Week 2: Cybersecurity Operations Lab and Simulations

Introduction

This document provides a hands-on guide to multiple cybersecurity domains, including threat hunting, malware analysis, vulnerability management, incident response, and risk assessment. Clear and structured documentation is essential for effective cybersecurity operations, enabling analysts to understand, replicate, and validate security processes efficiently

1. Threat Hunting with Open-Source Tools

1.1. Overview

This section focuses on proactive threat detection using open-source Security Information and Event Management (SIEM) and rule-based detection frameworks.

1.2. Tool Used

- **Elastic Security**
- **Sigma Rules**

1.3. Activity Description

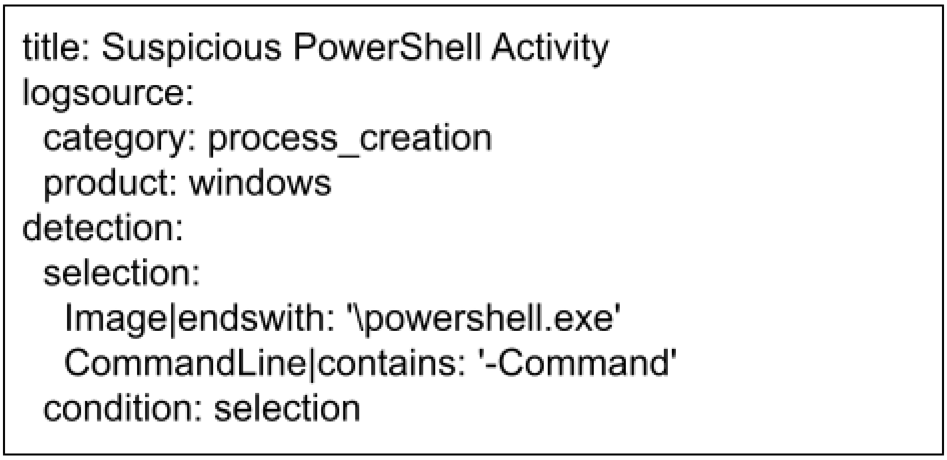
Windows process creation logs are ingested into Elastic Security, with emphasis on Event ID 4688, which records new process execution events. Threat hunting is performed by identifying suspicious PowerShell usage patterns commonly associated with malicious activity.

1.4. Sigma Rule Creation

A Sigma rule is created to detect PowerShell executions that use inline command execution.

Detection Logic

- Process image ending with *powershell.exe*
- Command-line arguments containing the -Command flag.



```
title: Suspicious PowerShell Activity
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    Image|endswith: '\powershell.exe'
    CommandLine|contains: '-Command'
  condition: selection
```

The rule is validated by executing a harmless command in a Windows virtual machine using:

powershell -Command "Write-Host Test"

1.5. Threat Hunting Query

PowerShell-related process creation events are queried in Elastic Security.

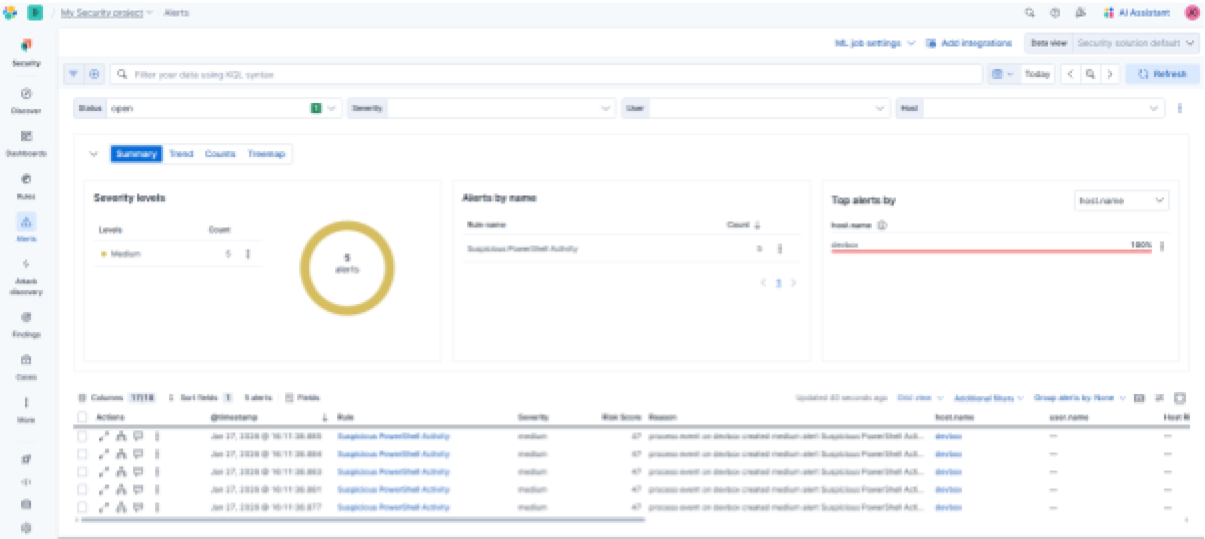


Figure 1: Rule alert of the powershell activity

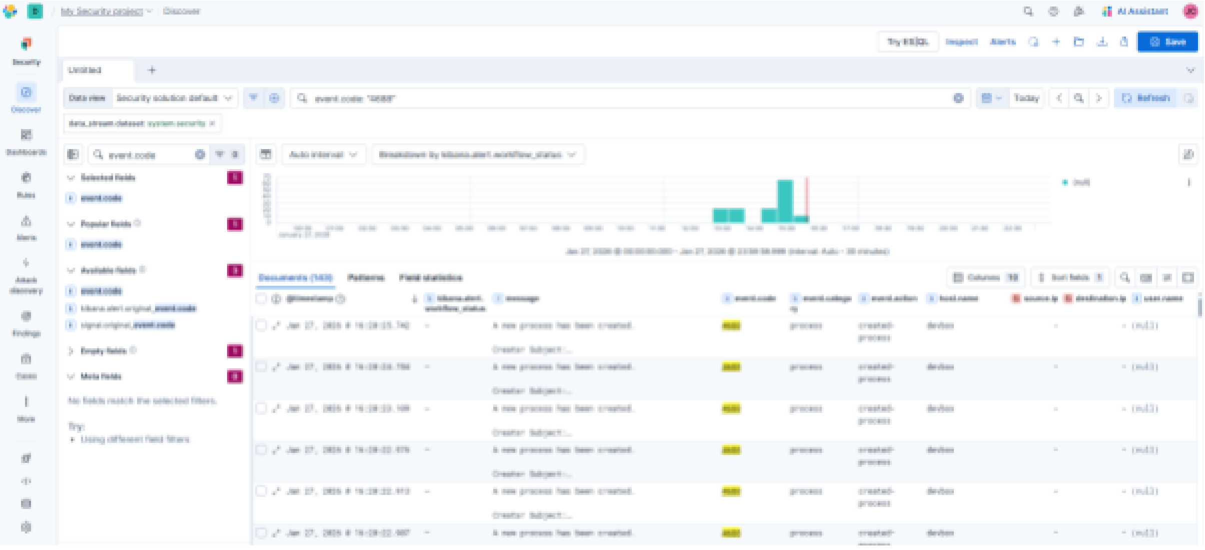


Figure 2: Elastic Security Event for ID 4688

2. Malware Analysis Basics

2.1. Overview

This section introduces fundamental malware analysis techniques using a benign executable to ensure a safe learning environment.

2.2. Tool Used

- **REMnux**
- **Hybrid Analysis**

2.3. Static Analysis

The benign file *calc.exe* is analyzed using the strings utility in **REMnux**. The command output is redirected to a text file and reviewed to identify human-readable strings such as system references or function names.

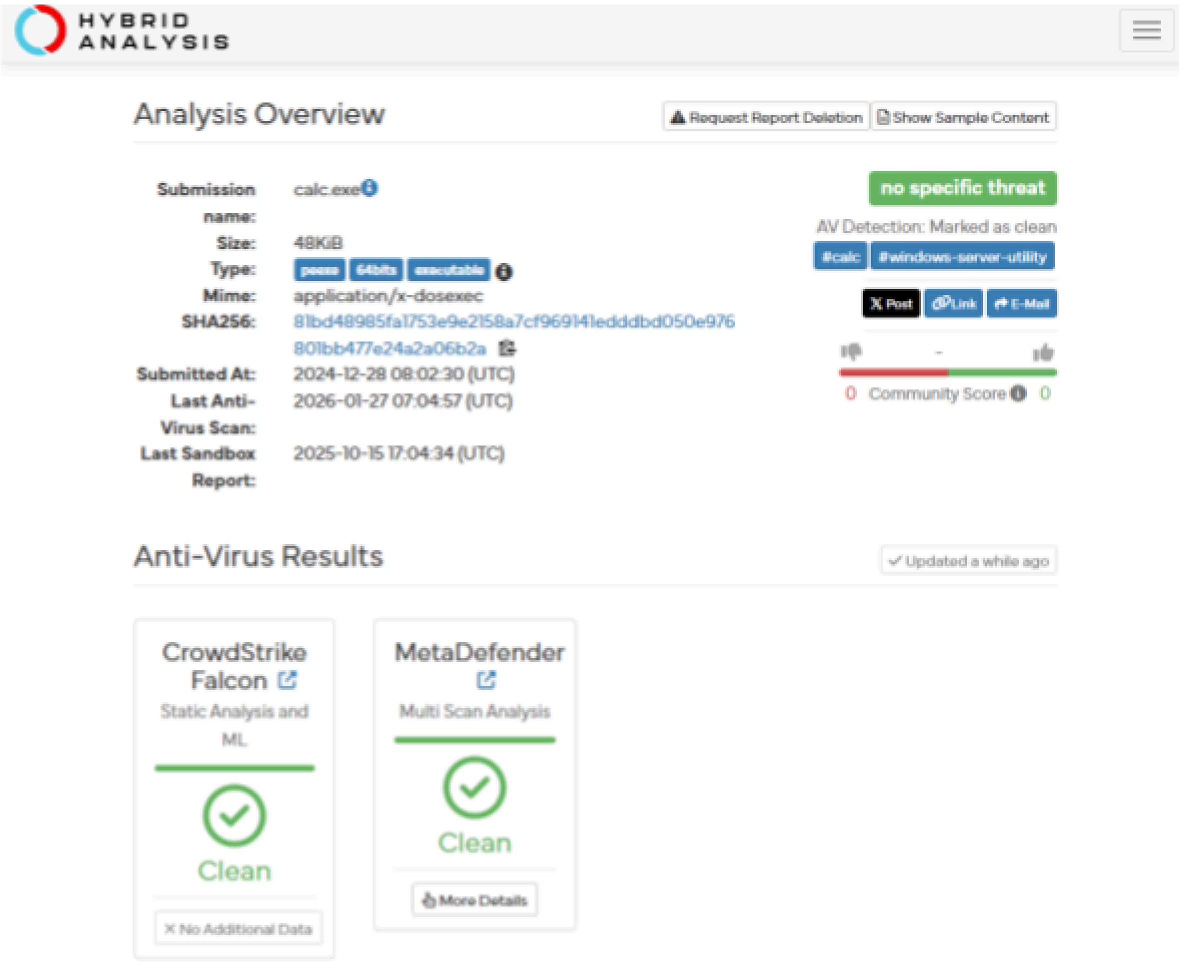
The Debugging Footprint (calc.pdb): The presence of calc.pdb is a classic artifact. A PDB (Program Database) file holds debugging information used during development. While the file itself isn't in your output, this string acts as a pointer for the Windows debugger to find symbols, confirming this is a standard Microsoft-compiled NT binary.

Telemetry and Logging (EventWriteTransfer): Strings like ETW0 and EventWriteTransfer indicate that Calculator isn't just a "silent" tool; it uses Event Tracing for Windows. This allows the OS to log when the app starts or crashes, helping Microsoft gather telemetry data to monitor the application's performance and stability across millions of devices.

The Modern Shell Transition (ShellExecuteW): Seeing ShellExecuteW and the manifest XML is fascinating because modern calc.exe is often just a "wrapper." In newer Windows versions, this small binary doesn't do the math itself; it uses these instructions to launch the full UWP (Universal Windows Platform) Calculator app interface from the Windows Shell environment.

2.4. Dynamic Analysis

The same executable is submitted to **Hybrid Analysis** for behavioral inspection. Observed runtime behaviors are compared with static findings from REMnux to highlight the strengths and limitations of each analysis method. While static analysis will only provide insight into the executable in the case of signature, strings, and assembly, dynamic analysis shows the behavior when the executable runs.



Analysis Overview

Submission name: calc.exe
Size: 48KiB
Type: peexe 64bits executable
Mime: application/x-dosexec
SHA256: 81bd48985fa1753e9e2158a7cf969141edddbd050e976801bb477e24a2a06b2a
Submitted At: 2024-12-28 08:02:30 (UTC)
Last Anti-Virus Scan: 2026-01-27 07:04:57 (UTC)
Last Sandbox Report: 2025-10-15 17:04:34 (UTC)

AV Detection: Marked as clean

0 Community Score 0

Anti-Virus Results

CrowdStrike Falcon
Static Analysis and ML
Clean

MetaDefender
Multi Scan Analysis
Clean

Figure 3: Hybrid Analysis Results

3. Vulnerability Management Pipeline

3.1. Overview

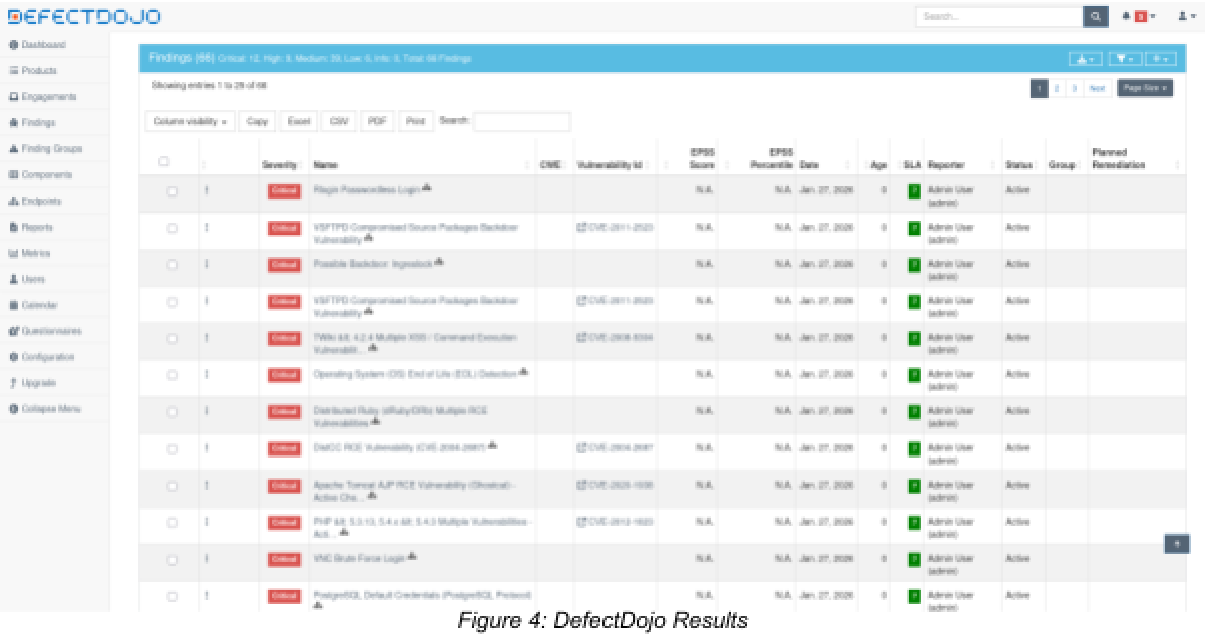
This section demonstrates a complete vulnerability management workflow from discovery to remediation planning.

3.2. Tool Used

- **OpenVAS**
- **DefectDojo**

3.3. Vulnerability Scanning

An OpenVAS scan is performed against a *Metasploitable2* virtual machine. Scan results are exported and imported into DefectDojo for centralized vulnerability tracking.



Findings (68) Critical: 12, High: 8, Medium: 39, Low: 6, Info: 3, Total: 68 Findings

Showing entries 1 to 25 of 68

□		Severity	Name	CWE	Vulnerability Id	EPSS Score	EPSS Percentile	Date	Age	SLA	Reporter	Status	Group	Planned Remediation
□	!	Critical	Rlogin Passwordless Login			N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	VSFTPD Compromised Source Packages Backdoor Vulnerability		CVE-2011-2523	N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	Possible Backdoor: Ingreslock			N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	VSFTPD Compromised Source Packages Backdoor Vulnerability		CVE-2011-2523	N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	TWiki &lt; 4.2.4 Multiple XSS / Command Execution Vulnerabili...		CVE-2008-5304	N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	Operating System (OS) End of Life (EOL) Detection			N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities			N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	DistCC RCE Vulnerability (CVE-2004-2687)		CVE-2004-2687	N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	Apache Tomcat AJP RCE Vulnerability (Ghostcat) - Active Che...		CVE-2020-1938	N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	PHP &lt; 5.3.13, 5.4.x &lt; 5.4.3 Multiple Vulnerabilities - Act...		CVE-2012-1823	N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	VNC Brute Force Login			N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		
□	!	Critical	PostgreSQL Default Credentials (PostgreSQL Protocol)			N.A.	N.A.	Jan. 27, 2026	0	[illegible]	Admin User (admin)	Active		

Figure 4: DefectDojo Results

3.4. Prioritized Vulnerabilities

Vulnerability	**CVSS Score**	**Description**
VSFTPD Backdoor	10.0	Allows remote access
Rlogin Passwordless Login	7.5	Allows remote login
TWiki 4.2.4 XSS	4.3	Allows execution of arbitrary web script

3.5. Remediation Plan

- Patching or upgrading vulnerable services
- Disabling unnecessary services such as VSFTPD
- Applying firewall or network segmentation controls

4. Incident Response Simulation

4.1. Overview

This section simulates a real-world incident to practice detection, investigation, and response techniques.

4.2. Tools Used

- **MITRE Caldera**
- **Velociraptor**

4.3. Phishing Simulation

The attack chain commenced with the delivery of a simulated malicious payload to the target Windows endpoint. Upon execution, meant to mimic a user opening a weaponized attachment, the payload initiated a PowerShell process. This process downloaded the Caldera 'Sandcat' agent from the external Command and Control (C2) server. Once executed, the agent established a persistent C2 channel over port 8888, registering the endpoint as compromised. The adversary profile then proceeded to execute automated discovery commands, generating distinct process tree anomalies and network traffic artifacts subsequently captured by the forensic agents.

4.4. Artifact Collection and Analysis

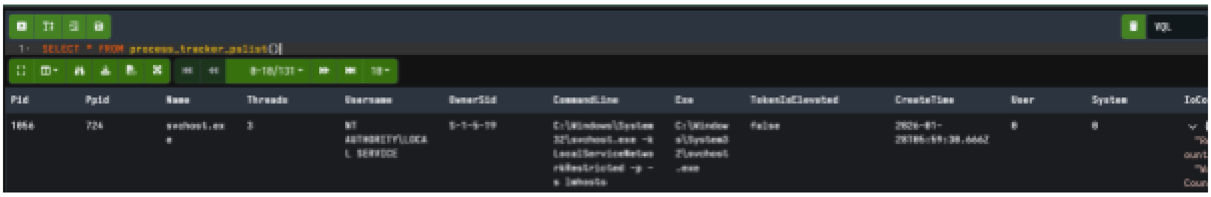
Velociraptor is used to collect endpoint artifacts using the following queries:

- Process listings
- Active network connections

VQL Queries Used:

- **SELECT * FROM process_tracker_pslist()**
- **SELECT * FROM netstat()**

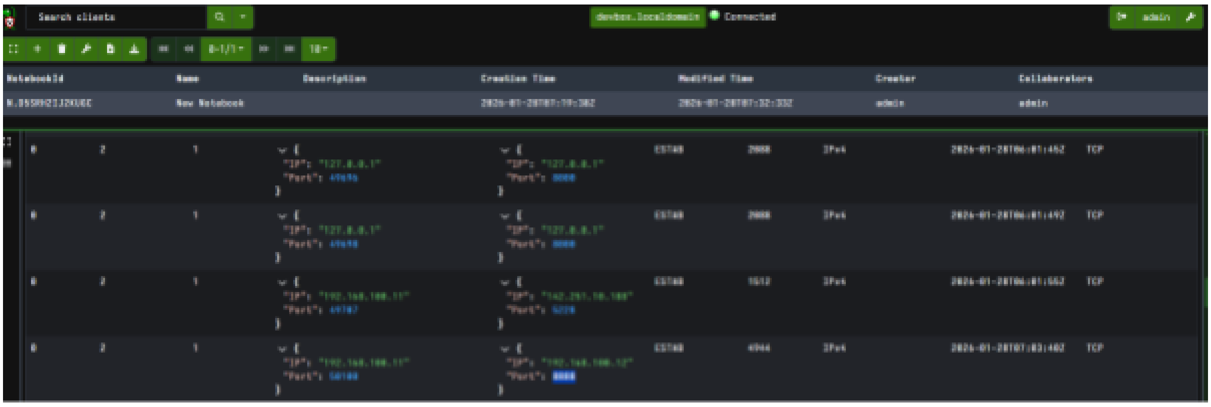
Collected data is exported to CSV format and analyzed to identify Indicators of Compromise (IOCs), such as suspicious processes or anomalous network activity.



```
SELECT * FROM process_tracker_pslist()
```

Pid	Ppid	Name	Threads	Username	OwnerSid	CommandLine	Exe	TokenIsElevated	CreateTime	User	System	IoCo
1056	724	svchost.exe	3	NT AUTHORITY\LOCAL SERVICE	S-1-5-19	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p -s lmhosts	C:\Windows\System32\svchost.exe	false	2026-01-20T05:59:30.666Z	0	0	[illegible]

Figure 5: Velociraptor results of process list query



NotebookId	Name	Description	Creation Time	Modified Time	Creator	Collaborators
N.D5SRH2IJ2KUCC	New Notebook		2026-01-28T07:19:38Z	2026-01-28T07:32:33Z	admin	admin

```
0   2   1   {                           {                            ESTAB   2008   IPv4   2026-01-28T06:01:45Z   TCP
              "IP": "127.0.0.1"           "IP": "127.0.0.1"
              "Port": 49696               "Port": 8888
            }                           }
0   2   1   {                           {                            ESTAB   2008   IPv4   2026-01-28T06:01:49Z   TCP
              "IP": "127.0.0.1"           "IP": "127.0.0.1"
              "Port": 49698               "Port": 8888
            }                           }
0   2   1   {                           {                            ESTAB   1512   IPv4   2026-01-28T06:01:55Z   TCP
              "IP": "192.168.100.11"      "IP": "142.251.10.188"
              "Port": 49787               "Port": 5228
            }                           }
0   2   1   {                           {                            ESTAB   4944   IPv4   2026-01-28T07:03:40Z   TCP
              "IP": "192.168.100.11"      "IP": "192.168.100.12"
              "Port": 50188               "Port": 8888
            }                           }
```

Figure 6: Velociraptor results of netstat query

When analysing the Velociraptor artifacts using VQL queries for IOC(Indicator of Compromise), the Caldera agent/payload can be seen in the process list and network connection.

5. Network Defense with Open-Source Tools

5.1. Overview

This section focuses on detecting and blocking malicious network activity.

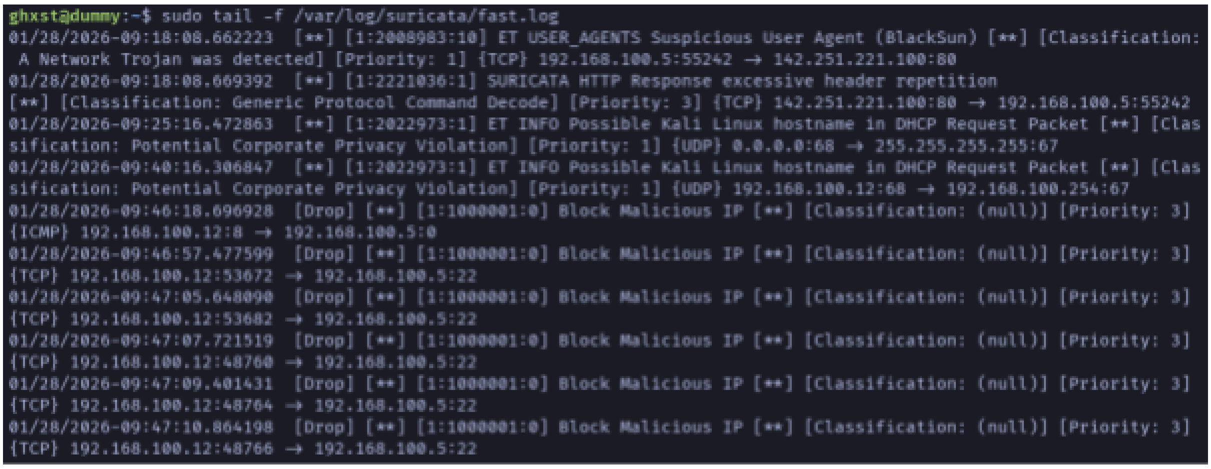
5.2. Tool Used

- **Suricata**

5.3. Suricata Rule Configuration

A custom Suricata rule is created to block traffic from a known malicious IP address. The rule is validated by generating traffic from a separate virtual machine and confirming the block.

```
drop ip 192.168.1.12 any -> any any (msg:"Block Malicious IP"; sid:1000001;)
```



```
ghxst@dummy:~$ sudo tail -f /var/log/suricata/fast.log
01/28/2026-09:18:08.662223  [**] [1:2008983:10] ET USER_AGENTS Suspicious User Agent (BlackSun) [**] [Classification:
 A Network Trojan was detected] [Priority: 1] {TCP} 192.168.100.5:55242 → 142.251.221.100:80
01/28/2026-09:18:08.669392  [**] [1:2221036:1] SURICATA HTTP Response excessive header repetition
[**] [Classification: Generic Protocol Command Decode] [Priority: 3] {TCP} 142.251.221.100:80 → 192.168.100.5:55242
01/28/2026-09:25:16.472863  [**] [1:2022973:1] ET INFO Possible Kali Linux hostname in DHCP Request Packet [**] [Clas
sification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 0.0.0.0:68 → 255.255.255.255:67
01/28/2026-09:40:16.306847  [**] [1:2022973:1] ET INFO Possible Kali Linux hostname in DHCP Request Packet [**] [Clas
sification: Potential Corporate Privacy Violation] [Priority: 1] {UDP} 192.168.100.12:68 → 192.168.100.254:67
01/28/2026-09:46:18.696928  [Drop] [**] [1:1000001:0] Block Malicious IP [**] [Classification: (null)] [Priority: 3]
{ICMP} 192.168.100.12:8 → 192.168.100.5:0
01/28/2026-09:46:57.477599  [Drop] [**] [1:1000001:0] Block Malicious IP [**] [Classification: (null)] [Priority: 3]
{TCP} 192.168.100.12:53672 → 192.168.100.5:22
01/28/2026-09:47:05.648090  [Drop] [**] [1:1000001:0] Block Malicious IP [**] [Classification: (null)] [Priority: 3]
{TCP} 192.168.100.12:53682 → 192.168.100.5:22
01/28/2026-09:47:07.721519  [Drop] [**] [1:1000001:0] Block Malicious IP [**] [Classification: (null)] [Priority: 3]
{TCP} 192.168.100.12:48760 → 192.168.100.5:22
01/28/2026-09:47:09.401431  [Drop] [**] [1:1000001:0] Block Malicious IP [**] [Classification: (null)] [Priority: 3]
{TCP} 192.168.100.12:48764 → 192.168.100.5:22
01/28/2026-09:47:10.864198  [Drop] [**] [1:1000001:0] Block Malicious IP [**] [Classification: (null)] [Priority: 3]
{TCP} 192.168.100.12:48766 → 192.168.100.5:22
```

Figure 7: Suricata dropping IP connections

5.4. MITRE ATT&CK Mapping

Suricata alerts are mapped to MITRE ATT&CK techniques to provide contextual threat intelligence.

Alert	Tactic	Technique	Notes
Block Malicious IP	Brute Force	T1110	SSH bruteforcing

6. Risk Assessment Practice

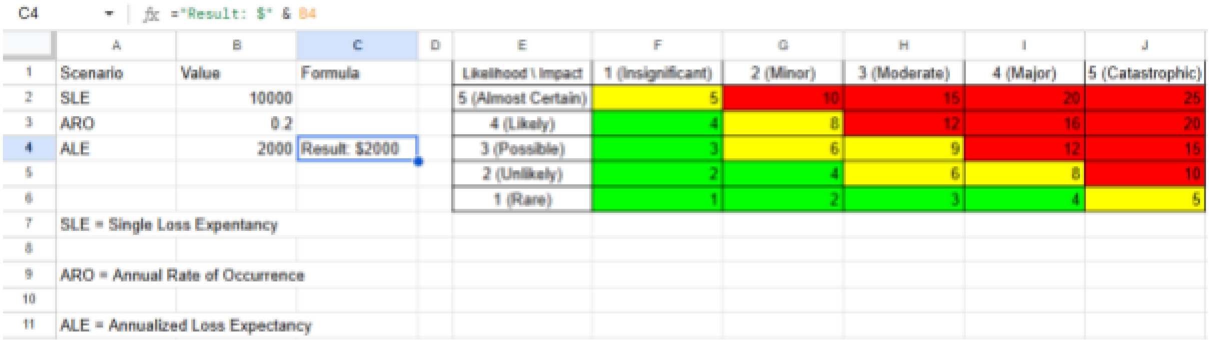
6.1. Overview

This section introduces both quantitative and qualitative risk assessment techniques.

6.2. ALE Calculation and Risk Matrix

A ransomware scenario is evaluated using the formula:

ALE = Single Loss Expectancy (SLE) X Annualized Rate of Occurrence (ARO)



C4 ▾ | fx ="Result: $" & B4

	A	B	C	D	E	F	G	H	I	J
1	Scenario	Value	Formula		Likelihood \ Impact	1 (Insignificant)	2 (Minor)	3 (Moderate)	4 (Major)	5 (Catastrophic)
2	SLE	10000			5 (Almost Certain)	5	10	15	20	25
3	ARO	0.2			4 (Likely)	4	8	12	16	20
4	ALE	2000	Result: $2000		3 (Possible)	3	6	9	12	15
5					2 (Unlikely)	2	4	6	8	10
6					1 (Rare)	1	2	3	4	5
7	SLE = Single Loss Expentancy									
8										
9	ARO = Annual Rate of Occurrence									
10										
11	ALE = Annualized Loss Expectancy									

Figure 8: ALE Calculation in Google Sheets

- **Likelihood Score:** An ARO of 0.2 means the event happens once every 5 years. This typically aligns with Score 2 (Unlikely).
- **Impact Score:** An SLE of $10,000 must be judged against your organization's total budget. If this loss is manageable but requires attention, it may be Score 2 (Minor) or Score 3 (Moderate).
- **Risk Score:** 2 x 3 = 6 (Medium Risk).

7. Incident Response Report

Date: Jan 29, 2026

Severity: High

Status: Closed

7.1. Executive Summary

On Jan 29 2026, the security team detected and neutralized a simulated phishing attack targeting the finance department. The attack utilized a malicious attachment to establish remote access. Defense systems successfully logged the activity, and the incident response team contained the threat within 15 minutes. No sensitive data was exfiltrated.

7.2. Incident Timeline

- 10:00 AM: Phishing email delivered to target inbox.
- 10:05 AM: User opened attachment; malicious payload executed.
- 10:06 AM: Velociraptor detected anomalous child process.
- 10:15 AM: Host isolated from network; C2 connection severed.

7.3. Mitigation Steps

- **Detection:** The incident was identified through a combination of user unusual network activity and unidentified process.
- **Containment:** Blocked the IP address and domain at the perimeter firewall and DNS filter. Isolated the machine from the network.
- **Eradication:** Identified and removed all the messages. Executed a Hard Wipe on the machine and mail environment.
- **Recovery:** Important documents recovered for reinstatement. Active sessions were revoked to invalidate potentially stolen session tokens. High Alert status placed for monitoring

7.4. Lesson Learned

- Security gaps in mail filtering identified
- New threat pattern and behaviour identified
- Schedule an updated threat awareness training for employees

7.5. Incident Response Process Flowchart

Detection
Containment
Eradication
Recovery
Lesson Learned

Figure 9: Flowchart of incident response process

8. Capstone Project: Full Incident Response Cycle

8.1. Overview

The capstone integrates multiple tools and skills into a complete attack-and-response scenario.

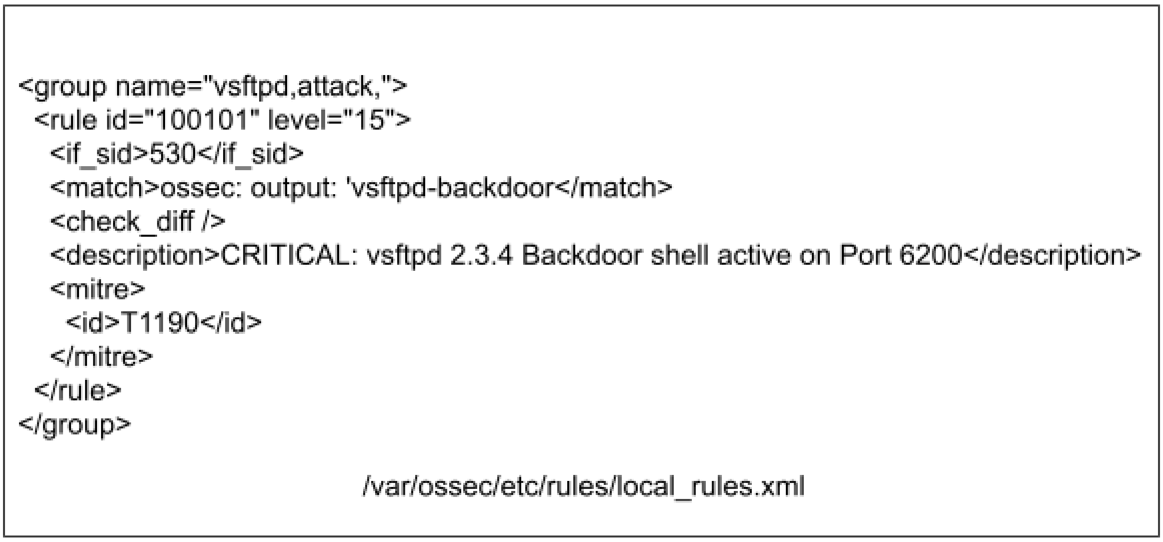
8.2. Tool Used

- **Metasploit**
- **Wazuh**
- **CrowdSec**

8.3. Attack Simulation and Detection

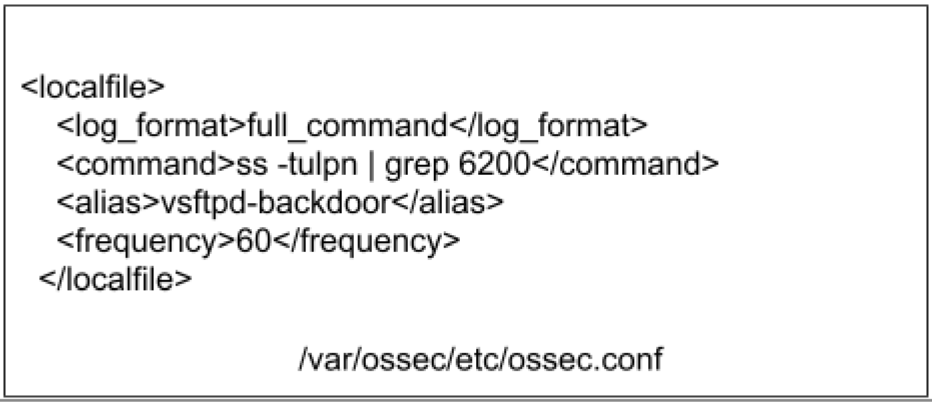
A known vulnerability in Metasploitable2 is exploited using Metasploit. Wazuh detects the activity and generates alerts with mapped MITRE ATT&CK techniques.

These rules are used to filter the detection:



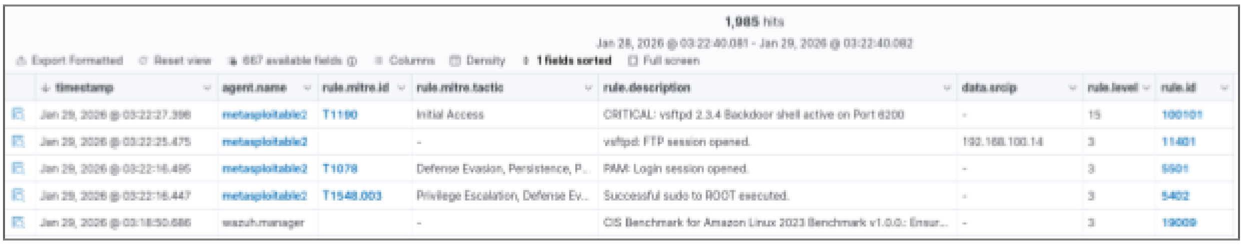
```
<group name="vsftpd,attack,">
  <rule id="100101" level="15">
    <if_sid>530</if_sid>
    <match>ossec: output: 'vsftpd-backdoor</match>
    <check_diff />
    <description>CRITICAL: vsftpd 2.3.4 Backdoor shell active on Port 6200</description>
    <mitre>
      <id>T1190</id>
    </mitre>
  </rule>
</group>
```

/var/ossec/etc/rules/local_rules.xml



```
<localfile>
    <log_format>full_command</log_format>
    <command>ss -tulpn | grep 6200</command>
    <alias>vsftpd-backdoor</alias>
    <frequency>60</frequency>
  </localfile>
```

/var/ossec/etc/ossec.conf



1,985 hits

Jan 28, 2026 @ 03:22:40.081 - Jan 29, 2026 @ 03:22:40.082

timestamp	agent.name	rule.mitre.id	rule.mitre.tactic	rule.description	data.srcip	rule.level	rule.id
Jan 29, 2026 @ 03:22:27.398	metasploitable2	T1190	Initial Access	CRITICAL: vsftpd 2.3.4 Backdoor shell active on Port 6200	-	15	100101
Jan 29, 2026 @ 03:22:25.475	metasploitable2		-	vsftpd: FTP session opened.	192.168.100.14	3	11401
Jan 29, 2026 @ 03:22:16.495	metasploitable2	T1078	Defense Evasion, Persistence, P...	PAM: Login session opened.	-	3	5501
Jan 29, 2026 @ 03:22:16.447	metasploitable2	T1548.003	Privilege Escalation, Defense Ev...	Successful sudo to ROOT executed.	-	3	5402
Jan 29, 2026 @ 03:18:50.686	wazuh.manager		-	CIS Benchmark for Amazon Linux 2023 Benchmark v1.0.0.: Ensur...	-	3	19009

Figure 10: Wazuh alert on the attack

8.4. Containment using CrowdSec

The attacker's IP address is blocked using CrowdSec, and connectivity tests confirm successful containment.

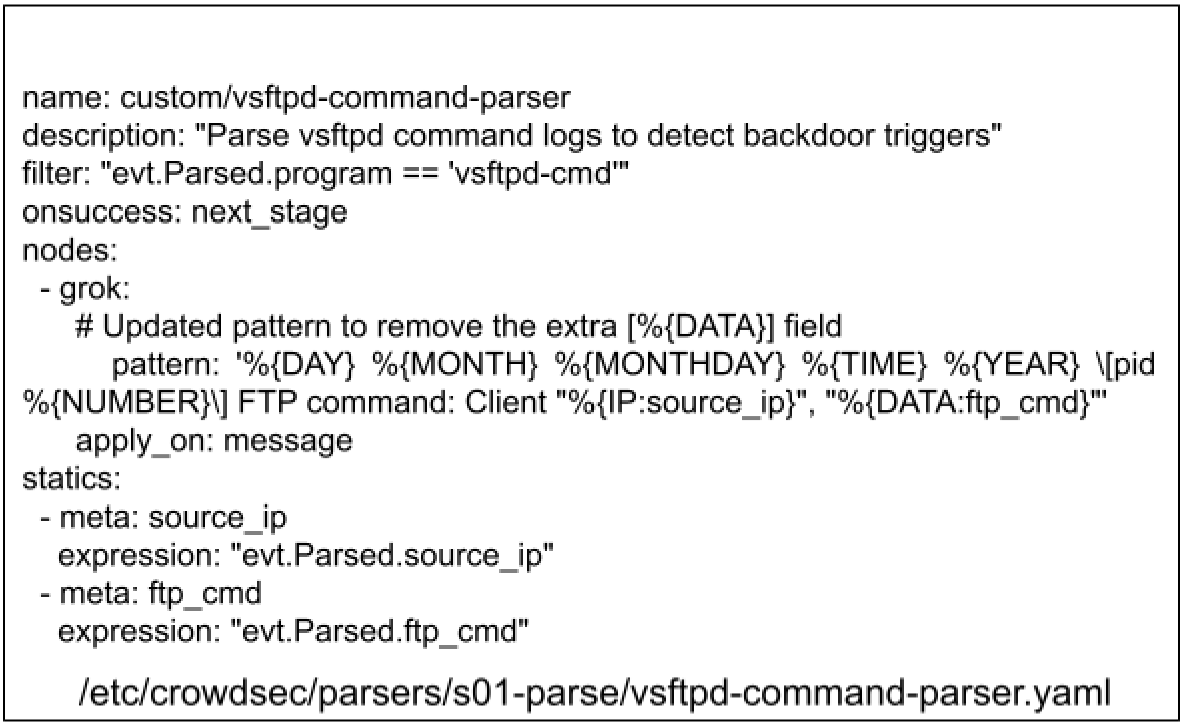
To install crowdsec on the client the following commands where ran,

- **curl -s https://install.crowdsec.net | sudo bash**
- **sudo apt install crowdsec -y**
- **sudo apt install crowdsec-firewall-bouncer-iptables -y**

The following configuration and rules are used,

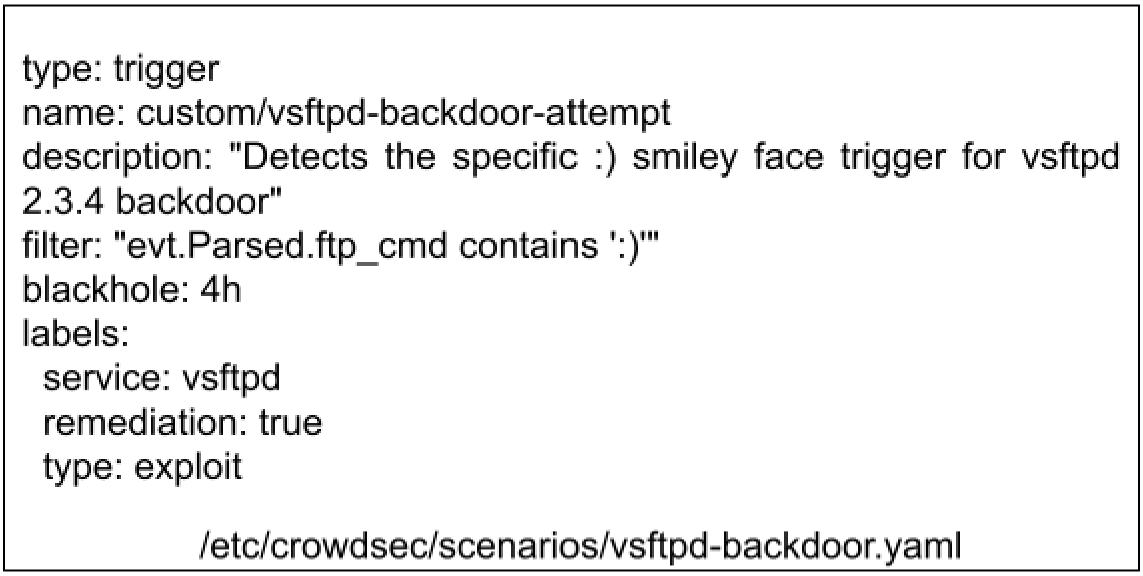
```
filenames:
  - /var/log/vsftpd.log
labels:
  type: vsftpd-cmd
```

/etc/crowdsec/acquis.d/vsftpd.yaml



```
name: custom/vsftpd-command-parser
description: "Parse vsftpd command logs to detect backdoor triggers"
filter: "evt.Parsed.program == 'vsftpd-cmd'"
onsuccess: next_stage
nodes:
  - grok:
     # Updated pattern to remove the extra [%{DATA}] field
        pattern: '%{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{YEAR} \[pid
%{NUMBER}\] FTP command: Client "%{IP:source_ip}", "%{DATA:ftp_cmd}"'
     apply_on: message
statics:
  - meta: source_ip
    expression: "evt.Parsed.source_ip"
  - meta: ftp_cmd
    expression: "evt.Parsed.ftp_cmd"
```

/etc/crowdsec/parsers/s01-parse/vsftpd-command-parser.yaml



```
type: trigger
name: custom/vsftpd-backdoor-attempt
description: "Detects the specific :) smiley face trigger for vsftpd
2.3.4 backdoor"
filter: "evt.Parsed.ftp_cmd contains ':)'"
blackhole: 4h
labels:
  service: vsftpd
  remediation: true
  type: exploit
```

/etc/crowdsec/scenarios/vsftpd-backdoor.yaml

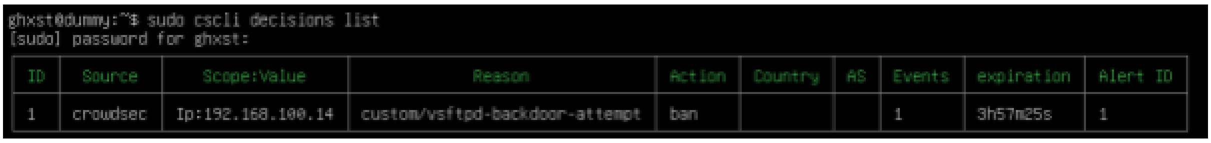
Crowdsec restarted to apply changes:

- sudo systemctl restart crowdsec

Since the testing is done with a private ipv4 address, these ipv4 ranges need to be whitelisted from the */etc/crowdsec/parsers/s02-enrich/whitelists.yaml* file.

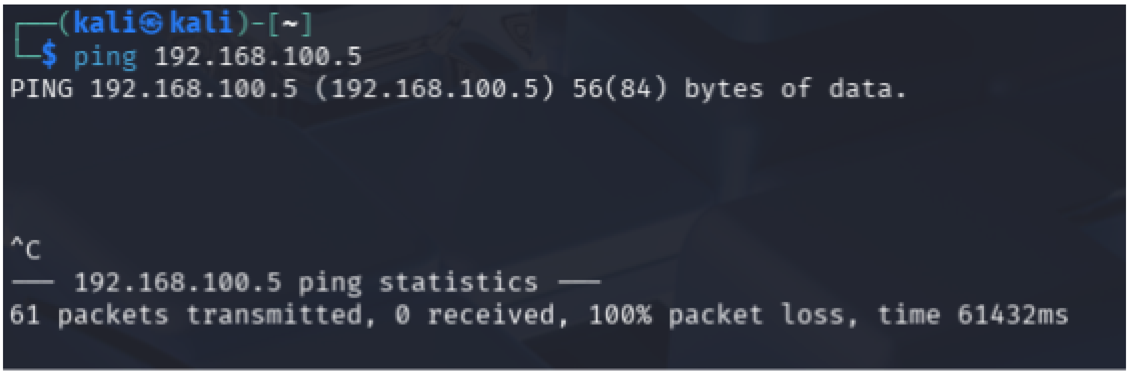
After running the metasploit exploit or straight up using the “:)” character while connecting the ip is banned using crowdsec. This can be confirmed with running this command,

- **sudo cscli decisions list**



```
ghxst@dummy:~$ sudo cscli decisions list
[sudo] password for ghxst:
| ID | Source   | Scope:Value       | Reason                          | Action | Country | AS | Events | expiration | Alert ID |
| 1  | crowdsec | Ip:192.168.100.14 | custom/vsftpd-backdoor-attempt  | ban    |         |    | 1      | 3h57m25s   | 1        |
```

Figure 11: Crowdsec ban list



```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.100.5
PING 192.168.100.5 (192.168.100.5) 56(84) bytes of data.


^C
── 192.168.100.5 ping statistics ──
61 packets transmitted, 0 received, 100% packet loss, time 61432ms
```

Figure 12: IP Ban confirmation with ping test

8.5. Incident Report: vsftpd 2.3.4 Backdoor Exploitation

Date: January 29, 2026

Severity: Critical

Subject: Detection and Containment of CVE-2011-2523

Incident Overview

A security breach simulation was conducted targeting the FTP service running *vsftpd 2.3.4*. The attack utilized a known backdoor vulnerability triggered by a specific username payload, intended to open an unauthorized root shell on the host system.

Findings

Forensic analysis revealed two primary Indicators of Compromise (IoCs):

- **Log Analysis:** The vsftpd logs recorded a connection attempt containing the malicious signature USER tttt:).
- **System State:** Post-exploitation monitoring identified an unauthorized listener active on TCP port 6200, confirming that the backdoor had successfully executed and opened a command shell.

Actions Taken

A multi-layered defense strategy was deployed using Wazuh and CrowdSec. A custom CrowdSec parser was configured to inspect FTP command logs for the "smiley face" pattern. Upon detection of the malicious payload, the CrowdSec decision engine immediately issued a ban, adding the source IP to the firewall blocklist and terminating the connection.

Recommendations

- **Patch Management:** Immediately upgrade vsftpd to a stable, non-vulnerable version.
- **Network Segmentation:** Configure firewall rules to explicitly block ingress traffic on non-standard ports (specifically port 6200).
- **Continuous Monitoring:** Maintain real-time log analysis for anomalous payload signatures to detect future exploitation attempts.

9. Conclusion

This document provides a structured, hands-on approach to practicing key cybersecurity operations using open-source tools. By combining technical execution, analysis, and professional documentation, learners gain practical experience aligned with real-world SOC workflows.

10. References

- Elastic Security
- Sigma Rules
- REMnux
- Metasploit Framework Documentation
- OpenVAS documentation
- DefectDojo
- Wazuh
- CrowdSec
- Velociraptor
- MITRE Caldera
- Suricata
- MITRE ATT&CK Matrix
- Other community tutorials and security blogs

By: Karthik R