh Service Detection
MEDIUM	6.5	3.6	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS
MEDIUM	6.5	-	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	-	57582	SSL Self-Signed Certificate

MEDIUM	6.5	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	42263	Unencrypted Telnet Server
MEDIUM	5.9	4.4	136808	ISC BIND Denial of Service

MEDIUM	5.9	4.4	89058	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption)
MEDIUM	5.9	4.4	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
MEDIUM	5.3	-	12085	Apache Tomcat Default Files
MEDIUM	5.3	-	12217	DNS Server Cache Snooping Remote Information Disclosure
MEDIUM	5.3	4.0	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	-	57608	SMB Signing not required
MEDIUM	5.3	-	15901	SSL Certificate Expiry
MEDIUM	5.3	-	45411	SSL Certificate with Wrong Hostname
MEDIUM	5.3	-	26928	SSL Weak Cipher Suites Supported
MEDIUM	4.0*	6.3	52611	SMTP Service STARTTLS Plaintext Command Injection
MEDIUM	4.3*	-	90317	SSH Weak Algorithms Supported
MEDIUM	4.3*	3.7	81606	SSL/TLS EXPORT_RSA <= 512-bit Cipher Suites Supported (FREAK)
LOW	3.7	3.6	70658	SSH Server CBC Mode Ciphers Enabled
LOW	3.7	-	153953	SSH Weak Key Exchange Algorithms Enabled
LOW	3.7	3.9	83875	SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam)
LOW	3.7	3.9	83738	SSL/TLS EXPORT_DHE <= 512-bit Export Cipher Suites Supported (Logjam)
LOW	3.4	5.1	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
LOW	2.1*	4.2	10114	ICMP Timestamp Request Remote Date Disclosure
LOW	2.6*	-	71049	SSH Weak MAC Algorithms Enabled
LOW	2.6*	-	10407	X Server Detection

INFO	N/A	-	10223	RPC portmapper Service Detection
INFO	N/A	-	21186	AJP Connector Detection
INFO	N/A	-	18261	Apache Banner Linux Distribution Disclosure
INFO	N/A	-	48204	Apache HTTP Server Version
INFO	N/A	-	39446	Apache Tomcat Detection
INFO	N/A	-	39519	Backported Security Patch Detection (FTP)
INFO	N/A	-	84574	Backported Security Patch Detection (PHP)
INFO	N/A	-	39520	Backported Security Patch Detection (SSH)
INFO	N/A	-	39521	Backported Security Patch Detection (WWW)
INFO	N/A	-	45590	Common Platform Enumeration (CPE)

Phase 4

Compliance Management Following ISO/IEC 27001:2013 Standards.

Introduction:

Compliance management ensures that the organization's information security practices align with ISO/IEC 27001:2013 standards. This phase focuses on establishing, implementing, maintaining, and continually improving an information security management system (ISMS).

Key Elements of Compliance Management:

1. Establishing the ISMS:

- **Scope Definition:** Identify the boundaries of the ISMS in terms of the organization, locations, assets, and technology.
- **Policy Development:** Create an information security policy aligned with ISO/IEC 27001:2013 requirements.

2. Risk Assessment and Treatment:

- **Risk Identification:** Identify potential security risks to information assets.
- **Risk Analysis:** Evaluate the likelihood and impact of identified risks.
- **Risk Treatment Plan:** Develop plans to mitigate, transfer, accept, or avoid risks.

3. Implementation of Controls:

- **Annex A Controls:** Implement applicable controls from Annex A of ISO/IEC 27001:2013.
- **Additional Controls:** Apply any additional controls identified during the risk assessment.

4. Monitoring and Review:

- **Performance Metrics:** Establish metrics to measure the effectiveness of implemented controls.
- **Internal Audits:** Conduct regular audits to ensure compliance with the ISMS requirements.
- **Management Review:** Hold periodic reviews by senior management to evaluate the ISMS's performance and make necessary adjustments.

5. Continual Improvement:

- **Non-Conformity Management:** Identify and address any non-conformities in the ISMS.
- **Corrective Actions:** Implement actions to correct and prevent the recurrence of non-conformities.
- **Feedback Loop:** Use feedback from audits, reviews, and incidents to continuously improve the ISMS.

Steps to Achieve Compliance

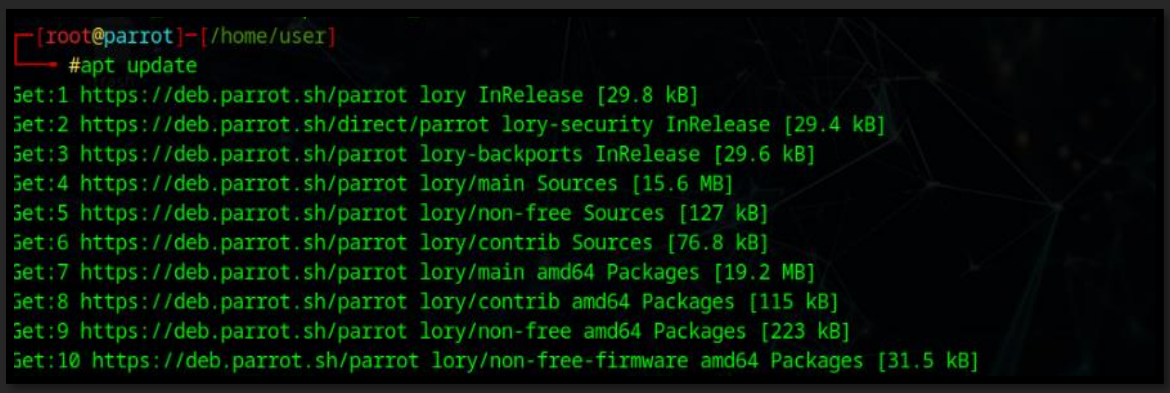
1. **Initial Gap Analysis:** Conduct a gap analysis to compare current practices with ISO/IEC 27001:2013 requirements.
2. **Develop an Action Plan:** Based on the gap analysis, create an action plan to address deficiencies and implement necessary controls.
3. **Training and Awareness:** Provide training to staff on information security policies, procedures, and their roles in maintaining compliance.
4. **Document Management:** Ensure all ISMS documentation is properly maintained, controlled, and accessible.
5. **Incident Management:** Establish an incident management process to detect, report, and respond to information security incidents.
6. **Supplier Management:** Ensure that third-party suppliers comply with the organization's information security requirements.
7. **Certification Audit:** Undergo a certification audit by an accredited certification body to achieve ISO/IEC 27001:2013 certification.

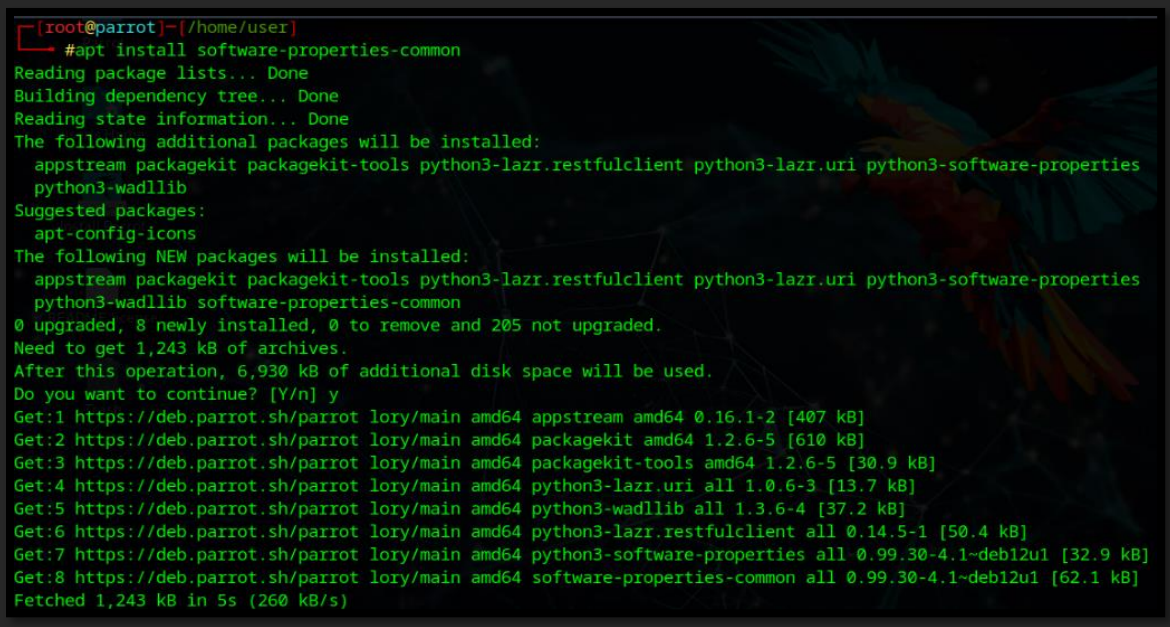
Conclusion

Compliance management following ISO/IEC 27001:2013 standards involves a systematic approach to managing information security risks and implementing controls. By adhering to these standards, the organization ensures that its information security practices are robust, effective, and continuously improving, thereby protecting valuable information assets and maintaining stakeholder trust.

Implementation of Suricata IDS/IPS:

Open up our Parrot endpoint and run the following commands:





```
┌─[root@parrot]─[/home/user]
└──╼ #apt update
Get:1 https://deb.parrot.sh/parrot lory InRelease [29.8 kB]
Get:2 https://deb.parrot.sh/direct/parrot lory-security InRelease [29.4 kB]
Get:3 https://deb.parrot.sh/parrot lory-backports InRelease [29.6 kB]
Get:4 https://deb.parrot.sh/parrot lory/main Sources [15.6 MB]
Get:5 https://deb.parrot.sh/parrot lory/non-free Sources [127 kB]
Get:6 https://deb.parrot.sh/parrot lory/contrib Sources [76.8 kB]
Get:7 https://deb.parrot.sh/parrot lory/main amd64 Packages [19.2 MB]
Get:8 https://deb.parrot.sh/parrot lory/contrib amd64 Packages [115 kB]
Get:9 https://deb.parrot.sh/parrot lory/non-free amd64 Packages [223 kB]
Get:10 https://deb.parrot.sh/parrot lory/non-free-firmware amd64 Packages [31.5 kB]
```

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #apt install suricata -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libhiredis0.14 libhtp2 libhyperscan5 libnetfilter-log1 suricata-update
Suggested packages:
  libtcmalloc-minimal4
Recommended packages:
  snort-rules-default
The following NEW packages will be installed:
  libhiredis0.14 libhtp2 libhyperscan5 libnetfilter-log1 suricata suricata-update
0 upgraded, 6 newly installed, 0 to remove and 205 not upgraded.
Need to get 4,633 kB of archives.
After this operation, 23.7 MB of additional disk space will be used.
Get:1 https://deb.parrot.sh/parrot lory/main amd64 libhyperscan5 amd64 5.4.0-2 [2,489 kB]
Get:2 https://deb.parrot.sh/parrot lory/main amd64 libhiredis0.14 amd64 0.14.1-3 [35.9 kB]
Get:3 https://deb.parrot.sh/parrot lory/main amd64 libhtp2 amd64 1:0.5.42-1 [70.4 kB]
Get:4 https://deb.parrot.sh/parrot lory/main amd64 libnetfilter-log1 amd64 1.0.2-3 [13.4 kB]
Get:5 https://deb.parrot.sh/parrot lory/main amd64 suricata amd64 1:6.0.10-1 [1,963 kB]
Get:6 https://deb.parrot.sh/parrot lory/main amd64 suricata-update amd64 1.2.7-1 [61.4 kB]
Fetched 4,633 kB in 15s (306 kB/s)
```

```
┌[root@parrot]-[/home/user]
└─► #suricata --build-info
This is Suricata version 6.0.10 RELEASE
Features: NFQ PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTP_URI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HAV
E_LUA HAVE_LUAJIT HAVE_LIBJANSSON TLS TLS_C11 MAGIC RUST
SIMD support: none
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 12.2.0, C version 201112
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: _Thread_local
compiled with LibHTP v0.5.42, linked against LibHTP v0.5.42

Suricata Configuration:
  AF_PACKET support:                       yes
  eBPF support:                            yes
  XDP support:                             yes
  PF_RING support:                         no
  NFQueue support:                         yes
  NFLOG support:                           yes
  IPFW support:                            no
  Netmap support:                          no  using new api: no
  DAG enabled:                             no
  Napatech enabled:                        no
  WinDivert enabled:                       no
```

```
##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.56.149/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
```

```
# Linux high speed capture support
af-packet:
  - interface: enp2s1
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
```

```
 GNU nano 7.2                                  /etc/suricata/suricata.yaml
pcap:
  - interface: enp2s1
    # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
    # as total memory used by the ring. So set this to something bigger
    # than 1% of your bandwidth.
    #buffer-size: 16777216
    #bpf-filter: "tcp and port 25"
    # Choose checksum verification mode for the interface. At the moment
    # of the capture, some packets may have an invalid checksum due to
    # the checksum computation being offloaded to the network card.
```

```
    # Community Flow ID
    # Adds a 'community_id' field to EVE records. These are meant to give
    # records a predictable flow ID that can be used to match records to
    # output of other tools such as Zeek (Bro).
    #
    # Takes a 'seed' that needs to be same across sensors and tools
    # to make the id less predictable.

    # enable/disable the community id feature.
    community-id: true
    # Seed value for the ID output. Valid values are 0-65535.
    community-id-seed: 0
```

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo apt-get install geoip-bin geoip-database
sudo apt-get install xtables-addons-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
geoip-bin is already the newest version (1.6.12-10).
geoip-database is already the newest version (20240708-1~bpo12+1).
0 upgraded, 0 newly installed, 0 to remove and 204 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnet-cidr-lite-perl xtables-addons-dkms
Recommended packages:
  linux-headers
The following NEW packages will be installed:
```


```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo apt-get install geoip-bin geoip-database
sudo apt-get install xtables-addons-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
geoip-bin is already the newest version (1.6.12-10).
geoip-database is already the newest version (20240708-1~bpo12+1).
0 upgraded, 0 newly installed, 0 to remove and 204 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnet-cidr-lite-perl xtables-addons-dkms
Recommended packages:
  linux-headers
The following NEW packages will be installed:
```

```
┌─[root@parrot]─[/home/user]
└──╼ #wget -O cn.zone http://www.ipdeny.com/ipblocks/data/countries/cn.zone
wget -O ru.zone http://www.ipdeny.com/ipblocks/data/countries/ru.zone
--2024-07-20 12:02:01--  http://www.ipdeny.com/ipblocks/data/countries/cn.zone
Resolving www.ipdeny.com (www.ipdeny.com)... 51.15.12.186
Connecting to www.ipdeny.com (www.ipdeny.com)|51.15.12.186|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 135039 (132K) [text/plain]
Saving to: 'cn.zone'

cn.zone                     100%[=================================================>] 131.87K   168KB/s    in 0.8s

2024-07-20 12:02:02 (168 KB/s) - 'cn.zone' saved [135039/135039]

--2024-07-20 12:02:02--  http://www.ipdeny.com/ipblocks/data/countries/ru.zone
Resolving www.ipdeny.com (www.ipdeny.com)... 51.15.12.186
Connecting to www.ipdeny.com (www.ipdeny.com)|51.15.12.186|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 171374 (167K) [text/plain]
Saving to: 'ru.zone'

ru.zone                     100%[=================================================>] 167.36K   264KB/s    in 0.6s

2024-07-20 12:02:04 (264 KB/s) - 'ru.zone' saved [171374/171374]
```

```
┌─[✗]─[root@parrot]─[/home/user]
└──╼ #sudo ipset create china hash:net
sudo ipset create russia hash:net
ipset v7.17: Set cannot be created: set with the same name already exists
ipset v7.17: Set cannot be created: set with the same name already exists
┌─[✗]─[root@parrot]─[/home/user]
└──╼ ## Add China IP ranges to the IP set
while read ip; do
  sudo ipset add china $ip
done < cn.zone

# Add Russia IP ranges to the IP set
while read ip; do
  sudo ipset add russia $ip
done < ru.zone
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
ipset v7.17: Element cannot be added to the set: it's already added
```

```
┌─[root@parrot]─[/home/user]
└──╼ #sudo iptables -I INPUT -m set --match-set china src -j DROP
sudo iptables -I INPUT -m set --match-set russia src -j DROP
┌─[root@parrot]─[/home/user]
└──╼ #sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  anywhere             anywhere             match-set russia src
DROP       all  --  anywhere             anywhere             match-set china src

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Create a rule file (/etc/suricata/rules/ipset.rules):

```
  GNU nano 7.2                                /etc/suricata/rules/ipset.rules
# /etc/suricata/rules/ipset.rules
alert ip any any -> any any (msg:"ALERT: Traffic from China IP"; content:"CN"; sid:1000001; rev:1;)
alert ip any any -> any any (msg:"ALERT: Traffic from Russia IP"; content:"RU"; sid:1000002; rev:1;)
```

A. **Create rules which restrict the user from specified websites**
B. **Administrator privileges rule**

Create a rule file /etc/suricata/rules/local.rules):

```
  GNU nano 7.2                                        local.rules
alert icmp any any -> $HOME_NET any (msg:"THIS IS AN ICMP Ping"; sid:1; rev:1;)
# /etc/suricata/rules/local.rules
drop http any any -> any any (msg:"DROP Access to Facebook"; content:"facebook.com"; http_host; sid:1000003; rev:1;)
# /etc/suricata/rules/local.rules
alert tcp any any -> any any (msg:"Admin Action Detected"; content:"admin"; http_uri; sid:1000004; rev:1;)
alert tcp any any -> any 22 (msg:"SSH connection attempt"; flow:to_server; sid:100004; rev:1;)
```

Include the rule file in suricata.yaml:

```
  GNU nano 7.2                                        /etc/suricata/suricata.yaml
##
## Configure Suricata to load Suricata-Update managed rules.
##
default-rule-path: /etc/suricata/rules

rule-files:
  - /etc/suricata/rules/ipset.rules
  - /etc/suricata/rules/local.rules

##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
```

Now we will save and close the YAML file then update the Suricata configuration with the following command

```
┌──[root@parrot]─[/home/user]
└─── #suricata-update
20/7/2024 -- 11:18:41 - <Info> -- Using data-directory /var/lib/suricata.
20/7/2024 -- 11:18:41 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/7/2024 -- 11:18:41 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
20/7/2024 -- 11:18:41 - <Info> -- Found Suricata version 6.0.10 at /usr/bin/suricata.
20/7/2024 -- 11:18:41 - <Info> -- Loading /etc/suricata/suricata.yaml
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol http2
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol modbus
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol dnp3
20/7/2024 -- 11:18:41 - <Info> -- Disabling rules for protocol enip
20/7/2024 -- 11:18:41 - <Info> -- No sources configured, will use Emerging Threats Open
20/7/2024 -- 11:18:41 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-6.0.10/emerging.rules.tar.gz.
  2% - 131072/4386085
```

We will verify the Suricata configuration file by using the built-in test command:

```
┌──[root@parrot]─[/home/user]
└─── #sudo suricata -T -c /etc/suricata/suricata.yaml
20/7/2024 -- 12:48:52 - <Info> - Running suricata under test mode
20/7/2024 -- 12:48:52 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
20/7/2024 -- 12:48:52 - <Notice> - Configuration provided was successfully loaded. Exiting.
┌──[root@parrot]─[/home/user]
└─── #
```

```
┌──[✗]─[root@parrot]─[/home/user]
└─── #sudo suricata -T -c /etc/suricata/suricata.yaml -v
20/7/2024 -- 13:01:37 - <Info> - Running suricata under test mode
20/7/2024 -- 13:01:37 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
20/7/2024 -- 13:01:37 - <Info> - CPUs/cores online: 2
20/7/2024 -- 13:01:37 - <Info> - fast output device (regular) initialized: fast.log
20/7/2024 -- 13:01:37 - <Info> - eve-log output device (regular) initialized: eve.json
20/7/2024 -- 13:01:37 - <Info> - stats output device (regular) initialized: stats.log
20/7/2024 -- 13:01:37 - <Info> - 2 rule files processed. 6 rules successfully loaded, 0 rules failed
20/7/2024 -- 13:01:37 - <Info> - Threshold config parsed: 0 rule(s) found
20/7/2024 -- 13:01:37 - <Info> - 6 signatures processed. 1 are IP-only rules, 2 are inspecting packet payload, 2 inspect appli
cation layer, 0 are decoder event only
20/7/2024 -- 13:01:38 - <Notice> - Configuration provided was successfully loaded. Exiting.
20/7/2024 -- 13:01:38 - <Info> - cleaning up signature grouping structure... complete
┌──[root@parrot]─[/home/user]
└─── #
```

Now we will enable and start Suricata and also check its status:

```
┌──[✗]─[root@parrot]─[/home/user]
└─── #sudo systemctl enable suricata
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
┌──[root@parrot]─[/home/user]
└─── #sudo systemctl start suricata
┌──[root@parrot]─[/home/user]
└─── #systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-07-20 12:57:43 UTC; 8min ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata-ids.org/docs/
   Main PID: 148234 (Suricata-Main)
      Tasks: 8 (limit: 3409)
     Memory: 46.9M
        CPU: 7.168s
     CGroup: /system.slice/suricata.service
             └─148234 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Jul 20 12:57:43 parrot systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
Jul 20 12:57:43 parrot suricata[148232]: 20/7/2024 -- 12:57:43 - <Notice> - This is Suricata version 6.0.10 RELEASE running i>
Jul 20 12:57:43 parrot systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
lines 1-16/16 (END)
```

Now we are going to verify that our Suricata is working and is able to track and log any threat on our PARROT endpoint.To do that we will first disable Suricata with the following command.

```
┌──[root@parrot]─[/home/user]
└─── #sudo ethtool -K ens33 gro off lro off
┌──[root@parrot]─[/home/user]
└─── #
```

Stop the Suricata with the following command.

sudo systemctl stop suricata

Remove the Suricata PID file so we can have a clean restart.

```
┌──[root@parrot]─[/home/user]
└─── #sudo rm -rf /var/run/suricata.pid
┌──[root@parrot]─[/home/user]
└─── #
```

Now we will run Suricata manually

```
┌──[root@parrot]─[/home/user]
└─── #sudo suricata -D -c /etc/suricata/suricata.yaml -i ens33
20/7/2024 -- 13:13:40 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode
┌──[root@parrot]─[/home/user]
└─── #
```

Now Verifying The Implimentation with The Ping Log Rules:

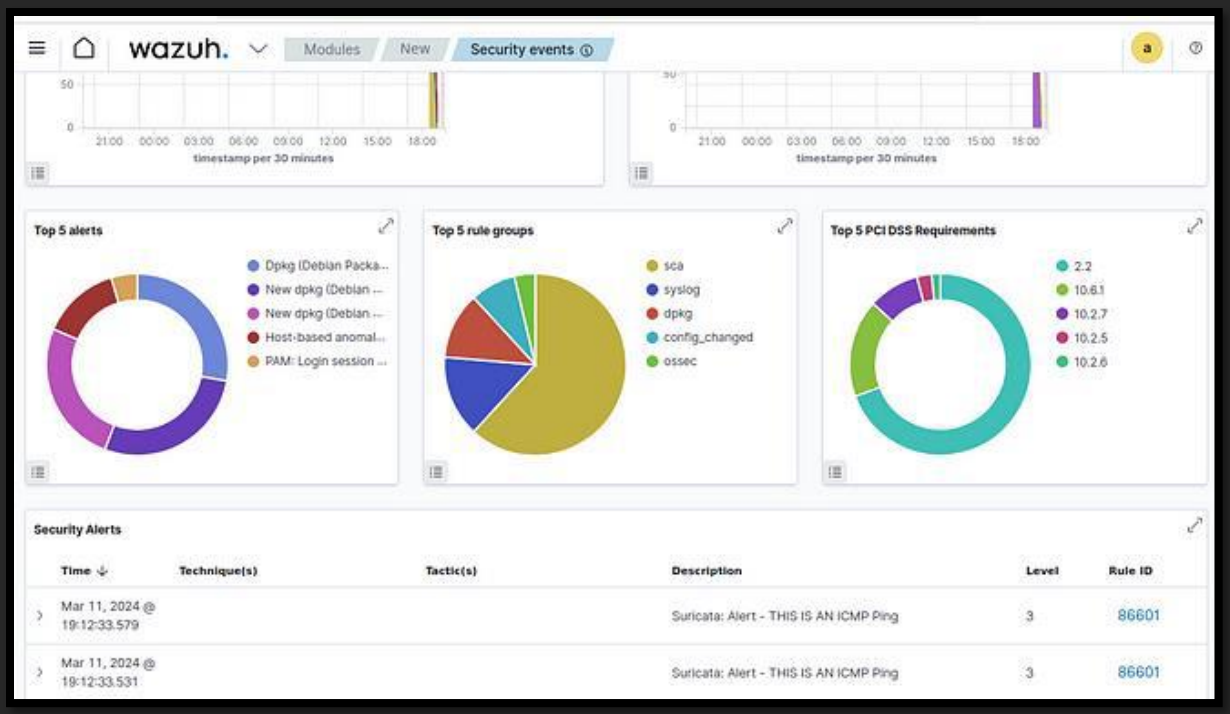
Now we will Ping our PARROT endpoint and see how Suricata tracks and logs the pings.I will log into my Kali Linux and preform the Ping on our PARROT endpoint.

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:6a:d7:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.142/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
       valid_lft 1615sec preferred_lft 1615sec
    inet6 fe80::8167:1e6:cc8e:8c6f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:ff:cd:ba:58 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping 192.168.56.149
PING 192.168.56.149 (192.168.56.149) 56(84) bytes of data.
64 bytes from 192.168.56.149: icmp_seq=1 ttl=64 time=3.67 ms
64 bytes from 192.168.56.149: icmp_seq=2 ttl=64 time=0.788 ms
64 bytes from 192.168.56.149: icmp_seq=3 ttl=64 time=0.698 ms
64 bytes from 192.168.56.149: icmp_seq=4 ttl=64 time=0.690 ms
64 bytes from 192.168.56.149: icmp_seq=5 ttl=64 time=0.627 ms
64 bytes from 192.168.56.149: icmp_seq=6 ttl=64 time=0.576 ms
^C
── 192.168.56.149 ping statistics ──
6 packets transmitted, 6 received, 0% packet loss, time 5050ms
rtt min/avg/max/mdev = 0.576/1.174/3.667/1.116 ms

┌──(kali㉿kali)-[~]
└─$
```

After the ping we will run the following command to view if Suricata logged the ping.

```
┌[root@parrot]-[/home/user]
└─ #sudo tail -f /var/log/suricata/fast.log
07/20/2024-13:20:57.896169  [**] [1:1:1] THIS IS AN ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.56.142:8 -> 192.168
.56.149:0
07/20/2024-13:20:57.896248  [**] [1:1:1] THIS IS AN ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.56.149:0 -> 192.168
.56.142:0
07/20/2024-13:20:57.896158  [**] [1:1:1] THIS IS AN ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.56.142:8 -> 192.168
.56.149:0
07/20/2024-13:20:57.896246  [**] [1:1:1] THIS IS AN ICMP Ping [**] [Classification: (null)] [Priority: 3] {ICMP} 192.168.56.149:0 -> 192.168
.56.142:0
```

We can see the last two lines from the fast.log files shows a ping that was logged in. The source IP address is from the Kali linux with IP address 192.168.56.142 and the ping is targeted at our PARROT endpoint with IP address 192.168.56.149.


wazuh.
Modules
New
Security events
timestamp per 30 minutes
timestamp per 30 minutes
Top 5 alerts
Dpkg (Debian Packa...
New dpkg (Debian ...
New dpkg (Debian ...
Host-based anomal...
PAM: Login session ...
Top 5 rule groups
sca
syslog
dpkg
config_changed
ossec
Top 5 PCI DSS Requirements
2.2
10.6.1
10.2.7
10.2.5
10.2.6
Security Alerts
Time
Technique(s)
Tactic(s)
Description
Level
Rule ID
Mar 11, 2024 @ 19:12:33.579
Suricata: Alert - THIS IS AN ICMP Ping
3
86601
Mar 11, 2024 @ 19:12:33.531
Suricata: Alert - THIS IS AN ICMP Ping
3
86601

Conclusion

This report outlines a comprehensive security assessment and hardening process for a small business network. Starting with asset discovery and inventory, we systematically identified all devices and software. The security hardening phase implemented crucial measures such as password policies, account lockout thresholds, and software updates to fortify the network against potential threats. Advanced scanning techniques were employed to uncover vulnerabilities, followed by compliance management to align with ISO/IEC 27001:2013 standards. The implementation of Suricata IDS/IPS added an additional layer of defense. These steps collectively enhanced the security posture of the network, ensuring ongoing protection and resilience against cyber threats.

About the Company

At Digital Empowerment, we envision a future where youth are equipped with the skills, knowledge, and mindset needed to thrive in a rapidly evolving world. Our mission is to bridge the digital divide, foster leadership development, and enhance academic growth, empowering young minds to realize their full potential.

To achieve this, we offer comprehensive virtual internships across various domains, providing students with invaluable hands-on experience and practical skills essential for success. Furthermore, we are committed to assisting exceptional students in securing positions at prestigious companies, helping to launch their careers and build a brighter future for themselves and future generations.

Digital Empowerment Pakistan

Author's Note

Throughout this task, I made significant advancements in cybersecurity knowledge and practical skills. I successfully completed tasks performing network scans with Nmap, and conducting vulnerability assessments with Nessus. The culmination of these efforts was a comprehensive security assessment and hardening project, which included meticulous asset discovery, security hardening, vulnerability scanning, and compliance management. This hands-on experience has equipped me with valuable skills and insights, emphasizing the importance of robust security practices in protecting digital assets and mitigating cyber threats.

This report represents a significant journey in exploring and implementing various aspects of cybersecurity. Each phase of the project provided invaluable insights into the complexities and challenges of securing a small business network. I extend my gratitude to everyone who contributed their expertise and support throughout this process. Their guidance and encouragement were instrumental in the successful completion of this project.