

Inside The Gentlemen: Undisclosed TukTuk C2 Framework and EDR Neutralization Research

 oasis-security.io/blog/The-Gentlemen-Ransomware-Hacker-Groups-TukTuk-Framework



August 31, 202626 min readCyber Threat Intelligence
By Threat Intelligence Unit

Overview

Analysis of server-side data associated with **The Gentlemen** identified the `eb.sys` file, along with materials related to TukTuk C2, DLL side-loading configurations, EDR neutralization, and vulnerable driver research.

Notably, the **complete project structure of the TukTuk framework was identified, for which no prior public disclosure or documentation could be confirmed.**

The same server also contained **224 Jira tickets and eight attachments assessed to have been exfiltrated from a global technology company's Jira environment, as well as credential and infrastructure-related information associated with a global healthcare company's Infrastructure-as-Code (IaC) Platform.**

Executive Summary

- The eb.sys sample matched **GentleKiller** by hash
- The complete **TukTuk C2** project was identified, including Windows Agent, Linux Agent, Backend, and Panel — indicating a cross-platform design
- **Greenshot** DLL sideloading configurations were identified, along with ProcMon, Slack, and Postman research
- Step-by-step materials covered **EDR neutralization, BYOVD, vulnerable driver hunting, and kernel-level research**
- **224 Jira tickets and eight attachments** were exfiltrated from a global technology company, including data related to its **U.S. defense and defense-industry customers**
- Credentials and infrastructure data from a global healthcare company's **laC Platform** were identified, including **AWS, production databases, Azure AD, Bitbucket, and infrastructure configuration management**

1. The Analyzed Server

Server Overview and Collection

The subject of this analysis is a single server detected by our threat intelligence service, AGATHA. All findings presented in this report are based on data collected from this server.

A set of files was collected from the server, and the subsequent analysis is based on this data.

Collected File Inventory

The collected data can be broadly divided into three categories.

Category

Components

Attack Tools & Frameworks

Complete TukTuk C2 project (tuktuk-v2.0_10.zip: agent / agent-linux / backend / panel),
TukTuk.exe

Greenshot DLL sideloading set (Greenshot.exe + malicious log4net.dll)

EDR killer family (eb.sys, EDRKiller, WarsawKiller, UnknownKiller, wsftprm.sys)

Research & Training Materials

Step-by-step EDR neutralization and vulnerable driver training materials (Lesson 1–4)

DLL sideloading research (sideload-research: ProcMon, Slack, Postman)

Exfiltrated Data

Global technology company Jira tickets (224) + attachments (8), including data related to its U.S. defense and defense-industry customers

Global healthcare company IaC platform credentials (AWS, Azure AD, Bitbucket, production DB)

This inventory serves as a map for the subsequent analysis, with each item examined in detail in the following sections. It shows that the server held (1) attack tools, (2) related research and training materials, and (3) exfiltrated data together in one place.

2. Attribution to The Gentlemen

The attribution of this server to The Gentlemen is based on multiple indicators found on the server that match existing public threat reports on The Gentlemen. The technical details of these indicators are covered in the TukTuk C2 Framework and EDR Neutralization sections; this section consolidates the basis for the attribution.

Indicator 1 — eb.sys hash match

The eb.sys file collected from the server matches, by its hash, the GentleKiller sample documented in public reports (SHA256:

97BD65E98CDC4E93D49EDD4EA905D43A61244DF0FD3323E6649330DE3B1BE091).

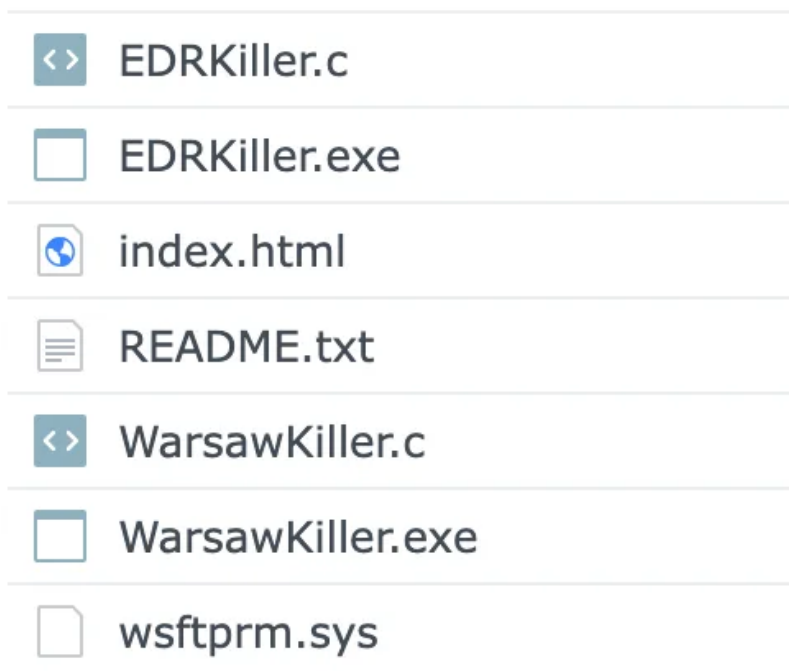


Figure 1. EDRKiller and WarsawKiller source files, executables, and the wsftprm.sys driver identified on the server

Indicator 2 — Greenshot DLL sideloading structure

The composition of the `greenshot_sideload` folder (legitimate `Greenshot.exe` + malicious `log4net.dll`) is consistent with the DLL sideloading technique known to have been used in The Gentlemen attack.

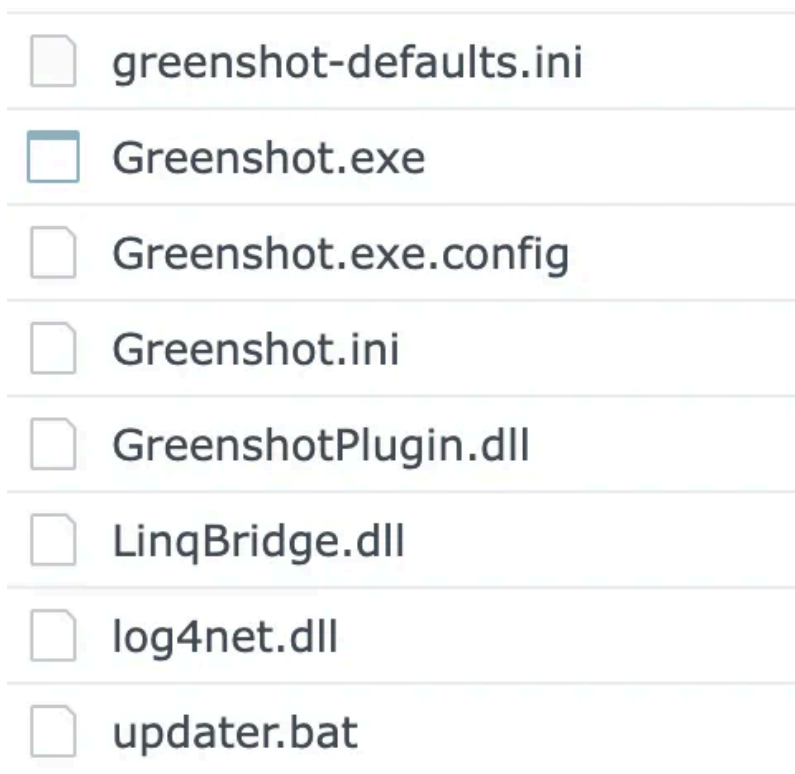


Figure 2. `greenshot_sideload` folder containing `Greenshot.exe` and malicious `log4net.dll`

Static analysis of `log4net.dll` further identified TukTuk C2-related code and C2 configuration values, linking the sideloading set directly to the TukTuk framework.

Indicator 3 — TukTuk dashboard identifiers

The hostname `DESKTOP-22EVPBQ` and the operator identifier `Neo`, observed in the TukTuk C2 dashboard in `screenshot2.png` on the server, match information referenced in The Gentlemen's public report.

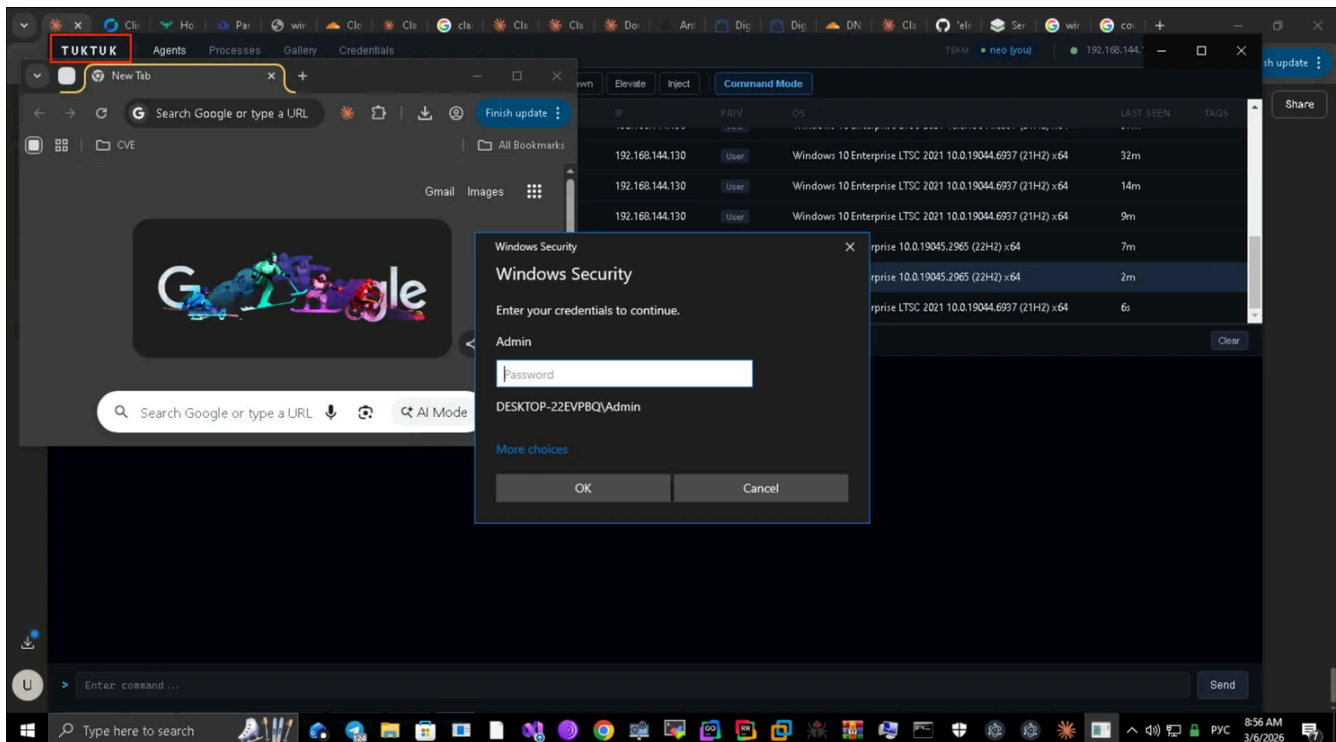


Figure 3. TukTuk C2 dashboard showing hostname DESKTOP-22EVPBQ

Referenced Public Sources

- The DFIR Report, "Flash Alert: EtherRAT and TukTuk C2 End in The Gentlemen Ransomware" (May 11, 2026)
- Related X (Twitter) post : <https://x.com/evangeorgevoug/status/2038198328631193700>

3. TukTuk C2 Framework

Identification

The TukTuk C2 was identified on the server through several artifacts: a dashboard screenshot, the executable, and indications of its development environment.

Dashboard

The TukTuk C2 management dashboard was found in screenshot2.png on the server. The dashboard displays connected agents along with their IP addresses, privilege levels, and operating system versions. (The hostname and operator identifier visible in this screenshot also match information in The Gentlemen's public report; this correspondence is treated as an attribution indicator in the **Attribution to The Gentlemen** section.)

TukTuk.exe

The executable `TukTuk.exe` and the compressed archive `tuktuk-v2.0_10.zip` were identified in the same path.

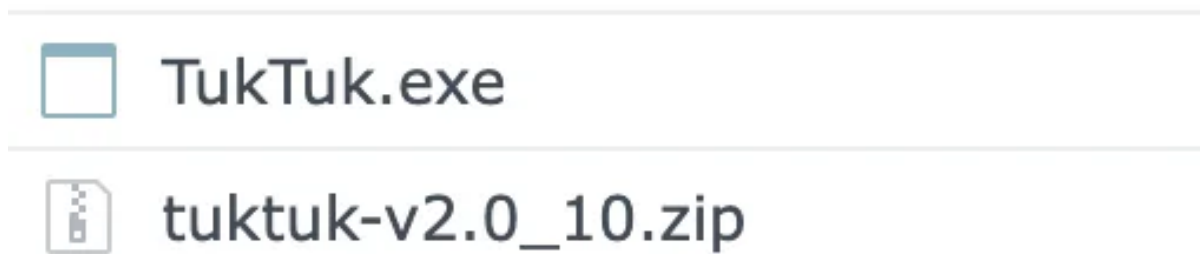


Figure 4. `TukTuk.exe` and `tuktuk-v2.0_10.zip` identified on the server

Based on its observed functionality, `TukTuk.exe` is assessed to facilitate access to and control of the TukTuk server; its login screen prompts for a server address, operator name, and password before connecting.

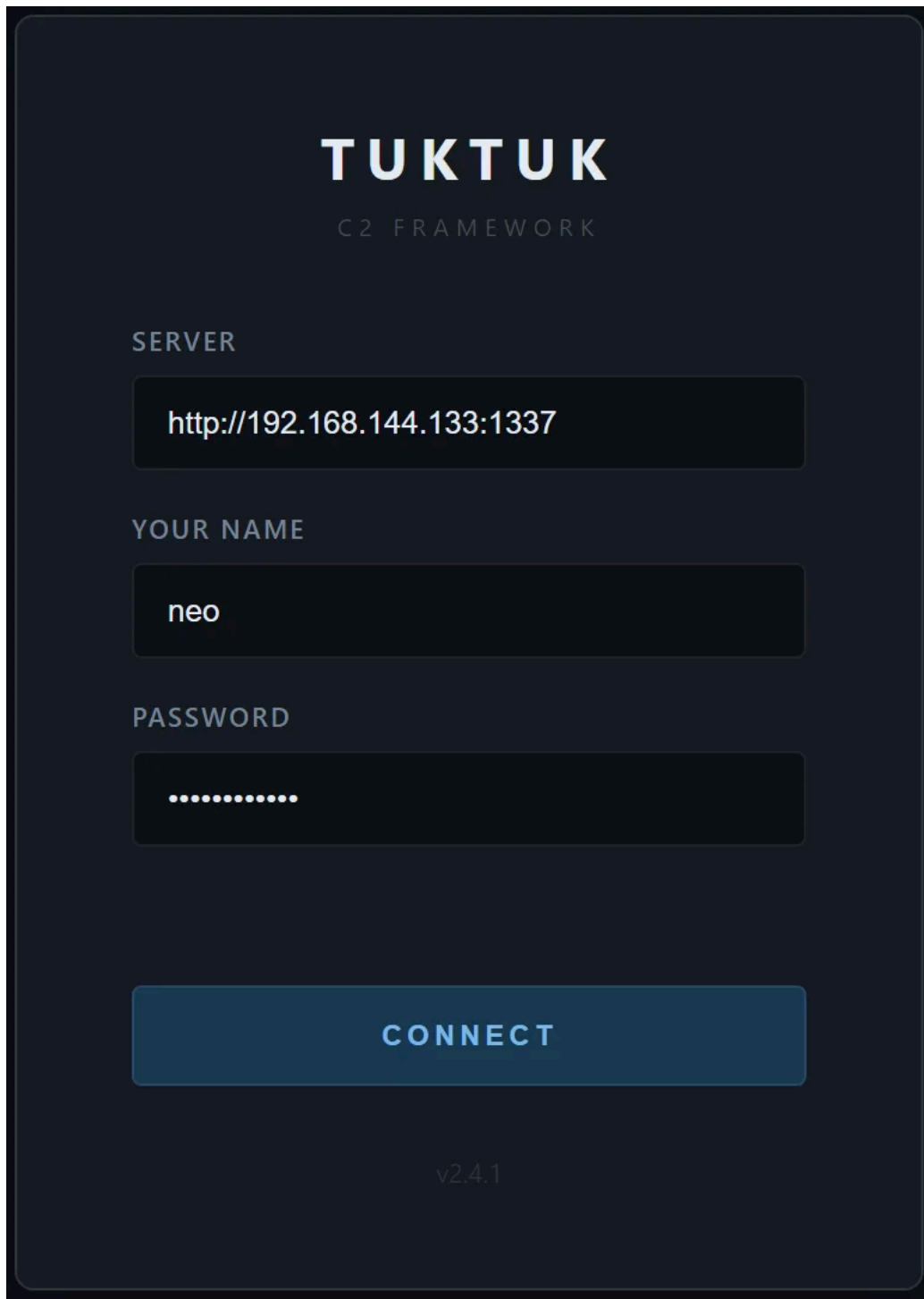


Figure 5. TukTuk C2 login screen used to access and control the TukTuk server

TukTuk.exe has a pre-existing submission history on VirusTotal and was flagged as malicious by 2 of 68 security vendors.

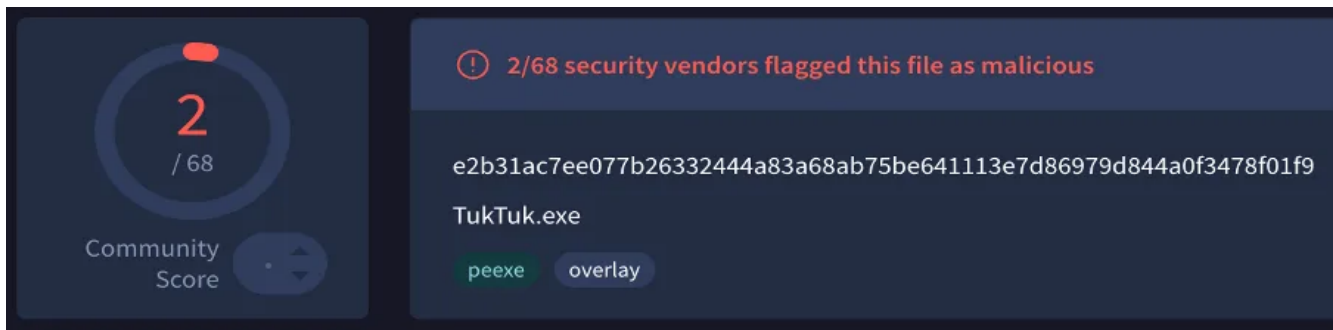


Figure 6. VirusTotal detections for TukTuk.exe, flagged as malicious by 2 of 68 vendors

Development Environment

screenshot.png contained indications that the development of TukTuk C2 was carried out using artificial intelligence (AI).

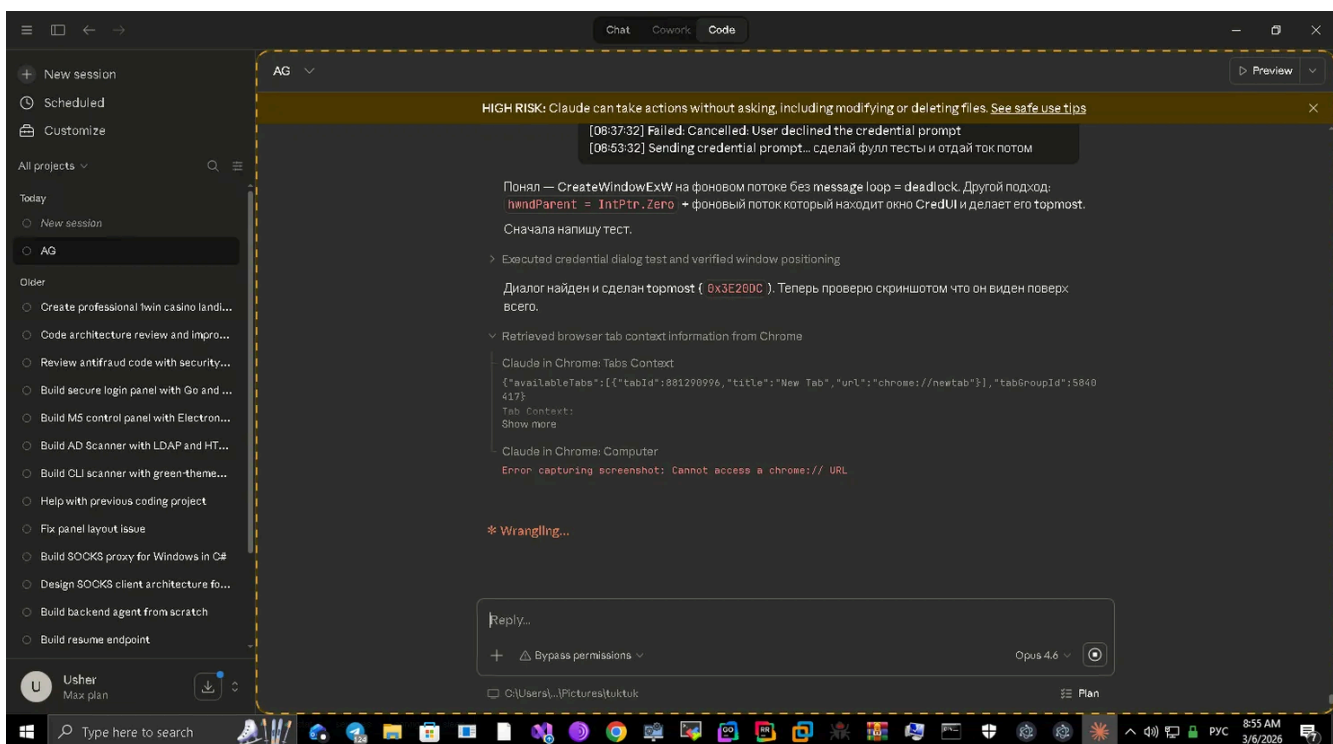


Figure 7. Screenshot indicating TukTuk C2 development was carried out using artificial intelligence

Project Structure

The **tuktuk-v2.0_10.zip** archive was confirmed to have no prior public disclosure.

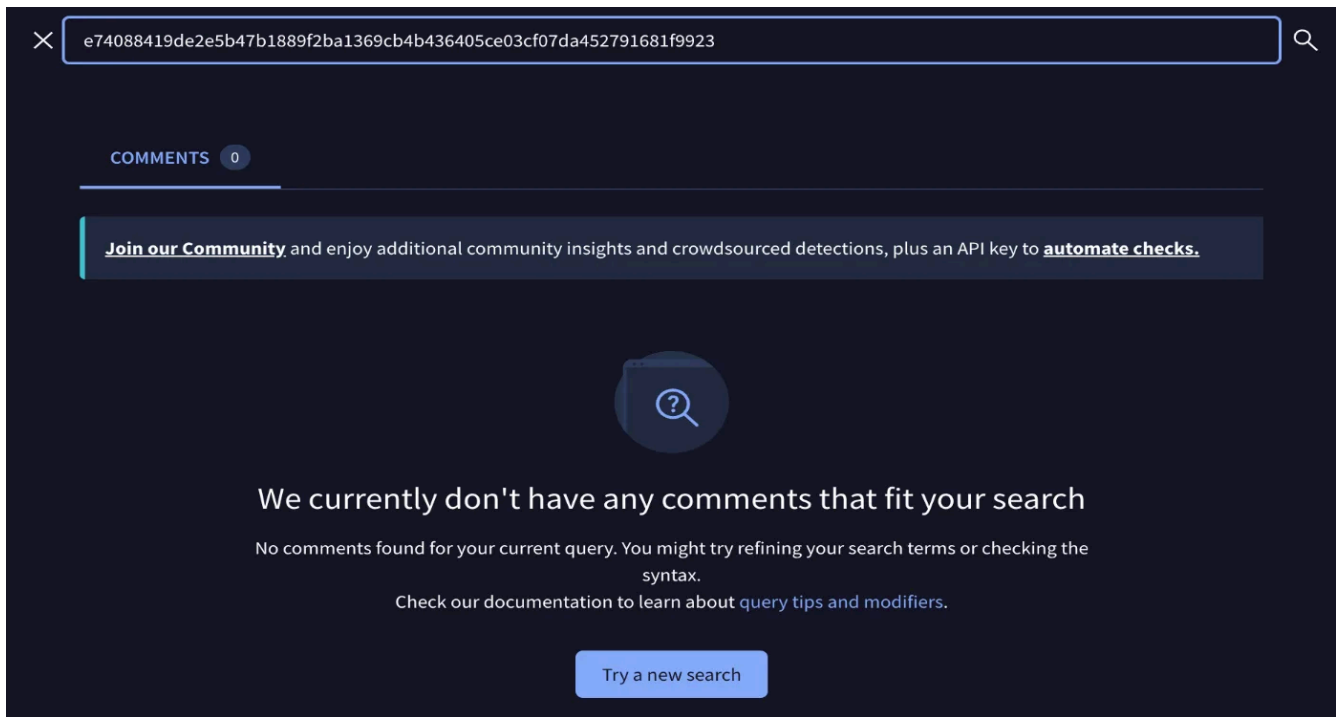


Figure 8. VirusTotal comments search for the `tuktuk-v2.0_10.zip` hash returning no prior public discussion

The archive is organized into four primary directories:

Collectively, these directories represent the **complete project structure of the TukTuk C2 framework identified on the server.**



Figure 9. TukTuk C2 project structure containing Windows Agent, Linux Agent, Backend, and Panel directories

Windows Agent

The `tuktuk-agent` directory contains the **source code of the TukTuk Agent operating in a Windows environment**.

The agent is executed on compromised systems, where it:

- Collects system information
- Communicates with the C2 server
- Executes commands delivered through the Panel

The source code is written in C# and corresponds to remote-control functionality provided through the Panel, including:

- File management
- Screen capture
- Command execution

Figure 10. Windows Agent `Poll()` function retrieving C2 commands and shell sessions

Linux Agent

A separate `tuktuk-agent-linux` directory was identified.

The presence of a Linux agent, independent of the Windows agent, supports the assessment that TukTuk was developed to operate across multiple operating systems rather than being confined to the Windows environment.

Figure 11. Linux Agent `poll()` function retrieving C2 commands

Backend

The `tuktuk-backend` directory contains the backend component of the framework.

Analysis confirmed that the **Backend processes agent connections according to Listeners created in the TukTuk Panel.**

The Backend is responsible for functions related to:

- Agent status
- Command delivery
- Result reception

Figure 12. TukTuk Backend `main()` initializing the database, API router, and HTTP server

Panel

The `tuktuk-panel` directory contains the management interface.

Implemented in Node.js with Electron, it can be packaged and run as a Windows executable (`.exe`).

The Panel serves as the management interface for TukTuk C2, providing functionality to monitor connected agents and execute remote commands.

The logo, menu layout, and overall interface differ from previously documented versions of the TukTuk panel available in public sources.

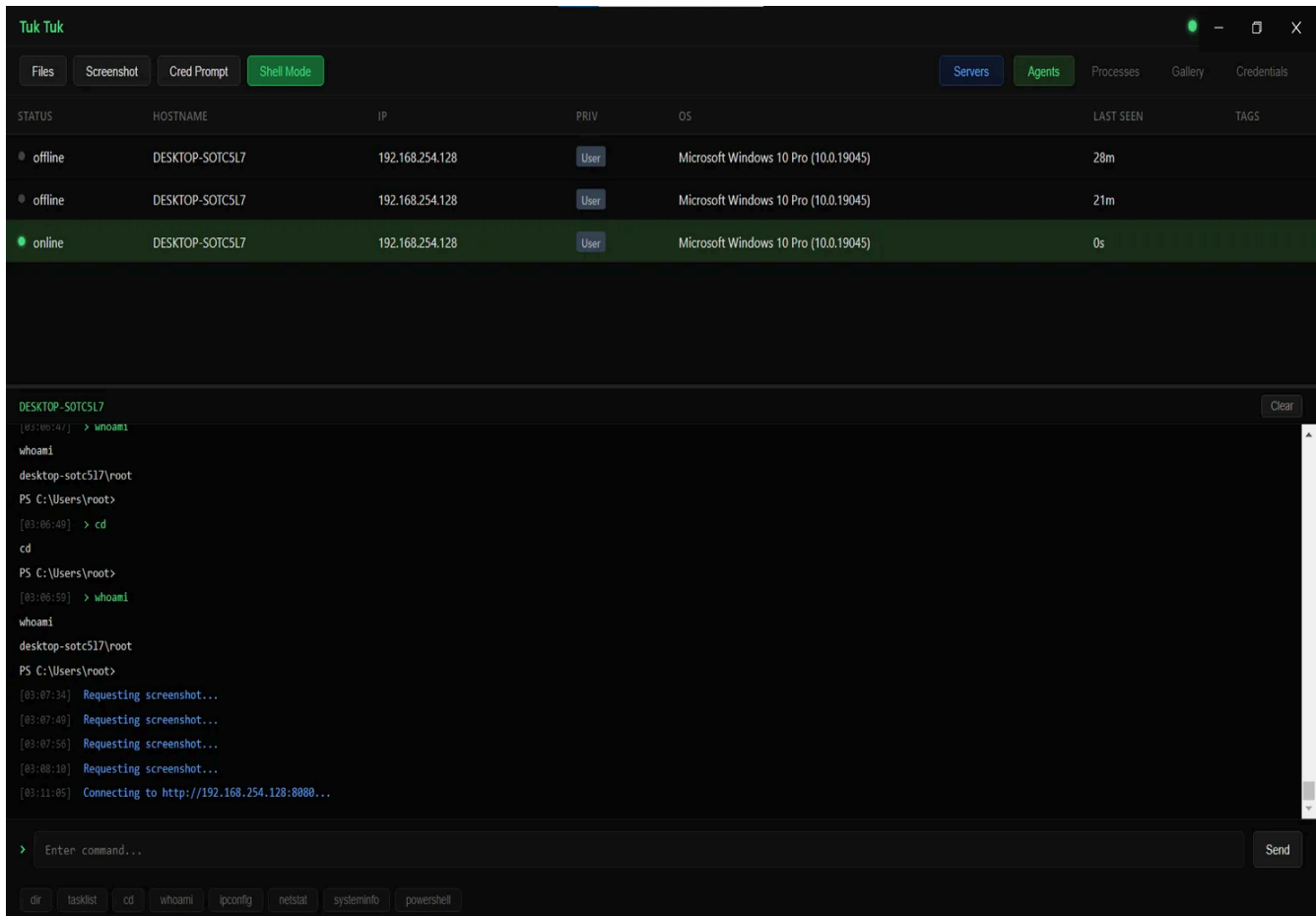


Figure 13. TukTuk Panel showing connected agents and Shell Mode remote command execution

Top-right menus

The top-right section of the TukTuk Panel contains:

- Servers
- Agents
- Processes
- Gallery
- Credential

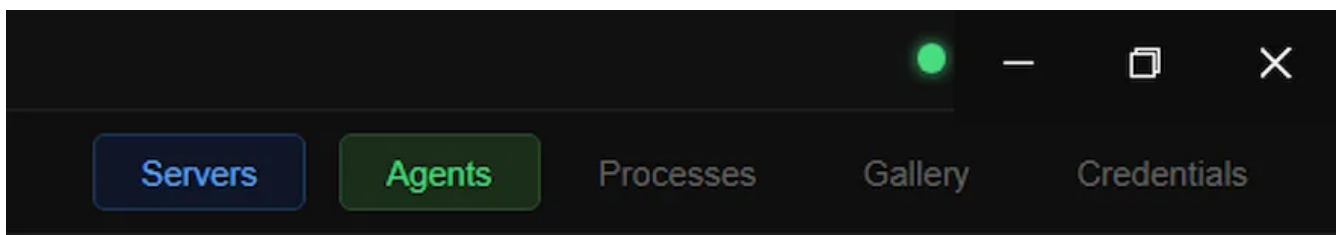


Figure 14. TukTuk Panel top-right menus: Servers, Agents, Processes, Gallery, and Credentials

Servers

The **Servers** menu provides access to the backend systems to which the agents are connected.

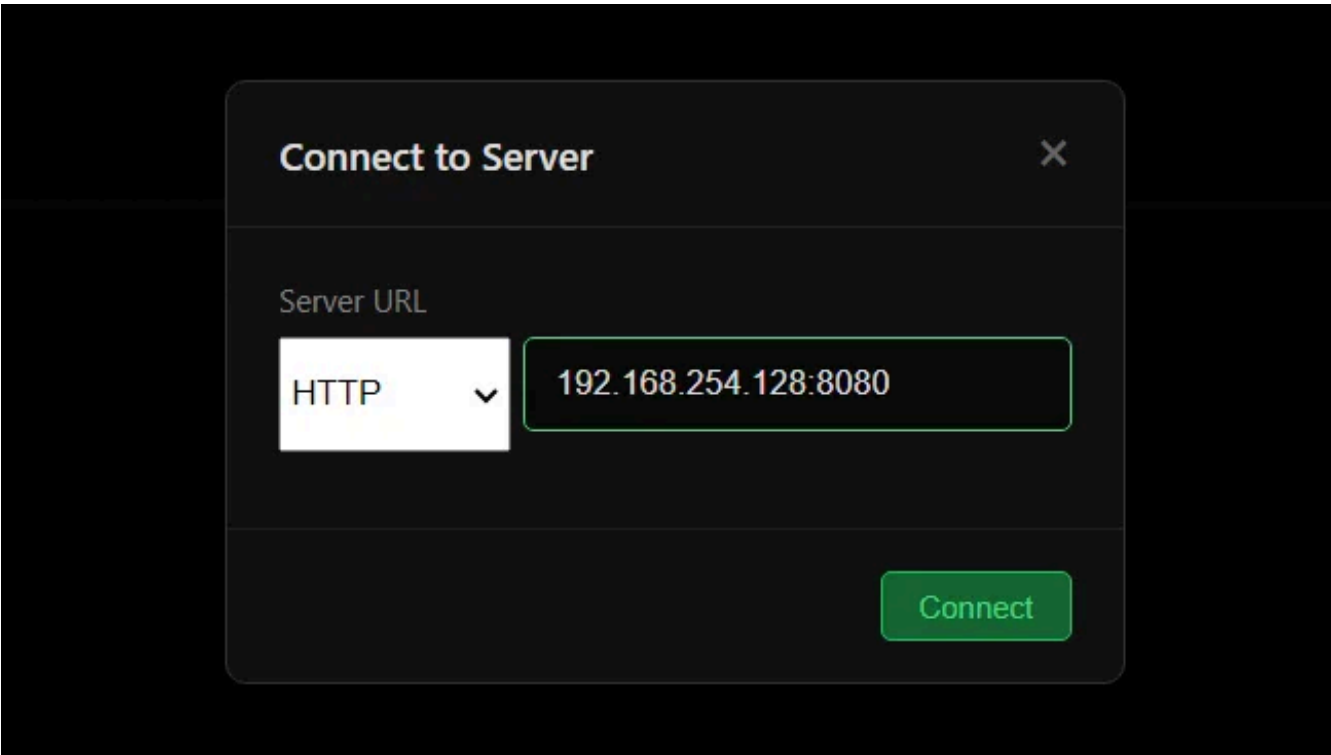


Figure 15. TukTuk Panel Servers menu Connect to Server dialog

Agent

The **Agent** menu displays the list of compromised hosts connected to the TukTuk framework and their activation status.

The image shows the 'Agents' menu in the TukTuk Panel. It features a toolbar with buttons for 'Files', 'Screenshot', 'Cred Prompt', and 'Shell Mode'. Below the toolbar is a table listing compromised hosts. The table has columns for STATUS, HOSTNAME, IP, PRIV, OS, LAST SEEN, and TAGS. There are three rows of data, with the last row highlighted in green as it is 'online'.

STATUS	HOSTNAME	IP	PRIV	OS	LAST SEEN	TAGS
offline	DESKTOP-SOTCSL7	192.168.254.128	User	Microsoft Windows 10 Pro (10.0.19045)	28m	
offline	DESKTOP-SOTCSL7	192.168.254.128	User	Microsoft Windows 10 Pro (10.0.19045)	21m	
online	DESKTOP-SOTCSL7	192.168.254.128	User	Microsoft Windows 10 Pro (10.0.19045)	0s	

Figure 16. TukTuk Panel Agents menu listing compromised hosts and activation status

Processes

The **Processes** menu enables remote control of processes running on compromised hosts.

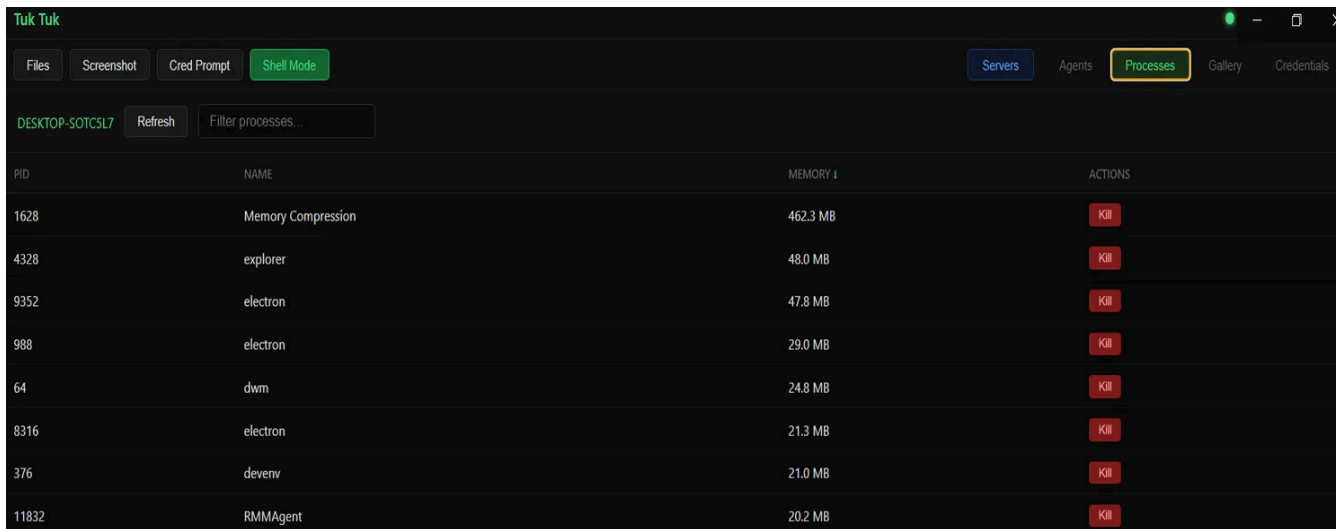


Figure 17. TukTuk Panel Processes menu showing remote process control

Gallery

The **Gallery** menu displays screenshots saved from compromised hosts.

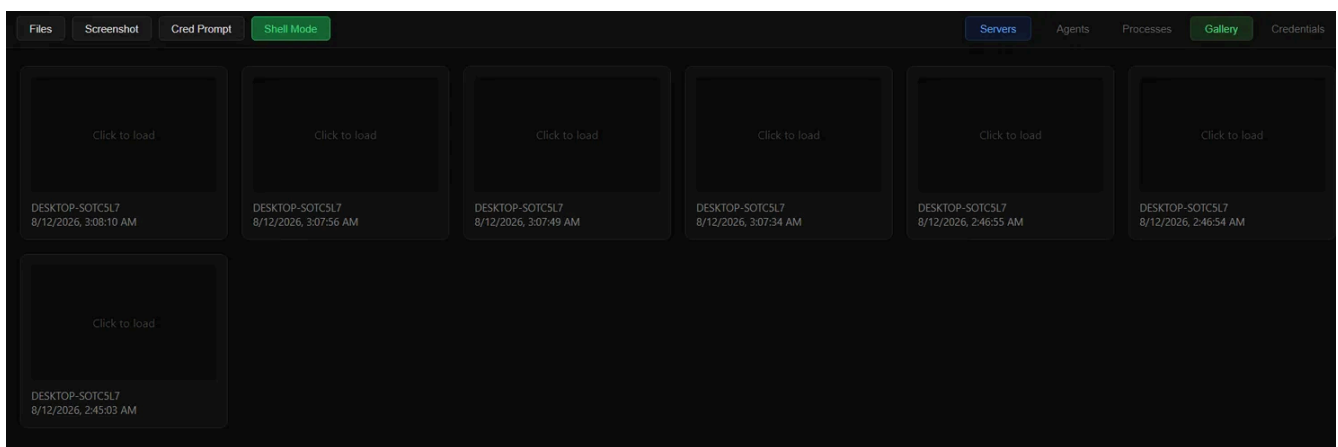


Figure 18. TukTuk Panel Gallery menu displaying screenshots saved from compromised hosts

Credential

The **Credential** menu displays the list of credentials received from compromised hosts.

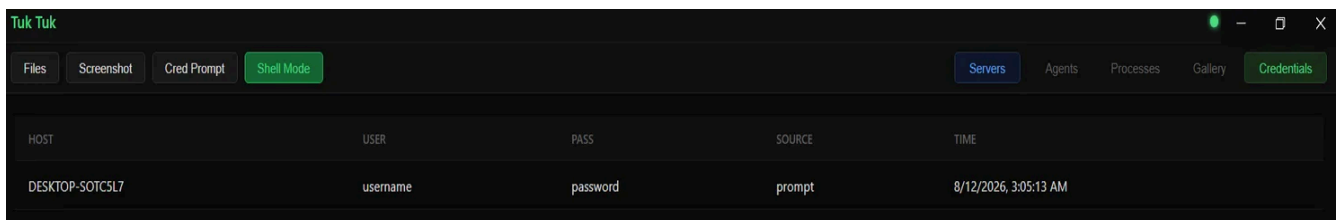


Figure 19. TukTuk Panel Credentials menu listing credentials received from compromised hosts

Top-left menus

The top-left section of the TukTuk Panel contains:

- Files
- Screenshot
- Cred Prompt
- Command Mode

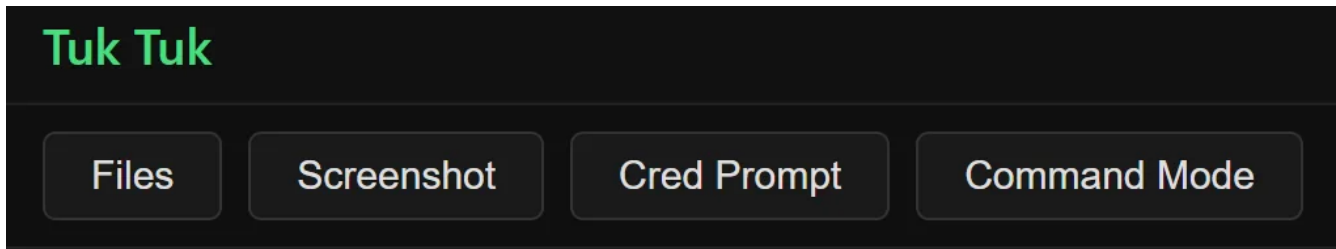


Figure 20. TukTuk Panel top-left menus: Files, Screenshot, Cred Prompt, and Command Mode

Files

The **Files** menu enables the upload of files from the local system to compromised hosts.

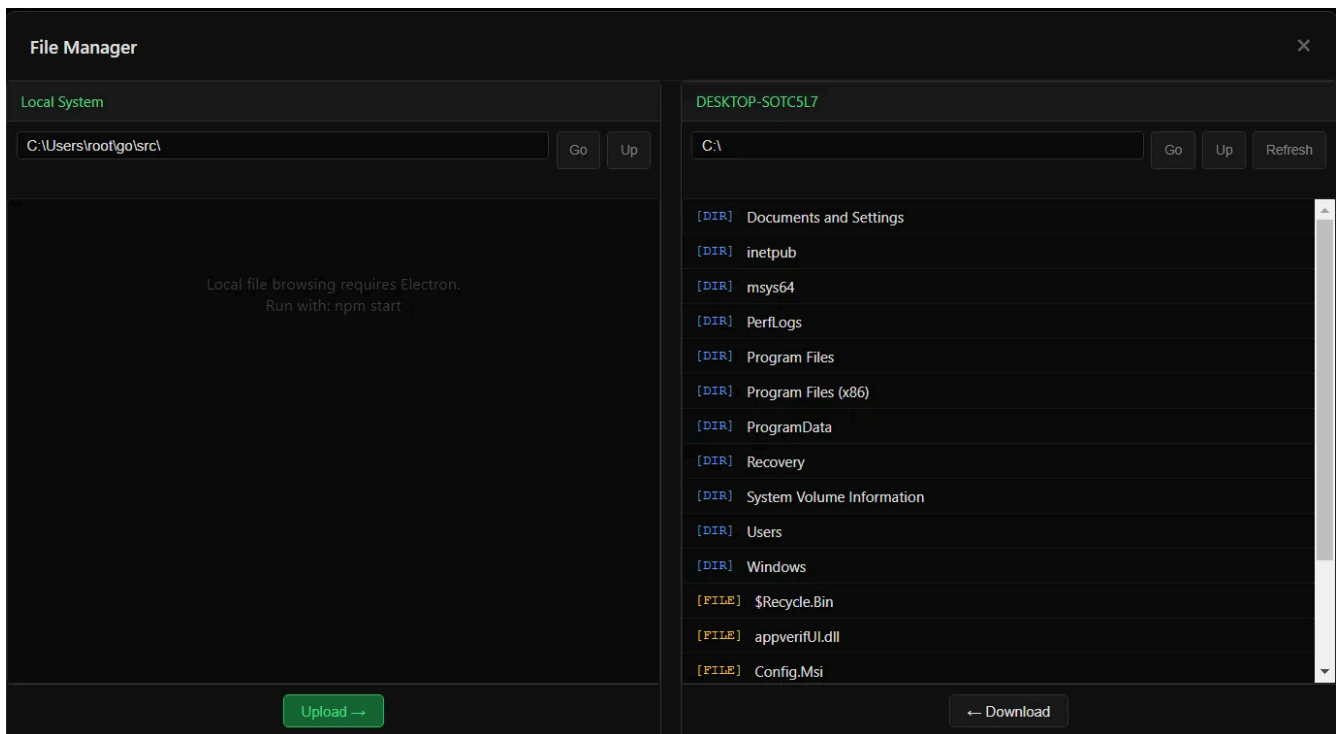


Figure 21. TukTuk Panel Files menu used to upload files to compromised hosts

Screenshot

The **Screenshot** menu enables the remote capture and storage of the current screen of connected compromised hosts.

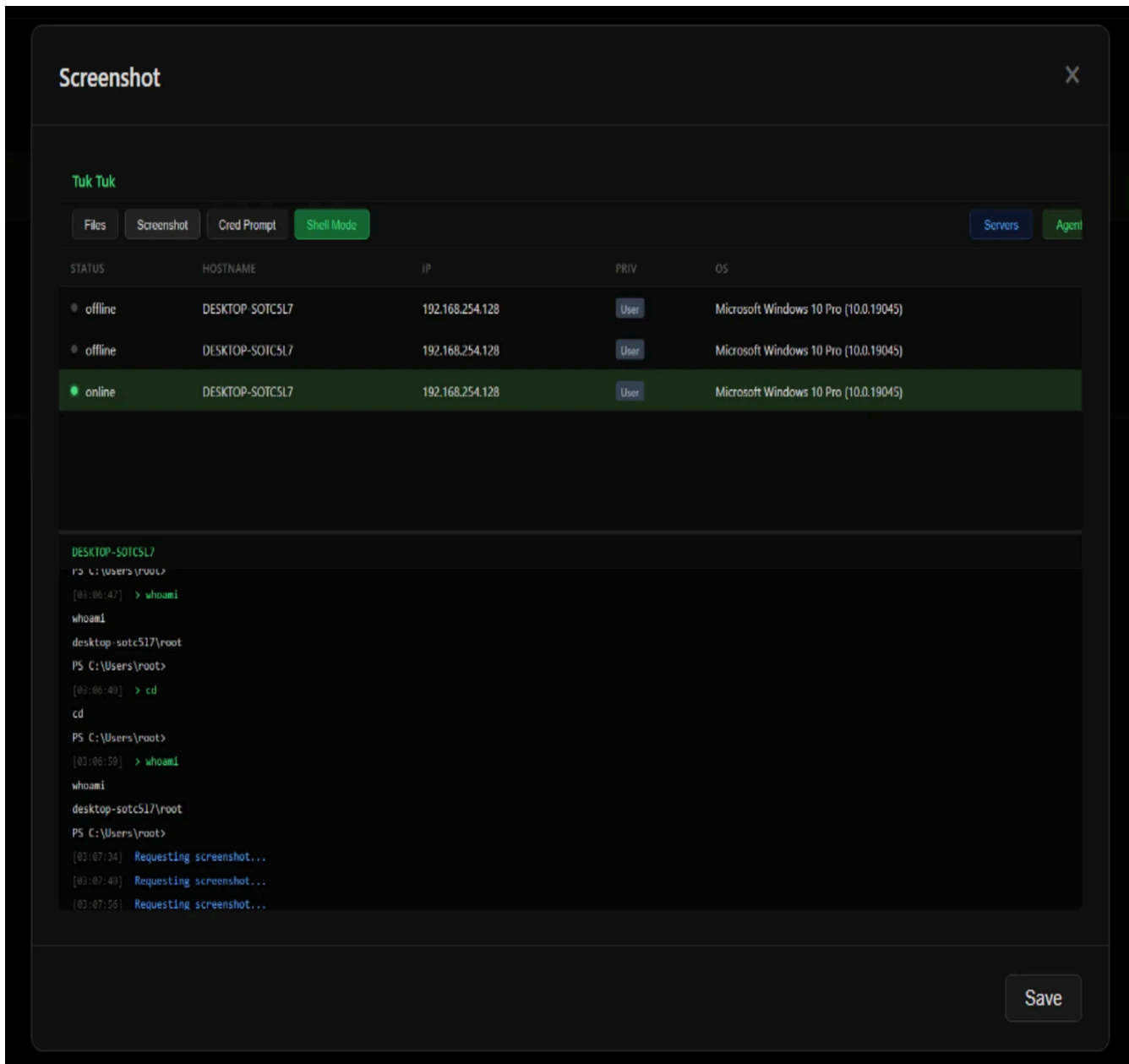


Figure 22. TukTuk Panel Screenshot menu requesting remote screen capture from a compromised host

Cred Prompt

The **Cred Prompt** menu elicits credentials by spoofing a Windows Security dialog.

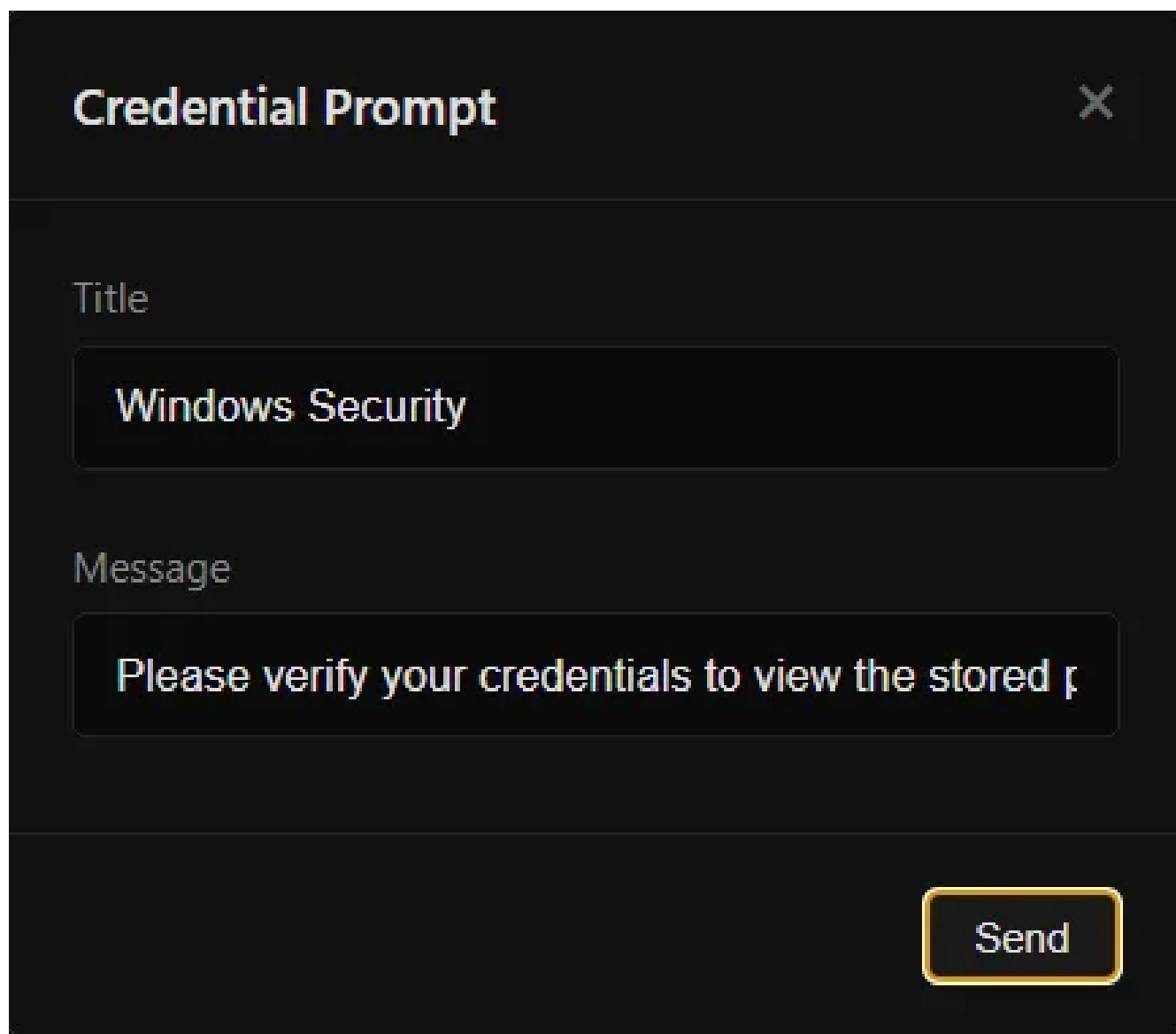


Figure 23. TukTuk Panel Cred Prompt menu configured to spoof a Windows Security dialog

Upon execution of the command, a window visually identical to the legitimate Windows Security interface appears on the compromised host.

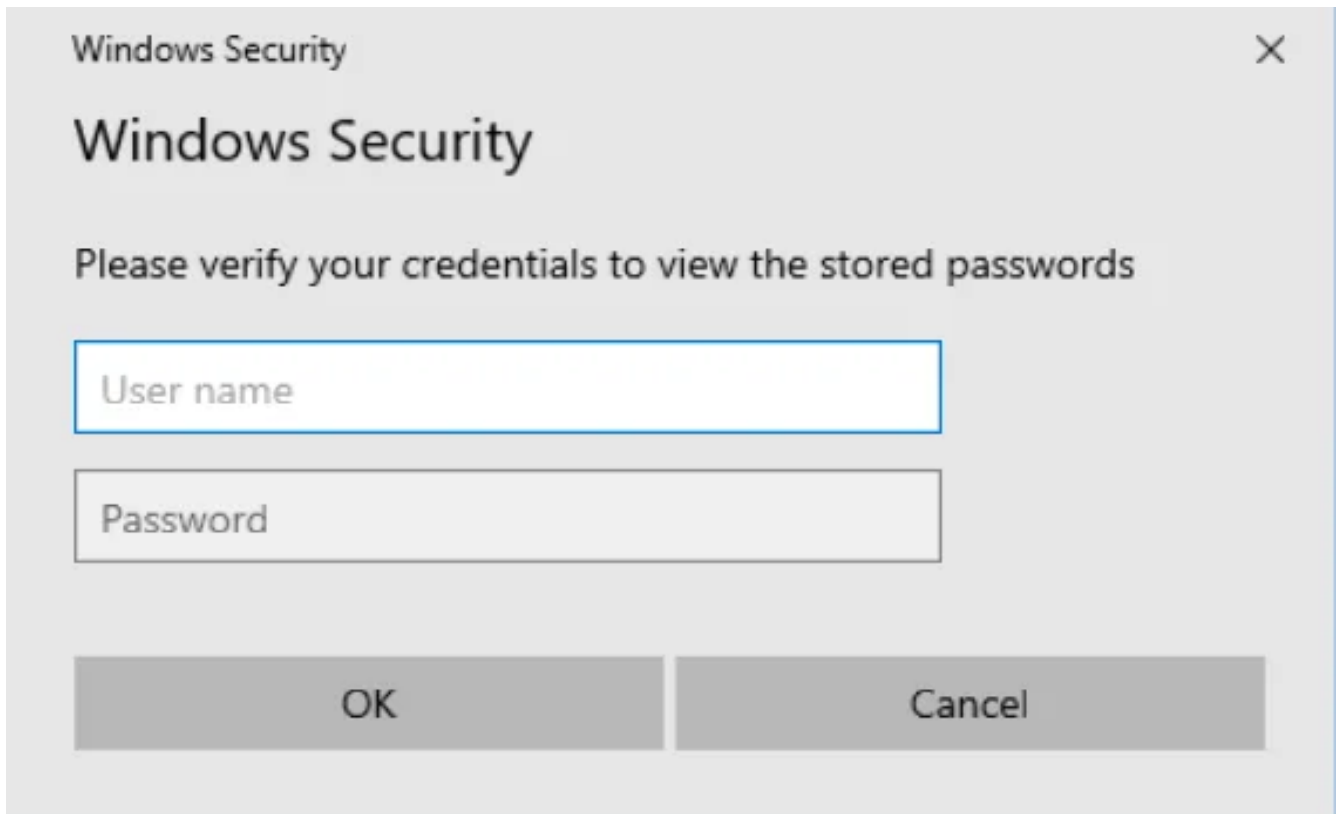


Figure 24. Spoofed Windows Security dialog displayed on the compromised host

Information entered by the user is recorded in the TukTuk framework.

```
[03:04:58] Sending credential prompt...  
[03:05:13] Username: username  
Password: password  
[03:05:13] Credential saved
```

Figure 25. Credentials entered in the spoofed prompt recorded in the TukTuk framework

Command Mode

The **Command Mode** menu provides predefined command shortcuts that enable an operator to remotely retrieve information from a compromised host.

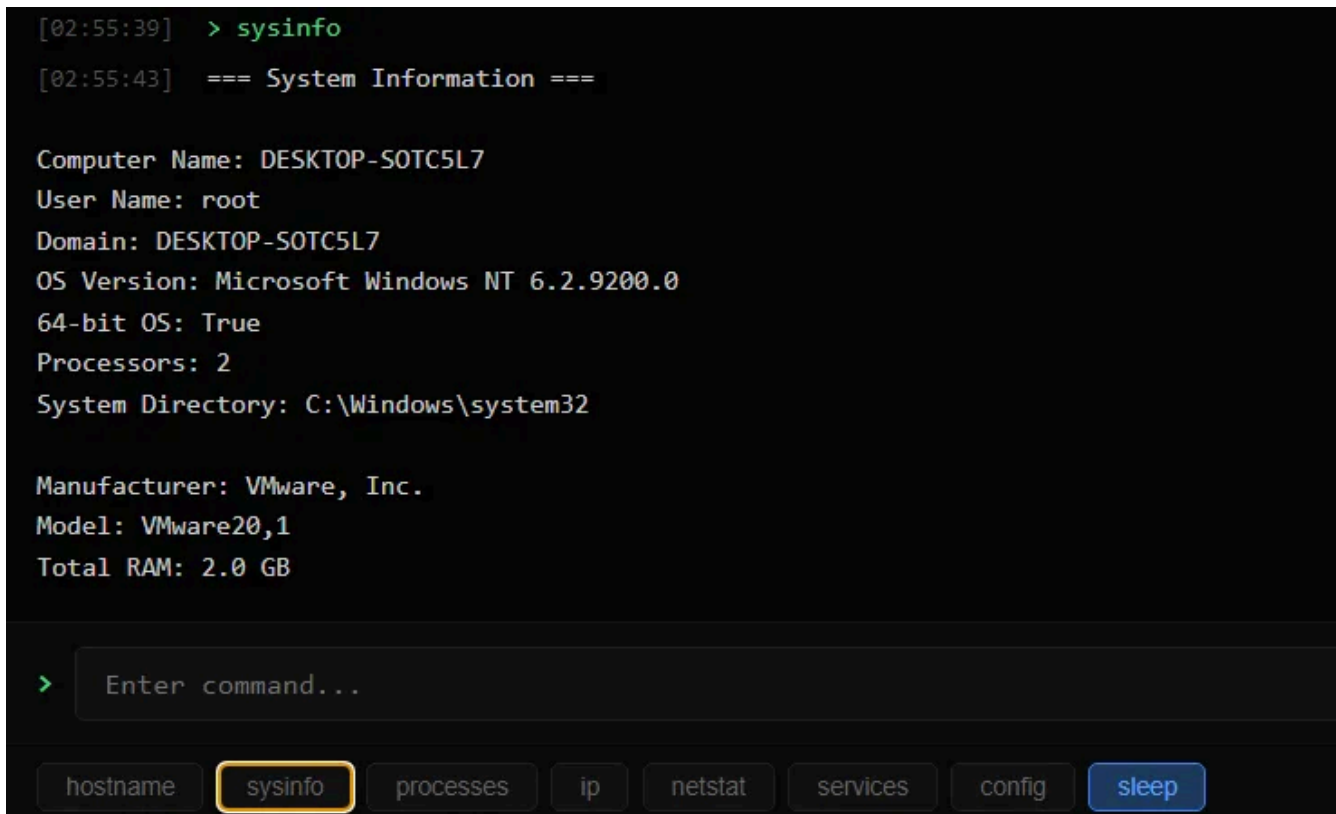


Figure 26. TukTuk Panel Command Mode showing predefined shortcuts including sysinfo

Clicking Command Mode again activates **Shell Mode**, which displays shortcuts for commands available in the Windows environment.

In addition to these predefined options, the operator can directly enter and execute arbitrary commands on the compromised host.

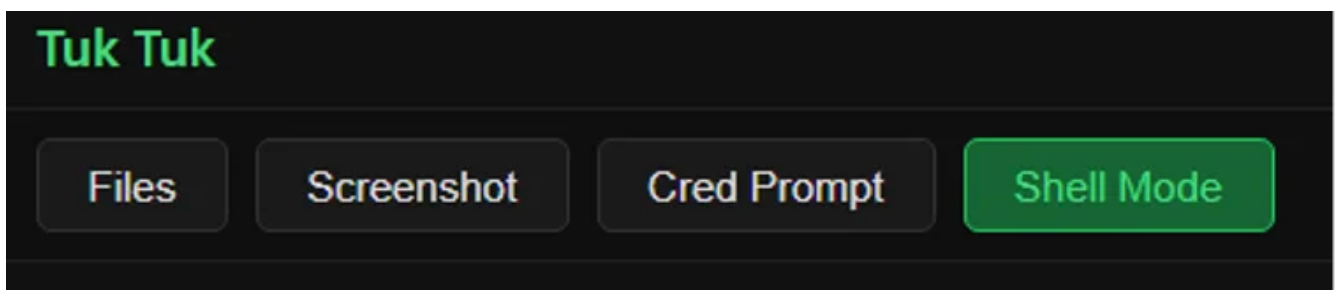


Figure 27. TukTuk Panel Shell Mode activated from Command Mode

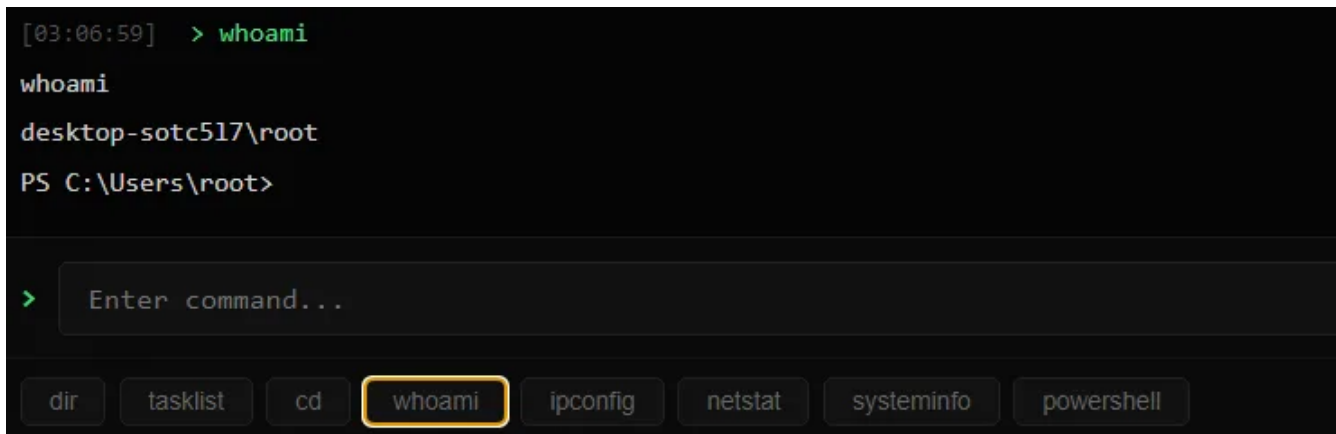


Figure 28. TukTuk Panel Shell Mode with Windows command shortcuts and arbitrary command execution

DLL Sideload via Greenshot

A malicious `log4net.dll` file known to have been used in the attack by The Gentlemen was identified.

The key files are:

- Legitimate `Greenshot.exe`
- Modified `log4net.dll`

The legitimate `Greenshot` executable loads the malicious `log4net.dll` from the same directory, thereby launching the **TukTuk Agent**.

The `log4net.dll` file has previously been submitted to VirusTotal and is currently detected as malicious by 17 vendors.

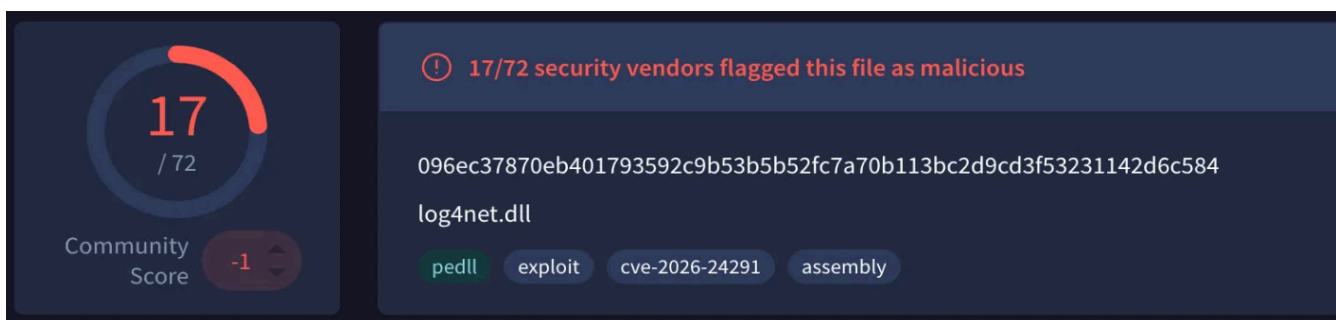


Figure 29. VirusTotal detections for malicious `log4net.dll`, flagged by 17 of 72 vendors

Static analysis of the malicious `log4net.dll` identified:

- Code associated with the TukTuk C2 framework
- Server addresses

- Configuration information
- Token-related information used for C2 communication

Identified services included:

- Slack
- GitHub
- Dropbox

Figure 30. TukTuk C2 configuration in `log4net.dll` including the domain `borjumaniya.store` and Slack, GitHub, and Dropbox settings

Notably, both are the domain and the GitHub repository that were previously identified as C2 infrastructure in the analysis report published on X.

DLL Sideload Research

A separate **sideLoad-research directory** was identified on the analyzed server.

The directory contains data analyzing and testing the potential for DLL sideloading targeting:

- Greenshot
- ProcMon
- Slack
- Postman

Greenshot	
Greenshot-Portable	
ProcMon	
Slack	
test_Greenshot	
test_Postman	
analyze.ps1	4KB
analyze_slack.ps1	3KB
analyze2.ps1	5KB
build_slack.bat	1KB
build_slack.ps1	2KB
check_dotnet.ps1	4KB
check_log4net_usage.ps1	2KB
check_slack.ps1	1KB
deep_scan.ps1	4KB
download_slack.ps1	2KB
extract_slack.ps1	2KB
full_slack_test.ps1	2KB
Greenshot_capture.pml	1KB
greenshot_log.pml	1KB
greenshot_results.csv	1KB
Greenshot-installer.exe	1,742KB
Greenshot-Portable-1.2.10.6.zip	663KB
hide_slack.ps1	3KB
ildasm_scan.ps1	1KB
index.html	3KB
package.ps1	2KB
Postman-installer.exe	145,608KB
ProcessMonitor.zip	2,944KB
test_launch.ps1	2KB
test_launch2.ps1	2KB
test_slack_dlls.ps1	3KB
test_slack_phantom.ps1	2KB
test_slack_proxy.ps1	3KB
test_slack_sideload.ps1	4KB

Figure 31. sideload-research directory containing DLL sideloading tests against Greenshot, ProcMon, Slack, and Postman

4. Analysis of Training and Research Materials on EDR Neutralization

Research materials on **EDR neutralization and vulnerable drivers** were identified on the server.

The materials were structured step-by-step from **Lesson 1 to Lesson 4**.

The collection progresses from learning the operating principles of existing EDR killers and proof-of-concept (PoC) BYOVD:

- Implementing live EDR neutralization tools
- Searching for new vulnerable drivers
- Researching kernel-level attack techniques



Figure 32. EDR neutralization research materials organized from Lesson 1 to Lesson 4

Lesson 1: GentleKiller and UnknownKiller

Identified materials include:

- eb.sys
- UnKnownKiller.c
- UnknownKiller.exe
- BYOCD_UnkownKiller.md

eb.sys has been analyzed in public reports as a driver associated with a variant of **GentleKiller**.

UnknownKiller source code and executable files utilizing the driver were identified.

☐	eb.sys
☐	LESSON_BYOVD_UnknownKiller.md
☐	step0_launch_target.bat
☐	step1_install_driver.bat
☐	step2_kill_process.bat
☐	step3_cleanup.bat
☒	UnknownKiller.c
☐	UnknownKiller.exe

Figure 33. Lesson 1 materials including eb.sys, UnknownKiller source code, and related scripts

A document describing **BYOVD-based EDR neutralization** was also included, explaining how protected processes are accessed and terminated using the eb.sys driver.

```
## 5. Kill Chain (последовательность атаки)
...
1. Initial Access (начальный доступ)
   └─ Любой метод: фишинг, эксплойт, украденные учётные данные
      |
2. Privilege Escalation (повышение привилегий)
   └─ Получение прав локального администратора
      |
3. Defense Evasion (обход защиты) ← Мы ЗДЕСЬ
   └─ Доставка eb.sys на диск
   └─ sc create eb type= kernel binPath= ...
   └─ sc start eb
   └─ UnknownKiller.exe MsMpEng.exe ← убиваем антивирус / EDR
      |
4. Execution (выполнение)
   └─ Запуск основной вредоносной нагрузки без противодействия
      |
5. Impact (воздействие)
   └─ Шифрование, эксфилтрация, закрепление...
...

**Реальные APT-группы, использующие BYOVD:**
- Lazarus Group
- BlackByte (ransomware)
- AvosLocker (ransomware)
- Cuba ransomware
---
```

Figure 34. BYOVD-based EDR neutralization kill chain using eb.sys and UnknownKiller

Lesson 2: EDRKiller and WarsawKiller

Identified materials include:

- EDRKiller
- WarsawKiller
- LESSON2_EDR_vs_BYOVD.md

The materials include educational content for understanding bypass techniques against actual EDR security products, as well as hands-on materials on EDR neutralization.

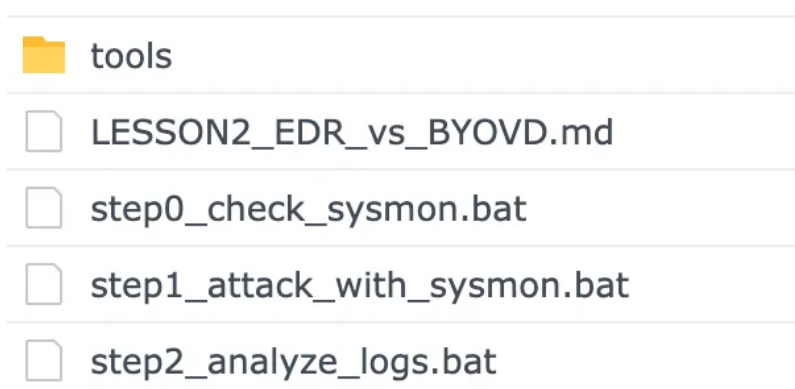


Figure 35. Lesson 2 materials including LESSON2_EDR_vs_BYOVD.md and related test scripts

LESSON2_EDR_vs_BYOVD.md compares **automatic recovery times following process termination for major EDR products**.

It appears to measure the window of opportunity for follow-on attacks by analyzing periods during which protection is active and inactive.

11.1 Визуальное сравнение: окно без защиты

...

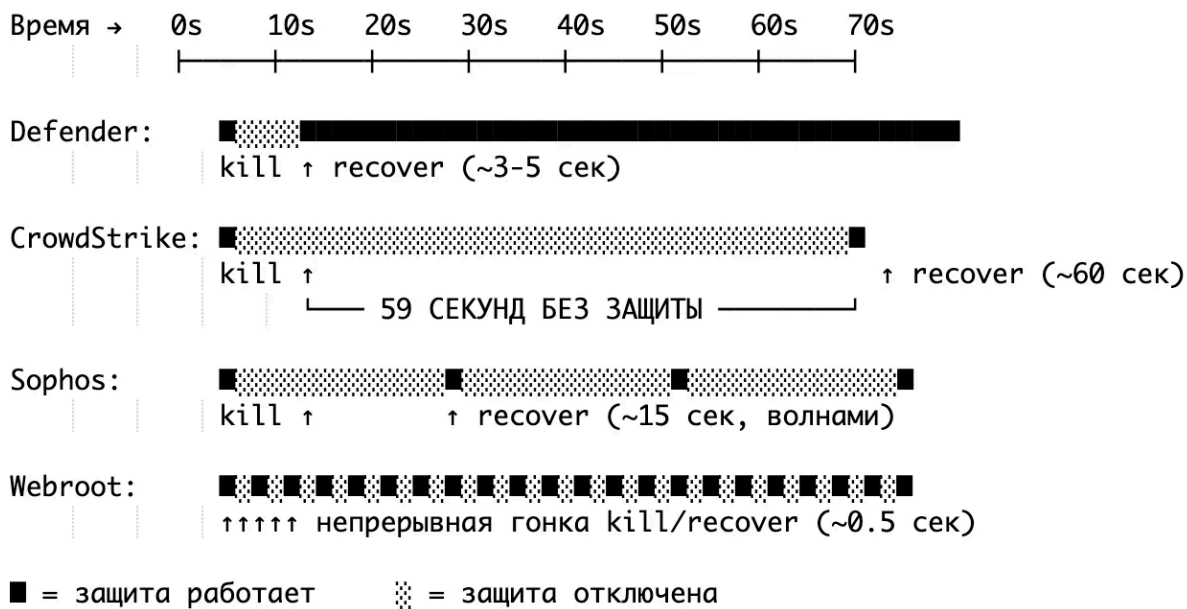


Figure 36. Comparison of EDR automatic recovery times following process termination

Lesson 3: Vulnerable Driver Hunting

Identified materials include:

- bd_test_k7old.bat
- bd_test_tfsysmon.bat
- bd_test_safetica.bat
- Driver_Hunting.md

Files for conducting repeated tests against multiple drivers using a previously known EDR Killer were identified.

- ☐ bd_test_batch.bat
- ☐ bd_test_k7old.bat
- ☐ bd_test_safetica.bat
- ☐ bd_test_tfsysmon.bat
- ☐ LESSON3_Driver_Hunting.md

Figure 37. Lesson 3 driver-hunting scripts including bd_test_k7old.bat, bd_test_tfsysmon.bat, and bd_test_safetica.bat

Driver_Hunting.md describes the testing procedure for screening drivers usable for EDR neutralization.

The method involves:

1. Registering each driver as a Windows kernel service
2. Verifying its operational status and file existence
3. Removing the service after the test is complete
4. Determining whether the driver can operate in a real-world environment based on driver loading results

3.2 Методология тестирования

Для каждого драйвера выполняем:

```
```batch
@echo off
REM === Шаг 1: Создаём сервис ===
sc create TestDriver type= kernel binPath= C:\tmp\driver.sys

REM === Шаг 2: Пытаемся загрузить ===
sc start TestDriver 2>&1
echo RESULT=%errorlevel%

REM === Шаг 3: Проверяем статус ===
sc query TestDriver

REM === Шаг 4: Проверяем, удалён ли файл EDR ===
dir C:\tmp\driver.sys

REM === Шаг 5: Очистка ===
sc stop TestDriver 2>nul
sc delete TestDriver 2>nul
```
```

****Коды ошибок загрузки:****

| Код | Название | Причина |
|------|-------------------|--------------------------------|
| 0 | SUCCESS | Драйвер загружен |
| 5 | ACCESS_DENIED | EDR заблокировал (blockhash) |
| 193 | BAD_IMAGE (0xC1) | EDR повредил/подменил файл |
| 577 | INVALID_SIGNATURE | Подпись не прошла проверку |
| 1275 | DRIVER_BLOCKED | MS Vulnerable Driver Blocklist |

Figure 38. Driver hunting methodology for loading, testing, and removing candidate kernel drivers

Lesson 4: Kernel-Level Research

Identified materials include:

`0day_Driver_Research.md`

Code for analyzing internal driver behavior or researching new kernel bypass techniques was identified.

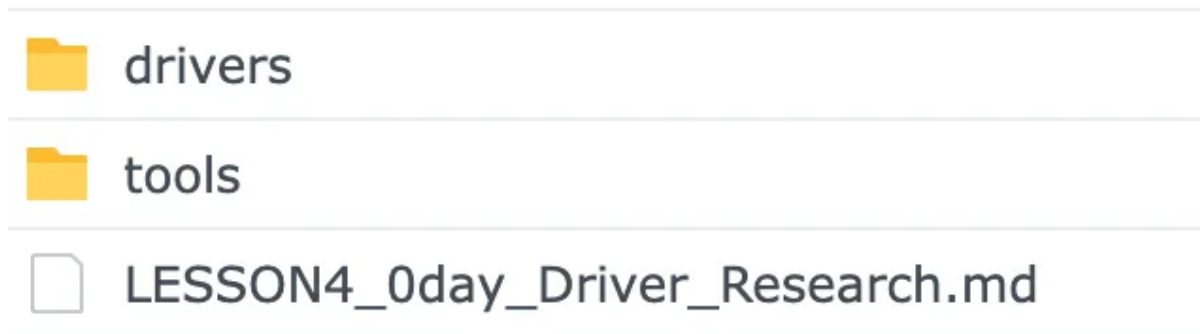


Figure 39. Lesson 4 materials including `0day_Driver_Research.md` and related driver research files

`0day_Driver_Research.md` indicates that The Gentlemen was conducting its own research into 0-day drivers.

The document describes a previously undisclosed driver vulnerability that had not yet been assigned a CVE and documents subsequent steps taken to verify the vulnerability.

5. Exfiltrated Data

Global Technology Company (Jira)

Overview

In addition to the TukTuk framework and related attack materials, the analyzed server contained **data assessed to have been exfiltrated from the Jira environment of a global technology company** that provides a range of test, measurement, and network solutions used in the development and validation of key products across the semiconductor, telecommunications, electronics, and defense industries.

A total of **224 Jira tickets** and **eight Jira attachments** were identified in the server's jira-
loot folder. **An attacker-created document** organized the collected Jira materials into
categories including internal technical support records, credentials, infrastructure-related
information, customer communications, vulnerability-related materials, and technical
attachments.

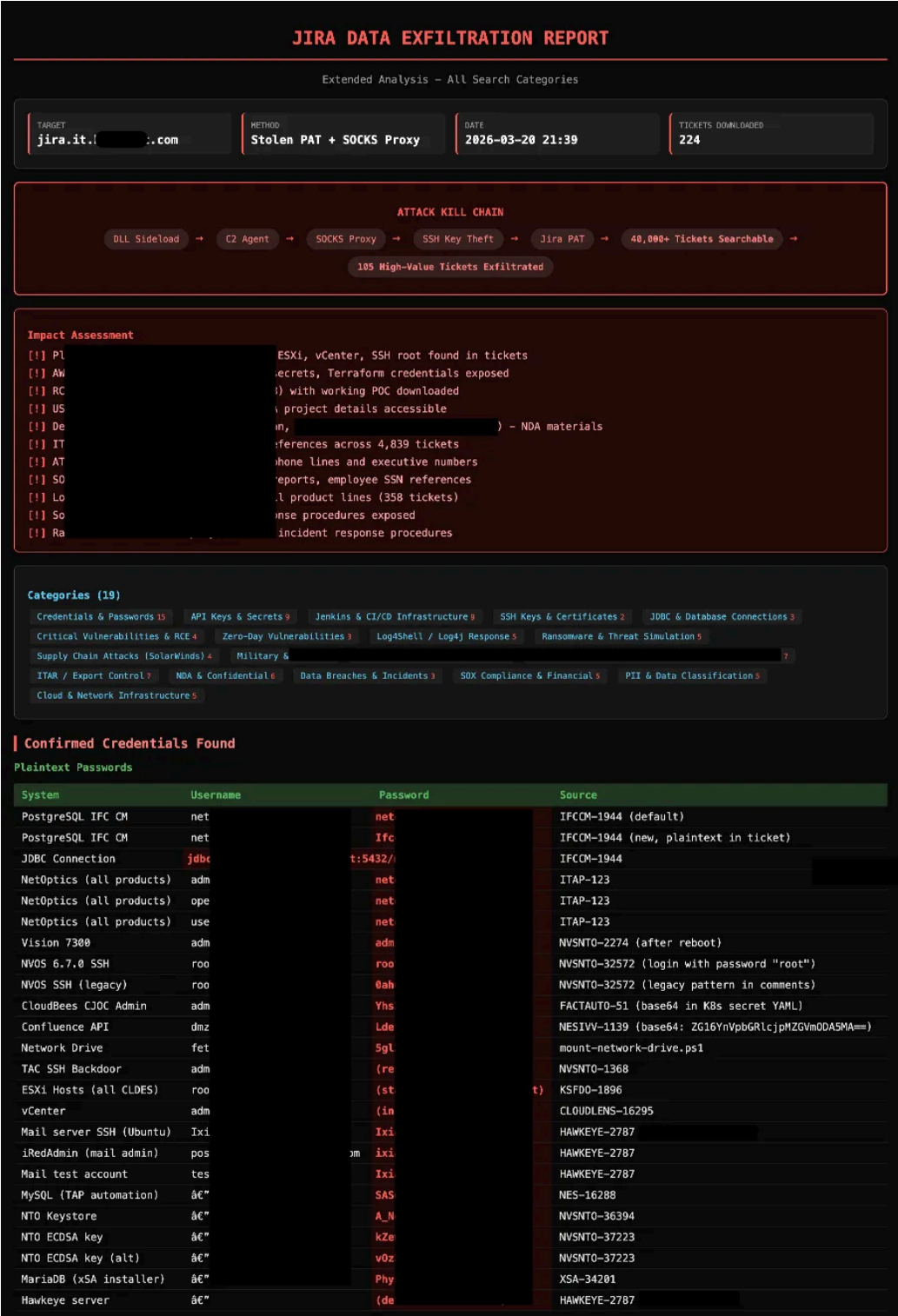


Figure 40. Jira data exfiltration report summarizing 224 downloaded tickets and 19 data categories

The **224 Jira tickets were organized into 19 categories**, including credentials, vulnerabilities, defense-related information, and infrastructure.

Some attachments contained **penetration testing reports, confidential PoC code, and internal technical documents associated with external organizations.**

Indications were also found that **PowerShell scripts were used to collect Jira system information and search Jira tickets for shared credentials and infrastructure-related data**, including information associated with cloud infrastructure, VPN environments, source code management, and other enterprise systems.

| AWS IAM Access Keys (Account:) | | | | |
|---|--|----------|----------|-----------------|
| Access Key ID | | IAM User | Status | Source |
| AKIA | | ter | Active | KSFDD-1783 |
| AKIA | | ter | Active | LABSDEVOPS-914 |
| AKIA | | ter | Active | KSFDD-793 |
| AKIA | | hep | Active | KSFDD-982 |
| AKIA | | kpc | Rotated | LABSDEVOPS-1658 |
| AKIA | | kpc | Rotated | LABSDEVOPS-1658 |
| AKIA | | ter | Inactive | KSFDD-1783 |
| AKIA | | ter | Inactive | KSFDD-1783 |
| AKIA | | ter | Inactive | KSFDD-793 |
| AKIA | | ter | Inactive | LABSDEVOPS-914 |
| AKIA | | ter | Inactive | LABSDEVOPS-914 |
| AKIA | | hep | Inactive | KSFDD-982 |

AWS Account: | Region: us-west-2 | Service: 53 | 3 IAM users: terraform-cloud, heptio-ark, kpc-cicd | Keys stored in: LastPass Vault, ~/.aws/credentials, tfvars

| Type | Value (truncated) | Source |
|---------------------------|-------------------|---|
| RSA Private Key | ----- | FACTAUTO-51 (base64 in K8s secret) |
| JWT Token (ServiceDesk) | eyJ... | KSMNPIS-4645 |
| CloudBees CI Agent Secret | be... | CBCI (nes-rca. (FLT-PC-GNCF53) .com, node |
| Confluence API Key | ZG... | NESIVV-1139 |
| AWS IAM (Jenkins) | ar... | ESQA-5925 |
| Ansible Vault (AES256) | ;l... | Ansible config (t WinRM+Kerberos) .COM, |
| Grafana admin | (r... | KSFDD-2039 |
| asdc-jenkins | (L... | KSFDD-391 |
| Bitbucket (scmPassword) | Qk... | FACTAUTO-51 |
| AWS ThreatSim agent | ey... | THREATSIM-601 (base64 encoded token) |

Search Results Summary (55 Categories)

| Category | Tickets Found |
|-------------------|---------------|
| confidential nda | 4270 |
| api key token | 3066 |
| revenue financial | 1795 |
| default password | 1043 |
| jenkins secrets | 1019 |
| database creds | 658 |
| hardcoded secrets | 381 |
| data breach | 265 |
| vpn credentials | 188 |
| pentest results | 176 |
| ssh root password | 175 |
| vcenter password | 175 |
| aws keys | 105 |
| esxi root creds | 14 |
| cve exploit | 0 |

Downloaded Attachments

| | |
|------|---|
| 2275 | |
| NVSA | .docx (2515.6 KB) |
| NVSA | n testing steps.docx (1312 KB) |
| SYS1 | _modeling_r02.pptx (1599.7 KB) |
| SYS1 | Demo .msg (98 KB) |
| ZDI- | marshal Deserialization of Untrusted Data.txt (14.9 KB) |
| ZDI- | |
| ZDI- | |

Figure 41. Exfiltrated AWS keys, tokens, search results, and Jira attachments identified in the collected data

The findings indicate that the exposed Jira environment contained not only the organization's internal information, but also **technical data shared by customers and partners through support and business activities.**

U.S. Defense & Defense-Industry Data

Among the exfiltrated Jira data, **technical and operational information associated with U.S. defense organizations, defense contractors, and aerospace and defense companies** was identified.

The exposed materials included:

- System and software configurations
- Equipment and system identifiers
- Host and network information
- Authentication and access issues
- Operational logs
- Software license management
- Asset and order management information
- Technical support and troubleshooting records
- Technical requirements
- Access-restricted attachments
- Confidential Proof-of-Concept materials

Records associated with **U.S. defense organizations** included authentication issues, system identifiers, host information, software versions, and technical support logs.

US Vision One: Cannot login to the admin UI and SSH password policies are enabled

US Vision password policies

Description

Customer (End-User): Cl Mac

Partner (if any): N/A

Model: 6 94

Cards installed: N/A

OS version: 5.12.0.12-20220614-034201-e3b696

Reason for Assistance:
Question:

The customer is unable to login to the CLI (via SSH) and the web UI at the sai policy is enabled.
Is this expected behavior?

Figure 42. Jira ticket documenting a U.S. defense-related authentication issue affecting admin UI and SSH access

Large File Admin 2023-02-07

Files ["6: W1_20230112_080226_5-12-0-12_logs_1_of_1.zip"] were uploaded at transfer, with the comment:

"Customer: US
Case 1226380
Hostname: S NV1
IP Address: 150. ,242
System ID: 6: 194
NVOS Version: 5.12.0.12

Figure 43. Uploaded system logs containing hostname, IP address, system identifier, and software version

Other records covered **software license issuance and equipment account management.**

| Item Number | System Name | Serial Number | Item Description | Order Number | Order PO Number | Instance Number |
|-------------|-------------|---------------|------------------|--------------|-----------------|-----------------|
| 954-4 | *AN | 202* | vTAP, P | 33 | 45 | *1592 |
| 991-0 | *AN | 202* | 6322 Ixia Visi | 33 | 45 | *1592 |
| 991-2 | *AN | 202* | Vision C | 33 | 45 | *1592 |
| 991-2 | *AN | 202* | Vision C | 33 | 45 | *1592 |
| 991-3 | *AN | 202* | 420 Ixia Visi | 33 | 45 | *1592 |
| 991-3 | *AN | 202* | 420 Ixia Visi | 33 | 45 | *1592 |
| 992-0 | *AN | 202* | SSC Ixia Visi | 33 | 45 | *1592 |
| 993-0 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 993-0 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 993-0 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 993-0 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 993-0 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 993-1 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 993-1 | *AN | 202* | Ixia Visi | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | NTO 10 | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | NTO 10 | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | NTO 10 | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | NTO 10 | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | NTO 10 | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | NTO 10 | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | XVK Ixia Visi | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | XVK Ixia Visi | 33 | 45 | *1592 |
| 995-0 | *AN | 202* | XVK Ixia Visi | 33 | 45 | *1592 |

Figure 44. Asset and license records including equipment identifiers, order information, and software licenses

Materials associated with **defense contractors and aerospace organizations** included equipment configurations, system logs, asset records, and **technical cooperation discussions**. **Confidential PoC documents and access-restricted attachments** were also identified.

Global Healthcare Company (laC)

The analyzed server also contained **credential and infrastructure-related information associated with a global healthcare company's Infrastructure-as-Code (laC) Platform**.



Figure 45. Security assessment report for the healthcare organization's IaC platform, covering AWS, Azure AD, Bitbucket, and infrastructure configuration environments

The identified information was associated with multiple enterprise environments, including:

- AWS and cloud storage
- Production-related databases
- Bitbucket
- Infrastructure configuration management
- Development and production environments

Administrator-level AWS Access Key information was identified along with information associated with cloud storage resources and databases.

1. AWS Credentials in Plaintext .env File

CRITICAL

AWS access keys with AccountAdministrator privileges were found stored in plaintext in `C:\projects\app\.env` on the DevOps workstation.

| Parameter | Value |
|-------------------|---|
| Access Key ID | AKIA |
| Secret Access Key | 2zj2 |
| Account ID | 1499 |
| IAM User | itx- |
| User Path | /pro |
| Created | 2022-09-12 |
| Tag | "jbpb-jenkins - IaC team local execution" |

Group Memberships

| Group | Access Level |
|-------------|-------------------|
| AccountAdm | Full Admin |
| AccountMon | Monitoring / Read |
| AccountMeml | Base access |
| AccountMeml | Owner level |

S3 Bucket Access

93 S3 buckets enumerated, 83 accessible at bucket level. Key buckets include:

| Bucket | Category | Risk |
|---------|--------------------|----------|
| 1499037 | Audit Logs | CRITICAL |
| asx-jhy | Terraform State | CRITICAL |
| itx-cew | SAP HANA Backup | CRITICAL |
| itx-cew | DB Backups | HIGH |
| itx-cew | Kubernetes Backups | HIGH |
| itx-cew | Terraform Configs | HIGH |

Note: Resource-based bucket policies (explicit deny) prevented content listing/download. This is a positive security control, but the IAM-level access remains overprivileged.

```

$ aws sts get-caller-identity
{
  "UserId": "AIDA",
  "Account": "149",
  "Arn": "arn:aws
  
```

Impact: Full administrative access to AWS account 149. Attacker could provision infrastructure, modify IAM policies, access CloudTrail logs, and potentially pivot accounts via cross-account roles.

Recommendation: Rotate AWS access keys immediately. Migrate to IAM Roles with temporary credentials (STS AssumeRole). Store secrets in AWS Secrets Manager or . Implement IP-based access restrictions on IAM policies.

2. Production Database Credentials Exposed

CRITICAL

Production MSSQL RDS database credentials stored in plaintext in `.env` file.

| Parameter | Value |
|-----------|-------------|
| Host | cloudx-iac- |
| Database | IaC |
| Username | iac |
| Password | |

Impact: Direct access to production IaC database containing infrastructure provisioning data, user accounts, and configuration metadata.

Recommendation: Rotate database password. Use IAM database authentication. Restrict network access to known CIDR ranges only. Store credentials in Secrets Manager with automatic rotation.

Figure 46. Administrator-level AWS access keys and production database credentials identified in a plaintext environment file

Additional materials contained authentication and development environment credentials, infrastructure configuration information, cross-account cloud data, VPC endpoint information, and certificate-related information.

An environment configuration file contained **12 pieces of entitlement information**, and a separate summary documented the access scope associated with the identified administrator-level cloud credentials.

9. Insecure Credential Storage Pattern

MEDIUM

All credentials (12+ secrets) are stored in a single plaintext `.env` file without encryption, access controls, or secret rotation policies.

Recommendation:

- Migrate all secrets to [REDACTED]
- Implement automatic secret rotation (90-day maximum lifetime)
- Use environment-specific secret stores (separate DEV/PROD)
- Add `.env` to `.gitignore` and scan repositories for leaked secrets
- Implement pre-commit hooks to prevent secret commits

10. PermissionsBoundary Provides Partial Mitigation

INFO

Despite AccountAdministrator group membership, AWS PermissionsBoundary policy (`vpco` [REDACTED]) effectively blocks S3 content access operations. This is a positive security control that limited the impact of the credential exposure.

| Allowed | Blocked |
|------------------------------------|--------------------------------------|
| <code>sts:GetCallerIdentity</code> | <code>s3:ListBucket</code> (content) |
| <code>iam:GetUser</code> | <code>s3:GetObject</code> |
| <code>iam:ListGroupsForUser</code> | <code>s3:GetBucketPolicy</code> |
| <code>s3:ListAllMyBuckets</code> | <code>s3:PutObject</code> |

Recommendation: Maintain and strengthen PermissionsBoundary policies. Apply similar boundaries to all service accounts. Consider adding IP-based conditions to further restrict access.

Figure 47. Environment file containing more than 12 secrets, with a summary of access scope for the exposed cloud credentials

The identified materials differed from previously disclosed leaked data associated with the same organization, **suggesting the possibility of separate or additional intrusion activity.**

Conclusion

Analysis of the server identified **eb.sys**, the **TukTuk C2 framework**, **DLL sideloading configurations**, and **EDR neutralization research**, showing an association with The Gentlemen. The **complete TukTuk v2 project**, with no known prior public disclosure, was found alongside step-by-step training materials.

The server also contained **data assessed to have been exfiltrated from a global technology company**, including information related to its **U.S. defense and defense-industry** customers, as well as separate credentials associated with a **global healthcare company's IaC platform**.

Relevant findings have been shared with the affected organizations and institutions.