The Wreath Network Penetration Test

Client – Thomas Wreath

Penetration Tester – Siddharth Ray Chaudhuri

Table of Contents

Confidentiality Statement

This document is the exclusive property of Thomas Wreath and Siddharth Ray Chaudhuri. This document contains proprietary and confidential information. Duplication, redistribution, or use in whole or part, in any form, requires consent of Thomas Wreath and Siddharth Ray Chaudhuri.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside that period.

Time-limited test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not in changes or modifications made outside that period.

Time-limited engagements do not allow for a full evaluation of all the security controls. Siddharth Ray Chaudhuri prioritized the assessment to identify the weakest security controls that an attacker would exploit. Siddharth Ray Chaudhuri recommends conducting similar assessments on an annual basis by internal basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

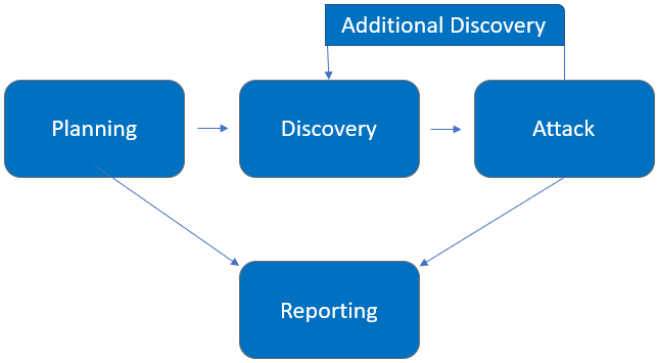
Name	Title	Contact Information
Thomas Wreath	**Client**	**Email: me@thomaswreath.thm**
Siddharth Ray Chaudhuri	**Penetration Tester**	**Email: siddharthraychaudhuri@gmail.com**

Assessment Overview

From 14th July, 2025 to 19th July, 2025, Thomas Wreath engaged Siddharth Ray Chaudhuri to evaluate the security posture of its infrastructure compared to industry best practices that include a penetration test of both the external and internal networks (combined named as the Wreath Network. All testing performed is based on the NIST SP 800-115 Technical Guide to the Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks.

Phases of penetration testing activities include the following:

- **Planning** – Customer goals are gathered and rules of engagement obtained
- **Discovery** – Perform scanning and enumeration to identify potential vulnerabilities, weak areas and exploits.
- **Attack** – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access
- **Reporting** – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

The penetration test encompassed the following components:

- **External Penetration Testing (EPT):**
  Targeted the organization's publicly accessible infrastructure. This phase included enumeration, vulnerability identification, and exploitation of external-facing services to gain initial access.
- **Internal Penetration Testing (IPT):**
  Simulated a scenario where an attacker had gained a foothold within the internal network. This included privilege escalation, lateral movement, pivoting through internal systems, and extraction of sensitive credentials and files.
- **Persistence and Post-Exploitation:**
  Evaluated the extent to which an attacker could maintain access and further compromise the network, including the creation of user accounts, installation of backdoors, and service abuse.
- **Tool Utilization:**
  A combination of manual testing techniques and open-source tools such as Nmap, Mimikatz, Evil-WinRM, sshuttle, Chisel, and PowerShell Empire were employed to ensure thorough assessment coverage.
- **Adherence to Frameworks and Standards:**
  The engagement was conducted in alignment with industry standards, including the MITRE ATT&CK framework, OWASP Top 10, and NIST SP 800-115.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: **Likelihood** and **Impact**.

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential of a vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
Wreath Network Penetration Test	**10.200.180.200**

Scope Exclusions

As per client request, Siddharth Ray Chaudhuri did not perform any of the following attacks during the test:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Thomas Wreath.

Client Allowances

Thomas Wreath provided Siddharth Ray Chaudhuri the following allowances:

Access to the IP of a public-facing(discoverable outside the subnet) host.

Executive Summary

A comprehensive external and internal penetration test was conducted against the Wreath Network to assess the security posture of its infrastructure, services, and internal systems. The assessment followed a goal-oriented methodology, simulating real-world adversarial behavior, with the objective of identifying vulnerabilities, misconfigurations, and the potential impact of an attacker gaining unauthorized access to the network.

Scope and Methodology

The test began with an **External Penetration Test (EPT)** targeting internet-exposed assets. Upon successfully exploiting a known **Webmin Remote Code Execution vulnerability (CVE-2019-15107)** on the external host prod-serv, full root shell access was obtained. The attacker pivoted into the internal network using **SSH tunneling**, reconnaissance tools, and post-exploitation techniques. From there, an **Internal Penetration Test (IPT)** was conducted with escalating levels of access, ultimately resulting in full domain compromise of internal systems, including git-serv and wreath-pc.

Key Findings

- Multiple **critical vulnerabilities** were exploited, including remote code execution, insecure service configurations, and privilege escalation paths.
- **Credential reuse, weak passwords**, and exposed private keys were identified, enabling lateral movement and persistent access.
- The attacker was able to perform **multi-level pivoting**, leveraging poor network segmentation and improperly filtered ports to access deep internal assets.
- Post-exploitation techniques such as **Mimikatz, secretsdump**, and **evil-winRM** enabled the extraction of hashes, escalation to SYSTEM privileges, and full control of target systems.
- **Web application flaws**, such as improper file extension filtering and metadata injection, were exploited to gain web shells and reverse shells.

Impact

The test demonstrated that an external attacker could penetrate the network perimeter, gain a foothold, and traverse internal systems with increasing privileges. The final outcome included:

- SYSTEM-level shell access on internal hosts
- Exfiltration of SAM and SYSTEM hives
- Offline hash dumping of both user and administrator credentials

Conclusion

The penetration test revealed **serious security gaps** across the Wreath Network's external and internal environments. If left unaddressed, these issues could allow real attackers to compromise critical systems, access sensitive data, and maintain long-term persistence within the network. Immediate remediation of the identified vulnerabilities is strongly recommended, along with a review of privilege management, segmentation, monitoring, and web application security controls.

Tester Notes and Recommendations

This penetration test was conducted to assess the security posture of the Wreath Network, simulating a real-world adversarial scenario targeting externally exposed and internally connected systems.

The test successfully demonstrated how an attacker could exploit a single exposed vulnerability to gain a foothold in the network, escalate privileges, and compromise multiple internal systems.

The initial access vector was a publicly reachable Webmin service that had not been patched against a known remote code execution vulnerability. This reflects a common security oversight — failing to keep internet-facing services updated with the latest security patches. From this foothold, the tester was able to pivot into the internal network using simple tunneling techniques, leveraging the compromised external server as a relay.

Once inside, weak internal configurations made it possible to escalate privileges and access critical systems. For example, poor password hygiene and local administrator password reuse enabled lateral movement without significant resistance. Additionally, service misconfigurations such as unrestricted file uploads, WDigest credential caching, and disabled SMB signing further lowered the barrier for compromise. These are not merely technical oversights but systemic weaknesses in hardening practices.

No significant intrusion detection or monitoring controls were observed during the test. This allowed the attacker to operate freely without detection, elevating the risk of persistent compromise. The presence of powerful tools like Mimikatz and remote access utilities on the system without any response indicates a lack of defensive depth and incident readiness.

However, it is also worth noting some strengths. The network had minimal external exposure, and the user's personal PC appeared to be well protected and segregated from the more vulnerable internal systems. The use of version control and Git workflows shows an awareness of basic development best practices.

Moving forward, it is recommended that the environment adopt stricter patch management practices, enforce a strong password policy, segment the internal network, and disable insecure legacy features. Additionally, basic intrusion detection and alerting mechanisms should be put in place to ensure that future compromises, if attempted, are detected in time.

Overall, this engagement highlighted how small oversights in basic security hygiene can quickly escalate into complete system compromise — but it also offers a clear roadmap for securing the environment more effectively.

Key Strengths and Weaknesses of the Wreath Network (Security Perspective)

Strengths

- **Network Filtering on Internal Host**: The wreath-pc (10.200.180.100) host was configured with network filtering, limiting direct communication beyond the internal network. This measure provided an initial layer of protection against unauthorized access, though it was ultimately bypassed through pivoting techniques.
- **Limited External Attack Surface**: Only one external host (prod-serv, 10.200.180.200) was directly accessible from outside the network, reducing the number of entry points for potential attackers.

Weaknesses

- **Unpatched Critical Vulnerabilities**: Exploitable vulnerabilities in Webmin (CVE-2019-15107) and GitStack (EBD-43777) on prod-serv and git-serv allowed attackers to gain root and administrative access, respectively, posing a severe security risk.
- **Weak Access Controls**: Insufficient privilege management enabled the creation of unauthorized accounts with elevated permissions on git-serv, facilitating persistence and privilege escalation.
- **Insecure Credential Management**: Weak passwords (e.g., for user "Thomas") and exposed SSH keys in default locations allowed unauthorized access and credential dumping, undermining authentication security.
- **Vulnerable Web Application**: The /resources image uploader on wreath-pc lacked proper file validation, enabling attackers to upload and execute malicious code via double-extension exploits.

- **Misconfigured Services and Directories**: An unquoted service path and writable service directory on wreath-pc allowed attackers to escalate privileges to SYSTEM level, indicating poor service hardening.
- **Insufficient Network Segmentation**: Network filtering was bypassed through pivoting from prod-serv and git-serv to wreath-pc, demonstrating inadequate isolation of internal systems and enabling lateral movement.
- **Exposure of Sensitive System Files**: Unrestricted access to SAM and SYSTEM files on wreath-pc allowed extraction and offline dumping of credential hashes, risking widespread system compromise.

Vulnerability Summary & Report Card

External Penetration Test Findings

1	3	`4	1	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
External Penetration Test		
EPT-001: Webmin Remote Code Execution (CVE-2019-15107)	Critical	Update Webmin to patched version. Implement WAF , restrict access via IP Whitelisting
EPT-002: Exposed Private SSH Key in Default location ~/.ssh/id_rsa	High	Rotate all keys. Implement strict key management. Use encrypted key storage.
EPT-003: Externally Accessible SSH with Tunnel Capability	High	Restrict SSH access using IP-based firewall rules.
EPT-004: Insecure File Upload to /tmp Directory on prod-serv	Moderate	Limit executable permission in temporary directories.
EPT-005: Internal Network Reconnaissance via Compromised Host	Moderate	Segment DMZ and internal network firewalls.
EPT-006: Open Internal Ports Accessible from prod-serv	Moderate	Review and enforce strict Internal network access controls
EPT-007: HTTP Directory Listing Enabled on Port 80	Low	Disable Directory Listing and improve error handling configurations
EPT-008: Gitstack Admin Interface	Moderate	Restrict Access to Admin Interfaces to Internal IPs or VPNs.
EPT-009: Gitstack RCE-EBD-43777	High	Patch GitStack to the latest version.

Internal Penetration Test Findings

5	10	4	0	0
Critical	High	Moderate	Low	Informational

Internal Penetration Test		
Findings	Severity	Recomendations
IPT-001: Unauthorized Local User Creation with Elevated Privileges	High	Use LAPS or centralized identity providers to avoid unmanaged admin accounts
IPT-002: Remote Management Access via WinRM and RDP	High	Restrict WinRM and RDP access to authorized subnets only.
IPT-003: Credential Dumping via Mimikatz on git-serv	Critical	Disable WDigest unless required.
IPT-004: Pass-the-Hash Attack for Admin Shell	High	Utilize local Administrator password Randomization (LAPS).
IPT-005: Arbitary File Write to Documents Directory	Moderate	Apply least privilege permissions to sensitive folders.
IPT-006: Tool Sharing via evil-winRM	Moderate	Inspect remote PS sessions for abnormal behaviour
IPT-007: Internal Port Scanning via Powershell	Moderate	Monitor Powershell activity. Apply endpoint firewall rules.
IPT-008: wreath-pc Exposed Critical Port	Moderate	Implement strict ACLs and isolate sensitive hosts
IPT-009: Double Pivot to Internal Host via SSH and Chisel	High	Apply host based firewalls and control inter-host connectivity
IPT-010: Weak Password on “Thomas” Account	High	Enforce strong password policies and perform periodic audits.
IPT-011: Insecure File Extension Validation in Upload Endpoint	High	Implement robust MIME-type and content-based upload validation
IPT-012: Metadata-based PHP Exploit via Image Upload	Crtical	Scan uploads with static/dynamic analyzers
IPT-013: Web Shell Access via GET Parameter	High	Audit server for unauthorized logic.

IPT-014:Reverse shell as Thomas via Backdoor	High	Rotate all user credentials. Investigate shell activity.
IPT-015: Unquoted Service Path in System Explorer	High	Correct path quotations in service definitions
IPT-016: Writable Service Directory	High	Enforce NTFS permissions on service paths
IPT-017: SYSTEM Shell via Malicious Service Binary	Critical	Monitor for abnormal service restarts
IPT-018: SAM and SYSTEM Hive Exfiltration	Critical	Disable unnecessary shares. Restrict access to registry files.
IPT-019: secretsdump used to extract Password Hashes	Critical	Salt and hash credentials. Rotate all credentials post-compromise

TECHNICAL FINDINGS

External Penetration Test findings

Finding EPT-001: Webmin RCE – CVE-2019-15107 (Critical)

Description:	The external host prod-serv was found to be running a vulnerable version of Webmin. Exploiting CVE-2019-15107 allowed unauthenticated remote command execution with root privileges.
Risk:	Likelihood: – High – The vulnerability is publicly known and easily exploitable with minimal prerequisites. Impact: Critical – Full system compromise of an externally accessible server.
System:	prod-serv (10.200.180.200)
Tools Used:	Nmap, curl, chmod, CVE-2019-15107.py
References:	https://github.com/squid22/Webmin_CVE-2019-15107 https://www.exploit-db.com/exploits/47293

Evidence:

```
┌──(kali㉿kali)-[~/thm/wreath_network/wreath_revise]
└─$ nmap -T 4 10.200.180.200 -p 1-15000
Starting Nmap 7.94 ( https://nmap.org ) at 2025-07-14 06:19 EDT
Nmap scan report for 10.200.180.200
Host is up (0.18s latency).
Not shown: 14735 filtered tcp ports (no-response), 260 filtered tcp ports (host-unreach)
PORT      STATE  SERVICE
22/tcp    open   ssh
80/tcp    open   http
443/tcp   open   https
9090/tcp  closed zeus-admin
10000/tcp open   snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 255.60 seconds

┌──(kali㉿kali)-[~/thm/wreath_network/wreath_revise]
└─$
```

Figure 1: Nmap scan of external host prod-serv

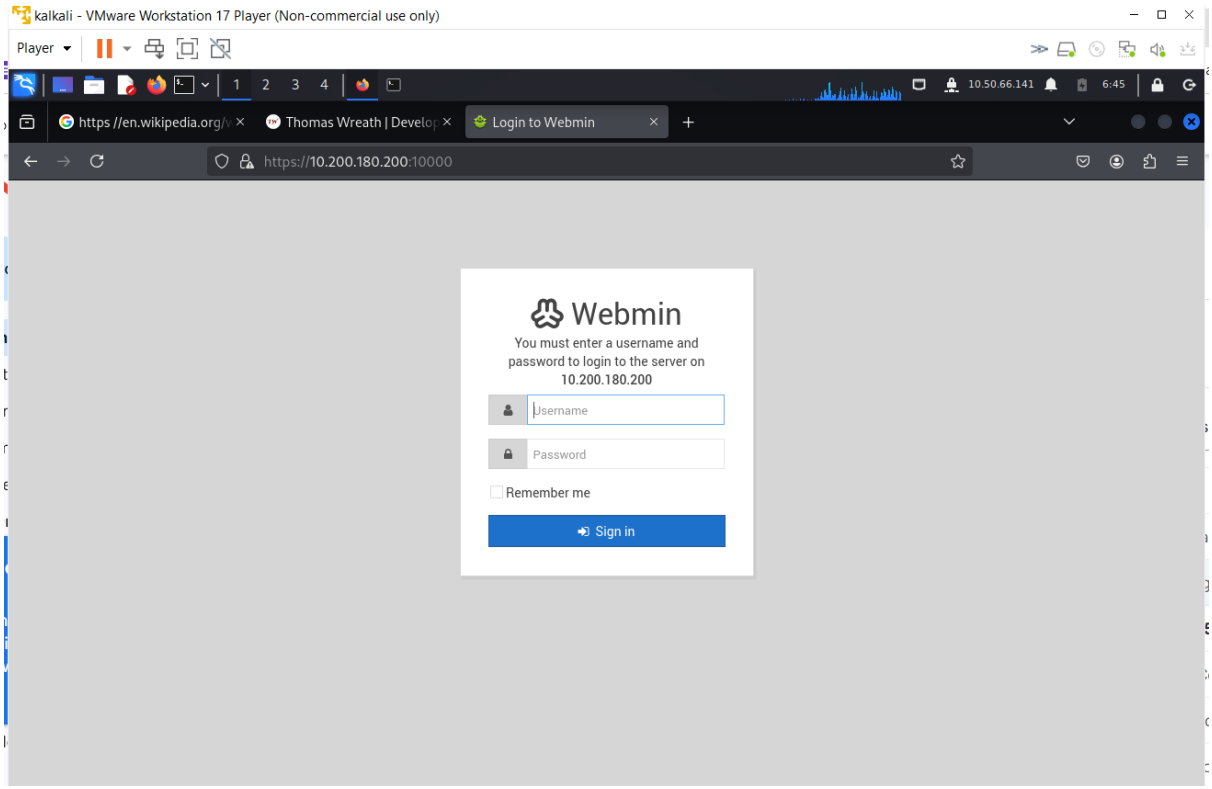


Figure 2: Webmin log-in portal found

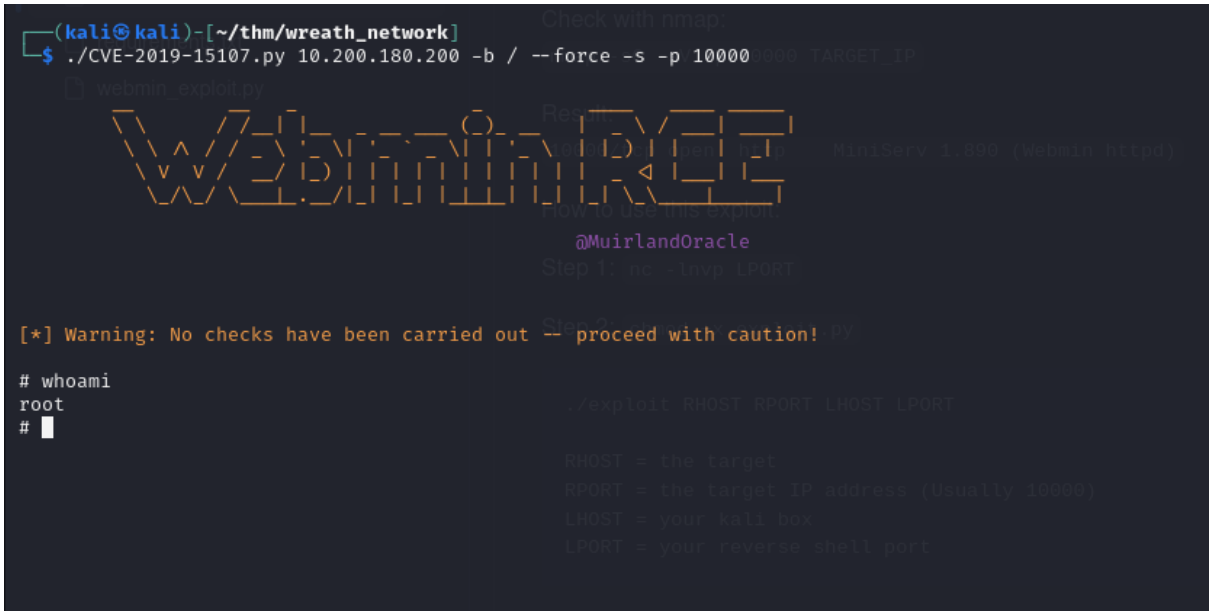


Figure 3: shell of prod-serv by Webmin RCE

Remediation: Update Webmin to a patched version. Implement Web Application Firewalls (WAF) and restrict access to Webmin via IP whitelisting

Finding EPT-002: Exposed Private SSH Key in Default Location ~/.ssh/id_rsa (High)

Description:	A private SSH Key was discovered in the default user directory on prod-serv. Such keys can be used to authenticate to other systems without a password
Risk:	Likelihood: – High – Private keys in predictable locations are easily discoverable and exploitable. Impact: High – Lateral movement and unauthorized access within internal network.
System:	prod-serv (10.200.180.200)
Tools Used:	find, cat, ssh
References:	

Evidence:

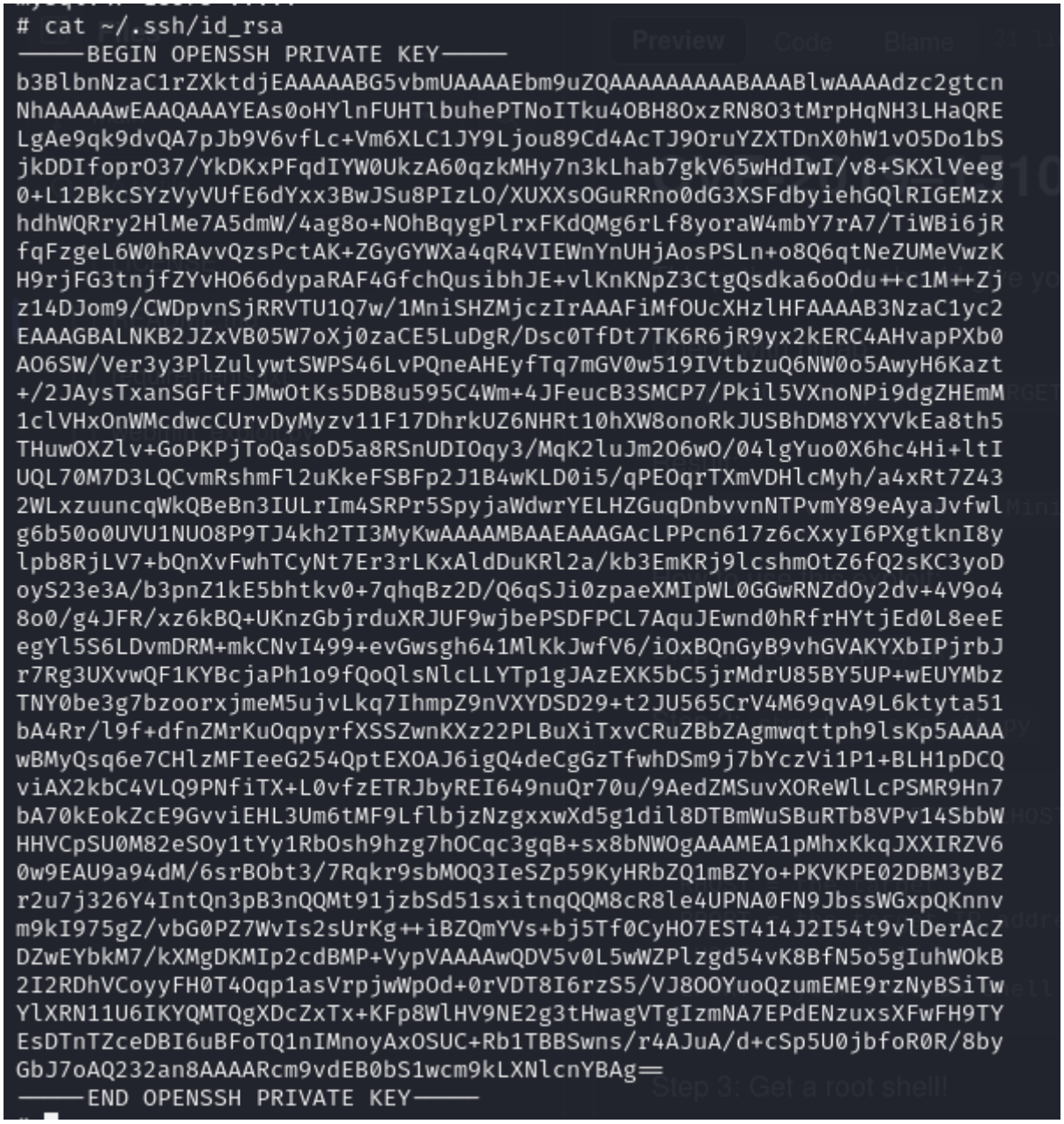


Figure 4: id_rsa key of prod-serv discovered

Remediation: Rotate all affected keys. Implement strict SSH key management policies. Use encrypted key storage and audit access permissions.

Finding EPT-003: Externally Accessible SSH with Tunnel Capability (High)

Description:	Port 22 was found open and accessible on prod-serv, allowing remote SSH access. Combined with stolen keys, this could be used to tunnel into internal network via tools like sshuttle.
Risk:	Likelihood: – High – SSH access combined with key compromise provides reliable attack vector. Impact: High – Network segmentation bypass and covert tunnelling.
System:	prod-serv(10.200.180.200), 10.200.180.200/24 subnet
Tools Used:	ssh, sshuttle
References:	

Evidence:

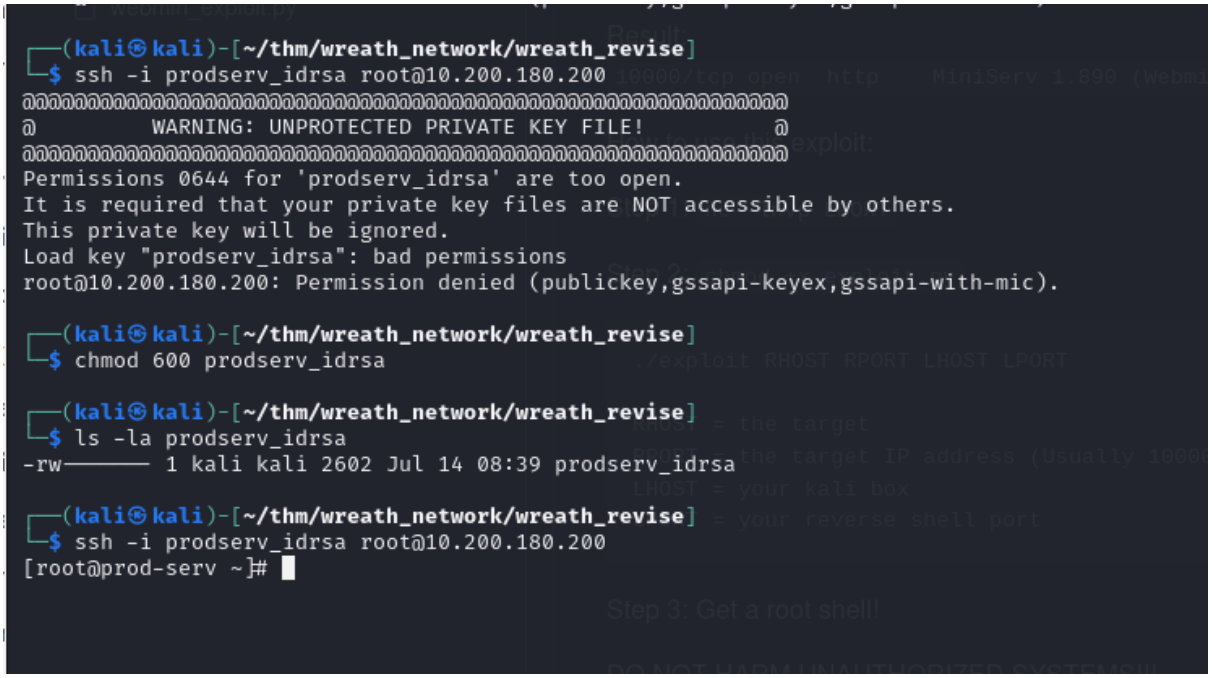


Figure 5:ssh connetion to prod-serv is successful

Figure 6:sshuttle subnet-wide tunnel, by using ssh creds

Remediation: Restrict SSH access using IP-based firewall rules. Monitor and alert on outbound SSH traffic from sensitive zones.

Finding EPT-004: Insecure File Upload to /tmp Directory on prod-serv (Moderate)

Description:	Arbitary binaries (eg., nmao, socat, chisel) were successfully transferred to /tmp on prod-serv.
Risk:	Likelihood: – Medium – Attack depends on prior access but lacks execution restrictions Impact: Moderate – Facilitates internal reconnaissance and proxying tools from compromised hosts.
System:	prod-serv (10.200.180.200)
Tools Used:	python3 http server, curl
References:	

Evidence:

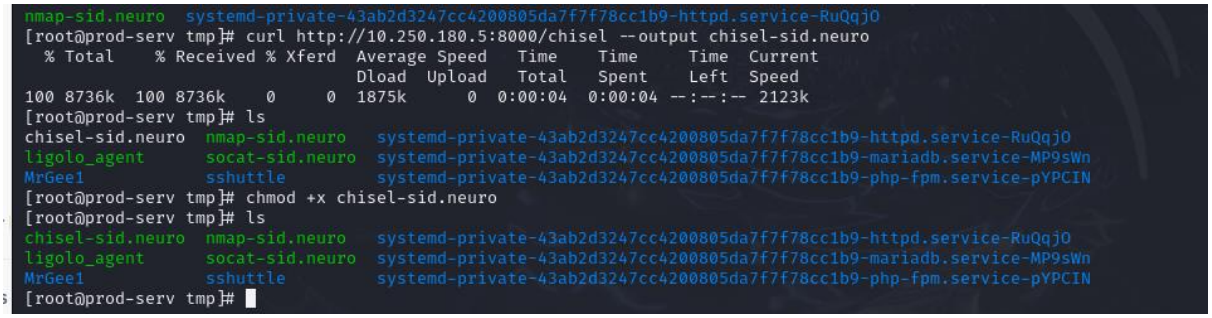


Figure 7: Planting binaries of nmap,socat and chisel in tmp dir of prod-serv

Finding EPT-005: Internal Network Reconnaissance via Compromised Host (Moderate)

Description:	From prod-serv, the attacker scanned the internal network and discovered hosts git-serv and wreath-pc
Risk:	Likelihood: – Moderate – Requires prior compromise but easily executed in flat networks. Impact: Moderate – Revealed internal attack surface due to flat network design.
System:	Internal network via prod-serv(10.200.18.200)
Tools Used:	Nmap binary
References:	

Evidence:

```
[root@prod-serv tmp]# ./nmap-sid.neuro -sn 10.200.180.200/24

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2025-07-15 09:38 BST
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-180-1.eu-west-1.compute.internal (10.200.180.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.18s latency).
MAC Address: 0A:AE:ED:6E:B8:CF (Unknown)
Nmap scan report for ip-10-200-180-100.eu-west-1.compute.internal (10.200.180.100)
Host is up (0.00031s latency).
MAC Address: 0A:A7:FA:B9:F2:CB (Unknown)
Nmap scan report for ip-10-200-180-150.eu-west-1.compute.internal (10.200.180.150)
Host is up (0.00034s latency).
MAC Address: 0A:4E:31:53:AA:09 (Unknown)
Nmap scan report for ip-10-200-180-250.eu-west-1.compute.internal (10.200.180.250)
Host is up (0.00022s latency).
MAC Address: 0A:DC:6C:52:20:C9 (Unknown)
Nmap scan report for ip-10-200-180-200.eu-west-1.compute.internal (10.200.180.200)
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 4.95 seconds
[root@prod-serv tmp]#
```

Figure 8: nmap scan of Internal network from prod-serv

Remediation: Segment DMZ and internal networks using firewalls. Implement egress filtering from DMZ to internal zones.

Finding EPT-006: Open Internal Ports Accessible from prod-serv (Moderate)

Description:	Ports 80 and 3389 on git-serv were accessible from prod-serv.
Risk:	Likelihood: – Moderate – Access was possible post-compromise and due to lack of internal filtering. Impact: Moderate – Enabled lateral movement into internal systems.
System:	git-serv (10.200.180.150), Internal host accessible via prod-serv
Tools Used:	Nmap binary, nmap with proxychains4, chisel
References:	

Evidence:

```
[root@prod-serv tmp]# ./nmap-sid.neuro 10.200.180.150 -p 1-15000

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2025-07-15 10:00 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-180-150.eu-west-1.compute.internal (10.200.180.150)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00054s latency).
Not shown: 14996 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi
5985/tcp open  wsman
MAC Address: 0A:4E:31:53:AA:09 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 122.68 seconds
[root@prod-serv tmp]#
```

Figure 9:nmap scan of git-serv individually, from prod-serv

Remediation: Restrict – Review and enforce strict internal network access controls.

Finding EPT-007: HTTP Directory Listing Enabled on Port 80

Description:	The web server on git-serv allowed directory names to be visible despite 404 errors.
Risk:	Likelihood: – Low – Requires access to internal HTTP service and is only useful for reconnaissance. Impact: Low – Information disclosure. Could assist in targeted attacks.
System:	git-serv(10.200.180.150)
Tools Used:	Firefox browser, manual, sshuttle
References:	

Evidence:

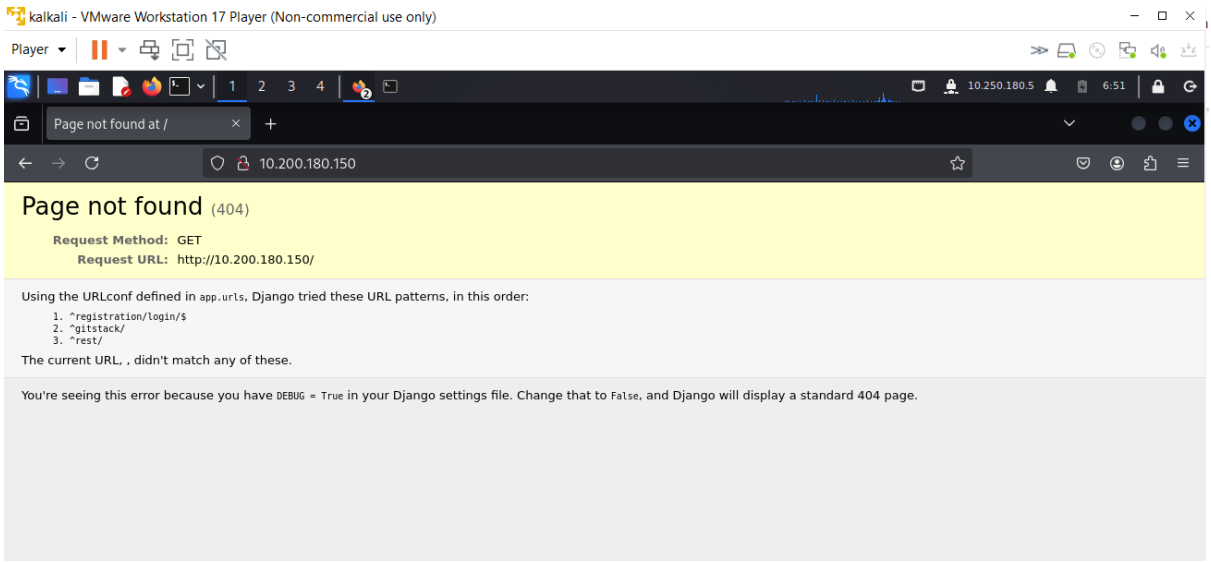


Figure 10: /gitstack discovered

Remediation: Disable Directory listing and improve error handling configurations.

Finding EPT-008: GitStack Admin Interface Exposed (Moderate)

Description:	The /gitstack login portal was exposed on git-serv
Risk:	Likelihood: Moderate - Exposed admin interfaces are frequently targeted and often lead to compromise if unpatched. Impact: Moderate – Increased attack surface and targeted exploitation
System:	git-serv (10.200.180.150)
Tools Used:	sshuttle, Firefox browser, manual
References:	https://stealthbits.com/blog/wdigest-clear-text-passwords-stealing-more-than-a-hash/

Evidence:

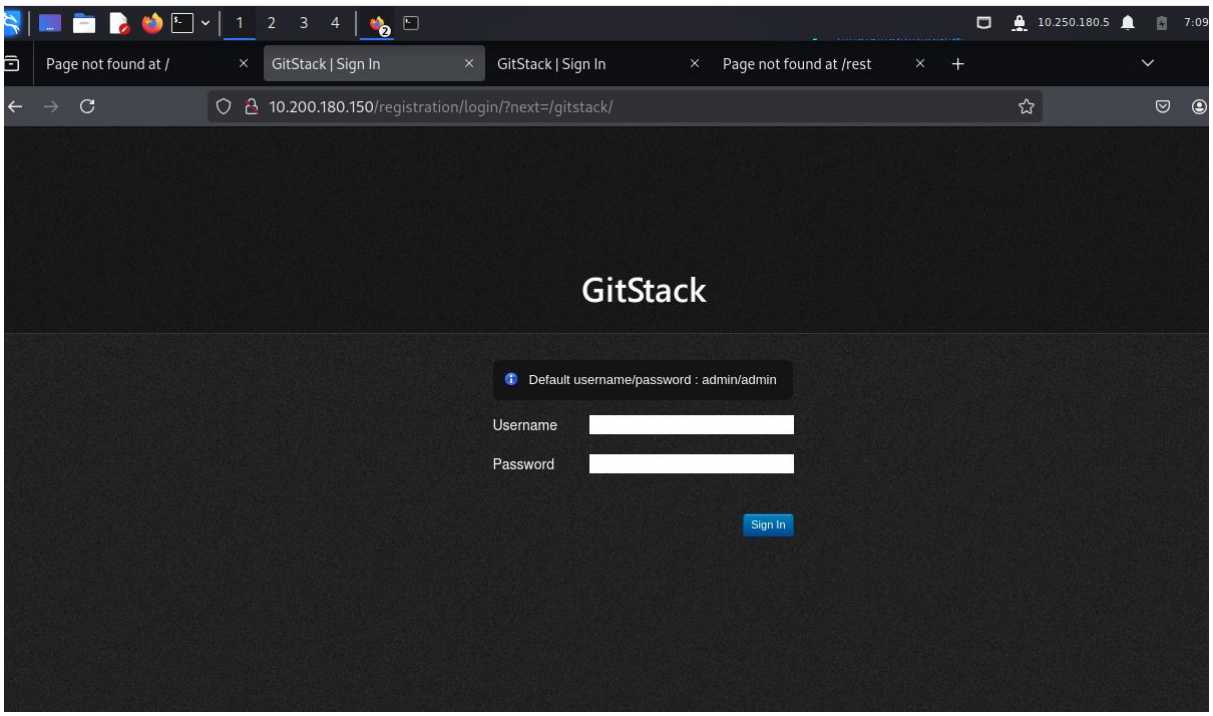


Figure 11: gitstack log in portal

Remediation: Restrict access to admin interfaces to internal IPs or VPN users.

Internal Penetration Test Findings

Finding IPT-001: GitStack RCE EBD-43777 (High)

Description:	Exploited a known GitStack vulnerability to obtain a reverse shell as NT AUTHORITY user.
Risk:	Likelihood: High – Public exploit available; target was unpatched and externally reachable. Impact: High – Unauthorized code execution with high privileges
System:	Git-serv (10.200.180.150)
Tools Used:	Searchsploit, sed, EBD-43777, netcat, socat binary,powershe.exe reverse shell code, url encoder, curl
References:	

Evidence:

```
┌──(kali㉿kali)-[~]
└─$ searchsploit gitstack
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
GitStack - Remote Code Execution                                              | php/webapps/44044.md
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)            | windows/remote/44356.rb
GitStack 2.3.10 - Remote Code Execution                                       | php/webapps/43777.py
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Figure 12:searchsploit results for gitstack

```
┌──(kali㉿kali)-[~/thm/wreath_network]
└─$ ./cleaned_43777.py
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by th
e Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
[+] Get user list
[+] Found user twreath
[+] Web repository already enabled
[+] Get repositories list
[+] Found repository Website
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
Your GitStack credentials were not entered correcly. Please ask your GitStack administrator to give you a username/password and give you access to this reposit
ory. <br />Note : You have to enter the credentials of a user which has at least read access to your repository. Your GitStack administration panel username/pa
ssword will not work.
[+] Execute command
"nt authority\system
"
```

Figure 13:shell of internal host git-serv, EBD-43777 exploit for gitstack

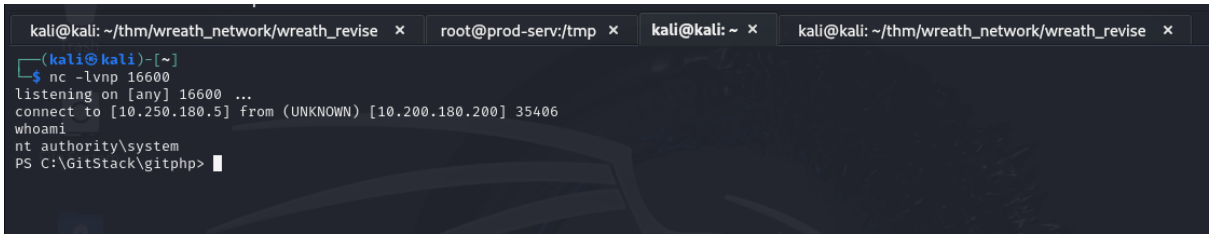


Figure 14: reverse shell of git-serv, as nt authority system

Remediation: Patch GitStack to the latest secure version. Monitor software vulnerability advisories.

Finding IPT-002: Unauthorized Local User Creation with Elevated Privileges(High)

Description:	A user "saptarshi: was created on git-serv and granted Administrator and Remote Management rights
Risk:	Likelihood: High – Achieved via prior RCE; no alerts or controls prevented unauthorized user creation. Impact: High – Persistent backdoor with elevated access
System:	Git-serv (10.200.180.150)
Tools Used:	net user, net localgroup, Windows command line via reverse shell
References:	

Evidence:

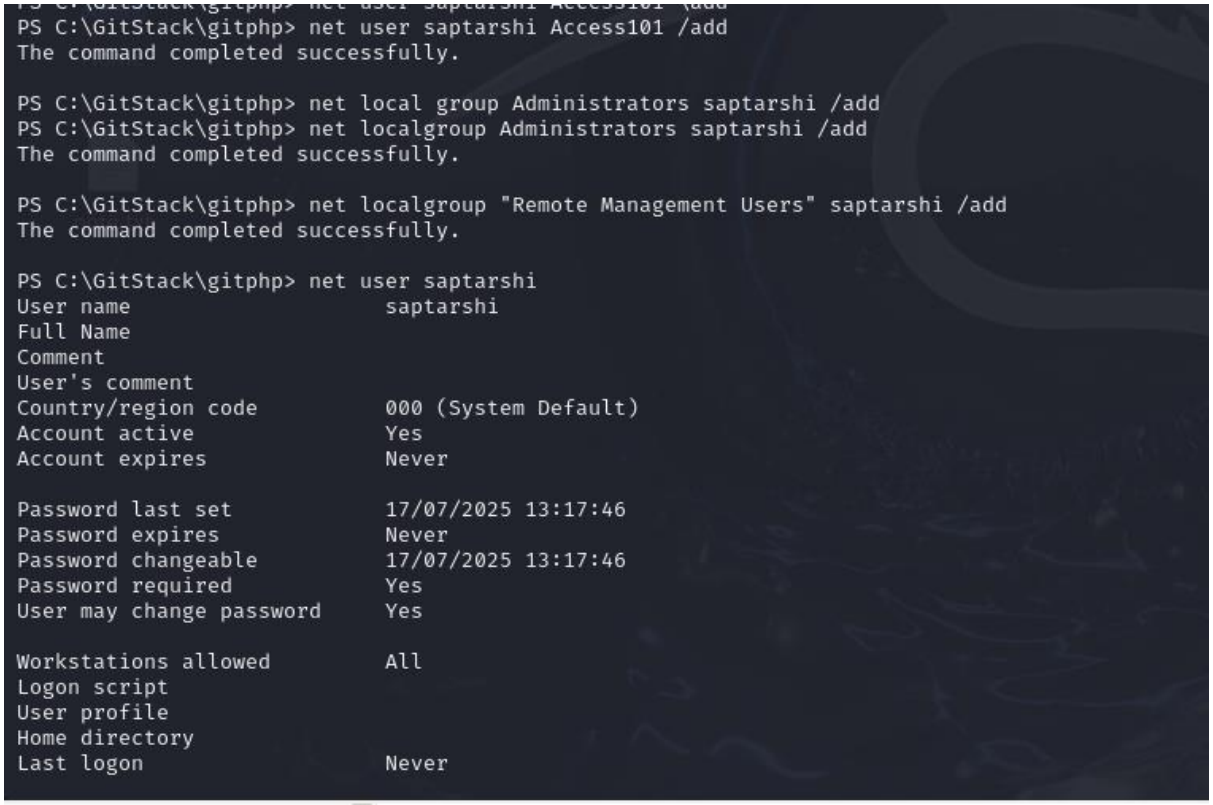


Figure 15: New user"saptarshi" created in git-serv and given local Administrator and rdp privilege

Remediation: Audit local user creation events. Use LAPS or centralized identity providers to avoid unmanaged admin accounts

Finding IPT-003: Remote Management Access via WinRM and RDP (High)

Description:	Remote shells and RDP sessions were successfully initiated using the created user account.
Risk:	Likelihood: High – Access was achieved using valid credentials and standard services open on internal systems. Impact: High – Enabled command execution and GUI-based control.
System:	Git-serv (10.200.180.150)
Tools Used:	Evil-winrm, xfreerdp

References:	

Evidence:

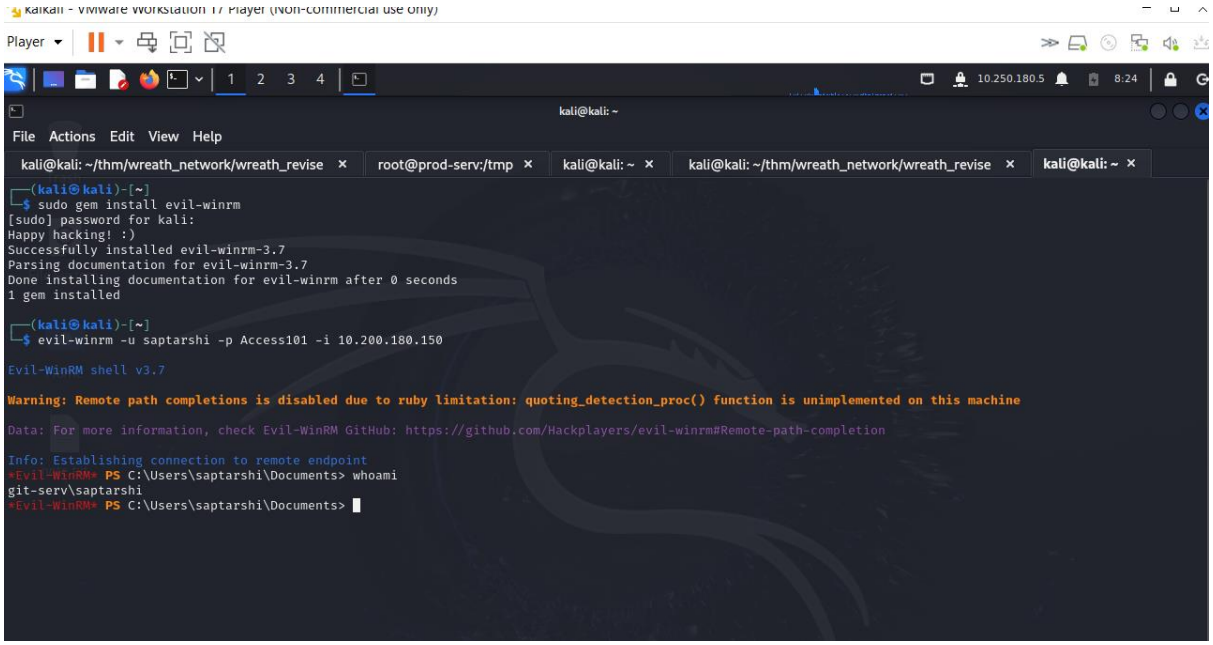


Figure 16:evil-winRM shell as created user "saptarshi"

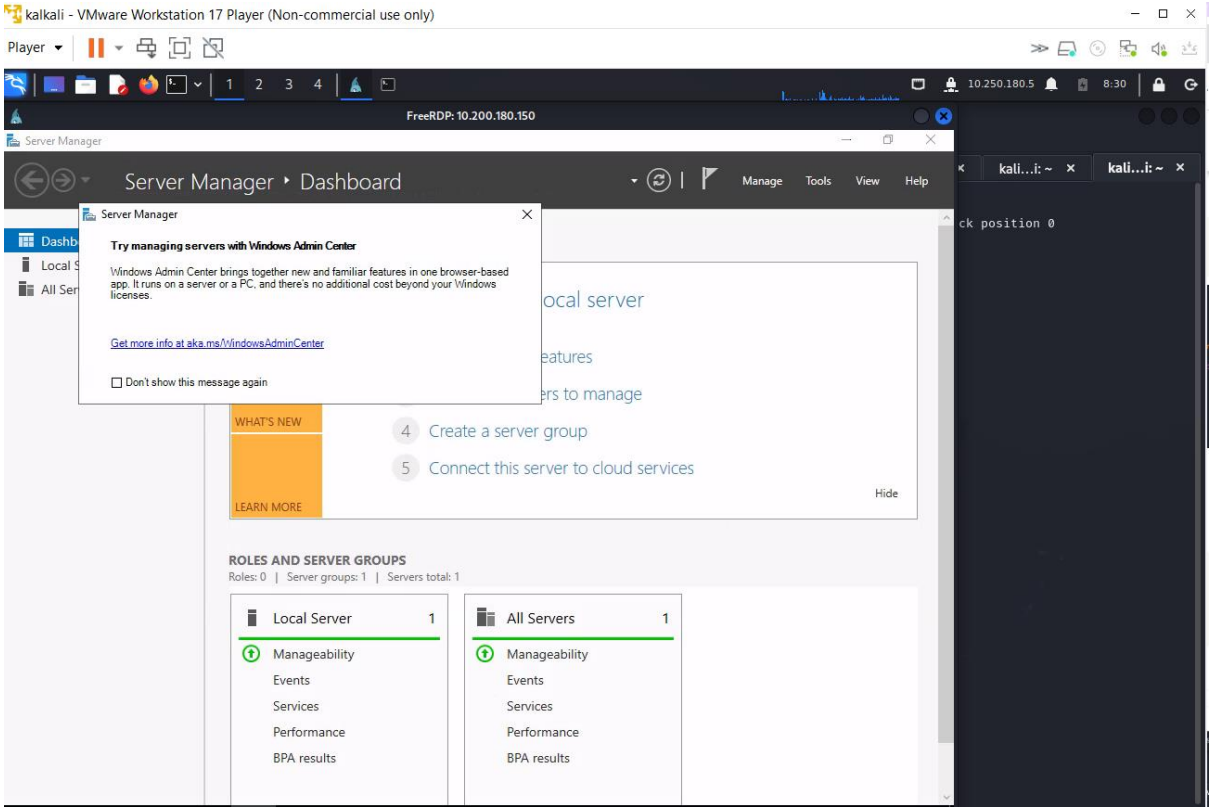


Figure 17:RDP connection into git-serv, as created user "saptarshi"

Remediation: Restrict WinRM and RDP access to authorized admin subnets only. Enforce MFA and session logging.

Finding IPT-004: Credential Dumping via Mimikatz on git-serv (Critical)

Description:	NTLM hashes, including that of the Administrator, were dumped using Mimikatz
Risk:	Likelihood: Very High – Credential dumping with Mimikatz is highly effective in misconfigured Windows environments lacking LSASS protections. Impact: Critical – Complete compromise of authentication secrets
System:	git-serv (10.200.180.150)
Tools Used:	Xfreerdp share, Mimikatz, Powershell(target)
References:	

Evidence:

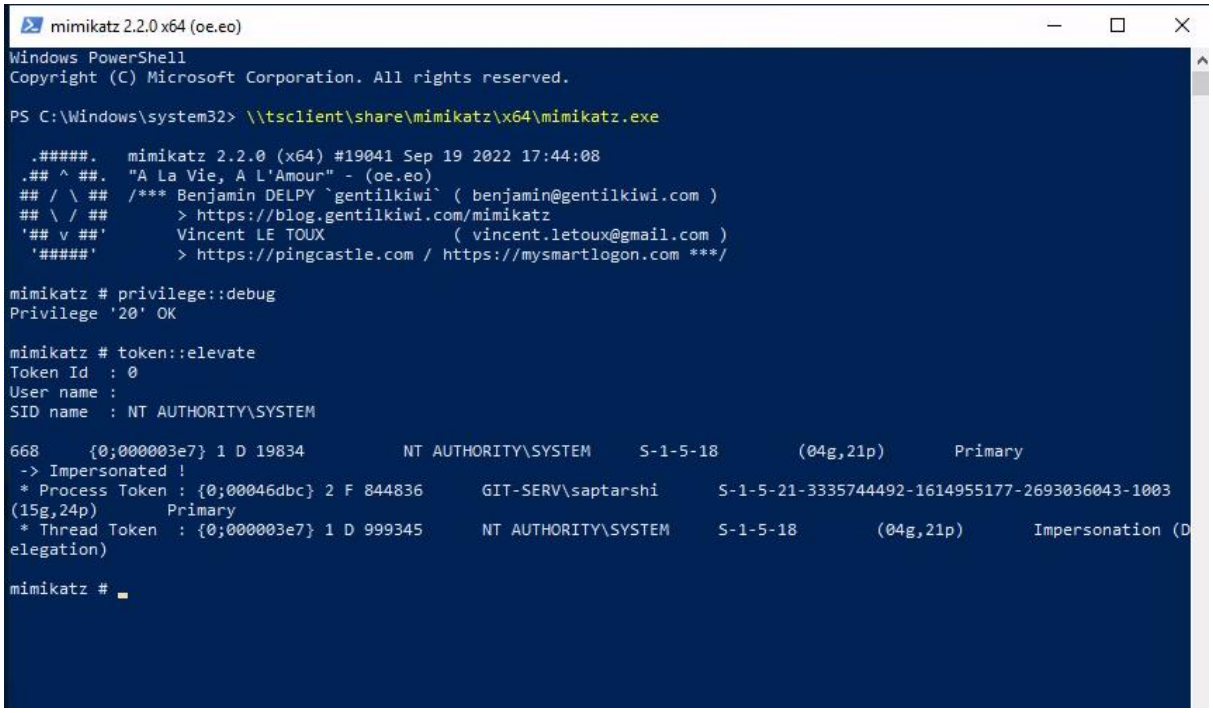


Figure 18:Accessing mimikatz.exe(through git-serv powershell terminal) through shared drive

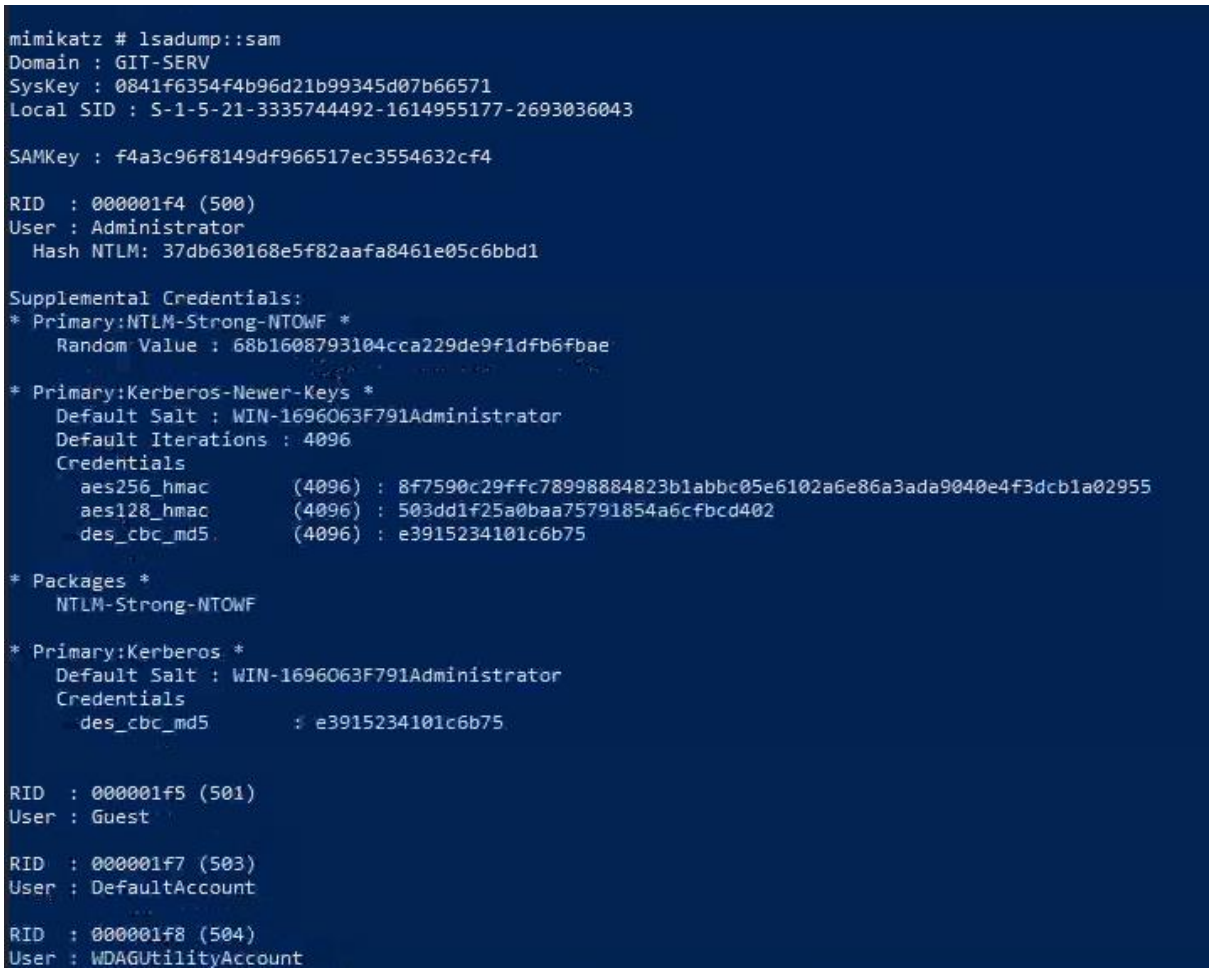


Figure 19: Dumping Administrator hash of git-serv with mimikatz.exe

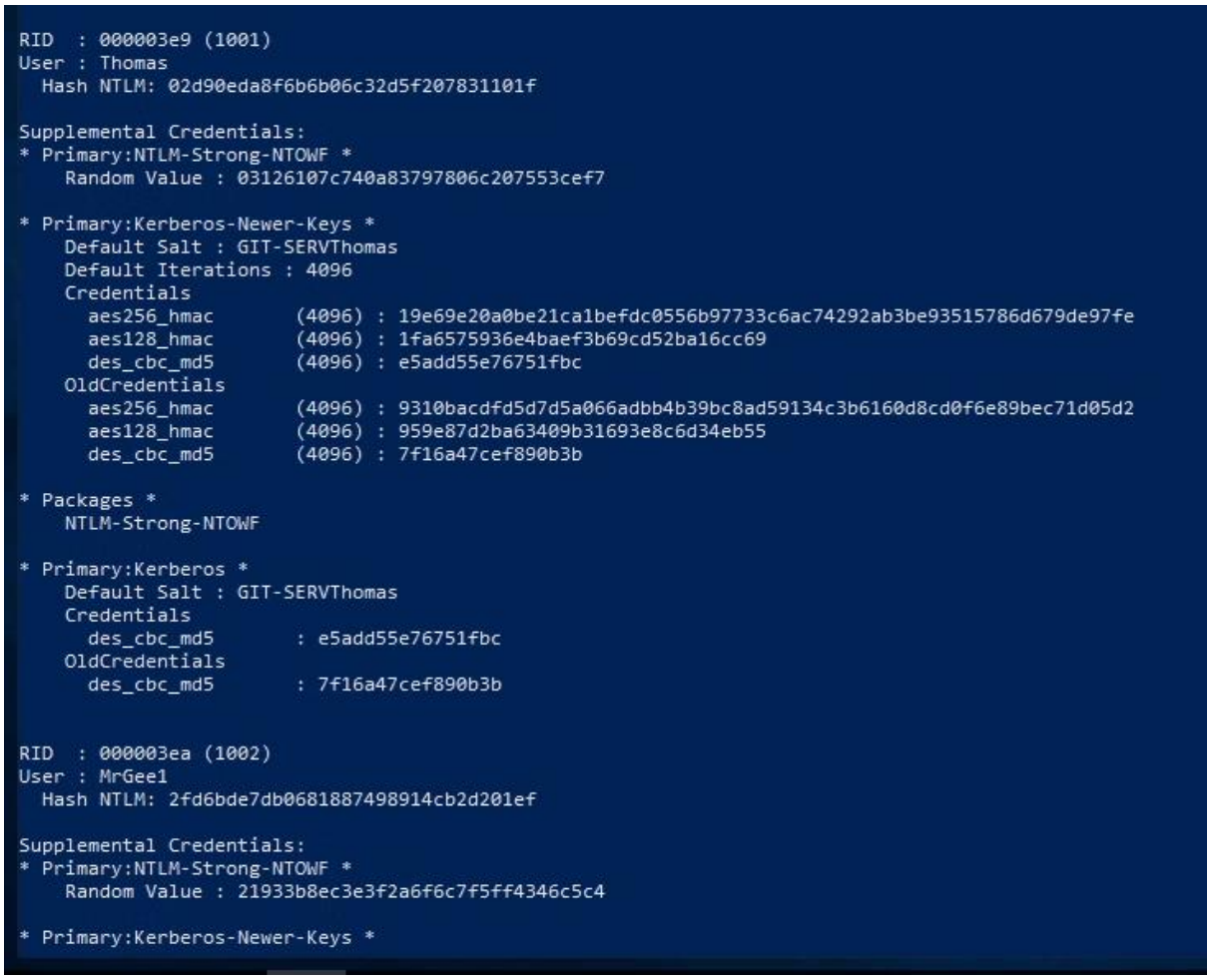


Figure 20:Dumping "Thomas" and other user hashes of git-serv with mimikatz

Remediation: Disable WDigest unless required. Enforce LSASS protections and deploy EDR tools.

Finding IPT-005: Use the dumped Administrator hash for Admin Shell via evilWinRm(High)

Description:	The extracted Administrator hash was reused to gain another elevated shell
Risk:	Likelihood: High – Admin hash is a potent credential for privilege escalation and commonly used malicious attackers. Impact: High – Credential replay and privilege escalation.
System:	git-serv (10.200.180.150)
Tools Used:	Mimikatz, Evil-winrm
References:	

Evidence:

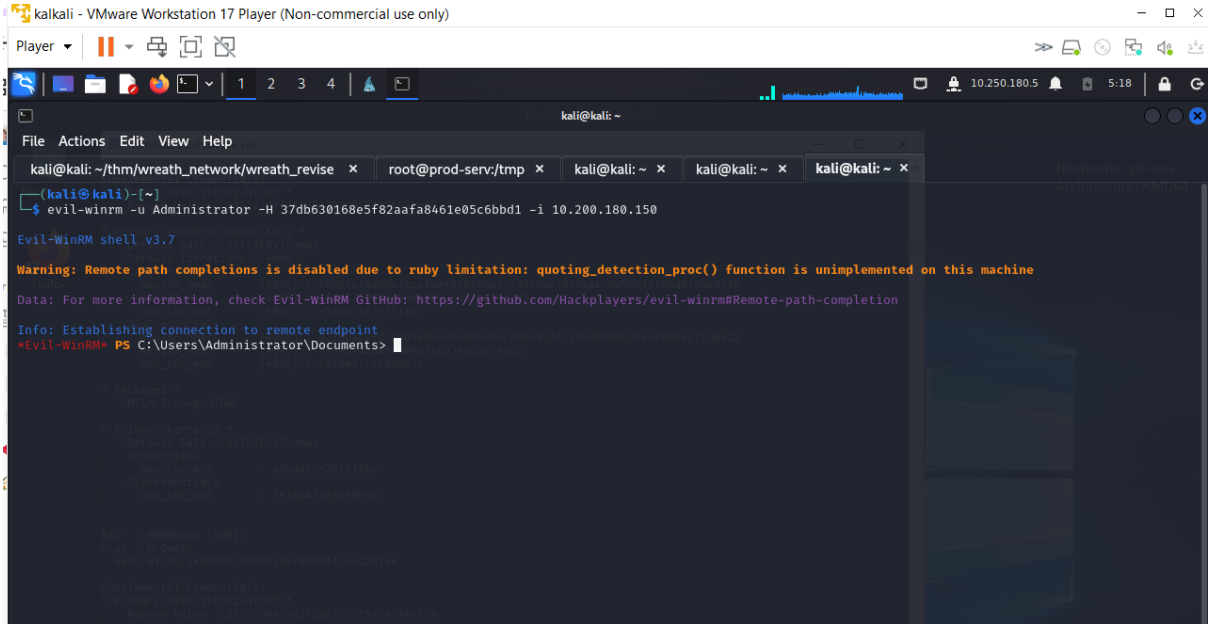


Figure 21:evil-winRM connection using Administrator hash

Remediation: Utilize Local Administrator Password Randomization (LAPS). Monitor lateral movement patterns.

Finding IPT-006: Arbitrary File Write to Documents Directory (Medium)

Description:	chisel.exe and other tools were written to the Documents folder on git-serv.
Risk:	Likelihood: Moderate – Attackers with write access can easily stage tools if folder permissions are not hardened. Impact: Tool staging for pivoting and post-exploitation.
System:	git-serv (10.200.180.150)
Tools Used:	Python3 http server, evil-winrm, Powershell (of target)
References:	

Evidence:

Figure 22:Planting chisel.exe through evil-winRM shell

Remediation: Apply least privilege permissions to sensitive folders.

Finding IPT-007: Tool sharing via evil-winRM (Moderate)

Description:	Powershell Empire modules were made accessible via shared evil-winRm sessions
Risk:	Likelihood: Moderate – WinRM misuse is common once internal foothold is established and tools like evil-winRM make exploitation easier. Impact: Moderate – Facilitated exploitation framework operations
System:	git-serv (10.200.180.150)
Tools Used:	Evil-winRm, PowerShellEmpire
References:	https://stealthbits.com/blog/wdigest-clear-text-passwords-stealing-more-than-a-hash/

Evidence:

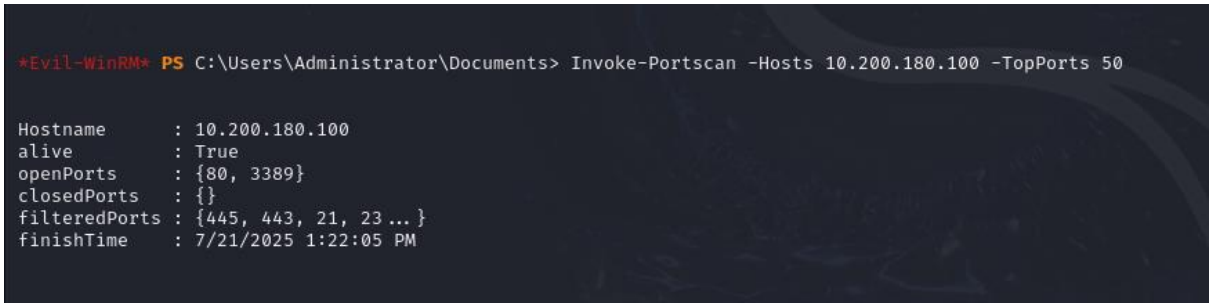


Figure 23:Using the shared Powershell Empire Invoke-Portscan tool to scan open ports of wreath-pc

Remediation: Disable or closely monitor WinRM access. Inspect remote PS sessions for abnormal behaviour

Finding IPT-008: Internal Port Scanning via Powershell Empire (Moderate)

Description:	The Invoke-PortScan script was used to discover open ports on internal hosts
Risk:	Likelihood: Moderate – PowerShell-based port scanning is a common, low-noise method for reconnaissance once an attacker gains internal access. Impact: Moderate – Network service mapping for lateral movement
System:	wreath-pc
Tools Used:	Evil-winRM, PowerShellEmpire (Invoke-PortScan)
References:	

Evidence:

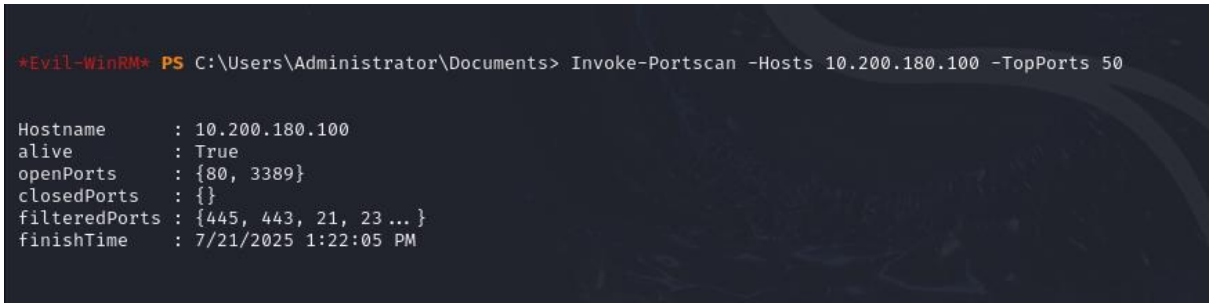


Figure 24:Open ports 80, 3389 discovered of wreath-pc

Remediation: Monitor Powershell activity. Apply endpoint firewall rules. (note: is Powershell Empire same as standard Powershell activity)

Finding IPT-009: wreath-pc Exposed Critical Ports (Moderate)

Description:	Wreath-pc revealed open ports 80 and 3389 that are HTTP and RDP ports from the internal network
Risk:	Likelihood: Moderate – Internal hosts often expose management and web services without proper segmentation or filtering. Impact: Moderate – Targeted system accessible without additional restrictions
System:	wreath-pc (10.200.180.100)
Tools Used:	Evil-winRM, PowerShellEmpire (Invoke-PortScan)
References:	

Evidence:

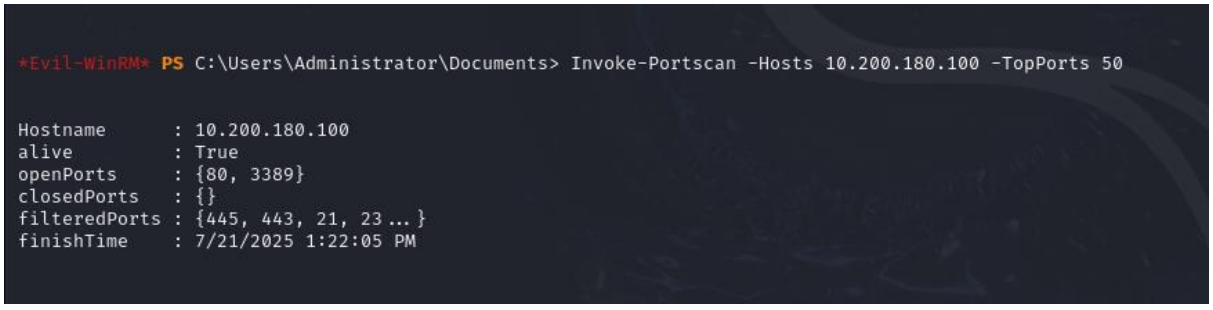


Figure 25:Powershell Empire tool via evil-WinRm easily able to bypass and discover open ports on wreath-pc which is otherwise not discoverable outside the internal network

Remediation: Implement strict ACLs and isolate sensitive hosts.

Finding IPT-010: Double Pivot to Internal host via SSH and Chisel (High)

Evidence:

Description:	Attacker created a chained pivot from prod-serv → git-serv → wreath-pc, accessing internal Git interface.
Risk:	Likelihood: High – Chained pivots using tools like SSH and Chisel are common in flat or poorly segmented networks. Impact: High – Full compromise of isolated network zone
System:	wreath-pc accessed via git-serv and prod-serv
Tools Used:	Chisel, Socat binary, sshuttle, ssh, Firefox browser
References:	

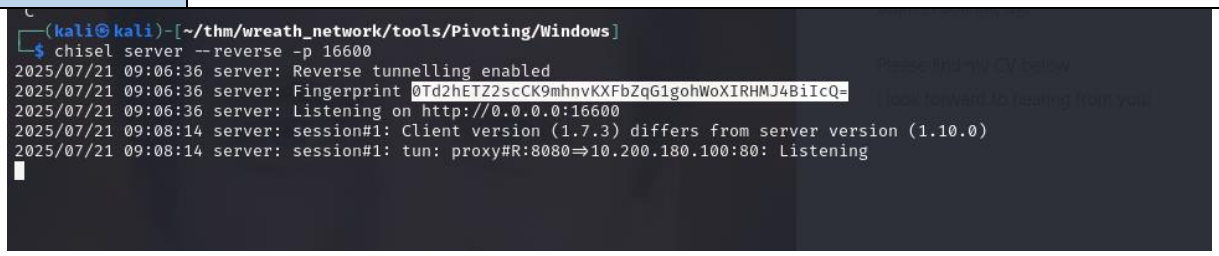


Figure 26:Pivoting:Chisel server started on attack box

Figure 27:Pivoting: socat binary on prod-serv being used to fork traffic

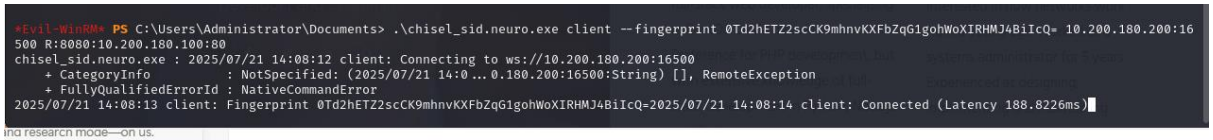


Figure 28:Pivoting:chisel client started on git-serv and port 8080 of attack box forwarded to port 80 of wreath-pc

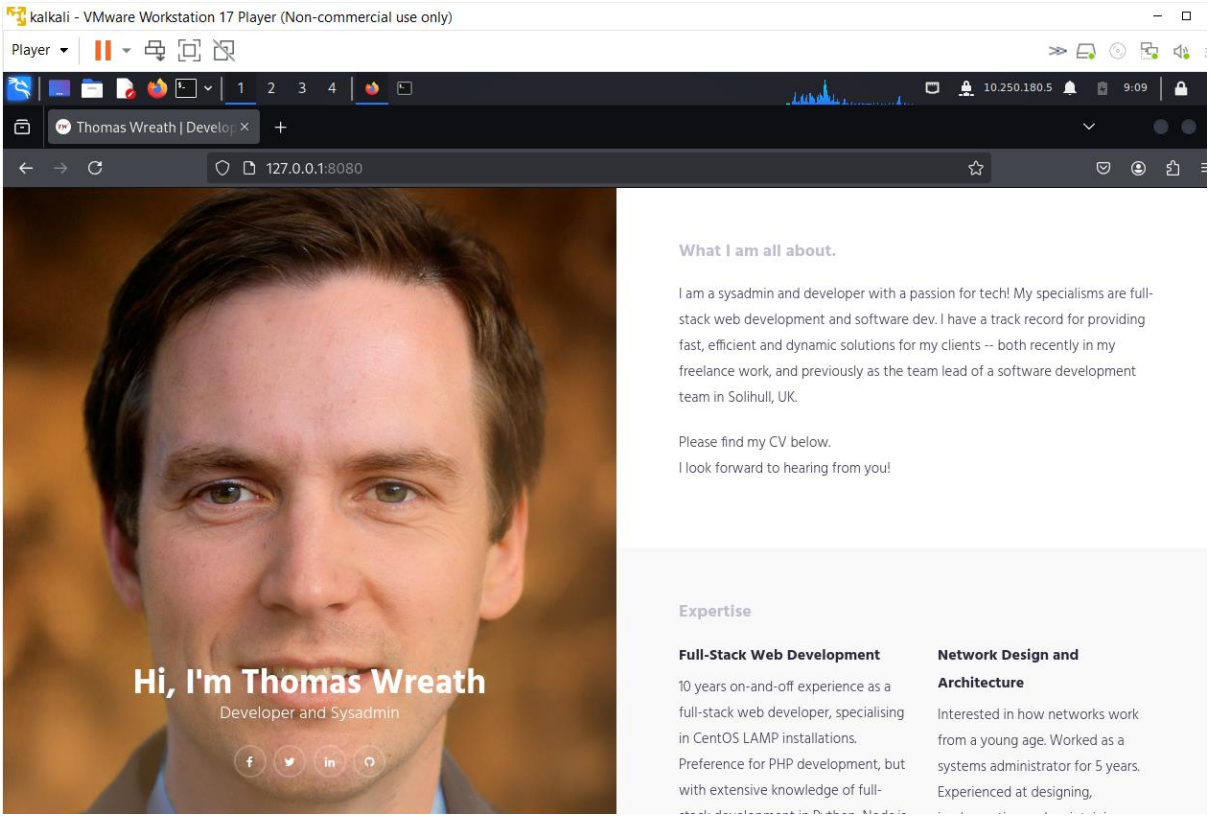


Figure 29:url which is a git-repository hosted on port 80 of wreath-pc, accessed by attack box through double pivot (see earlier)

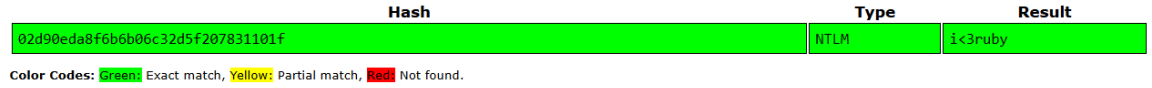
Remediation: Apply host-based firewalls and control inter-host connectivity

Finding IPT-011: Weak Password on "Thomas" account (High)

Description:	**The Hash of "Thomas" was cracked easily due to poor complexity**

Risk:	**Likelihood:** High – Password cracking using dumped NTLM hashes is highly feasible with modern GPU hardware and commonly used weak passwords. **Impact:** High – Unauthorized access to restricted Git resources
System:	wreath-pc
Tools Used:	Mimikatz, Crackstation(url)
References:	

Evidence:



Hash	Type	Result
02d90eda8f6b6b06c32d5f207831101f	NTLM	i<3ruby

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Figure 30:User "thomas" hash cracked to reveal plain-text password

Remediation: Implement robust MIME-type and content-based upload validation

Finding IPT-012: Insecure File Extension Validation in Upload Endpoint (High)

Description:	Application only validated extension after first dot (eg. Jpeg, php bypassed filter)

Risk:	**Likelihood: High – Extension-based bypasses are a well-known evasion technique, frequently exploited when input validation is improperly implemented.** **Impact: High – Enabled web shell upload**
System:	**wreath-pc**
Tools Used:	**Exiftool, Php reverse-shell, curl**
References:	

Evidence:

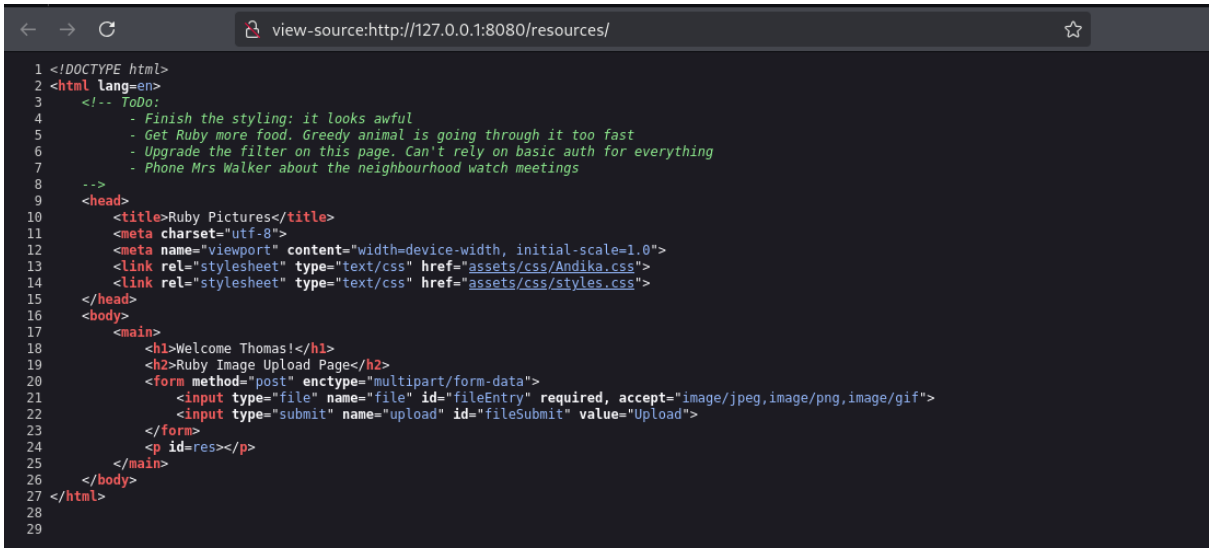


Figure 31:source code of the git-repository, revealing filtering method for image upload

Remediation: Strip all metadata. Scan uploads with static/dynamic analyzers

Finding IPT-013: Metadata-based PHP Exploit via Image Upload (Critical)

Description:	Malicious payload in image metadata triggered command execution using GET parameter wreath
Risk:	Likelihood: High – Metadata-based execution is an emerging attack vector, particularly effective when user-supplied files are not properly sanitized or scanned. Impact: Critical – Remote Code Execution
System:	wreath-pc
Tools Used:	exiftool, php-reverse-shell, curl, browser-based trigger
References:	

Evidence:

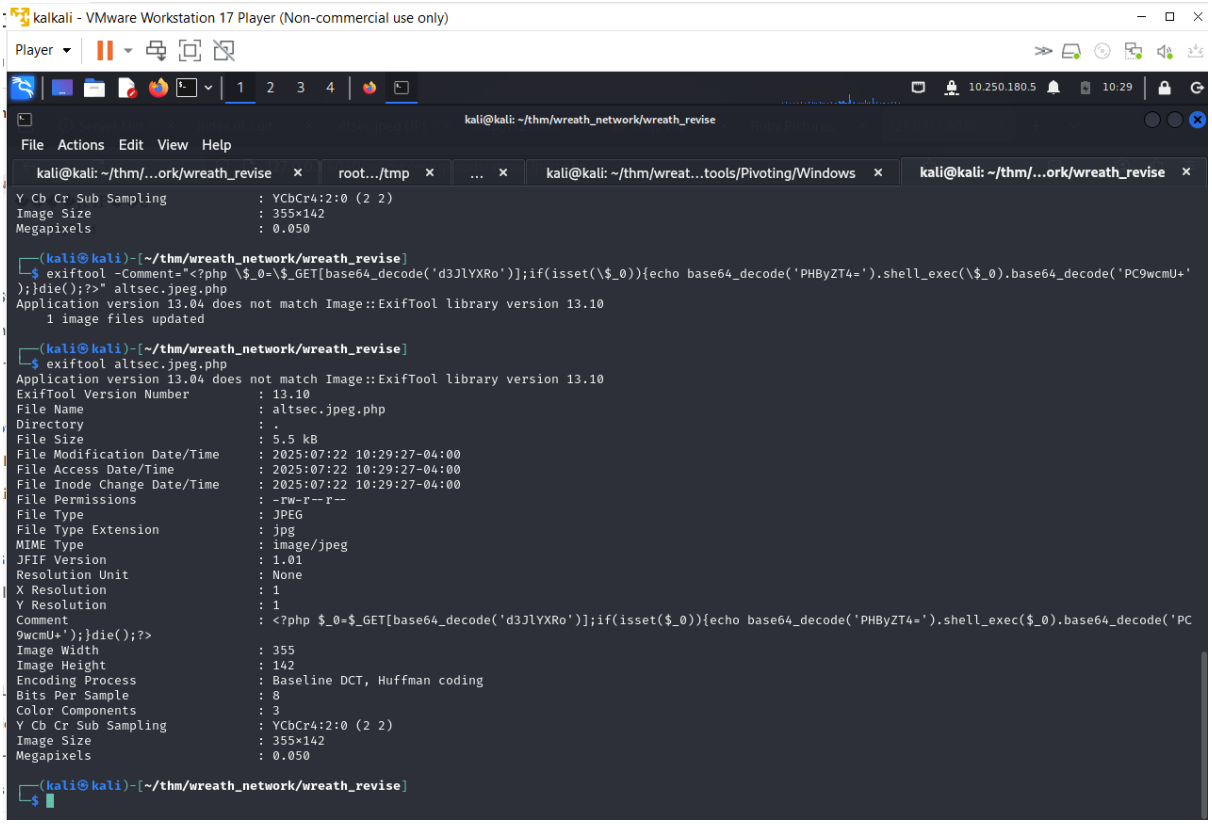


Figure 32:Using exiftool to plant shell code in the system

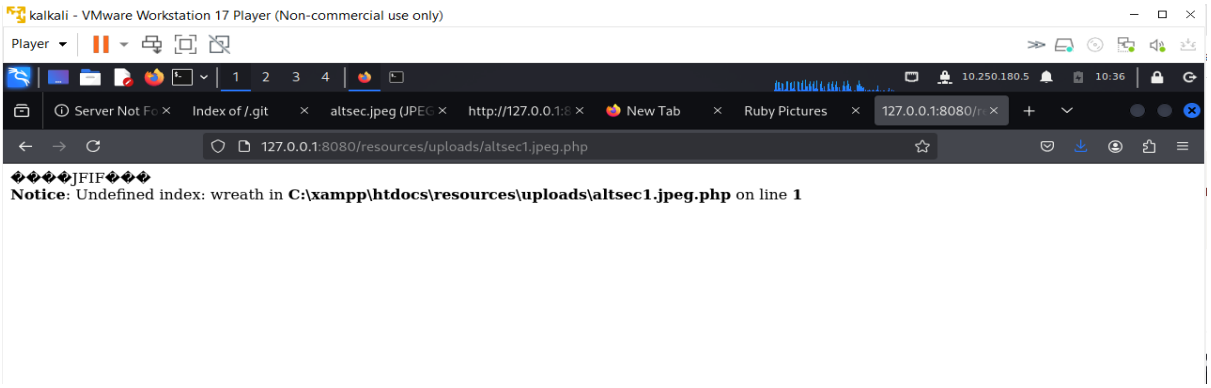


Figure 33: shell of wreath-pc confirmed on browser

Remediation: Strip all metadata. Scan uploads with static/dynamic analyzers.

Finding IPT-014: Web Shell Access via GET Parameter (High)

Description:	**Backdoor provided persistent command execution via web interface**
Risk:	**Likelihood: High – Web shells accessed via parameters are commonly used for persistent access post-exploitation, especially when server-side controls are weak.** **Impact: High – Continuous attacker presence**
System:	**wreath-pc**
Tools Used:	**custom PHP web shell, browser, PowerShell, curl**
References:	

Evidence:

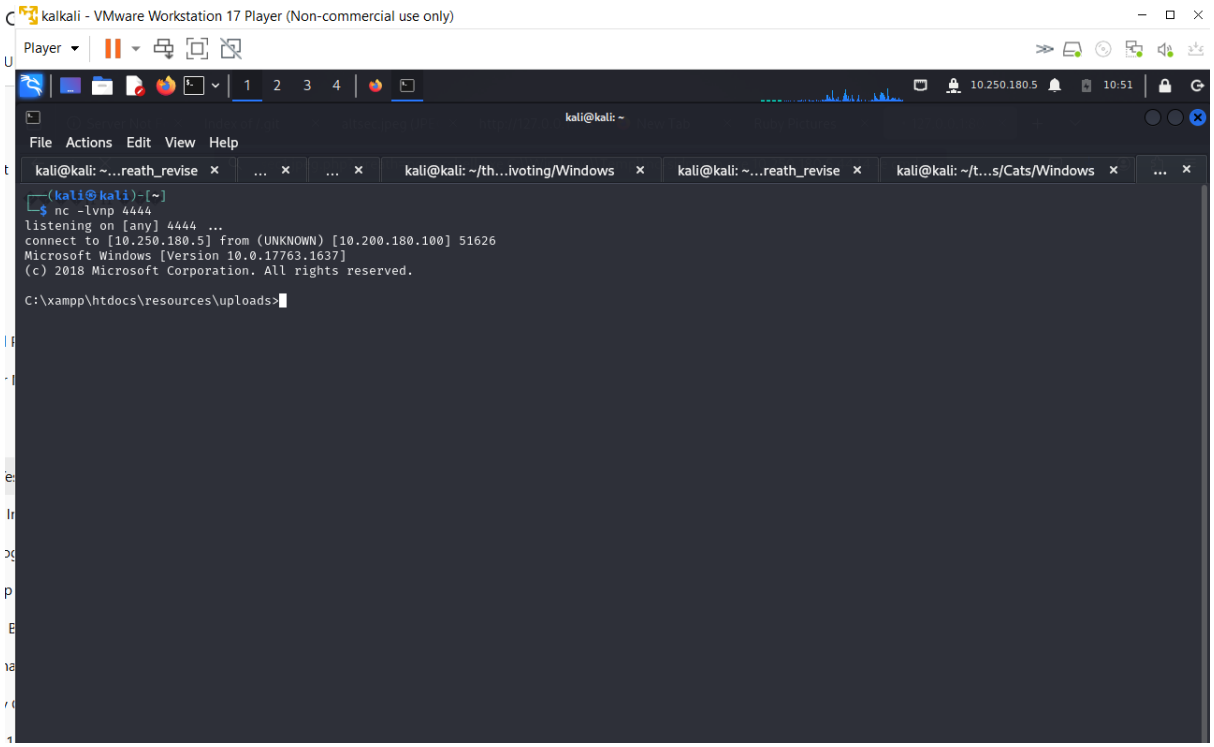


Figure 34:Reverse shell of wreath-pc with terminal access

Remediation: Audit server code for unauthorized logic. Remove all unverified web scripts.

Finding IPT-015: Reverse shell as Thomas via Backdoor (High)

Description:	Exploited PHP backdoor to gain shell as “Thomas”
Risk:	Likelihood: – Impact: High – Shell access via PHP backdoors is a frequent post-exploitation tactic, especially when combined with weak upload or validation logic.
System:	wreath-pc
Tools Used:	reverse shell payload, nc, custom PHP backdoor
References:	

Evidence:

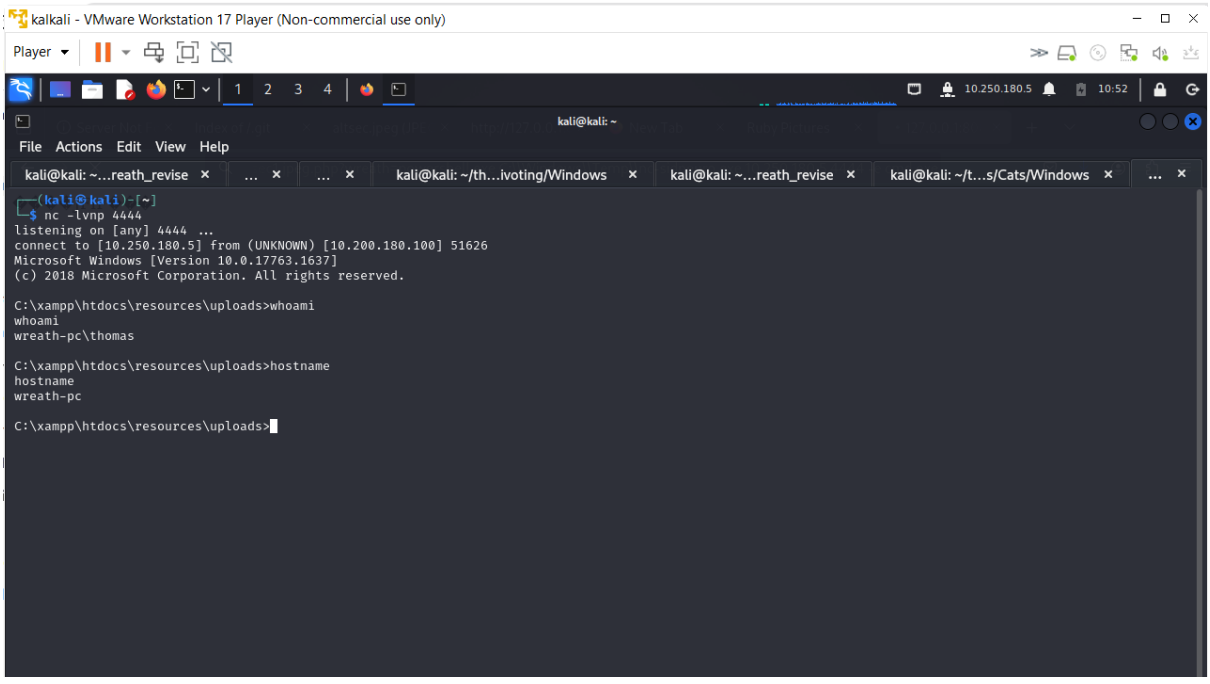


Figure 35:Reverse of wreath-pc as user "thomas"

Remediation: Rotate all user credentials. Investigate shell activity.

Finding IPT-016: Unquoted Service Path in System Explorer (High)

Description:	Identified unquoted service path exploitable for privilege escalation.
Risk:	Likelihood: High – Unquoted service paths are a common misconfiguration and easily exploitable if write access is available on parent directories. Impact: High – Service level abuse to escalate to SYSTEM
System:	wreath-pc
Tools Used:	Manual inspection via sc qc
References:	

Evidence:

```
C:\xampp\htdocs\resources\uploads>wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
DisplayName                                                 StartMode  Name                        PathName
Amazon SSM Agent                                            Auto       AmazonSSMAgent              "C:\Program Files\Amazon\SSM\a
mazon-ssm-agent.exe"
Apache2.4                                                   Auto       Apache2.4                   "C:\xampp\apache\bin\httpd.exe
" -k runservice
AWS Lite Guest Agent                                        Auto       AWSLiteAgent                "C:\Program Files\Amazon\XenTo
ols\LiteAgent.exe"
LSM                                                         Unknown    LSM
Mozilla Maintenance Service                                 Manual     MozillaMaintenance          "C:\Program Files (x86)\Mozill
a Maintenance Service\maintenanceservice.exe"
NetSetupSvc                                                 Unknown    NetSetupSvc
Windows Defender Advanced Threat Protection Service         Manual     Sense                       "C:\Program Files\Windows Defe
nder Advanced Threat Protection\MsSense.exe"
System Explorer Service                                     Auto       SystemExplorerHelpService   C:\Program Files (x86)\System
Explorer\System Explorer\service\SystemExplorerService64.exe
Windows Defender Antivirus Network Inspection Service       Manual     WdNisSvc                    "C:\ProgramData\Microsoft\Wind
ows Defender\platform\4.18.2011.6-0\NisSrv.exe"
Windows Defender Antivirus Service                          Auto       WinDefend                   "C:\ProgramData\Microsoft\Wind
ows Defender\platform\4.18.2011.6-0\MsMpEng.exe"
Windows Media Player Network Sharing Service                Manual     WMPNetworkSvc               "C:\Program Files\Windows Medi
a Player\wmpnetwk.exe"
```

Figure 36: Enumeration of wreath-pc reveals Unquoted Service Path vulnerability

Remediation: Correct path quotations in service definitions

Finding IPT-017: Writable Service Directory (High)

Description:	The service binary directory was writable by all users
Risk:	Likelihood: High – Writable service directories are often overlooked and easily exploited by attackers with local access. Impact: High – Malicious binaries could be dropped and executed
System:	wreath-pc
Tools Used:	Reverse shell, PowerShell Get-Acl command for directory permission inspection
References:	

Evidence:

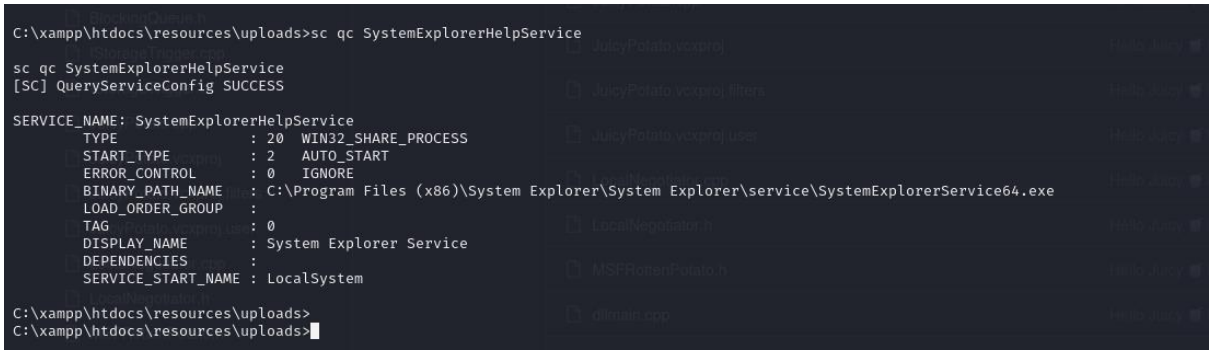


Figure 37: Configuration informtion of the vulnerable Service

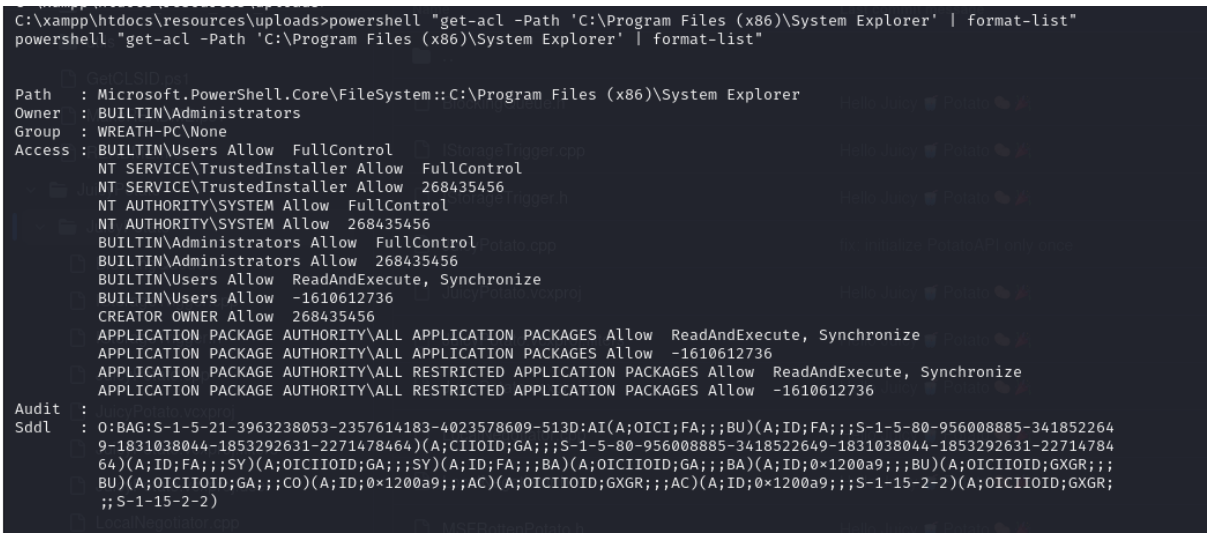


Figure 38: Enumeration permissions of the target directory(of the vulnerable system path)

Remediation: Enforce NTFS permissions on service paths

Findings IPT-018: SYSTEM shell via Malicious Service Binary (Critical)

Description:	Attacker restarted service with the malicious payload, gaining SYSTEM shell
Risk:	Likelihood: High – Writable service paths combined with weak access controls are a known privilege escalation vector and were actively exploited. Impact: Critical – Full system-level control
System:	wreath-pc
Tools Used:	Reverse shell, custom malicious binary, sc and net commands for service control.
References:	

Evidence:

Figure 39: Exploit script wrapper.exe transferred to wreath-pc

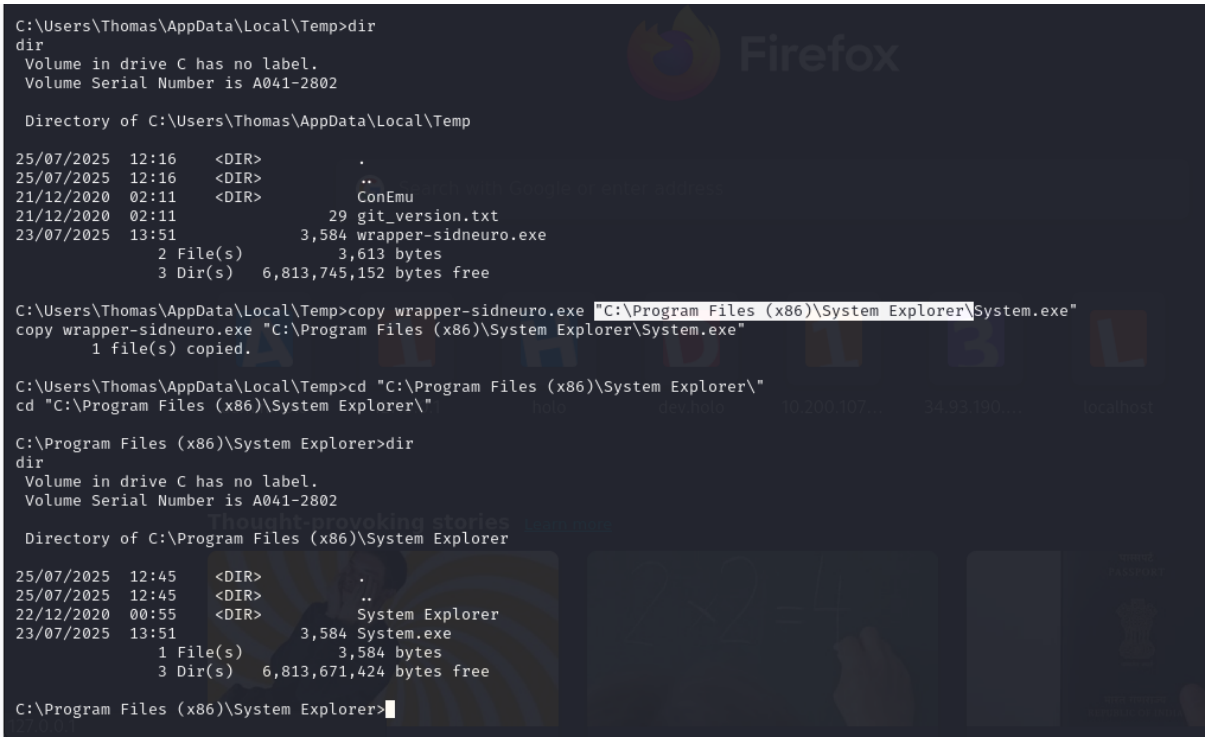


Figure 40:Exploit script wrapper.exe(now named wrapper-sidneuro.exe) planted in target directory of wreath-pc

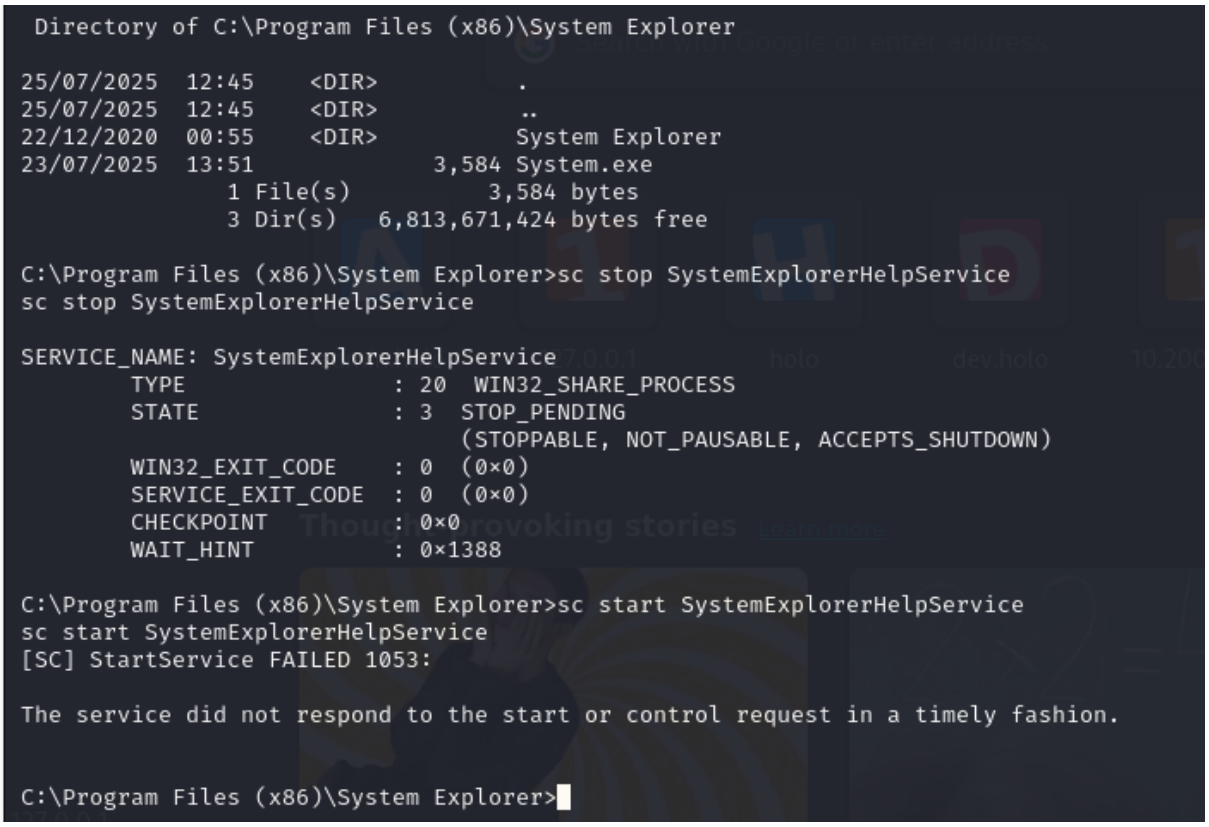


Figure 41:SystemExplorerHelpService started with motive to trigger the reverse shell

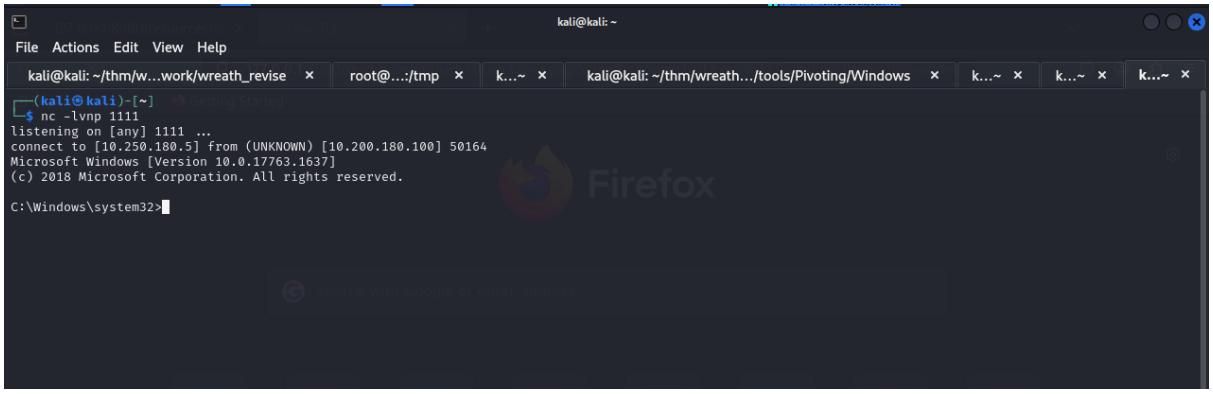


Figure 42: Successful reverse of shell of wreath-pc as LocalSystem

Remediation: Harden services. Monitor for abnormal service restarts.

Finding IPT-019: SAM and SYSTEM Hive Exfiltration (Critical)

Description:	SAM and SYSTEM files were copied out via SMB share
Risk:	Likelihood: High – Attackers frequently target SAM and SYSTEM hives once privileged access is achieved. Exfil via SMB is common in flat or weakly monitored internal networks. Impact: Critical – Offline credential dumping possible
System:	wreath-pc
Tools Used:	Built-in Windows file copy tools, SMB share, secretsdump.py (Impacket) for offline hash extraction.
References:	https://stealthbits.com/blog/wdigest-clear-text-passwords-stealing-more-than-a-hash/

Evidence:

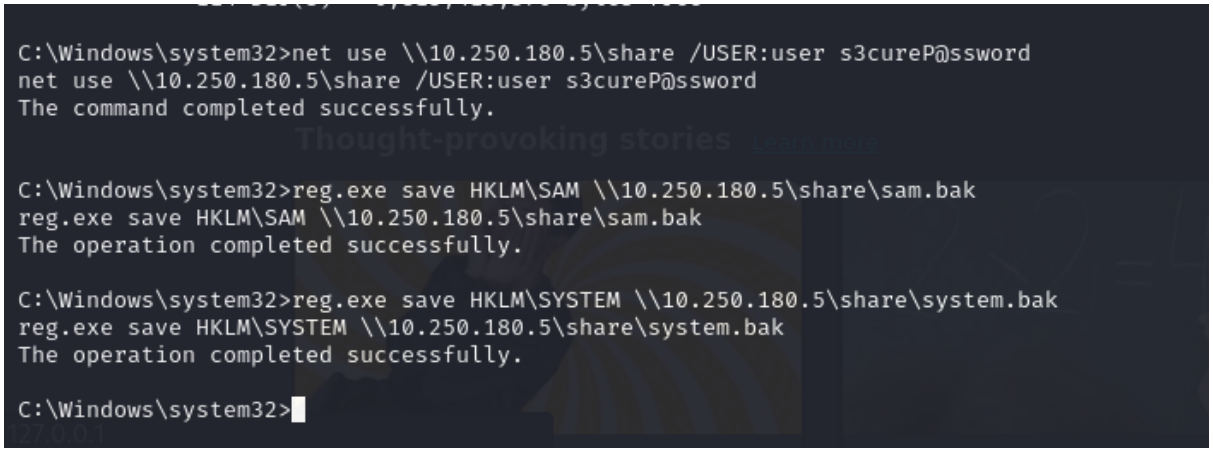


Figure 43: SAM and SYSTEM hive saved and transferred back to attack box

Remediation: Disable unnecessary shares. Restrict access to registry files

Finding IPT-020: secretsdump Used to Extract Password Hashes (Critical)

Description:	Extracted and cracked user and admin hashes from exfiltrated hives
Risk:	Likelihood: High – secretsdump is a well-known and reliable technique used in almost all post-exploitation phases when registry hives are accessible. Impact: Critical – Offline password compromise and potential reuse
System:	Wreath Network (prod-serv, git-serv and wreath-pc)
Tools Used:	secretsdump.py (Impacket)
References:	

Evidence:

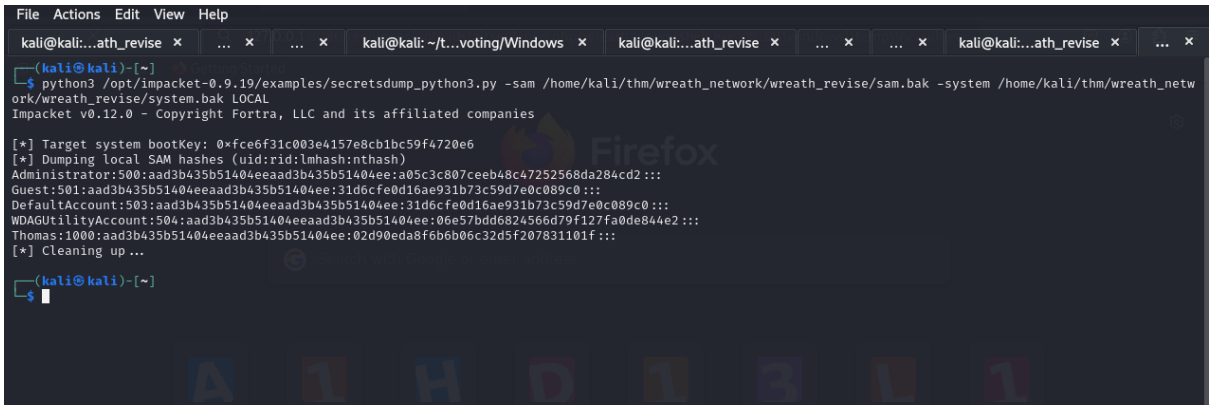


Figure 44: SAM and SYSTEM hives used to dump sam hashes of wreath-pc. Done in locally in attack box.

Remediation: Salt and hash credentials. Rotate all credentials post-compromise.

Steps to compromise the Wreath Network

The following section describes the steps taken by the Penetration Tester to compromise all the three hosts i.e. prod-serv(external host), git-serv and wreath-pc of the "Wreath Network".

Step	Action	Remediation
1	Initial access to prod-serv via exposed SSH service	Restrict SSH to authorized IPs. Enforce strong key-based authentication.
2	Deployed Socat, Chisel, and Nmap binaries on prod-serv to assist in scan, pivot into and gain a foothold in the Internal network.	Monitor and restrict outbound tunneling tools. Apply egress filtering.
3	Pivoted to git-serv and created a local user. Used the same to have shell via WinRM.	Restrict WinRM to admin-only subnets. Enforce MFA and audit remote sessions.

Step	Action	Remediation
4	Gained RDP access to git-serv using created user	Restrict RDP to trusted IPs. Enforce session logging and strong auth.
5	Dumped credentials via mimikatz on git-serv	Enforce LSASS protection. Deploy EDR. Disable WDigest unless required.
6	Used dumped Administrator hash to get shell via WinRm.	Rotate admin hashes. Implement LAPS. Monitor for lateral movement.
7	Uploaded post-ex tools and Empire modules using evil-winRM	Disable WinRM where not needed. Monitor file transfers and PowerShell usage.
8	Internal port scanning via PowerShell to enumerate accessible services	Monitor PowerShell script activity. Apply strict endpoint firewall rules.
9	Exploited insecure file upload to gain persistent shell via web app	Implement strict content-type and extension validation. Strip metadata.
10	Uploaded backdoor and achieved reverse shell as Thomas	Audit and sanitize all file upload logic. Rotate all affected user credentials.
11	Identified unquoted service path and writable directory	Review all service definitions. Harden NTFS permissions.
12	Planted malicious binary and restarted service to get SYSTEM shell	Monitor for unauthorized service restarts. Deploy EDR for behavior detection.
13	Exfiltrated SAM and SYSTEM hives via SMB share	Disable SMB where not needed. Restrict registry file access.
14	Used secretsdump to extract and crack hashes offline	Rotate all credentials. Enforce strong password policies and monitoring.