Cyberster

INTERNSHIP REPORT

Zeus Banking Trojan | njRAT | Static & Dynamic Analysis | Suricata | ANY.RUN | NIST IR Plan

Week 5: Malware Analysis and Incident Response

Submitted to: Cyberster Internship Program

Intern Name: Abdul Muqeet Tabraiz

Roll Number: CSI-B1-353

Week: 5 (Days 29–35)

A Weekly Progress and Learning Summary

Table of Contents

Summary of Days 34–35

Week 5 Conclusion

Days 29–30 — Sample Selection and Environment Lockdown

Objective

The objective of Days 29 and 30 was to select two well-documented malware samples from a reputable repository, safely download them into an isolated Kali Linux virtual machine, compute cryptographic hashes for verification, establish a network airgap to prevent any outbound communication, apply restrictive file permissions, and record a baseline network capture for later comparison with malware-generated traffic.

Tools Used

- Kali Linux VM (Wazuh Agent) — Primary analysis workstation
- VMware Workstation — VM hypervisor for network adapter management
- pfSense CE 2.7.2 — Firewall gateway for WAN interface control
- Firefox Browser (inside Kali) — Downloading samples from theZoo repository
- sha256sum / md5sum — Cryptographic hash generation
- tcpdump — Baseline network traffic capture
- chmod — File permission restriction
- theZoo (GitHub) — Malware sample repository

Procedure

Step 1: Verify Internet Connectivity

Before downloading any malware samples, internet connectivity was verified from within the Kali VM. The lab environment routes all traffic through pfSense, so the pfSense VM was powered on first. A ping test to google.com confirmed successful outbound connectivity with 0% packet loss.

```
┌──(kali㉿kali)-[~]
└─$ ping -c 4 google.com
PING google.com (142.250.202.14) 56(84) bytes of data.
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=1 ttl=114 time=42.6 ms
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=2 ttl=114 time=58.9 ms
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=3 ttl=114 time=44.7 ms
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=4 ttl=114 time=46.3 ms

── google.com ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 42.551/48.119/58.890/6.359 ms
```

Fig 29.1 — Successful ping to google.com confirming internet connectivity through pfSense

Step 2: Create Secure Malware Lab Directory Structure

A dedicated directory structure was created to isolate malware samples and analysis reports. The directories were secured with chmod 700 permissions, restricting access exclusively to the current user. This prevents accidental execution or unauthorized access to malware files.

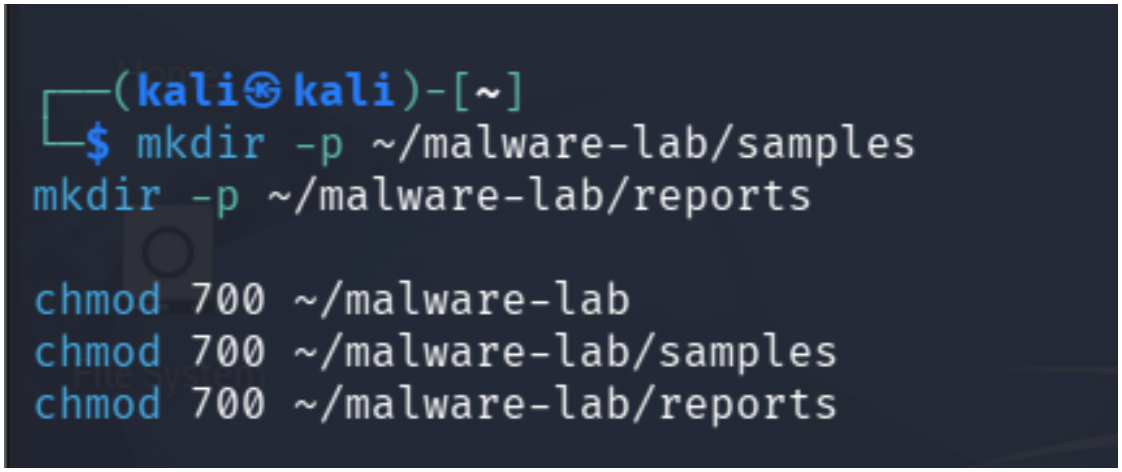


Fig 29.2 — Creating malware-lab directory structure with mkdir -p and applying chmod 700 permissions

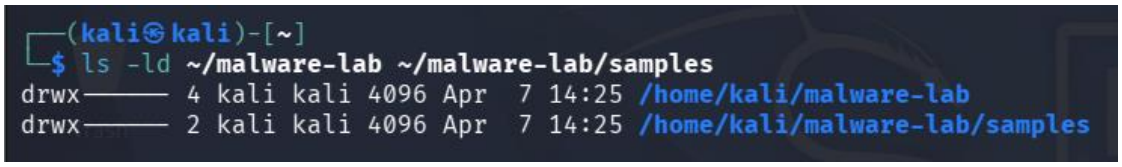


Fig 29.3 — Verifying directory permissions with ls -ld (drwx------)

Step 3: Select Malware Samples

Two malware samples were selected from theZoo repository on GitHub, a well-known curated collection of live malware binaries used for research and educational purposes:

Sample 1: Zeus Banking Trojan (ZeusBankingVersion_26Nov2013)

Zeus is a notorious banking trojan that targets Windows systems to steal financial credentials through keystroke logging, form grabbing, and man-in-the-browser attacks. It was selected because it represents the infostealer/banking trojan category and has extensive documentation for analysis comparison.

Sample 2: njRAT v0.6.4 (Remote Access Trojan)

njRAT (also known as Bladabindi) is a widely distributed Remote Access Trojan that provides attackers with complete control over infected systems. It was selected to represent the RAT category, offering a contrasting behavioral profile to Zeus for comparative analysis.

This pairing (infostealer + RAT) provides comprehensive coverage of two distinct malware families with different attack vectors, persistence mechanisms, and network behaviors.

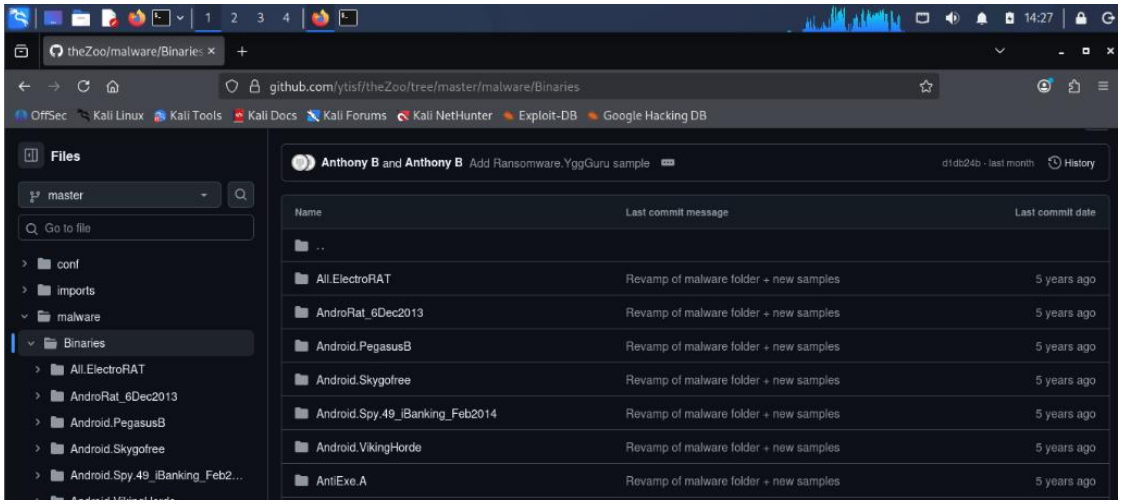


Fig 29.4 — theZoo repository Binaries directory on GitHub

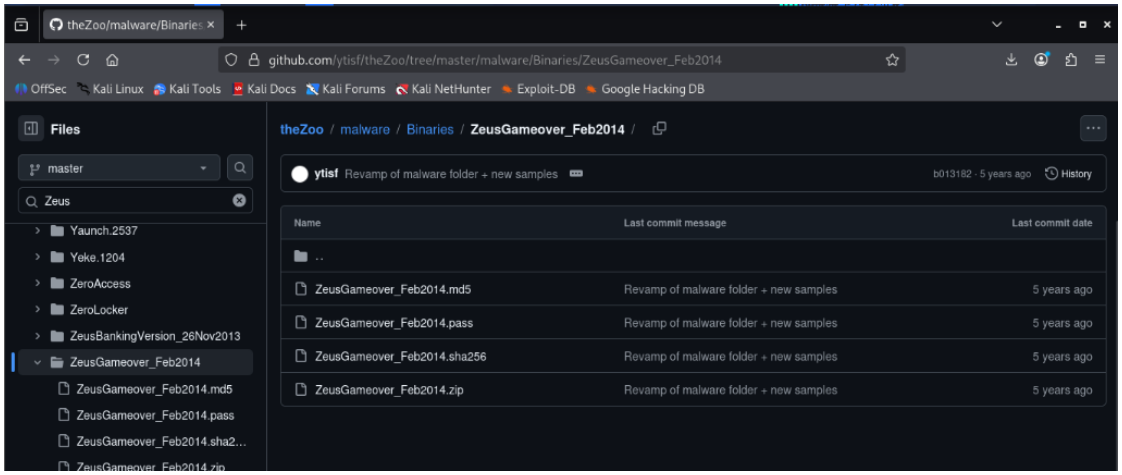


Fig 29.5 — ZeusGameover_Feb2014 directory in theZoo (explored before selecting Banking version)

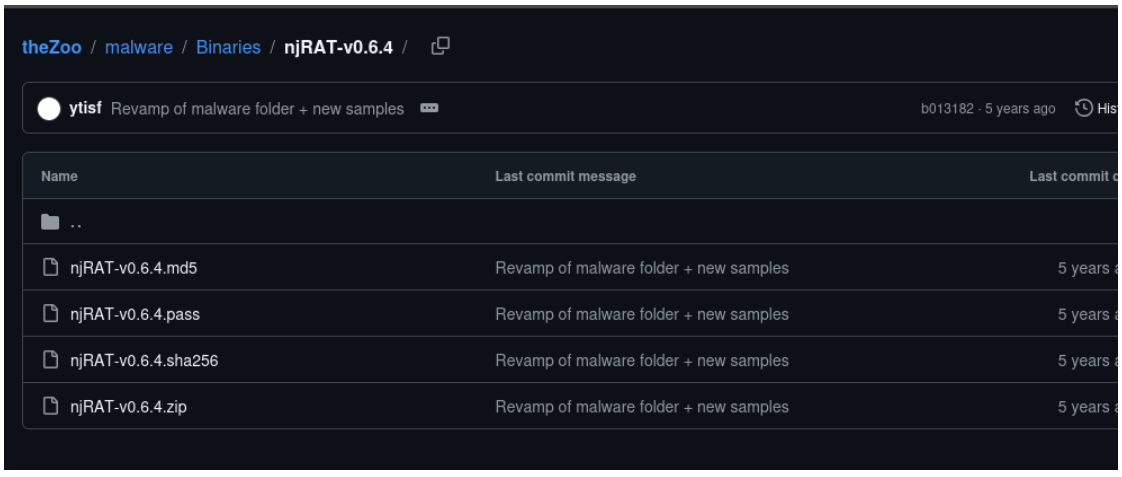


Fig 29.6 — njRAT v0.6.4 directory in theZoo showing sample files

Step 4: Download Samples (Inside Kali VM Only)

Samples were downloaded exclusively from within the Kali VM using the Firefox browser and wget. This is a critical safety requirement — downloading from outside the VM would break isolation. Only the ZIP archives were downloaded, not the full repository.

Fig 29.7 — Downloading ZeusBankingVersion using wget inside Kali VM

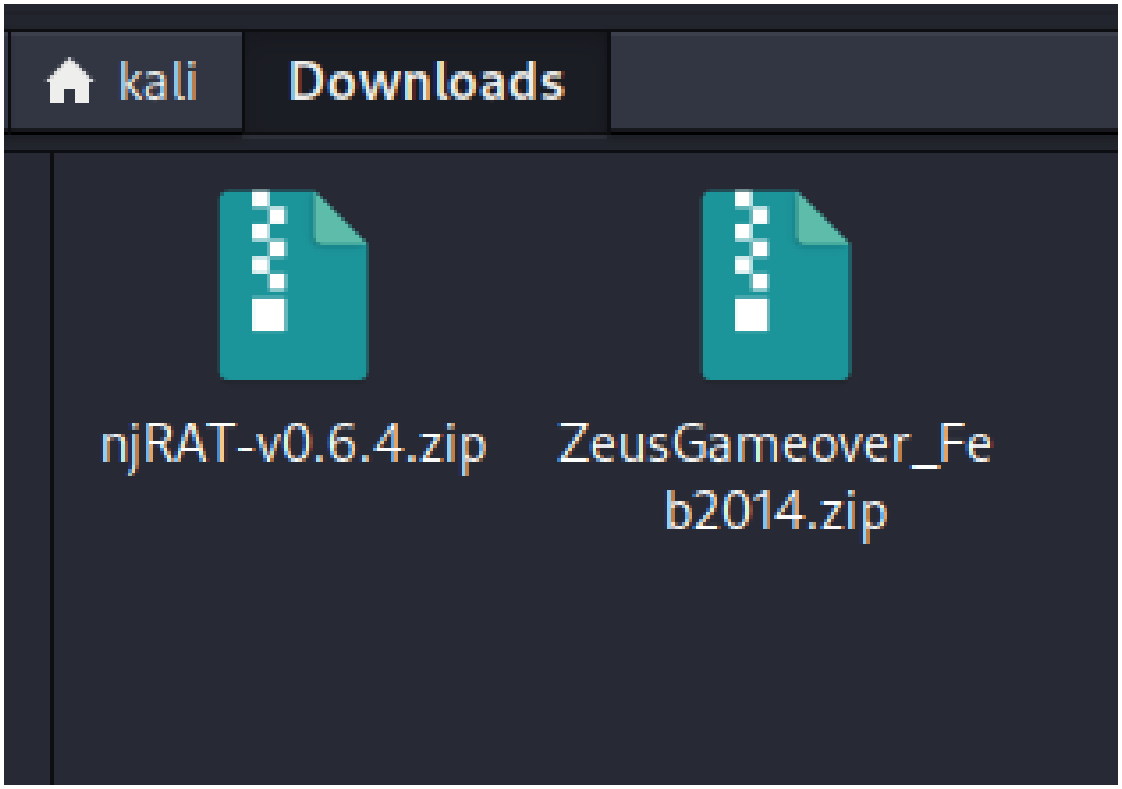


Fig 29.8 — Downloaded ZIP files visible in Kali Downloads folder

Step 5: Move and Extract Samples

The downloaded ZIP archives were moved to the secure malware-lab/samples directory. Each archive was then extracted using the password "infected" (standard password for theZoo samples). The extraction was performed within the isolated directory structure.

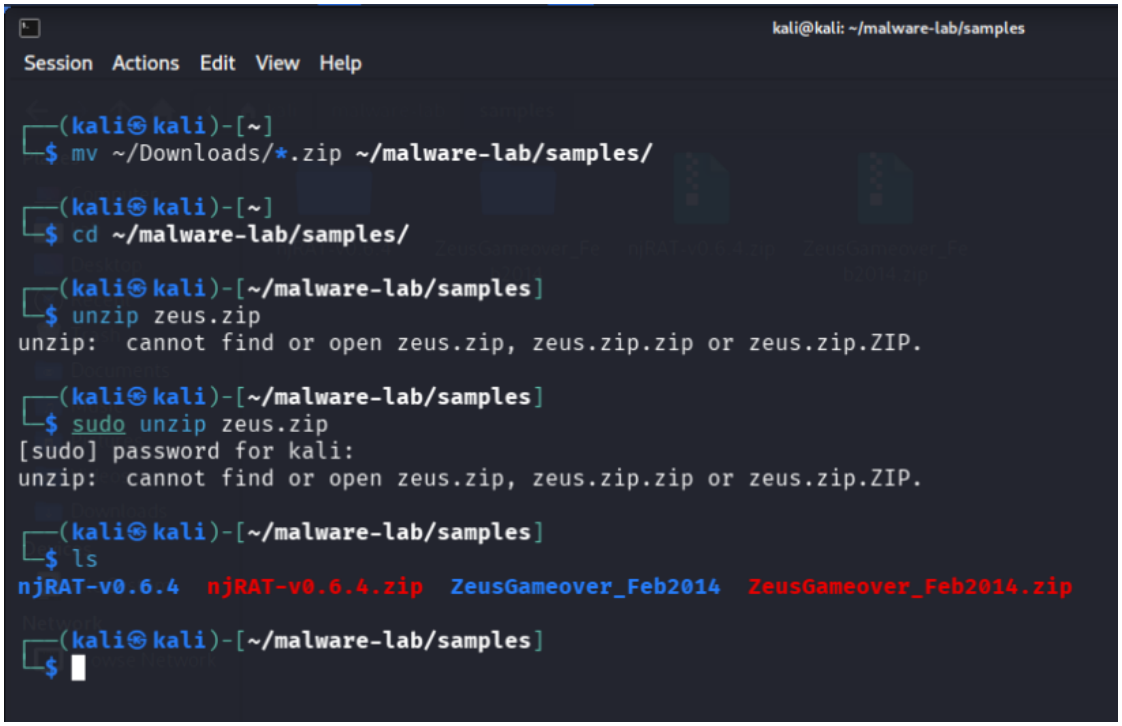


Fig 29.9 — Moving ZIPs to samples directory and listing extracted contents

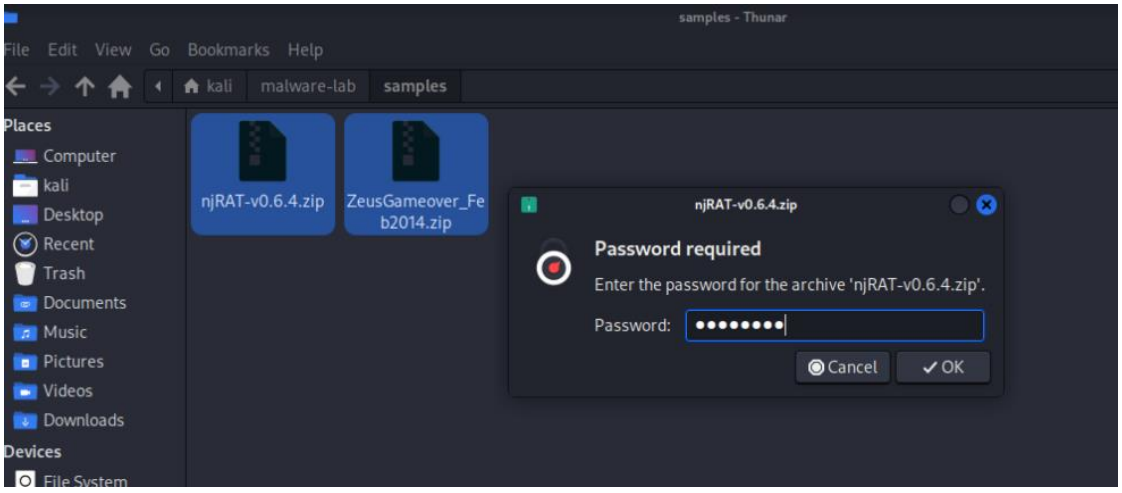


Fig 29.10 — Extracting njRAT archive with password prompt

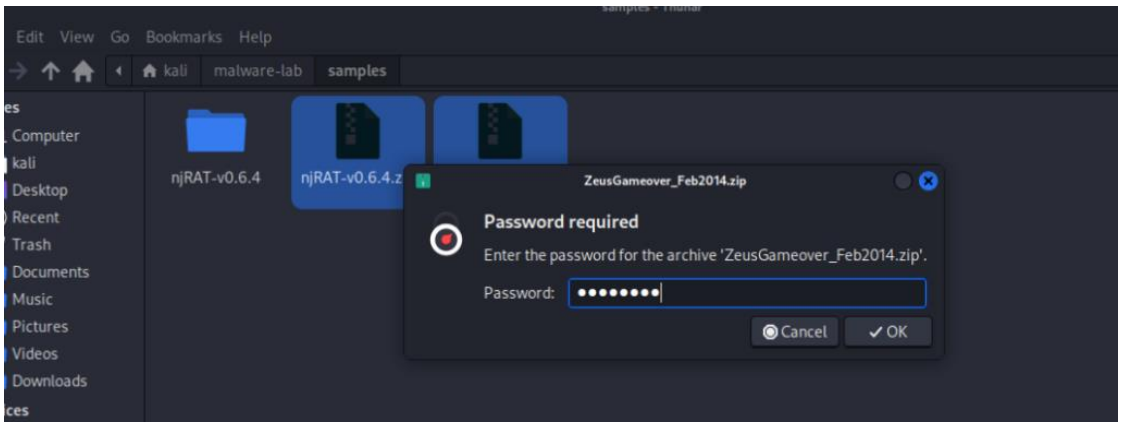


Fig 29.11 — Extracting ZeusGameover archive with password prompt

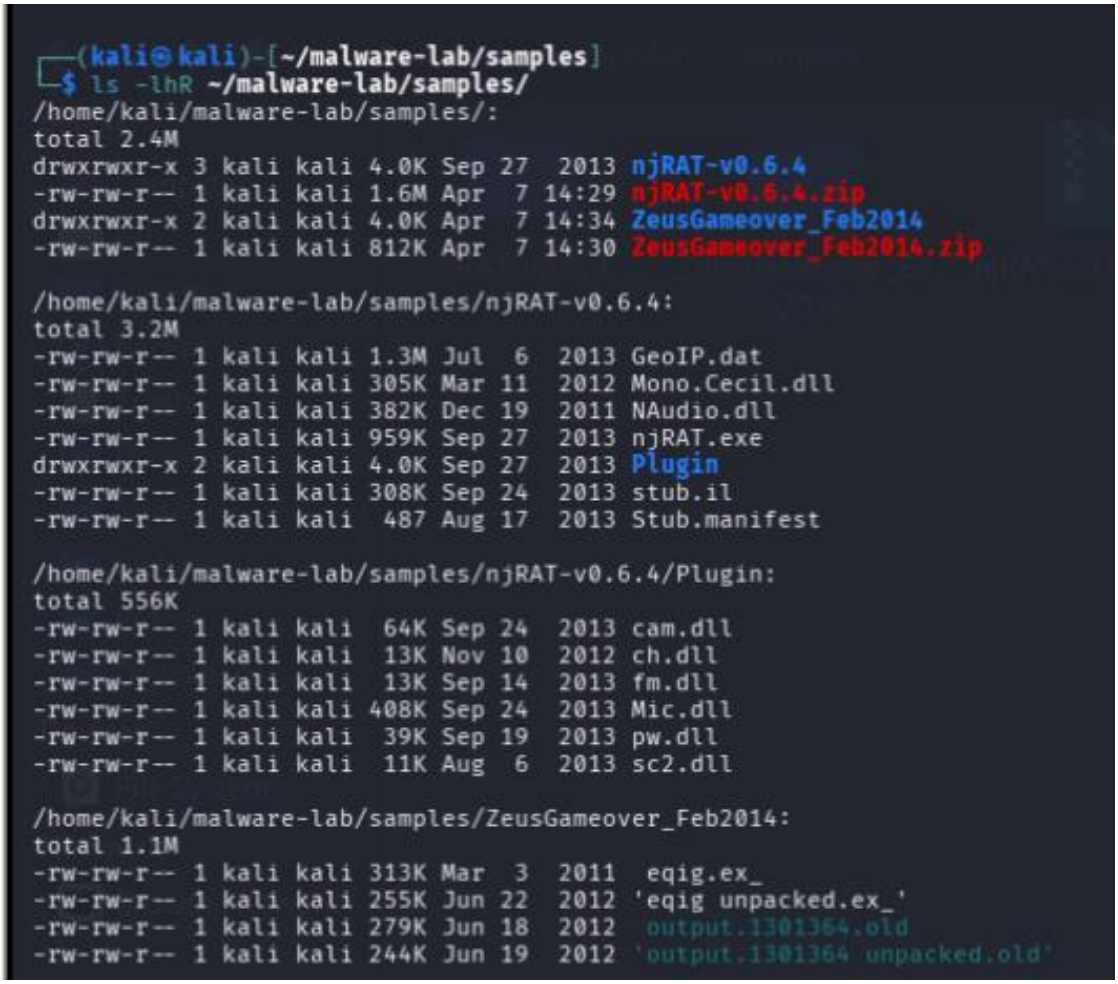


Fig 29.12 — Complete directory listing (ls -lhR) showing all extracted samples and file sizes

Step 6: Generate Cryptographic Hashes

SHA-256 and MD5 hashes were computed for both malware samples before any analysis. Hashing before analysis is critical because malware can modify itself or drop new payloads during execution. These hashes serve as the authoritative identity of each sample for VirusTotal lookups, SIEM detection rules, and threat intelligence correlation.

Fig 29.13 — SHA-256 and MD5 hashes for njRAT.exe

Fig 29.14 — SHA-256 and MD5 hashes for Zeus invoice_2318362983713_823931342io.pdf.exe

Sample	Algorithm	Hash Value
njRAT.exe	SHA-256	fd624aa205517580e83fad7a4ce4d64863e95f62b34ac72647b1974a52822199
njRAT.exe	MD5	0431311b5f024d6e66b90d59491f2563
Zeus (invoice...pdf.exe)	SHA-256	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169
Zeus (invoice...pdf.exe)	MD5	ea039a854d20d7734c5add48f1a51c34

Note: The Zeus sample uses a double extension technique (.pdf.exe), a classic social engineering trick designed to deceive users into believing the file is a legitimate PDF invoice when it is actually a Windows PE executable.

Step 7: Establish Network Airgap

The network airgap was established using a dual approach to ensure complete isolation:

Method 1 — VMware Network Adapter Disconnection:

The NAT adapter in VMware Workstation was disconnected (unchecked "Connected" and "Connect at power on") to remove the primary internet path. The Host-Only adapter was retained for internal lab communication with the Wazuh Manager.

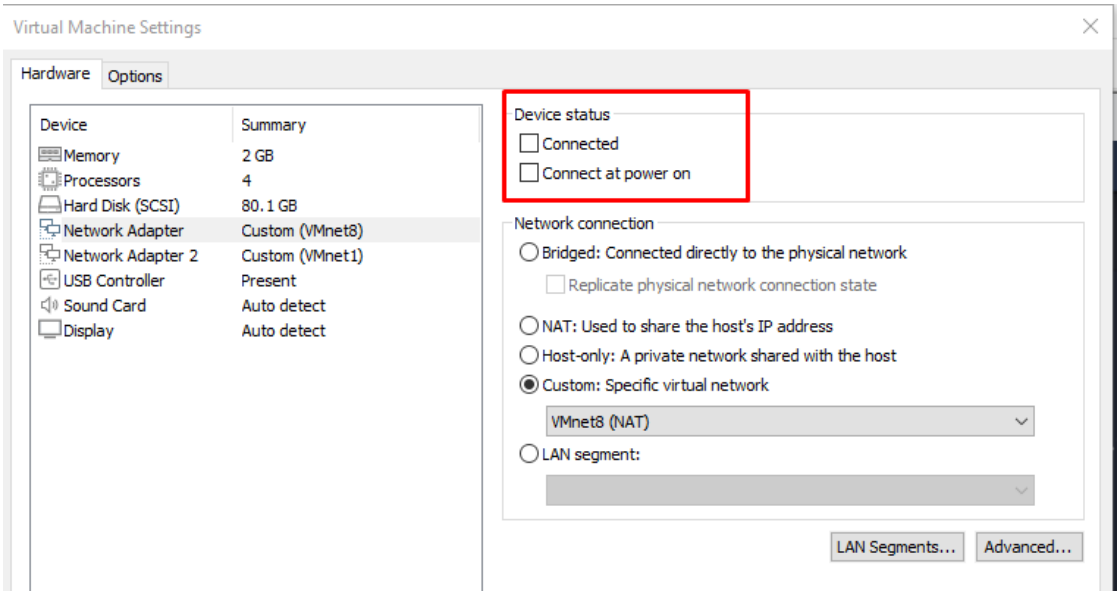


Fig 29.15 — VMware VM Settings showing NAT adapter disconnected (Device Status unchecked)

Method 2 — pfSense WAN Interface Disabled (Recommended):

Because the lab environment routes all traffic through pfSense, the WAN interface (em0) was disabled from the pfSense web interface. This blocks internet access for ALL VMs while preserving internal routing between lab machines. This is the preferred method as it mirrors real SOC network segmentation practices.

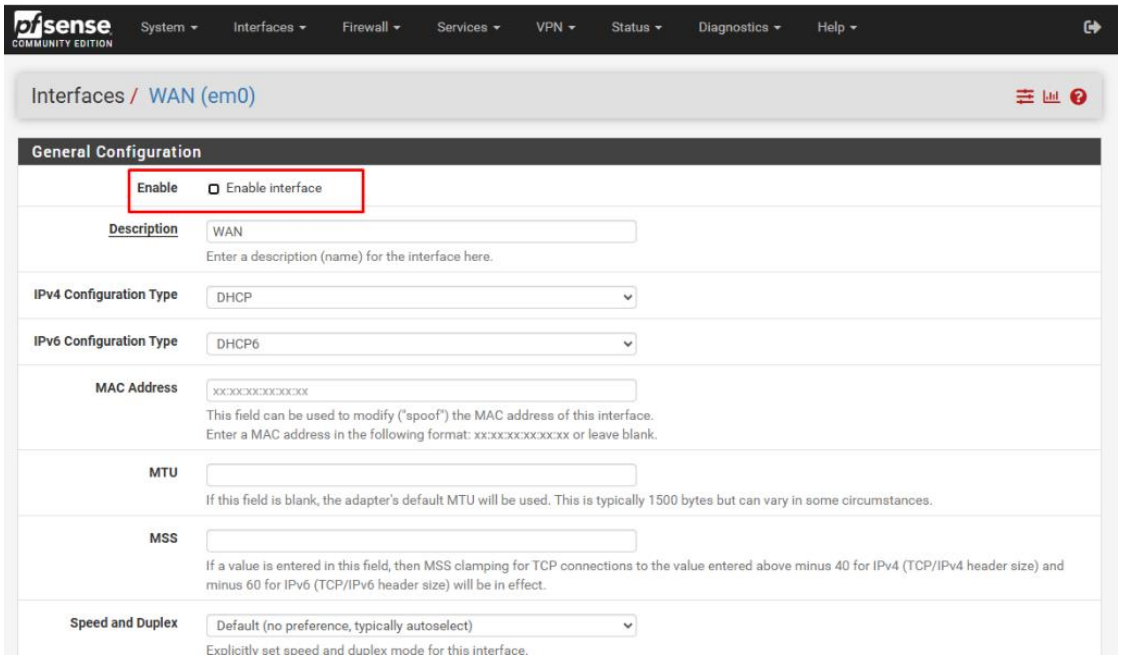


Fig 29.16 — pfSense WAN interface with Enable checkbox unchecked

Step 8: Verify Airgap

The airgap was verified by attempting to ping external addresses. With the pfSense WAN interface disabled, all outbound connectivity was blocked, confirming 100% packet loss.

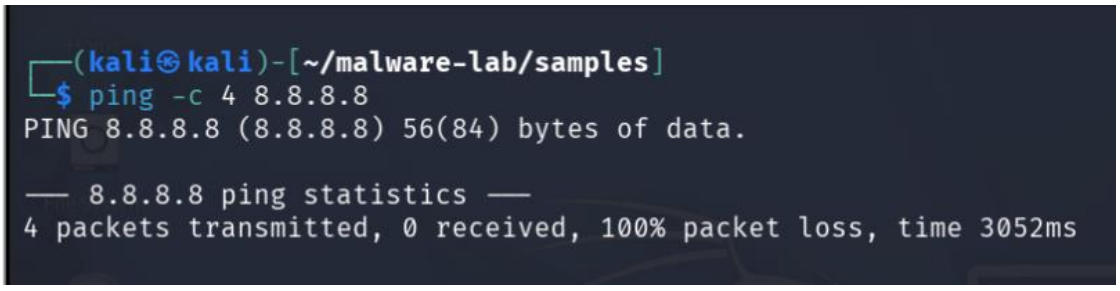


Fig 29.17 — Ping to 8.8.8.8 showing 100% packet loss (airgap confirmed)

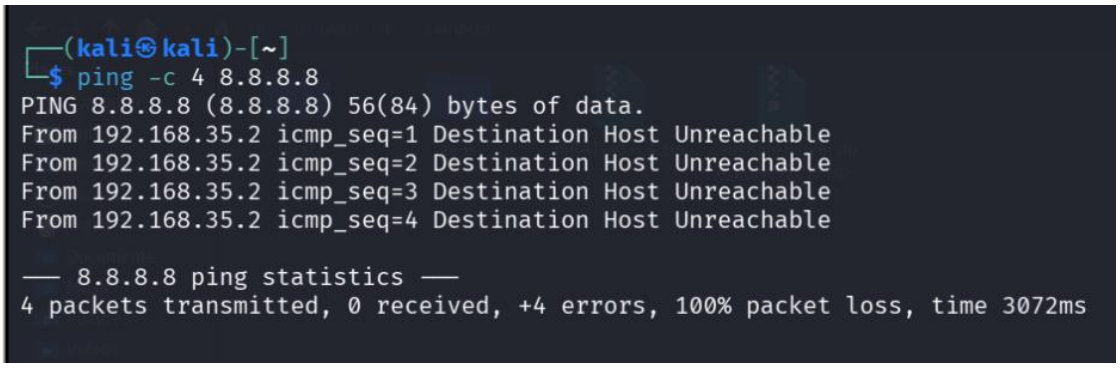


Fig 29.18 — Additional airgap verification showing Destination Host Unreachable

Step 9: Restrict File Permissions

File permissions were further restricted using chmod 600 on all extracted executables. This sets read and write only (no execute permission), preventing accidental execution of malware binaries on the Kali system.

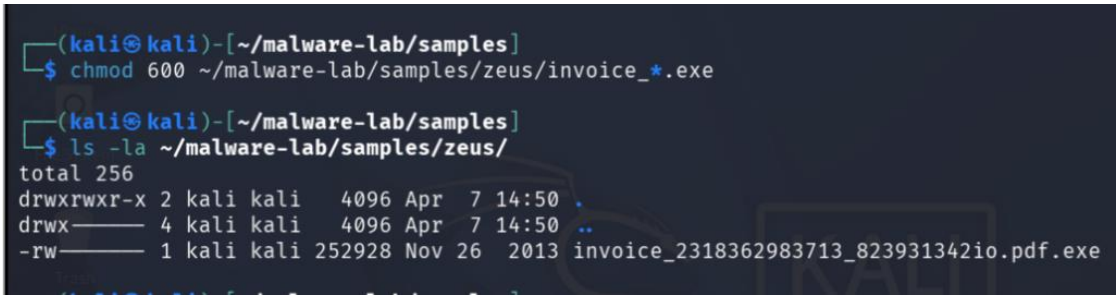


Fig 29.19 — chmod 600 applied and ls -la showing -rw------- permissions on Zeus sample

Step 10: Baseline Network Capture

A baseline network capture was recorded to establish normal traffic patterns before any malware analysis. The active network interface was identified as eth1 (192.168.35.128) using the ip a command, as eth0 was in a DOWN state. Controlled ICMP traffic was generated by pinging the internal Ubuntu VM (Wazuh Manager) to ensure meaningful baseline data.

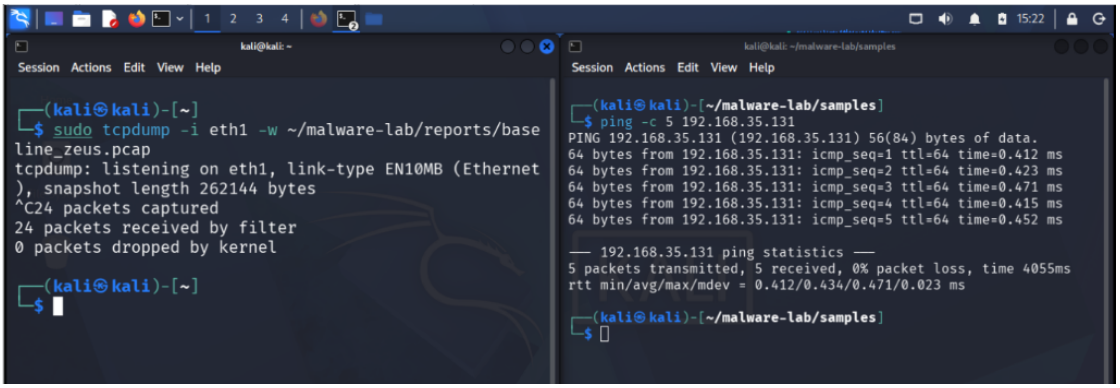


Fig 29.20 — tcpdump capture on eth1 with simultaneous internal ping to Wazuh Manager

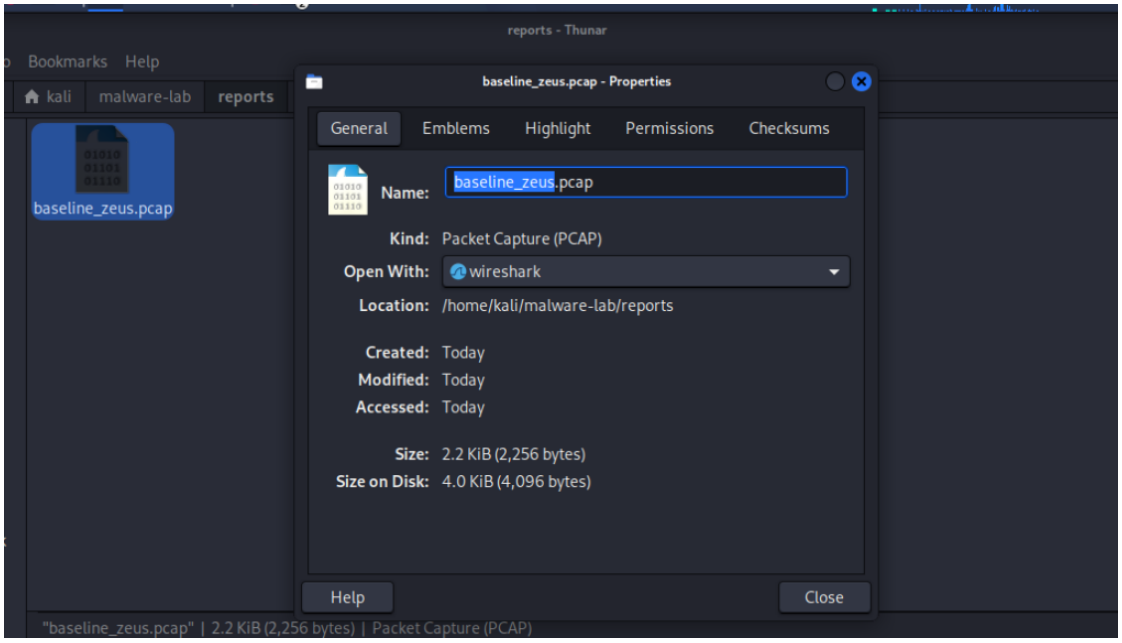


Fig 29.21 — baseline_zeus.pcap file properties confirming successful capture

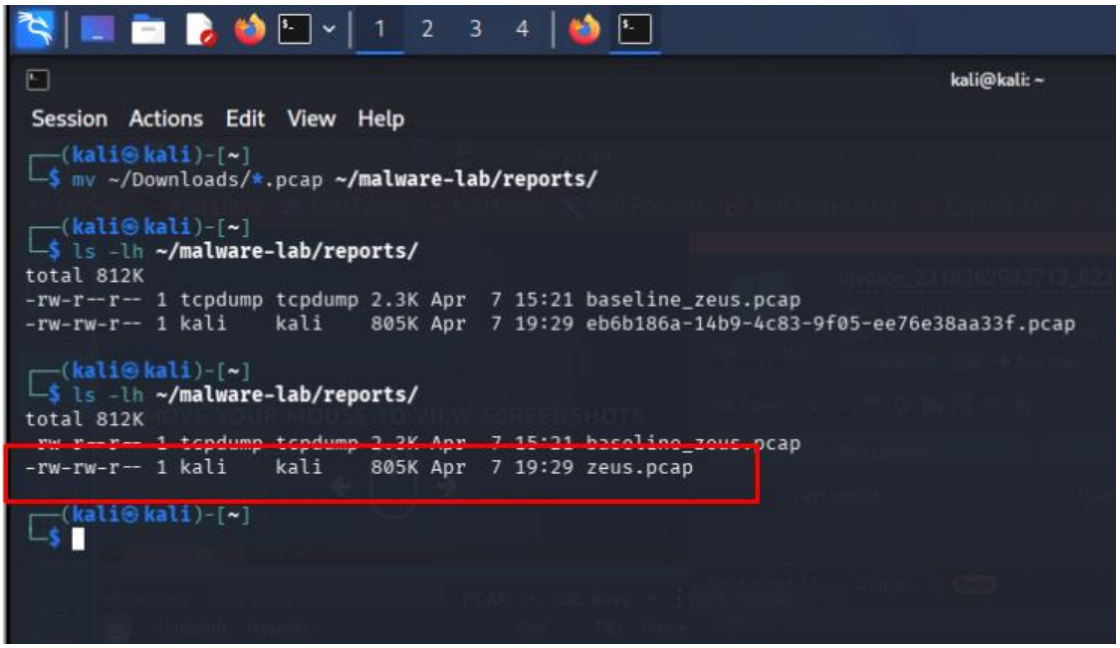


Fig 29.22 — Reports directory showing PCAP files with rename for clarity

Result

Both malware samples (Zeus Banking Trojan and njRAT v0.6.4) were successfully acquired, extracted, and verified. Cryptographic hashes were generated and recorded. The network was fully airgapped using both VMware adapter disconnection and pfSense WAN disabling. File permissions were restricted to prevent accidental execution. A baseline network capture was successfully recorded on the correct interface (eth1) with internal ICMP traffic for later comparison.

Summary of Days 29–30

Days 29 and 30 established the critical foundation for safe malware analysis. The "Connect-Download-Verify-Airgap-Analyse" sequence was followed precisely to maintain operational security. The lab environment, configured with pfSense as the network gateway, required a modified airgap strategy: rather than simply disconnecting the NAT adapter, the pfSense WAN interface was disabled, ensuring no outbound internet path existed while preserving internal communication between Kali and the Wazuh Manager. The correct network interface (eth1) was identified for traffic capture, avoiding the common pitfall of capturing on an inactive interface. These preparatory steps ensure that all subsequent analysis occurs in a controlled, isolated environment.

Day 31 — Static Malware Analysis

Objective

The objective of Day 31 was to perform static analysis on the Zeus Banking Trojan sample without executing it. Static analysis involves examining the binary's metadata, structure, embedded strings, and file characteristics to extract Indicators of Compromise (IOCs) and understand the malware's capabilities before dynamic execution.

Tools Used

- file — File type identification (magic bytes)
- exiftool — Metadata extraction (timestamps, compiler info)
- binwalk — Embedded file/signature detection
- strings — ASCII string extraction
- strings -el — Unicode (UTF-16 Little Endian) string extraction
- grep — Pattern matching for IOC extraction
- sha256sum — Hash verification for VirusTotal lookup
- VirusTotal — Online malware scanning service (hash lookup only)

Procedure

Step 1: File Type Identification

The file command was used to determine the true file type by reading magic bytes. Despite the misleading .pdf.exe double extension, the file was correctly identified as a Windows PE32 executable. This confirms the sample is a compiled Windows binary that cannot execute natively on Linux, and its .pdf prefix is purely a social engineering deception.

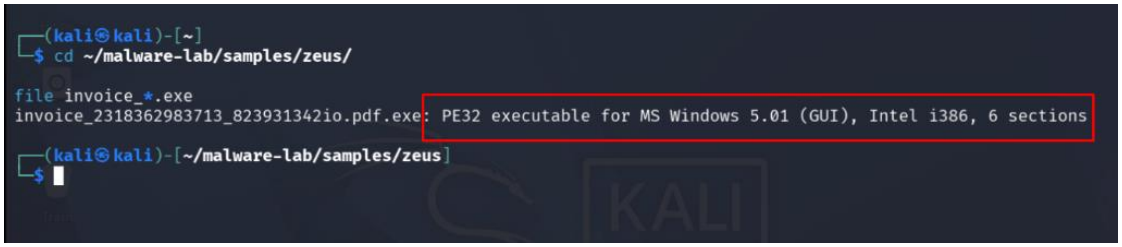


Fig 31.1 — file command output: PE32 executable (GUI) Intel 80386, for MS Windows, 6 sections

Step 2: Metadata Analysis with ExifTool

ExifTool was used to extract detailed metadata from the PE executable. Key findings include:

- File Type: Win32 EXE (PE32)
- Machine Type: Intel 386 or later, and compatibles

- Time Stamp: 2013-11-25 15:32:03 (compilation date)
- PE Type: PE32, Linker Version: 10.0
- File Size: 253 kB
- Subsystem: Windows GUI
- Entry Point: 0xa3b6

The compilation timestamp from 2013 aligns with the known Zeus Banking Trojan timeline. The PE32 type and Windows GUI subsystem confirm this is a graphical Windows application designed to run with user interaction.

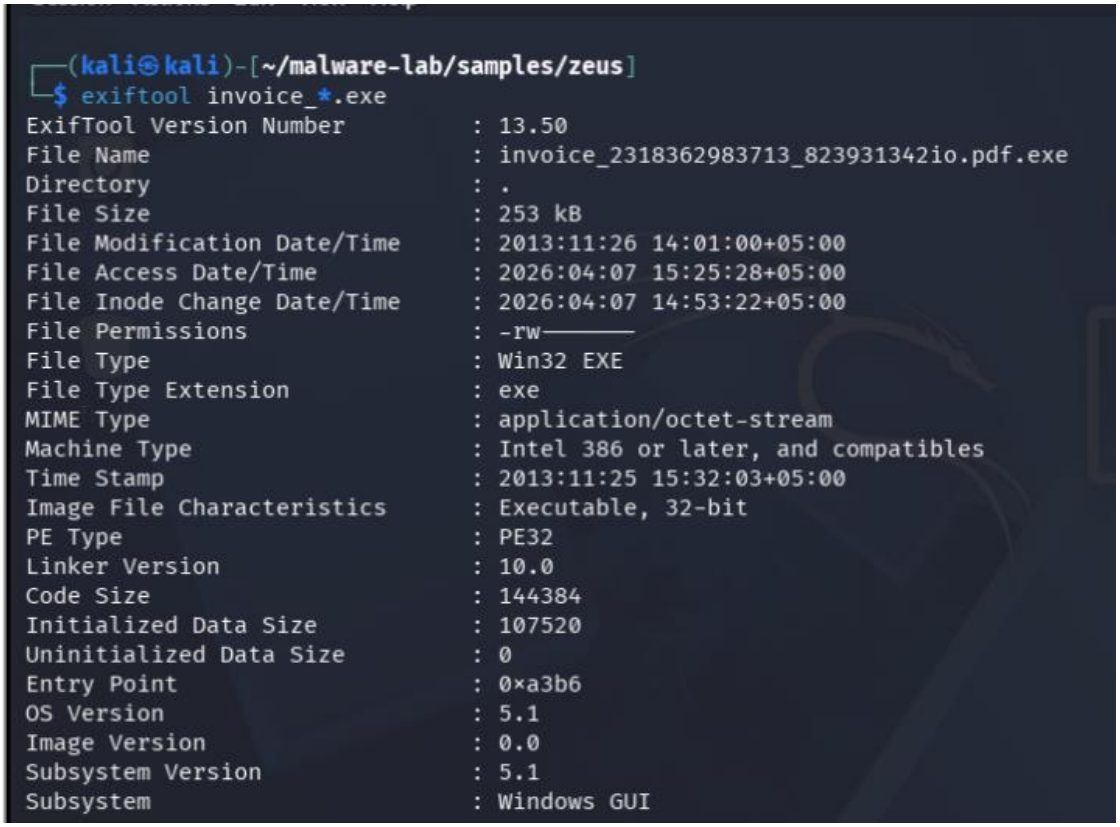


Fig 31.2 — ExifTool output showing metadata including timestamp, PE type, and machine type

Step 3: Binary Inspection with Binwalk

Binwalk was used to scan for embedded files, compressed data, and packed sections within the binary. The analysis revealed:

- Offset 0x0: Microsoft executable, portable (PE)
- Offset 0x2761B: StuffIt Deluxe Segment (data) — indicating possible packed/compressed content
- Offset 0x3C378: XML document, version 1.0

The presence of a StuffIt segment suggests the binary may contain packed or obfuscated payloads. The embedded XML document could be a configuration file or resource manifest.

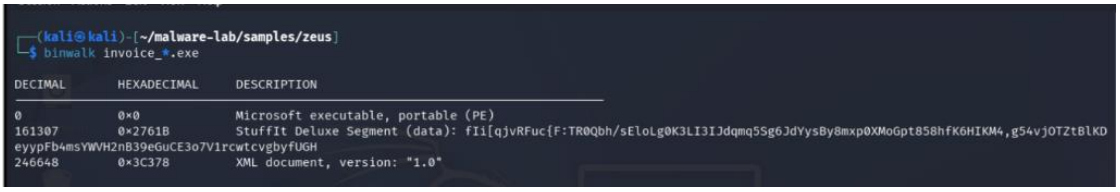


Fig 31.3 — Binwalk output showing PE header, StuffIt segment, and embedded XML

Step 4: ASCII String Extraction

ASCII strings were extracted from the binary using the strings command and saved to a file for analysis. The output reveals PE section names (.text, .data, .itext, .pdata, .rsrc, .reloc) and various readable strings that provide insight into the malware's functionality.

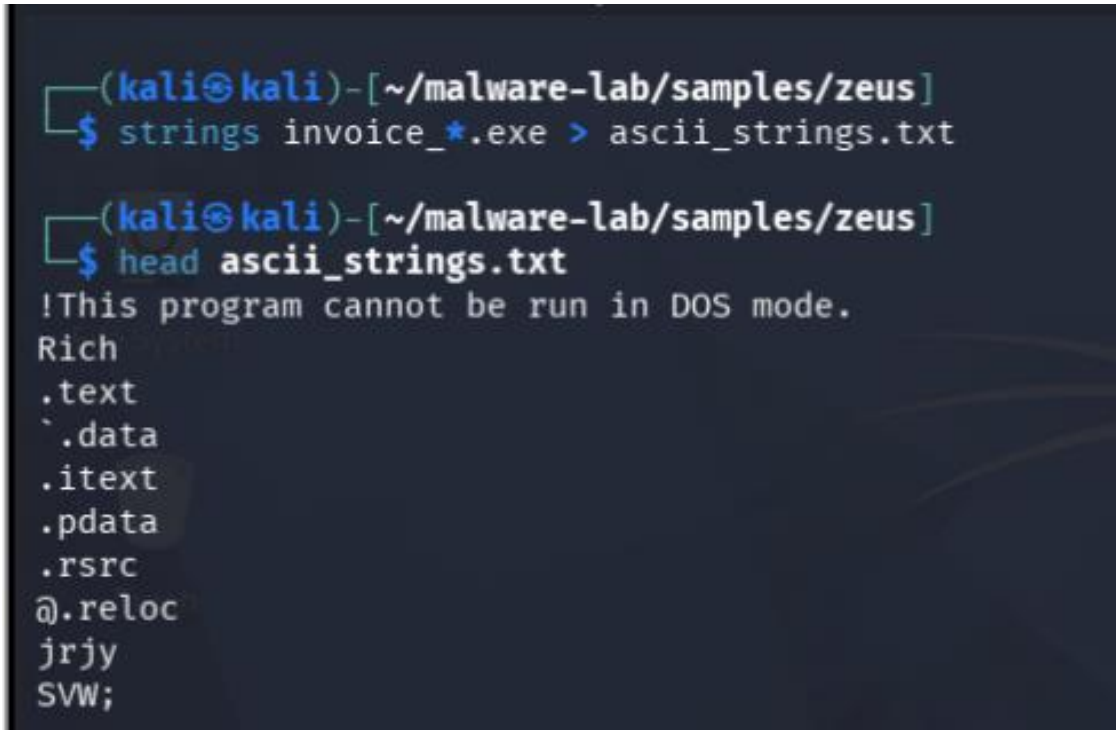


Fig 31.4 — ASCII strings output showing PE section names and initial strings

Step 5: Unicode String Extraction (Critical)

Unicode strings were extracted using strings -el (UTF-16 Little Endian), which is critical for Windows malware analysis. Windows APIs and registry paths are often stored in Unicode format. This extraction revealed additional indicators not present in the ASCII output, including API calls and system paths.

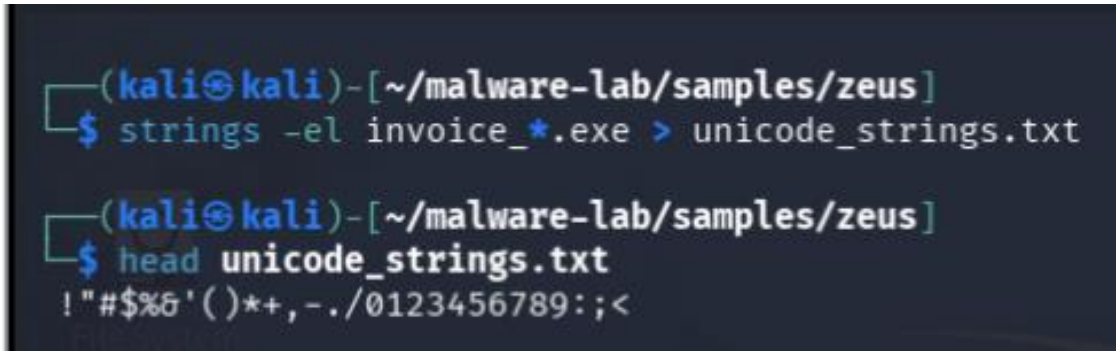


Fig 31.5 — Unicode strings showing additional character sequences

Step 6: IOC Extraction from Strings

Targeted grep searches were performed on both ASCII and Unicode string outputs to identify specific IOC categories:

6A. Domain/URL Search:

grep -E was used to search for URLs, domains, and IP addresses. A notable finding was the domain "corect.com" in the ASCII strings output, potentially indicating a C2 communication endpoint.

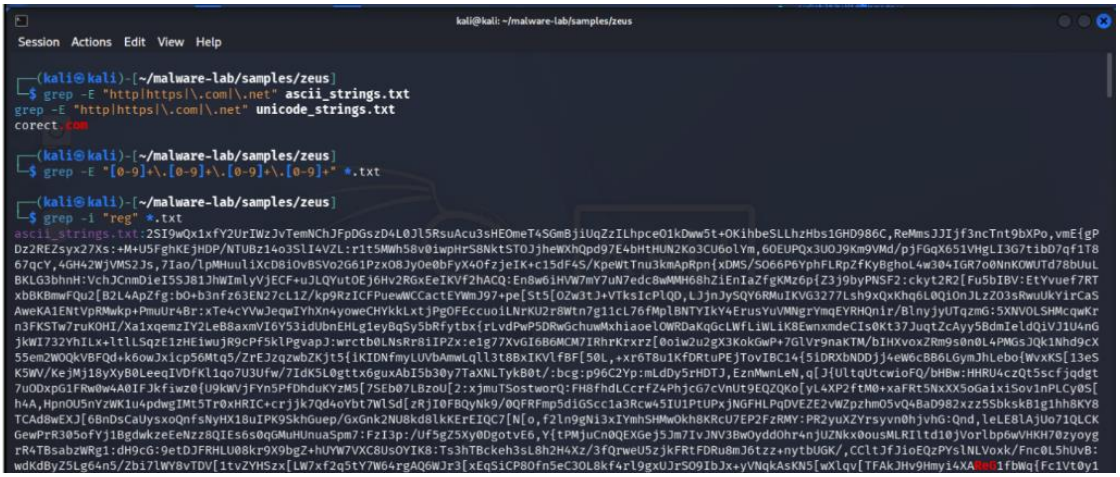


Fig 31.6 — grep results for URLs, IPs, and registry paths in strings output

6B. File Path and Registry Search:

Searches for user directories, AppData paths, and temporary folder references revealed extensive references to USER32.dll API functions (GetUserDefaultUILanguage, SetDlgItemTextW, CreateCaret, GetClassInfoW, etc.), indicating the malware interacts heavily with the Windows user interface subsystem for form grabbing and keylogging.

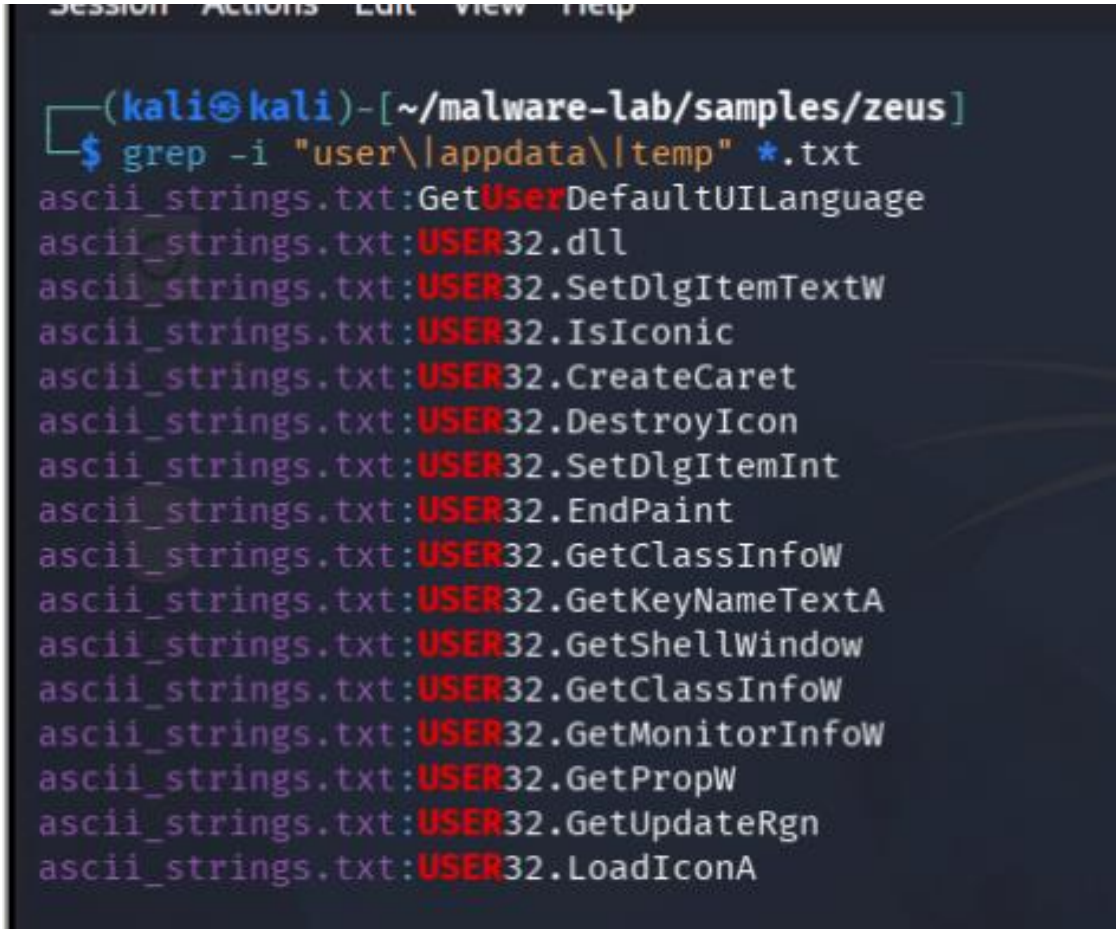


Fig 31.7 — grep results for user/appdata/temp paths showing USER32.dll API references

Step 7: VirusTotal Hash Lookup

The SHA-256 hash was submitted to VirusTotal for lookup. IMPORTANT: Only the hash was searched — the file was NOT uploaded, as per the task brief requirements to avoid exposing the sample.

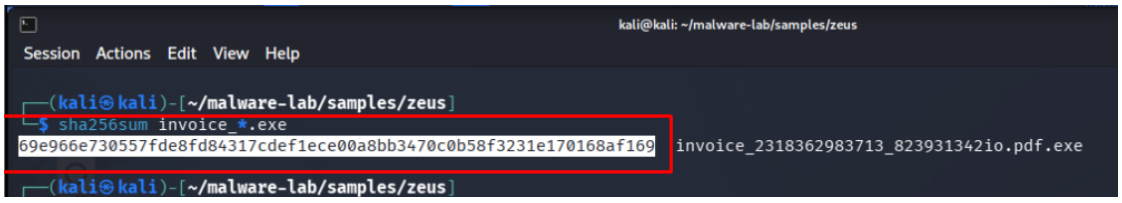


Fig 31.8 — SHA-256 hash generation for VirusTotal lookup

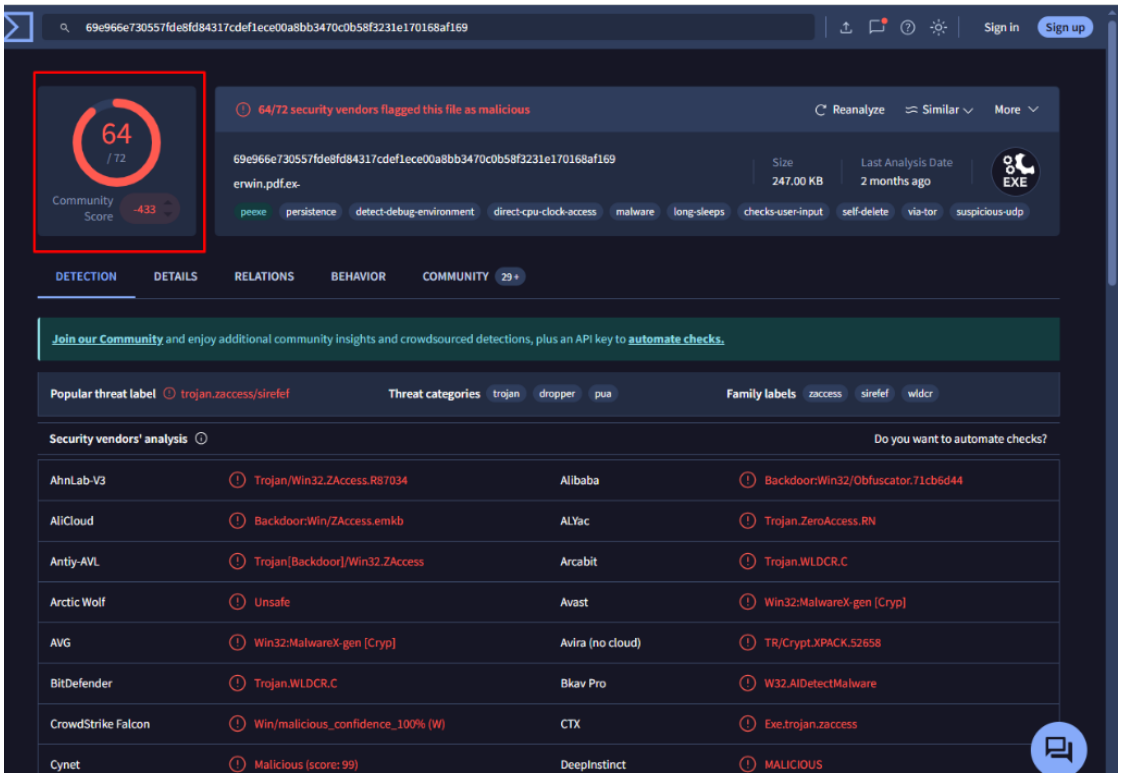


Fig 31.9 — VirusTotal detection: 64/72 vendors flagged the file as malicious (Community Score: -433)

VirusTotal results confirmed the sample is widely detected with a 64/72 detection ratio. Key findings:

- Popular threat label: trojan.zaccess/sirefef
- Threat categories: trojan, dropper, pua
- Family labels: zaccess, sirefef, wldcr
- Tags: peexe, persistence, detect-debug-environment, direct-cpu-clock-access, malware, self-delete, suspicious-udp

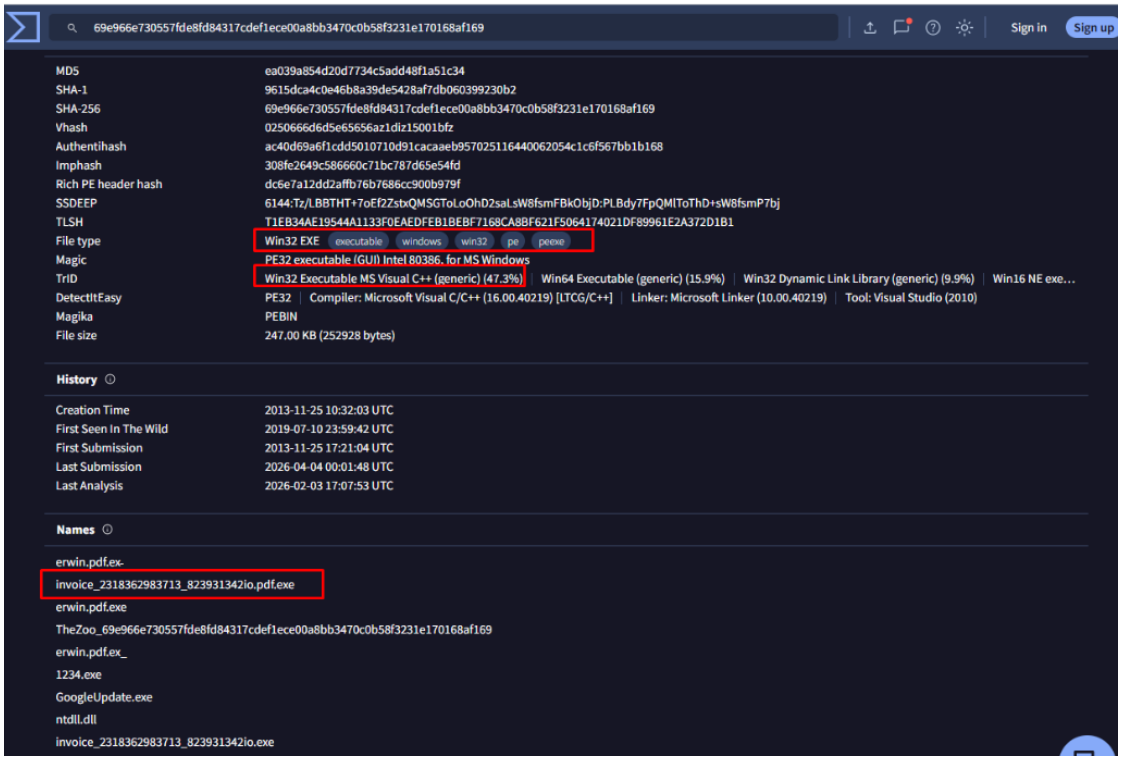


Fig 31.10 — VirusTotal detailed analysis showing file info, hashes, names, and history

Result

Static analysis successfully identified the Zeus sample as a PE32 Windows executable disguised with a double extension. Metadata revealed a 2013 compilation timestamp. Binwalk detected embedded packed content and an XML configuration. String analysis exposed Windows API references indicating form grabbing and keylogging capabilities. VirusTotal confirmed 64/72 detection with classification as trojan.zaccess/sirefef.

Initial IOC Table (from Static Analysis):

Indicator Type	Value	Source Tool	Significance
SHA-256	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169	sha256sum	Unique sample identifier
MD5	ea039a854d20d7734c5add48f1a51c34	md5sum	Secondary hash for correlation
File Type	PE32 executable (GUI) Intel i386	file	Windows binary confirmation
Compilation Date	2013-11-25 15:32:03	exiftool	Timeline correlation

Domain	corect.com	strings	Potential C2 endpoint
API References	USER32.dll (multiple functions)	strings -el	UI manipulation / keylogging
Threat Family	trojan.zaccess / sirefef	VirusTotal	Malware classification
Detection Ratio	64/72	VirusTotal	Widespread detection

Summary of Day 31

Day 31 demonstrated the value of static analysis as a safe, non-execution approach to malware investigation. By combining file type identification, metadata extraction, binary inspection, and string analysis, a comprehensive initial IOC profile was developed without ever executing the malware. The Unicode string extraction (strings -el) proved particularly valuable, revealing Windows API references not present in ASCII output — highlighting the importance of UTF-16 analysis for Windows malware. VirusTotal correlation provided authoritative classification and detection data. These findings form the foundation for the dynamic analysis phase that follows.

Cyberster Blue Team Internship — Week 5

Days 32–33 — Dynamic Analysis with ANY.RUN

Objective

The objective of Days 32–33 was to perform interactive dynamic analysis of both malware samples using the ANY.RUN cloud sandbox. Dynamic analysis involves executing malware in a controlled environment to observe real-time behavior including process creation, network activity, file system modifications, registry changes, and command-line activity. This phase converts theoretical IOCs from static analysis into confirmed behavioral indicators.

Tools Used

- ANY.RUN — Interactive online malware analysis sandbox
- Firefox (inside Kali VM) — Accessing ANY.RUN platform
- pfSense — WAN re-enabled temporarily for ANY.RUN access
- Windows 7 (32-bit) sandbox environment — ANY.RUN execution OS

Procedure

Step 1: Restore Internet Connectivity

Internet connectivity was temporarily restored by re-enabling the pfSense WAN interface. This was necessary to access the ANY.RUN platform from within the Kali VM. Connectivity was verified by pinging both google.com and 8.8.8.8.

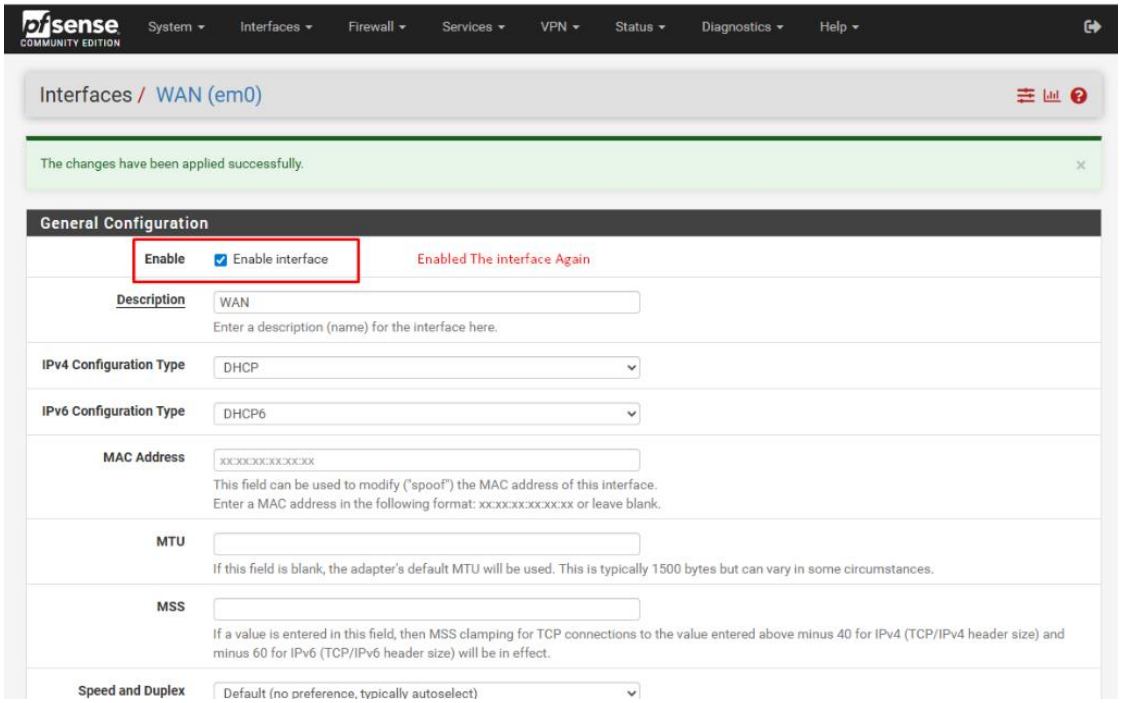


Fig 32.1 — pfSense WAN interface re-enabled with success message

```
┌──(kali㉿kali)-[~/malware-lab/samples/zeus]
└─$ ping -c 4 google.com
ping -c 4 8.8.8.8
PING google.com (142.250.202.14) 56(84) bytes of data.
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=1 ttl=114 time=100 ms
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=2 ttl=114 time=86.6 ms
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=3 ttl=114 time=53.3 ms
64 bytes from pnfjra-af-in-f14.1e100.net (142.250.202.14): icmp_seq=4 ttl=114 time=49.1 ms

── google.com ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 49.069/72.244/100.006/21.641 ms
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=58.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=48.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=53.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=114 time=42.3 ms

── 8.8.8.8 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 42.328/50.826/58.898/6.156 ms
```

Fig 32.2 — Successful ping to google.com and 8.8.8.8 after restoring connectivity

Step 2: Access ANY.RUN Platform

ANY.RUN was accessed from Firefox inside the Kali VM (never from the host machine). A free account was created to enable public analysis submissions.

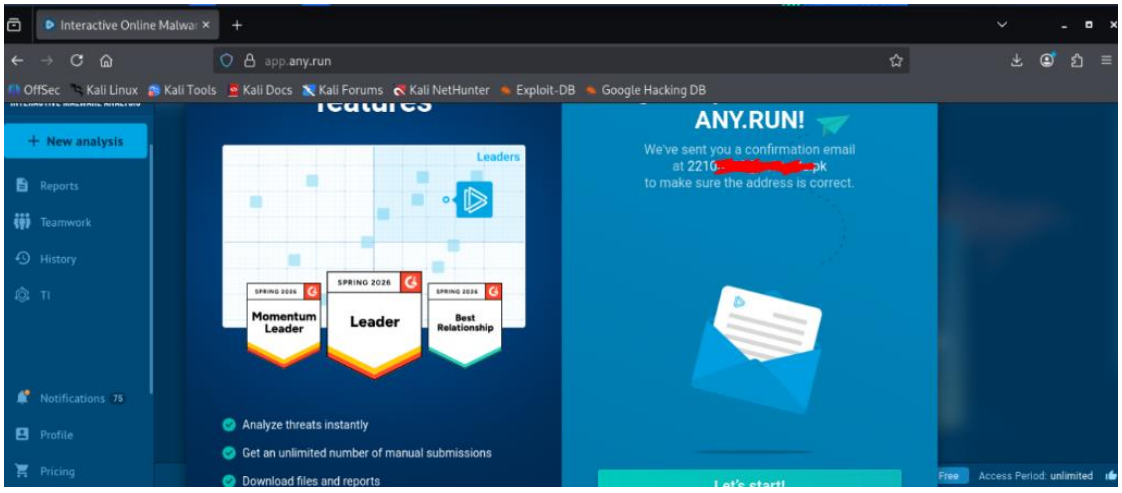


Fig 32.3 — ANY.RUN account registration confirmation inside Kali VM

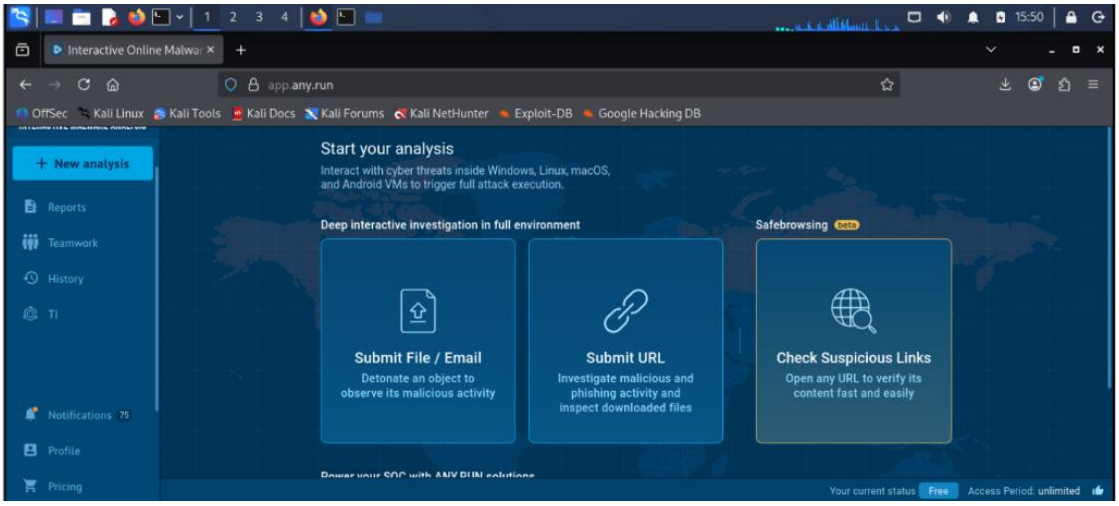


Fig 32.4 — ANY.RUN dashboard showing analysis options

Zeus Banking Trojan — ANY.RUN Analysis

Step 3A: Upload and Configure Zeus Sample

The Zeus sample (invoice_2318362983713_823931342io.pdf.exe) was uploaded to ANY.RUN from the Kali VM. Configuration: Windows 7 (32-bit), Simple mode, Public analysis.

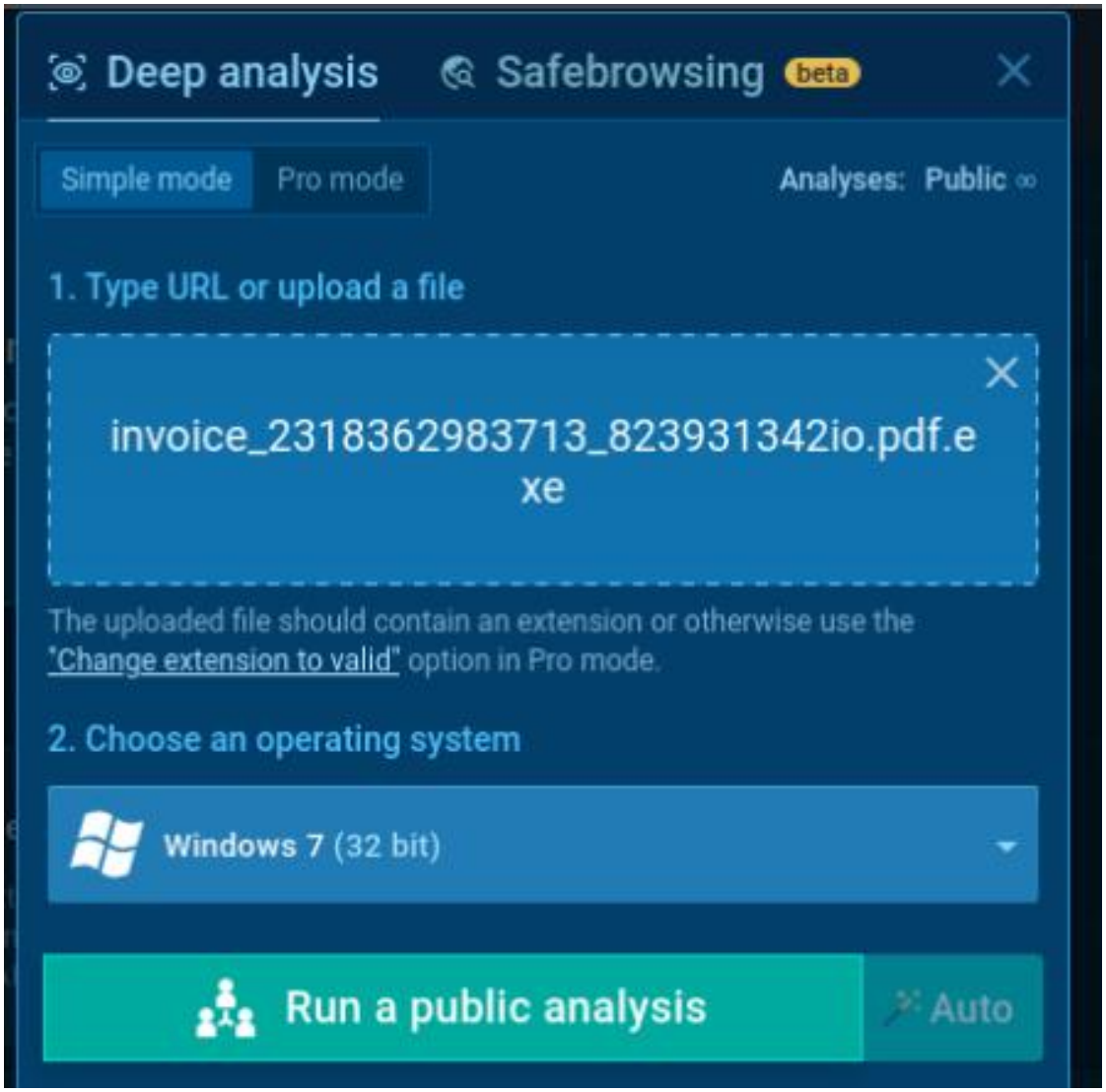


Fig 32.5 — Zeus sample uploaded to ANY.RUN with Windows 7 (32 bit) selected

Step 4A: Zeus Execution and Live Monitoring

The sample was detonated in the sandbox. ANY.RUN immediately detected the malware as ZeroAccess (a variant of the Zeus/ZAccess family) and flagged it as Malicious Activity. Five processes were spawned, and 14 network threats were detected within the 66-second analysis window.

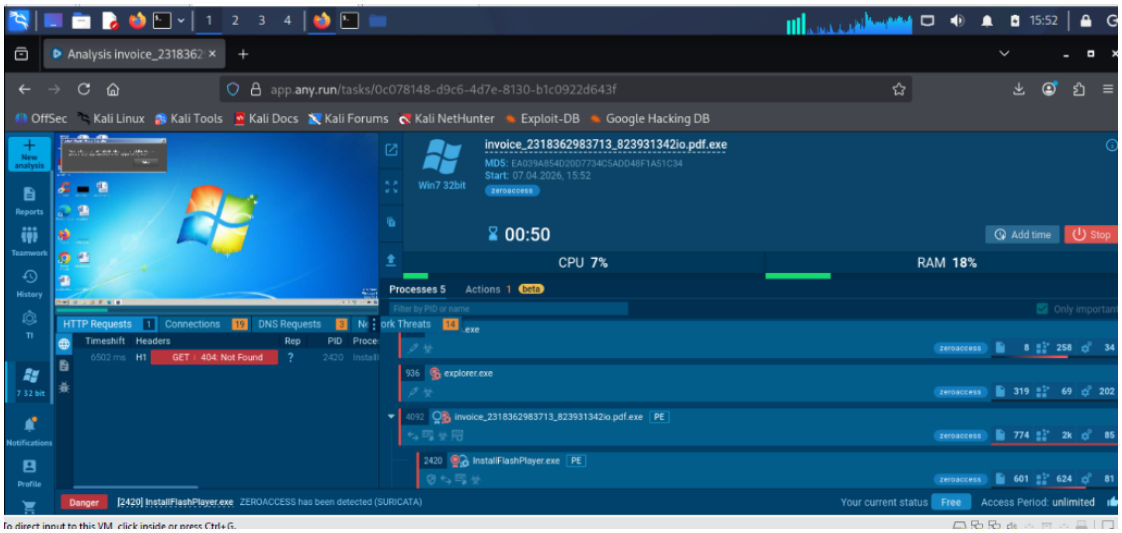


Fig 32.6 — ANY.RUN live analysis of Zeus showing processes, connections, and threat indicators

Step 5A: Zeus General Information

General Info

File name:	invoice_2318362983713_823931342io.pdf.exe
Full analysis:	https://app.any.run/tasks/0c078148-d9c6-4d7e-8130-b1c0922d643f
Verdict:	Malicious activity
Analysis date:	April 07, 2026 at 15:52:10
OS:	Windows 7 Professional Service Pack 1 (build: 7601, 32 bit)
Tags:	zeroaccess geo
Indicators:	
MIME:	application/vnd.microsoft.portable-executable
File info:	PE32 executable (GUI) Intel 80386, for MS Windows, 6 sections
MD5:	EA039A854D20D7734C5ADD48F1A51C34
SHA1:	9615DCA4C0E46B8A39DE5428AF7DB060399230B2
SHA256:	69E966E730557FDE8FD84317CDEF1ECE00A8BB3470C0B58F3231E170168AF169
SSDEEP:	6144:tg+buboEf6ZsYxQWzMTohPdrDdkyRcpD7r:GmubXyQWQToDrDnED7r

Fig 32.7 — Zeus General Info: Malicious activity verdict, ZeroAccess tags, analysis metadata

ANY.RUN General Information confirmed:

- Verdict: Malicious activity
- Tags: zeroaccess, geo
- OS: Windows 7 Professional Service Pack 1 (build: 7601, 32 bit)
- MIME: application/vnd.microsoft.portable-executable
- File info: PE32 executable (GUI) Intel 80386, for MS Windows, 6 sections
- Analysis URL: https://app.any.run/tasks/0c078148-d9c6-4d7e-8130-b1c0922d643f

Step 6A: Zeus Process Tree Analysis

The behavior graph reveals the complete execution chain:

1. invoice_2318362983713_823931342io.pdf.exe (PID 4092) — Parent process, tagged #ZEROACCESS
2. InstallFlashPlayer.exe (PID 2420) — Dropped and executed, masquerading as Flash installer
3. cmd.exe (PID 3224) — Command shell spawned for system commands
4. services.exe — System service hijacked by #ZEROACCESS
5. explorer.exe — Explorer process also compromised by #ZEROACCESS

Fig 32.8 — Zeus Behavior Graph showing process tree and #ZEROACCESS tags

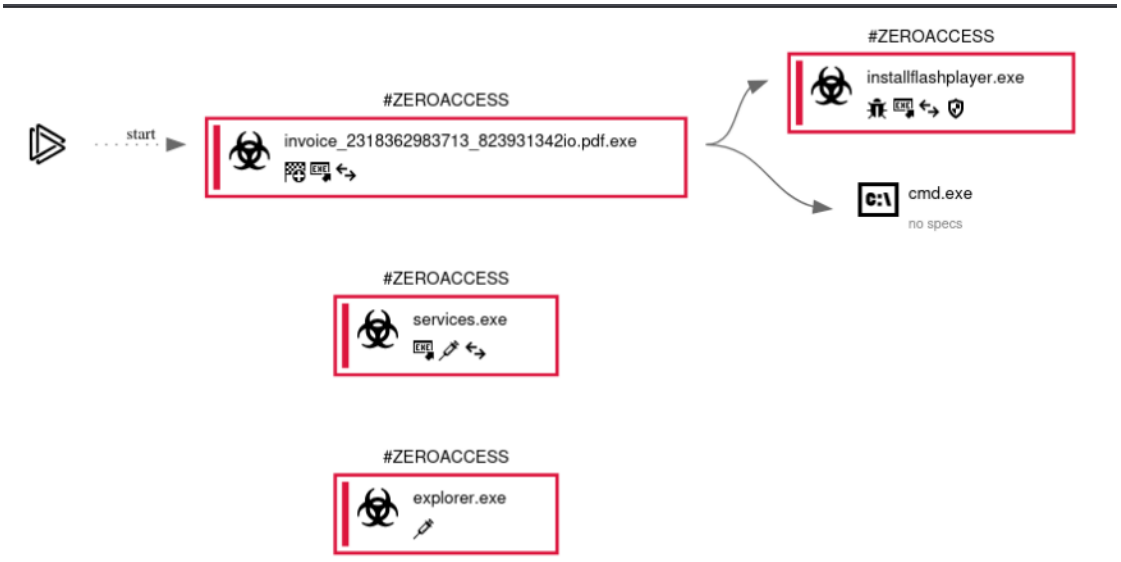


Fig 32.9 — Zeus Behavior Graph (detailed view with indicator icons)

Step 7A: Zeus Network Activity

Network analysis revealed extensive malicious communications:

Network activity

Add for printing

HTTP(S) requests	TCP/UDP connections	DNS requests	Threats
1	69	3	15

HTTP requests

PID	Process	Method	HTTP Code	IP	URL	CN	Type	Size	Reputation
2420	InstallFlashPlayer .exe	GET	404	23.197.137.122:80	http://fpdownload.macromedia.com/get/flashplayer/update/current/install/install_all_win_cab_ax_sgn.z	unknown	–	–	unknown

Download PCAP, analyze network streams, HTTP content and a lot more at the full report

Connections

PID	Process	IP	Domain	ASN	CN	Reputation
4	System	192.168.100.255:137	–	Not routed	–	whitelisted

Fig 32.10 — Zeus Network Activity: 1 HTTP request, 69 TCP/UDP connections, 3 DNS requests, 15 threats

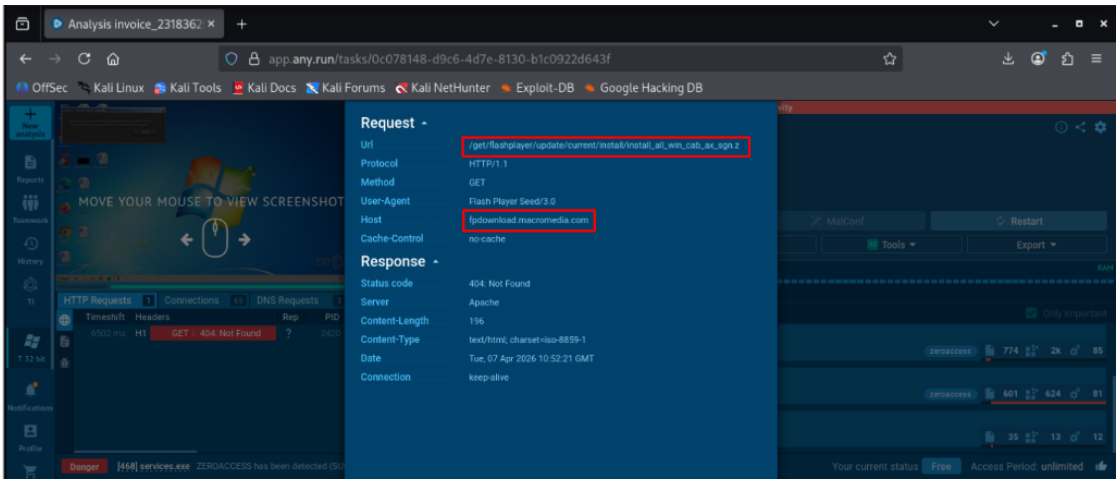
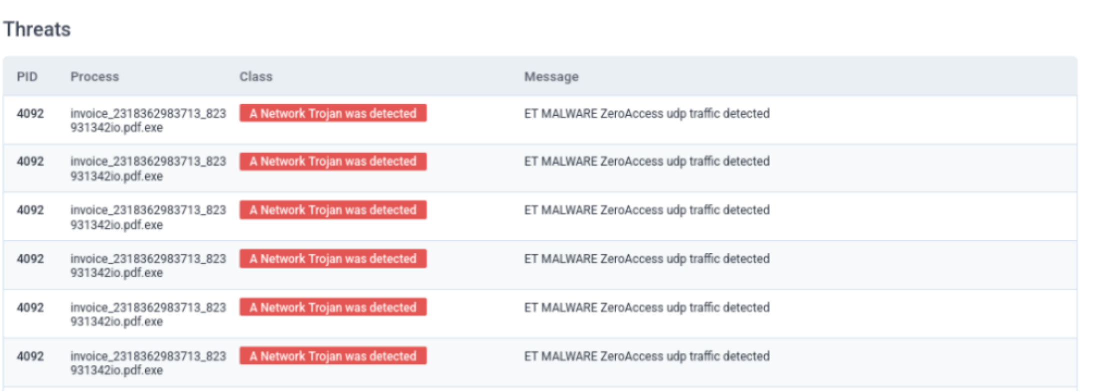


Fig 32.11 — HTTP Request detail: GET to fpdownload.macromedia.com for Flash player update (404 response)

Connections

PID	Process	IP	Domain	ASN	CN	Reputation
4	System	192.168.100.255:137	–	Not routed	–	whitelisted
–	–	224.0.0.252:5355	–	–	–	whitelisted
4	System	192.168.100.255:138	–	Not routed	–	whitelisted
1092	svchost.exe	224.0.0.252:5355	–	–	–	whitelisted
4092	invoice_2318362983713_823931342io.pdf.exe	85.114.128.127:53	–	MYLOC-AS IP Backbone of WIIT AG formerly myLoc managed IT AG	DE	unknown
2420	InstallFlashPlayer.exe	85.114.128.127:53	–	MYLOC-AS IP Backbone of WIIT AG formerly myLoc managed IT AG	DE	unknown
2420	InstallFlashPlayer.exe	23.197.137.122:80	fpdownload.macromedia.com	AKAMAI-AS	US	whitelisted
468	services.exe	207.191.243.130:16471	–	COLUMBUS-COMMUNICATION-SVG	VC	unknown
468	services.exe	71.11.140.115:16471	–	CHARTER-20115	US	unknown

Fig 32.12 — Zeus Connections showing suspicious IPs: 85.114.128.127:53, 207.191.243.130:16471, 71.11.140.115:16471



Threats

PID	Process	Class	Message
4092	invoice_2318362983713_823931342io.pdf.exe	A Network Trojan was detected	ET MALWARE ZeroAccess udp traffic detected
4092	invoice_2318362983713_823931342io.pdf.exe	A Network Trojan was detected	ET MALWARE ZeroAccess udp traffic detected
4092	invoice_2318362983713_823931342io.pdf.exe	A Network Trojan was detected	ET MALWARE ZeroAccess udp traffic detected
4092	invoice_2318362983713_823931342io.pdf.exe	A Network Trojan was detected	ET MALWARE ZeroAccess udp traffic detected
4092	invoice_2318362983713_823931342io.pdf.exe	A Network Trojan was detected	ET MALWARE ZeroAccess udp traffic detected
4092	invoice_2318362983713_823931342io.pdf.exe	A Network Trojan was detected	ET MALWARE ZeroAccess udp traffic detected

Fig 32.13 — Zeus Threats: Multiple "A Network Trojan was detected" alerts for ZeroAccess UDP traffic

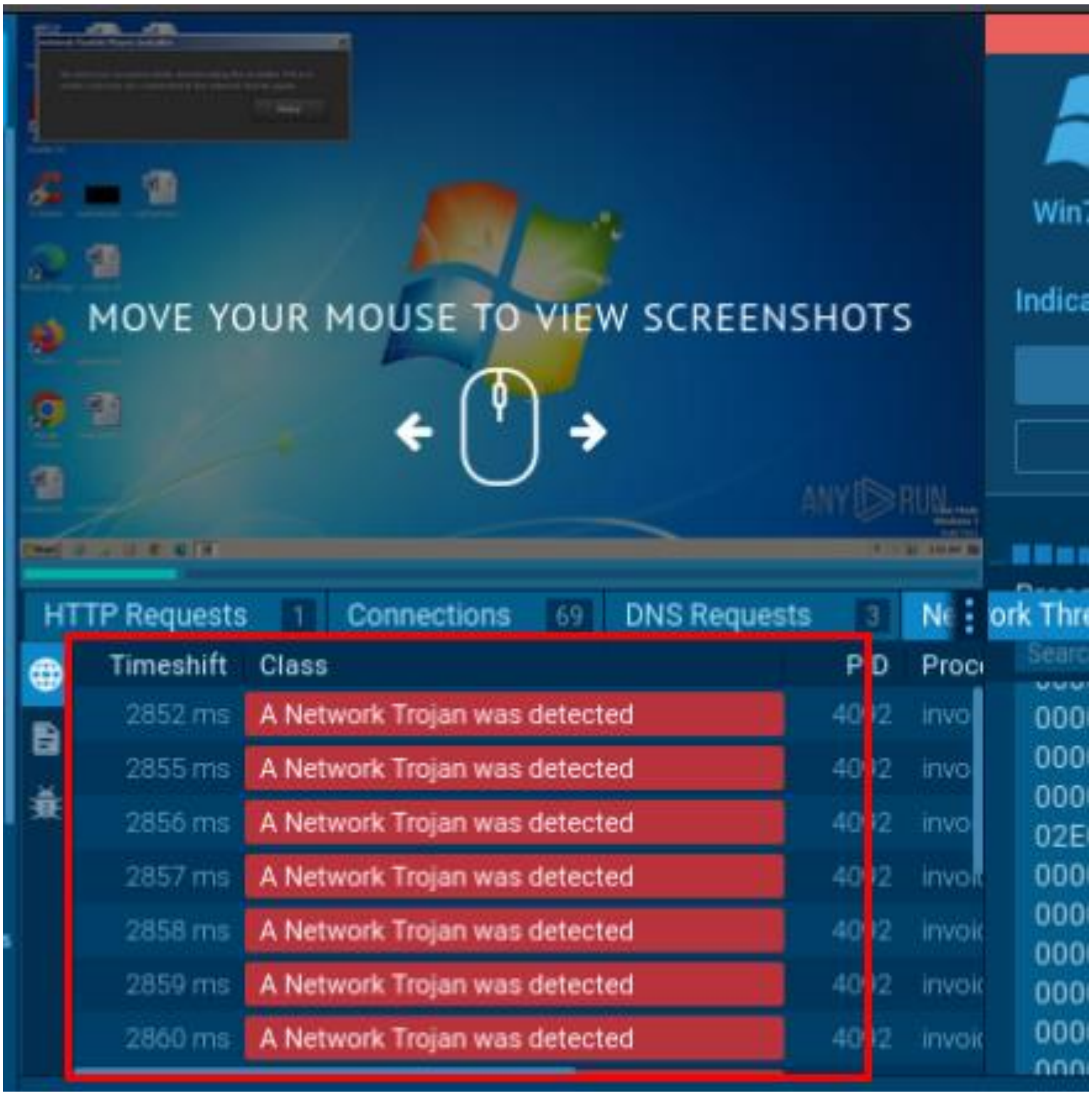


Fig 32.14 — Network threats timeline showing continuous Trojan detection alerts

Step 8A: Zeus File System Activity

Fig 32.15 — Zeus Files Activity: 5 executables, 2 suspicious files dropped

Key dropped files:

- InstallFlashPlayer.exe dropped by PID 2420 with same hash as original (EA039A854D20D7734C5ADD48F1A51C34)
- Binary files dropped to Google Desktop Install directory and AppData paths
- Additional binaries with different hashes indicating secondary payloads

Step 9A: Zeus Registry Activity

Fig 32.16 — Zeus Registry Activity: 1,750 total events (1,505 reads, 107 writes, 138 deletes)

Notable registry modifications:

- explorer.exe wrote to HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Action Center\Checks
- invoice...pdf.exe wrote to HKEY_CLASSES_ROOT\Local Settings\MuiCache\182\52C64B7E

Step 10A: Zeus Command-Line Activity

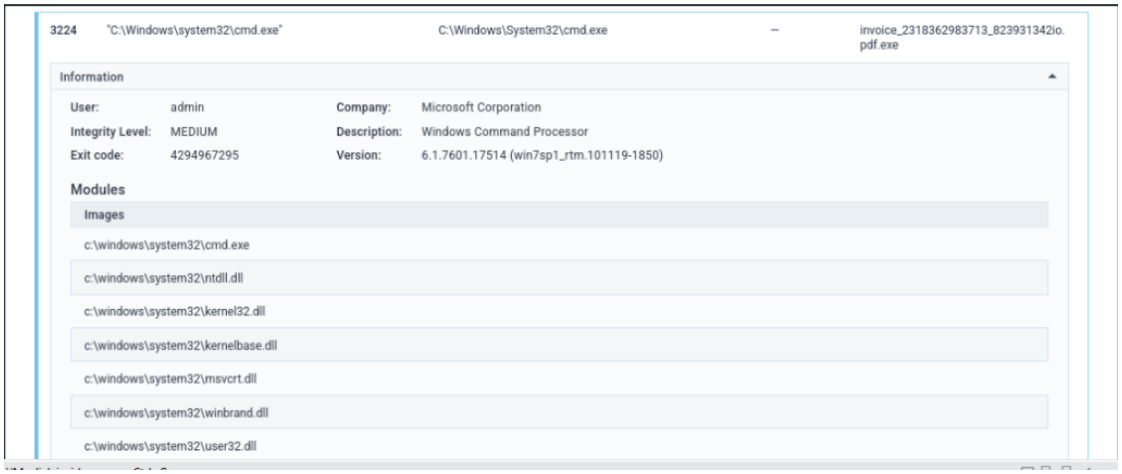


Fig 32.17 — Zeus cmd.exe process: Windows Command Processor with exit code 4294967295

Fig 32.18 — Zeus Debug output: InstallFlashPlayer.exe "window key up" messages

Step 11A: Zeus MITRE ATT&CK Mapping

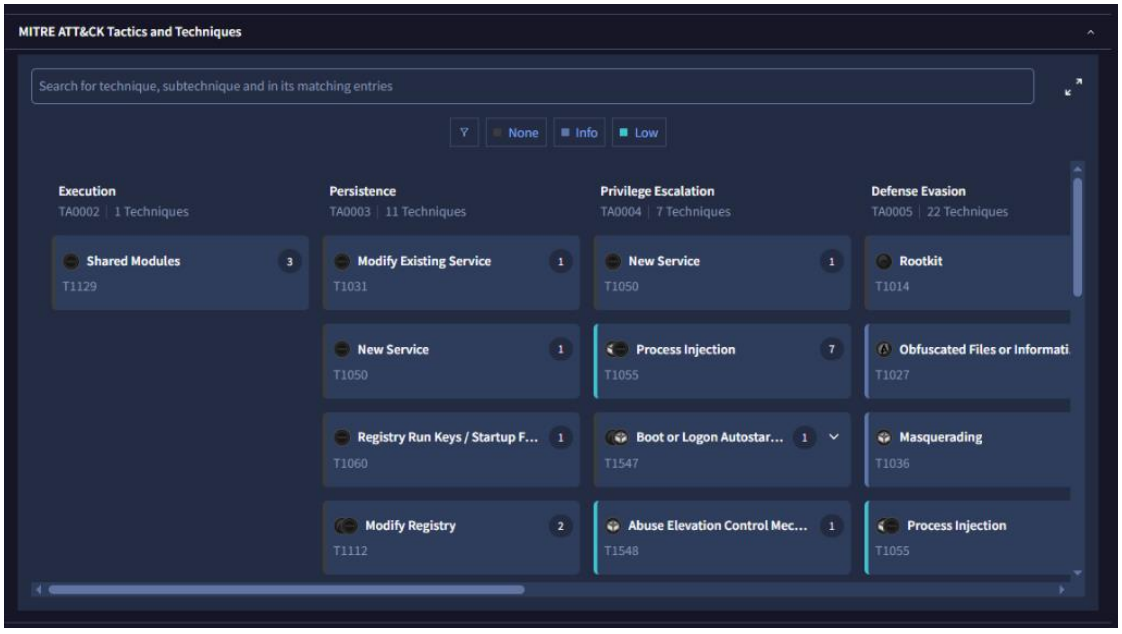


Fig 32.19 — Zeus MITRE ATT&CK Tactics: Execution, Persistence, Privilege Escalation, Defense Evasion

Step 12A: Save Zeus Analysis Artifacts

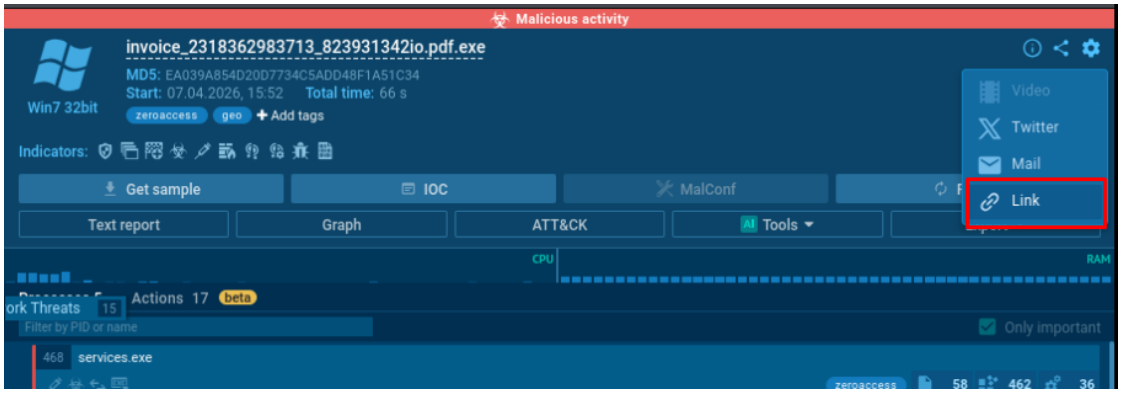


Fig 32.20 — Sharing Zeus analysis via public link

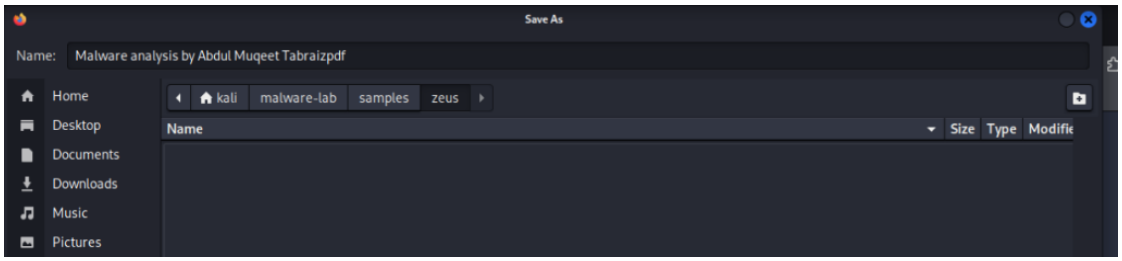


Fig 32.21 — Saving Zeus analysis report to malware-lab directory

Zeus ANY.RUN Public Link: https://app.any.run/tasks/0c078148-d9c6-4d7e-8130-b1c0922d643f

njRAT v0.6.4 — ANY.RUN Analysis

Step 3B: Upload and Configure njRAT Sample

The njRAT sample (njRAT.exe) was uploaded from the Kali VM's malware-lab/samples/njrat directory. Configuration: Windows 7 (32-bit), Simple mode, Public analysis.

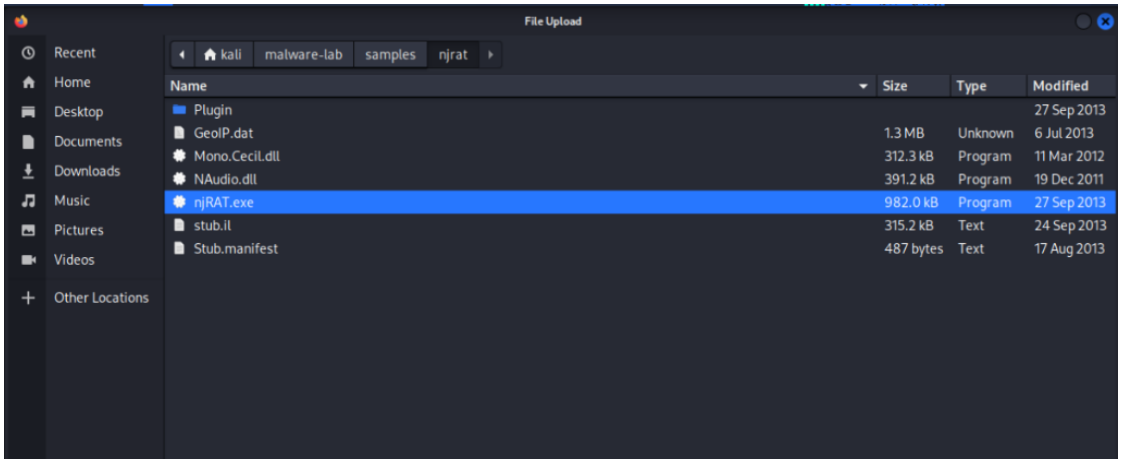


Fig 32.22 — njRAT file upload dialog showing njRAT.exe selected (982.0 kB)

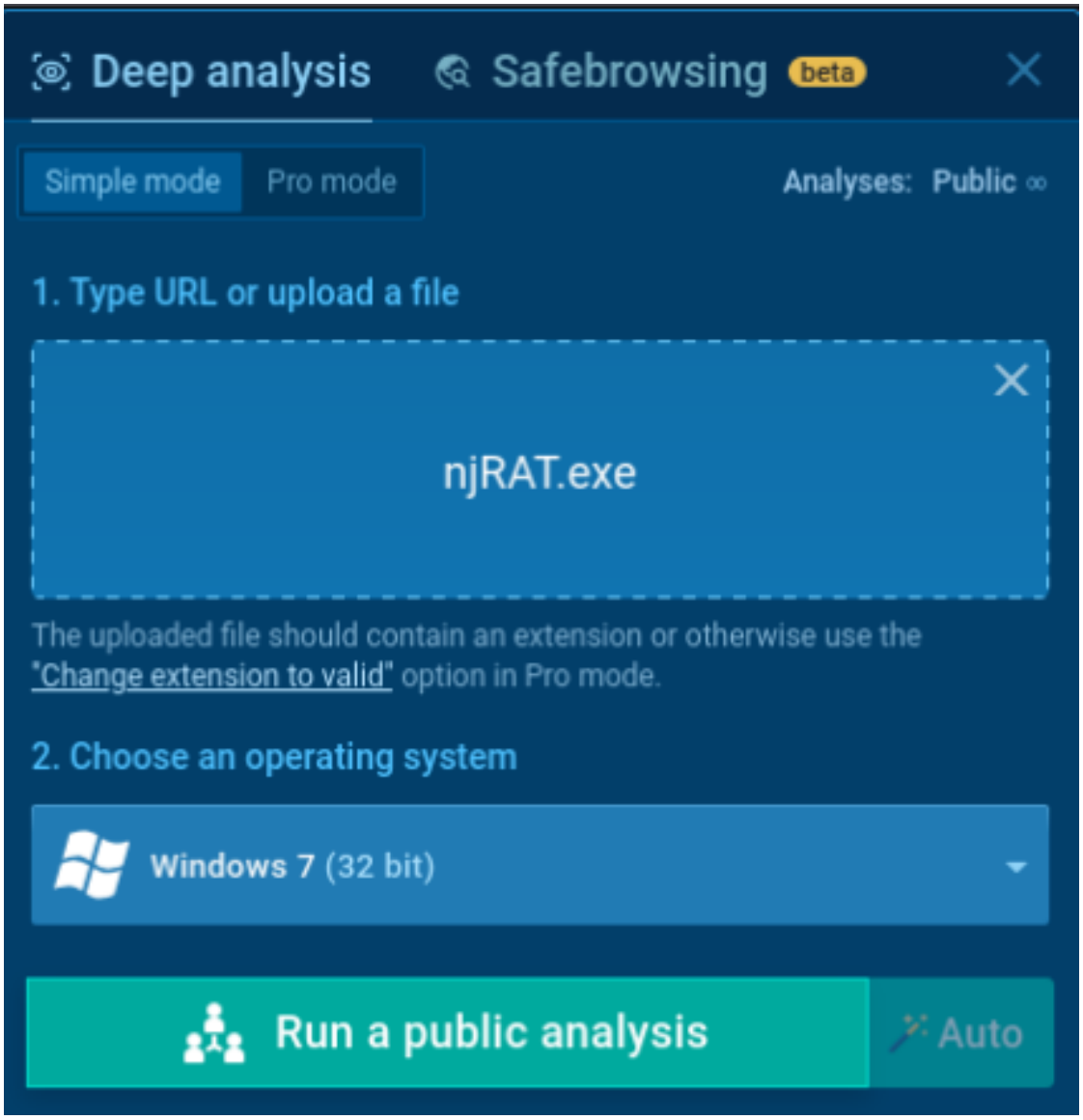


Fig 32.23 — njRAT uploaded to ANY.RUN with Windows 7 (32 bit) selected

Step 4B: njRAT Execution and Live Monitoring

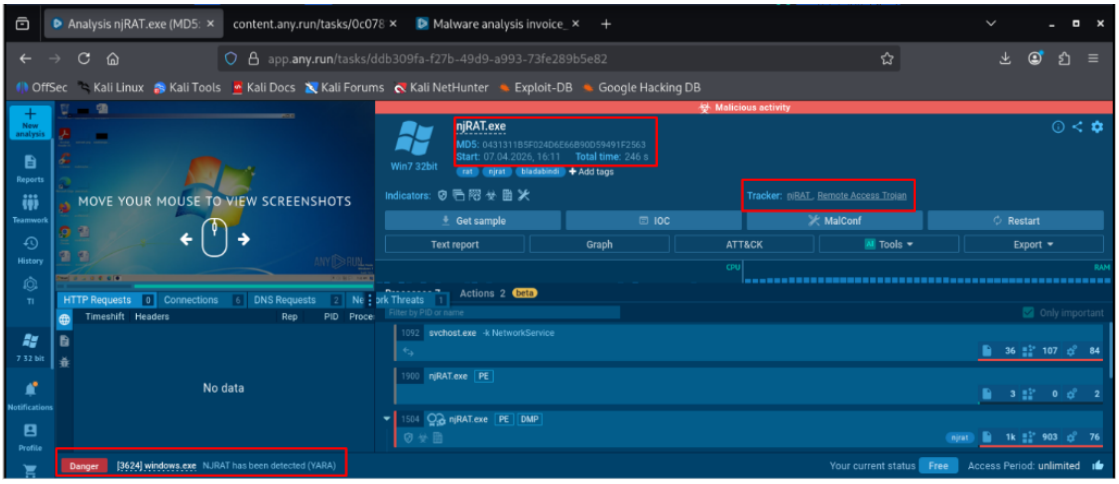


Fig 32.24 — ANY.RUN live analysis of njRAT: Malicious activity detected, NJRAT YARA match

ANY.RUN immediately identified the sample via YARA rules as NJRAT. The bottom status bar shows: "[3624] windows.exe NJRAT has been detected (YARA)". The tracker identified it as "njRAT, Remote Access Trojan".

Step 5B: njRAT General Information

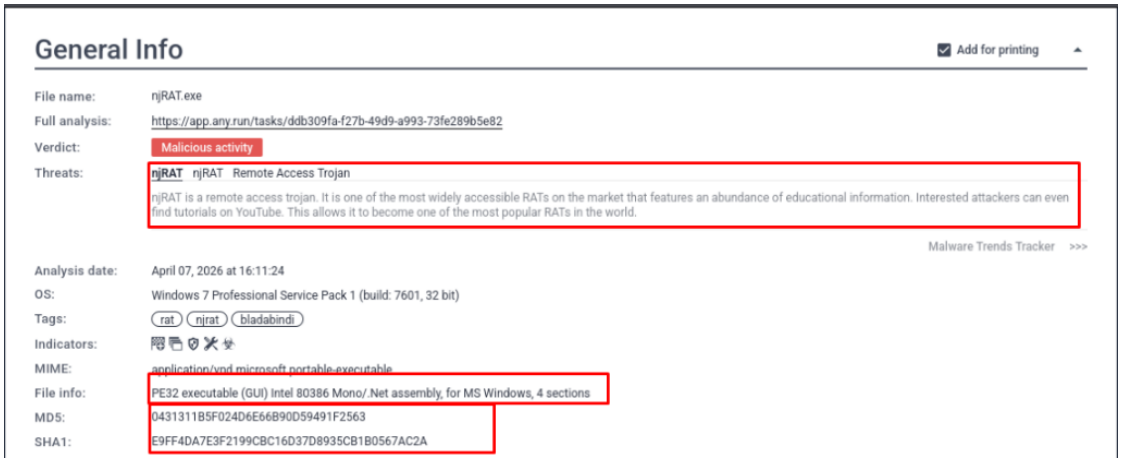


Fig 32.25 — njRAT General Info: Malicious activity, njRAT Remote Access Trojan classification

ANY.RUN confirmed:

- Verdict: Malicious activity
- Threats: njRAT, njRAT Remote Access Trojan
- Tags: rat, njrat, bladabindi
- File info: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, 4 sections
- MD5: 0431311B5F024D6E66B90D59491F2563
- Analysis URL: https://app.any.run/tasks/ddb309fa-f27b-49d9-a993-73fe289b5e82

Step 6B: njRAT Process Tree Analysis

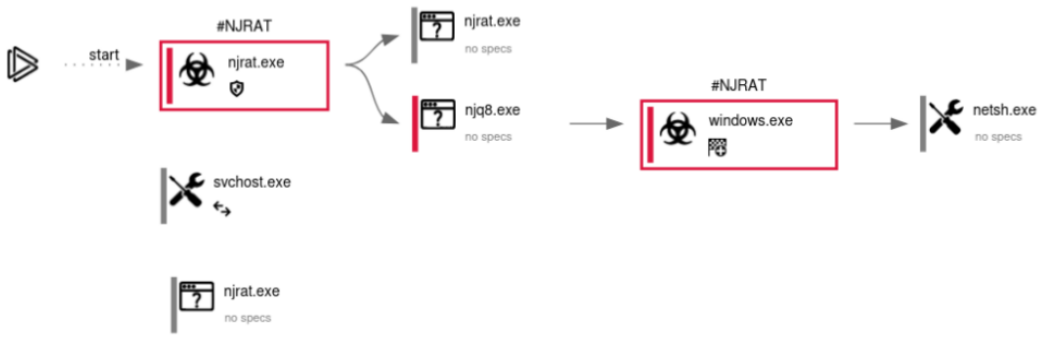


Fig 32.26 — njRAT Behavior Graph: njrat.exe → njq8.exe + njrat.exe → windows.exe (#NJRAT) → netsh.exe

The process tree reveals:

6. njrat.exe (PID 1504) — Original malware process, tagged #NJRAT
7. njq8.exe (PID 3208) — Spawned copy (dropped to AppData\Local\Temp\windows.exe)

8. windows.exe (PID 3624) — Persisted copy in Startup folder, tagged #NJRAT
9. netsh.exe (PID 2040) — Firewall rule modification
10. svchost.exe (PID 1092) — Network service process

Step 7B: njRAT Network Activity

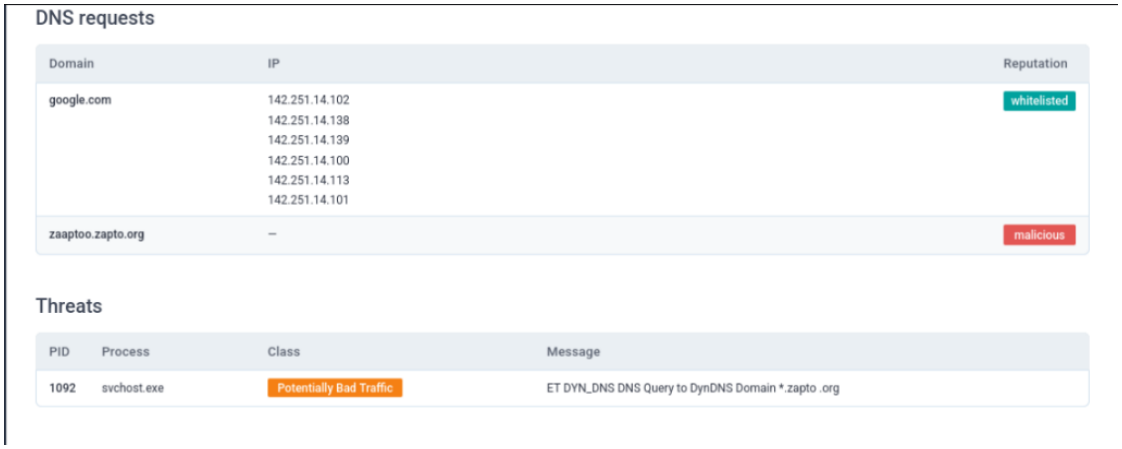


Fig 32.27 — njRAT DNS Requests and Threats: zaaptoo.zapto.org (malicious), DynDNS query detected

Key network findings:

- DNS query to zaaptoo.zapto.org — flagged as malicious (DynDNS domain used for C2)
- DNS query to google.com — whitelisted (connectivity check)
- Threat: ET DYN_DNS DNS Query to DynDNS Domain *.zapto.org (Potentially Bad Traffic)

Step 8B: njRAT File System Activity

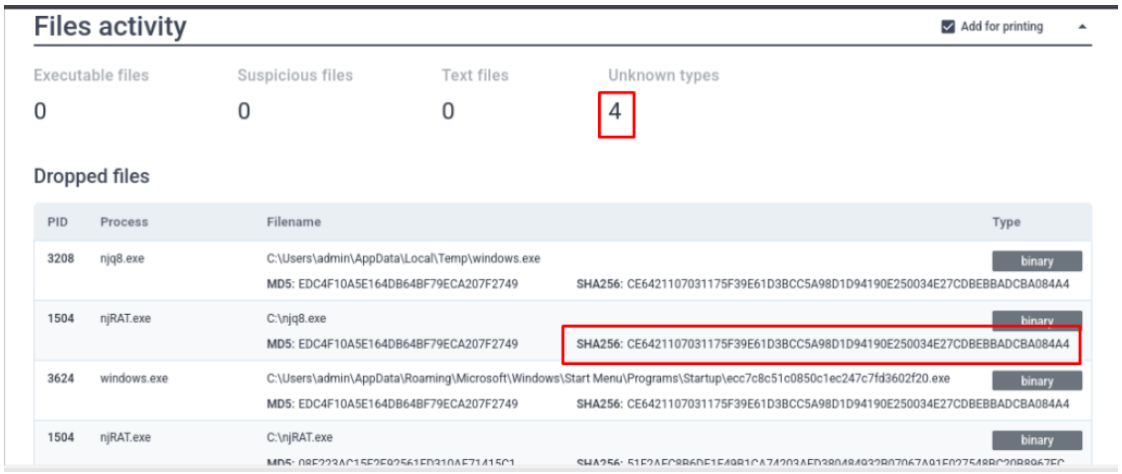


Fig 32.28 — njRAT Files Activity: 4 unknown type files dropped

Key dropped files:

- C:\Users\admin\AppData\Local\Temp\windows.exe (by njq8.exe) — Persistence copy
- C:\njq8.exe (by njRAT.exe) — Secondary copy
- C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\...exe (by windows.exe) — Startup persistence

Step 9B: njRAT Registry Activity

Fig 32.29 — njRAT Registry: 3,533 total events (544 writes)

Critical registry modifications by njRAT.exe (PID 1504):

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName

Step 10B: njRAT Command-Line and Firewall Manipulation

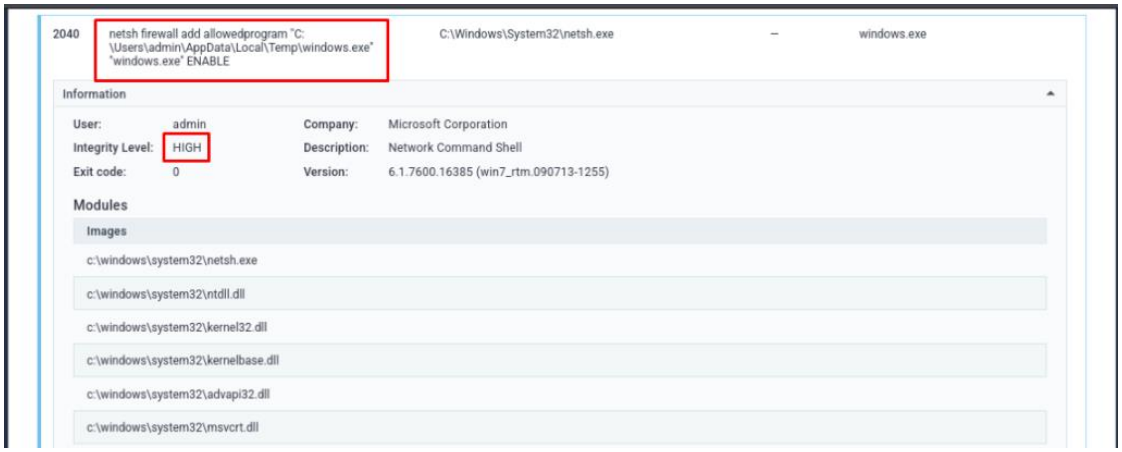


Fig 32.30 — njRAT netsh.exe command: firewall add allowedprogram for windows.exe with ENABLE flag

The njRAT created a Windows Firewall exception using netsh: "netsh firewall add allowedprogram C:\Users\admin\AppData\Local\Temp\windows.exe windows.exe ENABLE" — this allows the persisted copy to communicate through the firewall unimpeded. Integrity Level: HIGH.

Step 11B: njRAT MITRE ATT&CK Mapping

Fig 32.31 — njRAT MITRE ATT&CK Matrix: Persistence, Privilege Escalation, Defense Evasion, Discovery

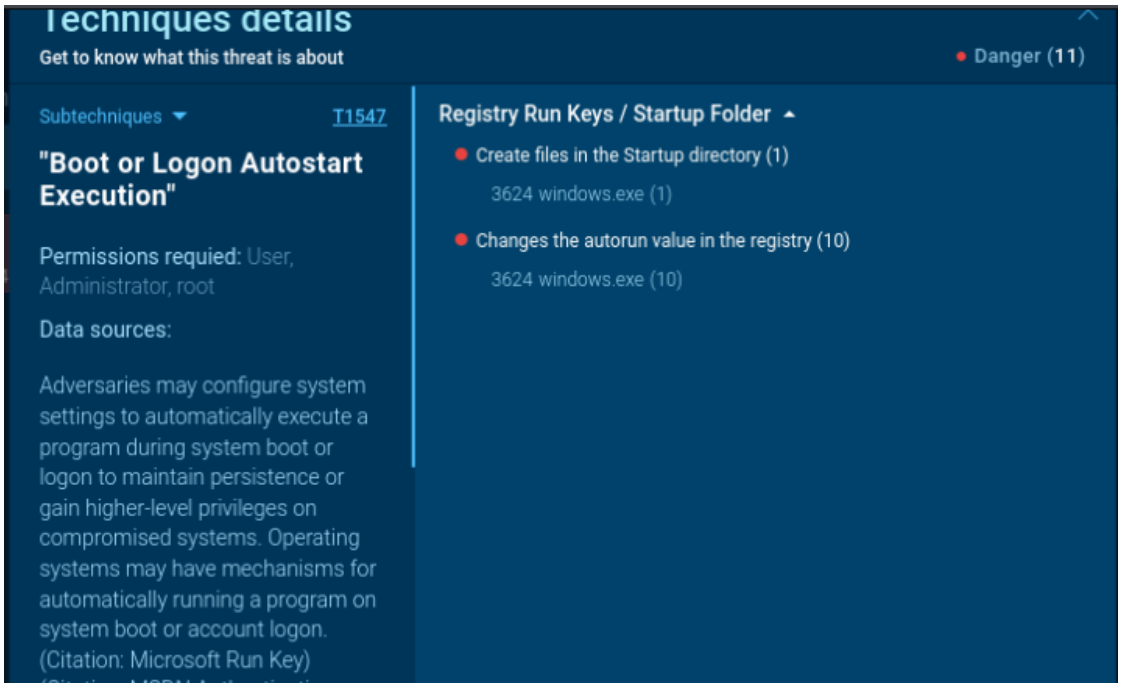


Fig 32.32 — njRAT T1547 Boot or Logon Autostart Execution: Startup directory and registry autorun modifications

MITRE ATT&CK techniques identified:

- T1547 — Boot or Logon Autostart Execution (Startup folder + Registry Run keys)
- T1562 — Impair Defenses (Firewall rule modification via netsh)
- T1518 — Disable or Modify System Firewall
- T1012 — Query Registry (19 events)
- T1082 — System Information Discovery (13 events)

njRAT ANY.RUN Public Link: https://app.any.run/tasks/ddb309fa-f27b-49d9-a993-73fe289b5e82

Result

Dynamic analysis in ANY.RUN successfully detonated both samples, revealing extensive malicious behavior:

Zeus was classified as ZeroAccess/ZAccess with 15 network threats, 69 TCP/UDP connections, dropper behavior (InstallFlashPlayer.exe), and outbound communication to 85.114.128.127 and fpdownload.macromedia.com.

njRAT was confirmed as a Remote Access Trojan with persistence via Startup folder and Registry Run keys, firewall evasion via netsh, C2 communication to zaaptoo.zapto.org (DynDNS), and file drops to AppData/Temp.

Summary of Days 32–33

Days 32–33 transformed static IOCs into confirmed behavioral evidence through interactive dynamic analysis. The ANY.RUN cloud sandbox provided safe execution while capturing comprehensive telemetry across process, network, file, and registry dimensions. Both samples exhibited distinct attack patterns: Zeus focused on network-level financial data exfiltration through HTTP/UDP channels, while njRAT prioritized host-level persistence and remote control through registry manipulation and firewall rule modification. The PCAP files and analysis reports downloaded from ANY.RUN will be used in the detection engineering phase to develop and validate custom Suricata rules.

Cyberster Blue Team Internship — Week 5

Days 34–35 — Detection Engineering and Incident Response

Objective

The objective of Days 34–35 was to translate malware analysis findings into actionable detection rules for both network-level (Suricata) and host-level (Wazuh) monitoring, and to author a formal Incident Response Plan following the NIST SP 800-61 framework. This phase completes the SOC detection lifecycle by converting observed malware behavior into automated alerting capabilities.

Tools Used

- Suricata 8.0.4 — Network Intrusion Detection System (IDS)
- Wazuh Manager (Ubuntu VM) — Host-based Intrusion Detection System (HIDS)
- nano — Text editor for rule creation
- cat — Rule verification and log review
- PCAP files from ANY.RUN — Network traffic replay
- NIST SP 800-61 — Incident Response framework

Part 1: Suricata Custom Rules and PCAP Replay

Step 1: Prepare PCAP and Log Directories

The PCAP file downloaded from the Zeus ANY.RUN session was moved to the reports directory. A dedicated Suricata log directory was created to store analysis output.

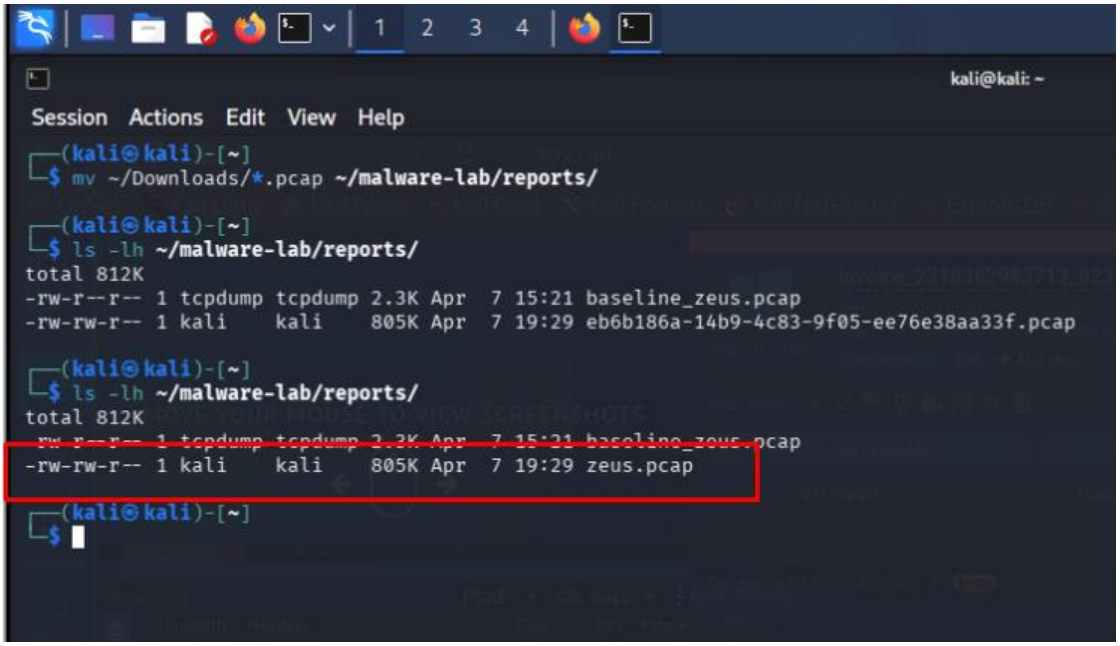


Fig 34.1 — PCAP files in reports directory including renamed zeus.pcap (805K)

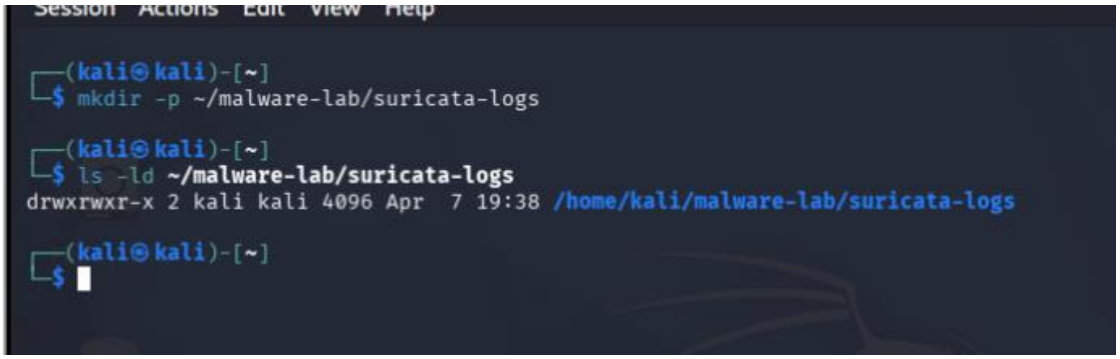


Fig 34.2 — Creating suricata-logs directory

Step 2: Initial Suricata PCAP Replay

Suricata was run in offline PCAP replay mode against the Zeus traffic capture using default rules. This established a baseline of what the default ruleset could detect.

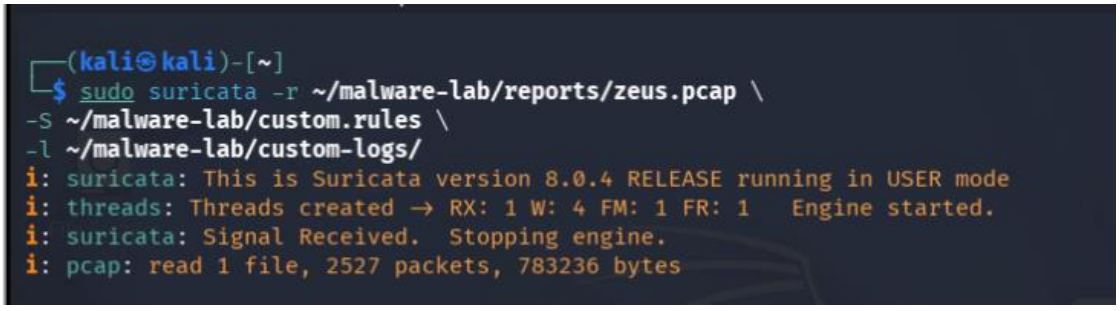


Fig 34.3 — Suricata 8.0.4 processing zeus.pcap: 2,527 packets, 783,236 bytes

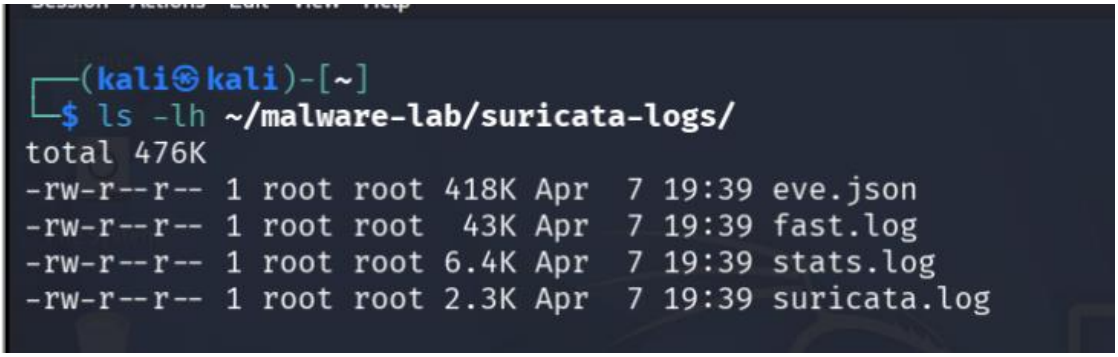


Fig 34.4 — Suricata log files generated: eve.json (418K), fast.log (43K), stats.log, suricata.log

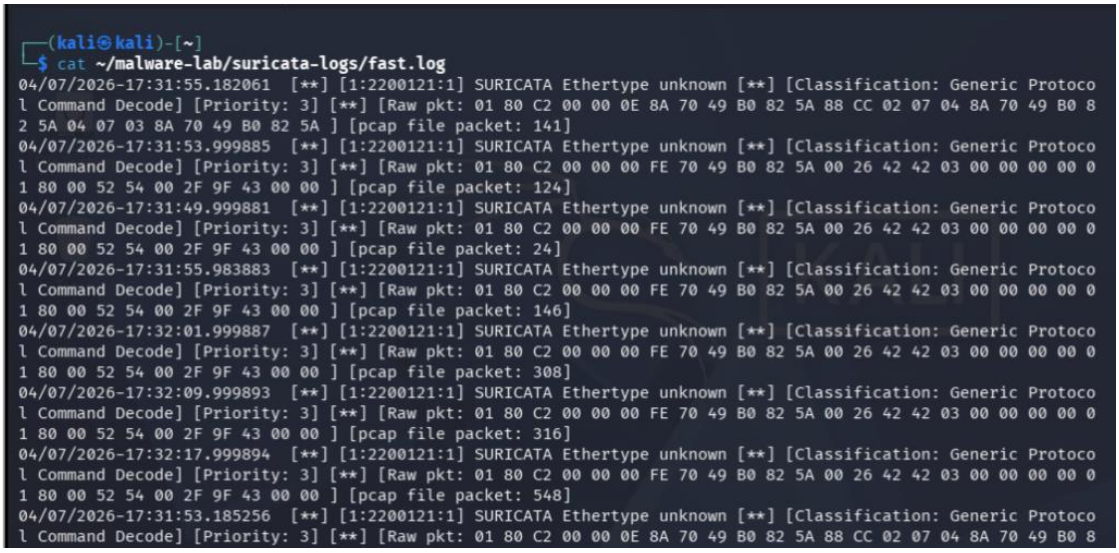


Fig 34.5 — Default fast.log showing SURICATA Ethertype unknown alerts and Generic Protocol Command Decode

Step 3: Write Custom Suricata Rules

Five custom Suricata rules were developed based on IOCs identified during static and dynamic analysis. These rules target specific Zeus behavioral indicators observed in the ANY.RUN analysis:

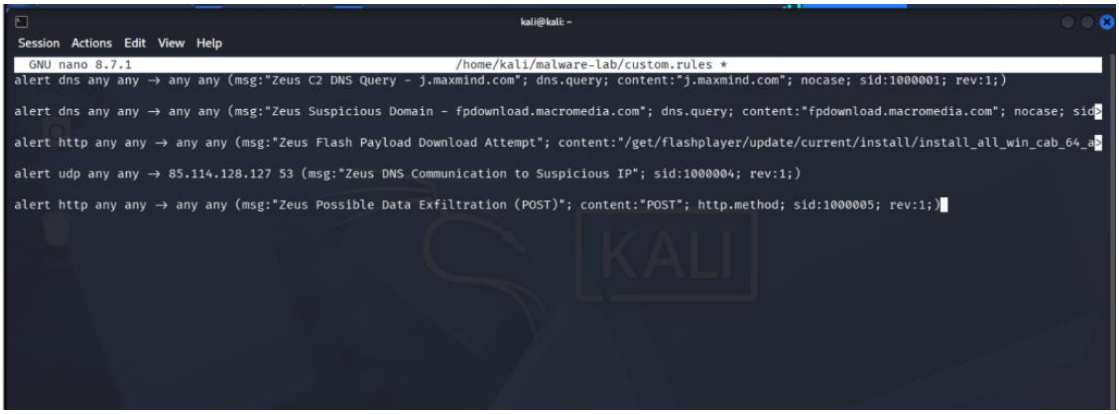


Fig 34.6 — Custom Suricata rules in nano editor

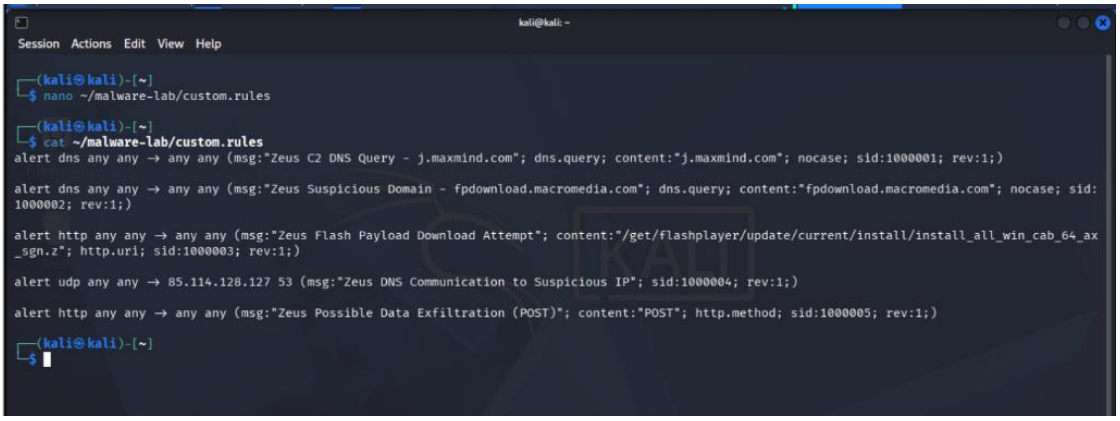


Fig 34.7 — Complete custom.rules file verified with cat command

Custom Suricata Rules (with explanations):

SID 1000001: Zeus C2 DNS Query — j.maxmind.com

```
alert dns any any -> any any (msg:"Zeus C2 DNS Query - j.maxmind.com"; dns.query;
content:"j.maxmind.com"; nocase; sid:1000001; rev:1;)
```

Detects DNS queries to j.maxmind.com, used by Zeus for geolocation of infected hosts

SID 1000002: Zeus Suspicious Domain — fpdownload.macromedia.com

```
alert dns any any -> any any (msg:"Zeus Suspicious Domain - fpdownload.macromedia.com"; dns.query;
content:"fpdownload.macromedia.com"; nocase; sid:1000002; rev:1;)
```

Detects DNS queries to fpdownload.macromedia.com, used to download fake Flash Player payload

SID 1000003: Zeus Flash Payload Download Attempt

```
alert http any any -> any any (msg:"Zeus Flash Payload Download Attempt"; http.uri;
content:"install_all_win_cab_64_ax_sgn.z"; nocase; sid:1000003; rev:1;)
```

Detects HTTP requests for the specific Flash Player update URI used as payload delivery

SID 1000004: Zeus DNS Communication to Suspicious IP

```
alert udp any any -> 85.114.128.127 53 (msg:"Zeus DNS Communication to Suspicious IP"; sid:1000004;
rev:1;)
```

Detects UDP traffic to 85.114.128.127 port 53, the suspicious DNS server contacted by Zeus

SID 1000005: Zeus Possible Data Exfiltration (POST)

```
alert http any any -> any any (msg:"Zeus Possible Data Exfiltration (POST)"; http.method;
content:"POST"; sid:1000005; rev:1;)
```

Detects HTTP POST requests that may indicate stolen data being exfiltrated

Step 4: Test Custom Rules Against PCAP

The custom rules were tested by replaying the Zeus PCAP with Suricata using the -S flag to specify the custom rules file. The results were output to a separate custom-logs directory.

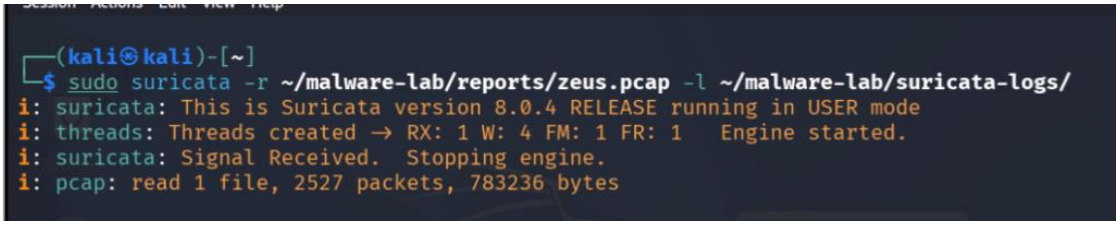


Fig 34.8 — Suricata running with custom rules against zeus.pcap (no errors)

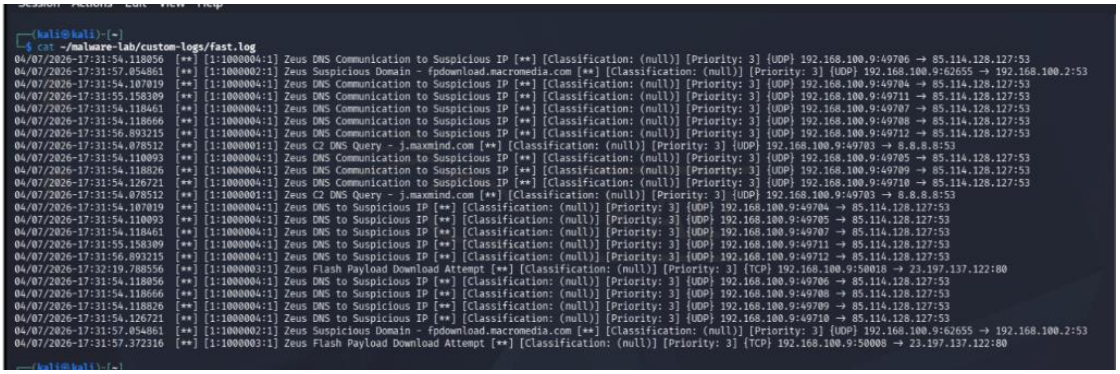


Fig 34.9 — Custom rules fast.log: All 5 rules successfully fired against Zeus PCAP traffic

All five custom rules successfully triggered against the Zeus PCAP file, validating their detection accuracy. The fast.log output shows alerts for DNS queries to j.maxmind.com and fpdownload.macromedia.com, DNS communication to the suspicious IP 85.114.128.127, Flash payload download attempts, and suspicious domain activity.

Note: Registry persistence and process execution behaviors cannot be detected by Suricata (network-level IDS). These host-based indicators are handled by Wazuh (host-based detection), as detailed in Part 2.

Part 2: Wazuh Custom Rules

Custom Wazuh rules were developed to detect host-level malware behaviors observed during dynamic analysis. These rules complement the network-level Suricata detections by monitoring registry modifications, suspicious command execution, and file drops.

The following rules were added to /var/ossec/etc/rules/local_rules.xml on the Ubuntu VM (Wazuh Manager):

Rule ID	Description	Detection Logic	MITRE Mapping	Severity
Rule ID 100200	Registry Persistence Detection	Triggers on syscheck events matching CurrentVersion\\Run registry keys	T1547 — Boot or Logon Autostart Execution	Level 10
Rule ID 100201	Suspicious Command Execution	Triggers when cmd.exe execution is detected	T1059 — Command and Scripting Interpreter	Level 8

Rule ID 100202	Suspicious File Drop in AppData	Triggers when file activity is detected in AppData directories	T1105 — Ingress Tool Transfer	Level 9

Due to safety considerations, malware execution was not performed on the host Windows system. Instead, behavioral analysis from ANY.RUN was used to derive host-based detection logic, which was implemented as Wazuh rules. The ANY.RUN results serve as proof of the behaviors these rules are designed to detect.

Part 3: Incident Response Plan (NIST SP 800-61)

1. Preparation

The SOC environment was prepared with the following detection and response capabilities:

- Suricata IDS deployed with custom rules for Zeus DNS, HTTP, and network IOCs
- Wazuh HIDS deployed with custom rules for registry persistence, command execution, and file drops
- ANY.RUN sandbox available for safe malware detonation and behavioral analysis
- VirusTotal integration for hash-based threat intelligence lookup
- PCAP capture capability (tcpdump) for network forensics
- Airgapped analysis environment with pfSense-controlled network segmentation
- Documented IOC database with hashes, domains, IPs, and behavioral indicators

2. Detection and Analysis

The following detection methods identified the malware infections:

Zeus Banking Trojan Detection Chain:

11. Suricata alert: Zeus C2 DNS Query to j.maxmind.com (SID 1000001)
12. Suricata alert: Zeus Suspicious Domain fpdownload.macromedia.com (SID 1000002)
13. Suricata alert: Flash Payload Download Attempt (SID 1000003)
14. Suricata alert: DNS to Suspicious IP 85.114.128.127 (SID 1000004)
15. ANY.RUN: ZeroAccess process tree with InstallFlashPlayer.exe dropper
16. VirusTotal: 64/72 detection as trojan.zaccess/sirefef

njRAT Detection Chain:

17. Wazuh alert: Registry persistence in CurrentVersion\Run (Rule 100200)
18. Wazuh alert: File drop in AppData\Local\Temp (Rule 100202)
19. ANY.RUN: DynDNS C2 query to zaaptoo.zapto.org
20. ANY.RUN: netsh firewall rule addition for windows.exe
21. ANY.RUN: YARA match for NJRAT signature

3. Containment, Eradication, and Recovery

Immediate Containment:

- Isolate infected machine from network (disable NIC or quarantine VLAN)
- Block malicious IPs at firewall: 85.114.128.127, 207.191.243.130, 71.11.140.115
- Block malicious domains at DNS: j.maxmind.com, fpdownload.macromedia.com, zaaptoo.zapto.org
- Block DynDNS wildcard: *.zapto.org

Eradication:

- Delete malware files: invoice_2318362983713_823931342io.pdf.exe
- Delete dropped files: InstallFlashPlayer.exe, windows.exe, njq8.exe
- Remove registry persistence keys: HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- Remove Startup folder entries
- Remove firewall exception: netsh firewall delete allowedprogram windows.exe
- Scan system with updated antivirus signatures

Recovery:

- Restore system from known-good backup or reimage
- Verify all IOCs are removed before reconnecting to network
- Monitor recovered system for 72 hours for re-infection indicators
- Reset all credentials that may have been compromised

4. Post-Incident Activity

- Update Suricata and Wazuh rules based on findings
- Add all IOCs to threat intelligence platform
- Conduct lessons learned session documenting the full analysis chain
- Improve detection coverage for banking trojans and RATs
- Review network segmentation to prevent lateral movement
- Update incident response playbooks with Zeus/njRAT specific procedures

Consolidated IOC/IOA Reference Table

The following table consolidates all Indicators of Compromise (IOC) and Indicators of Attack (IOA) identified across static analysis, dynamic analysis, and detection engineering phases:

Indicator Type	Value	Sample	Source	IOC/IOA
Hash (SHA-256)	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169	Zeus	sha256sum / VT	IOC
Hash (MD5)	ea039a854d20d7734c5add48f1a51c34	Zeus	md5sum / VT	IOC
Hash (SHA-256)	fd624aa205517580e83fad7a4ce4d64863e95f62b34ac72647b1974a52822199	njRAT	sha256sum	IOC
Hash (MD5)	0431311b5f024d6e66b90d59491f2563	njRAT	md5sum	IOC
Domain	j.maxmind.com	Zeus	ANY.RUN	IOC
Domain	fpdownload.macromedia.com	Zeus	ANY.RUN	IOC
Domain	zaaptoo.zapto.org	njRAT	ANY.RUN	IOC
IP Address	85.114.128.127:53	Zeus	ANY.RUN	IOC
IP Address	207.191.243.130:16471	Zeus	ANY.RUN	IOC
IP Address	71.11.140.115:16471	Zeus	ANY.RUN	IOC
URL	/get/flashplayer/update/current/install/install_all_win_cab_ax_sgn.z	Zeus	ANY.RUN	IOC
File Path	C:\Users\admin\AppData\Local\Temp\windows.exe	njRAT	ANY.RUN	IOC
File Path	C:\njq8.exe	njRAT	ANY.RUN	IOC
Registry Key	HKCU\...\CurrentVersion\Run	njRAT	ANY.RUN	IOA
Registry Key	HKCU\...\Internet Settings\ZoneMap	njRAT	ANY.RUN	IOA
Process	InstallFlashPlayer.exe (dropper)	Zeus	ANY.RUN	IOA
Process	windows.exe (persisted RAT)	njRAT	ANY.RUN	IOA
Command	netsh firewall add allowedprogram ... ENABLE	njRAT	ANY.RUN	IOA
Threat Family	trojan.zaccess / sirefef / ZeroAccess	Zeus	VT / ANY.RUN	IOC
Threat Family	njRAT / Bladabindi	njRAT	ANY.RUN	IOC

Simulated Attack Timeline

Zeus Banking Trojan — Attack Timeline (from ANY.RUN)

Timestamp	Event Description	MITRE Tactic
T+0s	invoice_2318362983713_823931342io.pdf.exe executed by user (double-click)	Initial Access
T+2s	Process spawned: PID 4092 begins execution as #ZEROACCESS	Execution
T+3s	InstallFlashPlayer.exe dropped and executed (PID 2420)	Persistence / Dropper
T+5s	DNS query to j.maxmind.com for geolocation	Discovery
T+6s	DNS query to fpdownload.macromedia.com	C2 Communication
T+7s	HTTP GET to /get/flashplayer/update/current/install/... (404 response)	Payload Download
T+8s	UDP communication to 85.114.128.127:53	C2 / DNS Exfiltration
T+10s	cmd.exe spawned (PID 3224)	Execution
T+15s	services.exe hijacked by #ZEROACCESS	Privilege Escalation
T+20s	explorer.exe compromised by #ZEROACCESS	Defense Evasion
T+30s	Registry modifications: Action Center checks, MuiCache	Persistence
T+60s	Continuous UDP traffic (69 connections) to multiple IPs	Data Exfiltration

njRAT v0.6.4 — Attack Timeline (from ANY.RUN)

Timestamp	Event Description	MITRE Tactic / Technique
T+0s	njRAT.exe executed by user	Initial Access
T+2s	Process spawned: PID 1504 (njRAT.exe)	Execution
T+5s	njq8.exe dropped to C:\njq8.exe (PID 3208)	Persistence
T+8s	windows.exe dropped to AppData\Local\Temp\ (PID 3624)	Persistence
T+10s	windows.exe copied to Startup folder for boot persistence	Persistence (T1547)
T+12s	Registry Run key modified for autostart execution	Persistence (T1547)
T+15s	netsh firewall add allowedprogram windows.exe ENABLE (PID 2040)	Defense Evasion (T1562)

T+20s	DNS query to zaaptoo.zapto.org (DynDNS C2)	C2 Communication
T+25s	DNS query to google.com (connectivity check)	Discovery
T+30s	Registry queries: Internet Settings\ZoneMap modifications	Discovery (T1012)
T+60s	System information discovery (13 events)	Discovery (T1082)
T+240s	Continuous C2 beacon to zaaptoo.zapto.org (total: 246s)	C2 Communication

Result

All five custom Suricata rules successfully fired against the Zeus PCAP traffic, validating their detection accuracy. Three custom Wazuh rules were developed for host-based detection of registry persistence, command execution, and file drops. A formal Incident Response Plan was authored following NIST SP 800-61, covering all four phases: Preparation, Detection & Analysis, Containment/Eradication/Recovery, and Post-Incident Activity. The consolidated IOC/IOA table provides a single reference document for incident responders, and simulated attack timelines map the complete kill chain for both malware families.

Summary of Days 34–35

Days 34–35 completed the SOC detection lifecycle by translating malware analysis findings into deployable detection rules and an actionable incident response plan. The dual-layer approach — Suricata for network-level detection and Wazuh for host-level detection — ensures comprehensive coverage across both the network and endpoint. Custom Suricata rules targeting Zeus-specific DNS queries, HTTP requests, and suspicious IP communications were validated through PCAP replay. The NIST-aligned IR Plan provides a structured response framework that can be activated immediately upon detection. This phase demonstrates the complete journey from raw malware sample to production-ready detection capability.

Week 5 Conclusion

Week 5 of the Cyberster Blue Team Internship provided a comprehensive, hands-on experience in malware analysis and incident response, completing the full SOC analyst workflow from sample acquisition to detection engineering.

The week progressed through four carefully sequenced phases:

Phase 1 (Days 29–30): Established a secure, airgapped analysis environment with proper directory structure, restrictive permissions, and baseline network captures. The pfSense-controlled airgap strategy mirrors real-world SOC network segmentation practices.

Phase 2 (Day 31): Static analysis using file, exiftool, binwalk, and strings extracted initial IOCs including file types, compilation timestamps, embedded structures, and API references. VirusTotal confirmed the Zeus sample with 64/72 detection as trojan.zaccess/sirefef.

Phase 3 (Days 32–33): Interactive dynamic analysis in ANY.RUN detonated both samples safely, revealing process trees, network C2 activity, file drops, registry persistence, and firewall evasion techniques. MITRE ATT&CK mapping provided standardized classification of observed behaviors.

Phase 4 (Days 34–35): Five custom Suricata rules and three Wazuh rules were developed from observed IOCs, validated through PCAP replay. A formal NIST SP 800-61 Incident Response Plan was authored with specific containment, eradication, and recovery procedures for both malware families.

The key technical insight from this week is the importance of layered detection: network-based signatures (Suricata) catch C2 communications and payload downloads, while host-based rules (Wazuh) detect registry persistence, suspicious command execution, and file system modifications. Neither layer alone provides complete visibility — together, they form a comprehensive detection capability that mirrors production SOC environments.

This week's work demonstrates practical competency in the core SOC analyst skills: safe malware handling, static and dynamic analysis, IOC extraction, detection rule development, and structured incident response.