Final VAPT Report

PREPARED BY: Esperanza Buitrago Díaz

Submitted To: Neuefische - Cybersecurity

Submission Date: February 6th 2026

TABLE OF CONTENTS

EXECUTIVE SUMMARY

A series of comprehensive security assessments and a forensic investigation were conducted for Neuefische between January and February 2026. This multi-phase engagement simulated real-world attack scenarios -from initial external exploitation and web application attacks to internal privilege escalation, credential compromise, and post-incident forensic analysis.

The purpose of this engagement was to assess the security posture across multiple layers of the environment, identify vulnerabilities, demonstrate exploitability, and provide actionable recommendations. There were identified a total of 18 vulnerabilities within the scope of the engagement which are broken down by severity in the table below.

CRITICAL	HIGH	MEDIUM	LOW
1	2	2	1

The highest severity vulnerabilities give potential attackers the opportunity to access the company's internal assets, hardware, network, data, etc. In order to ensure data confidentiality, integrity, and availability, security remediations should be implemented as described in the security assessment findings.

Phase 1: Internal Network penetration test

A simulated external attack against the corporate network identified **6 vulnerabilities**, including a critical EternalBlue (MS17-10) vulnerability. Attackers could remotely execute arbitrary code, gain full system control, steal credentials, and move laterally across the network.

Phase 2: Web application security assessment

An authenticated penetration test of the Damn Vulnerable Web Application (DVWA) revealed **4 high-risk vulnerabilities**, demonstrating risk to data confidentiality, integrity, and availability through common web-based attack vectors.

Phase 3: Credential security and password cracking assessment

Following simulated initial access via EtenalBlue, attackers successfully compromised **2 user accounts** via weak passwords, extracted password hashes from the SAM database, and demonstrated risks associated with weak password policies and credential reuse.

Phase 4: Digital forensic investigation

A forensic analysis of a compromised disk image verified evidence integrity, established Chain of Custody, and recovered hidden JPG files from deleted space, compressed archives, and obfuscated directories. Findings revealed systematic data concealment techniques, including file extension manipulation, NTFS metadata tampering, and malicious document integration.

Vulnerability summary by severity

PHASE	CRITICAL	HIGH	MEDIUM	LOW
Internal Network assessment	**3**	**2**	**1**	**0**
Web Application assessment	**-**	**4**	**-**	**-**
Credential Security assessment	**-**	**2***	**-**	**-**
Forensic Investigation	**-**	**1****	**-**	**-**

* Successful credential compromises via weak passwords.

** Systematic data concealment with high business impact.

Key findings and impact

The highest-severity vulnerabilities provide attackers with the ability to:

- Remotely execute code and gain full system control via unpatched EternalBlue vulnerability.
- Compromise web application through SQL injection, XSS, and command injection.
- Crack weak passwords and extract credential hashes, enabling lateral movement and privilege escalation.
- Conceal and exfiltrate sensitive data using obfuscated files, compressed archives, and NTFS metadata manipulation.

This finding highlights critical gaps in patch management, password policies, web application security, and forensic readiness. Implementing the recommended security controls is essential to ensuring the confidentiality, integrity, and availability of organizational data and systems.

Note that this assessment may not disclose all vulnerabilities that are present on the systems within the scope. Any changes made to the environment during the period of testing may affect the results of the assessment.

HIGH LEVEL ASSESSMENT OVERVIEW

Observed Security Strengths

The following strengths during the comprehensive security assessment, which demonstrate positive security practices that the company should continue to maintain and monitor.

Forensic and investigative capabilities.

- Proper evidence handling procedures:
  Established chain of custody and evidence integrity verification (findings **#17**, **#18**).
- Digital forensic readiness:
  Demonstrated capability to recover hidden files from deleted space and compressed archives.
- Systematic investigation methodology:
  Methodical approach to vulnerability identification and exploitation verification.

Security control implementation

- Network service identification:
  Comprehensive discovery of exposed services enabling target remediation.
- Vulnerability detection:
  Effective use of automated and manual testing methodologies.
- Technical documentation:
  Through evidence collection and documentation supporting findings.

Operational security practices

- Testing environment isolation:
  Proper segregation of assessment systems from production.
- Controlled exploitation:
  Ethical execution of penetration testing without business disruption.
- Findings validation:
  Multiple verification methods ensuring accurate risk assessment.

Areas for Improvement

It is recommended that the company takes the following actions to improve security posture across network, application, and endpoint layers. Implementing these recommendations will significantly reduce the likelihood of success attacks and minimize potential business impact.

Short Term Recommendations

Critical vulnerability remediation

1. Patch critical vulnerabilities:
    - Apply MS17-010 EternalBlue patches immediately.
    - Update all end-of-life Windows 7 systems.
    - Implements emergency patch management procedures.
2. Credential security enhancement:
    - Reset all compromised passwords with immediate effect.
    - Enforce minimum 12-character password policy.
    - Implement account lockout after 5 failed attempts.
    - Enable LSA protection on all Windows systems.
3. Web application security:
    - Fix SQL injection and XSS vulnerabilities in DVWA.
    - Implement input validation and output encoding.
    - Restrict file upload functionality with proper validation.
    - Deploy Web Application Firewall (WAF).
4. Network service hardening:
    - Disable SMBv1 protocol across all systems.
    - Restrict RPC, RTSP, and HTTPAPI services.
    - Implement network segmentation for critical systems.
    - Configure firewall rules to limit unnecessary service exposure.

Security control implementation

5. Endpoint protection:
    - Deploy endpoint detection and response (EDR) solutions.
    - Implement application whitelisting.
    - Enable Windows Defender Credentials Guard where supported.

6. Monitoring and detection:
    - Implement real-time alerting for brute-force attempts.
    - Monitor for hash extraction and pass-the-hash attacks.
    - Deploy file integrity monitoring for critical systems.

Long Term Recommendations

Strategic security initiatives

1. Architecture modernization:
    - Migrate from Windows 7 to supported operating systems.
    - Implement Zero Trust network architecture.
    - Deploy microsegmentation for critical assets.
    - Evaluate cloud-based securities solutions.
2. Identity and access management:
    - Implement Multi-Factor Authentication (MFA) for all accounts.
    - Deploy Privileged Access Management (PAM) solution.
    - Implement just-in-time administrative access.
    - Consider passwordless authentication solutions.
3. Advanced threat protection:
    - Deploy Security Information and Event Management (SIEM).
    - Implement threat intelligence integration.
    - Develop behavioral anomaly detection.
    - Establish security orchestration and automated response (SOAR).
4. Secure Development Lifecycle:
    - Integrate security testing into CI/CD pipelines.
    - Implement DevSecOps practices.
    - Conduct regular security code reviews.
    - Establish secure coding standards and training.
5. Forensic and incident response capability:
    - Develop a comprehensive incident response plan.
    - Establish a digital forensic investigation team.
    - Implement automated evidence collection.
    - Conduct regular incident response drills.

6. Compliance and Governance:
    - Align with ISO 27001/27037 standards.
    - Implement NIST Cybersecurity Framework.
    - Establish regular third-party security assessments.
    - Develop security metrics and reporting dashboard.

Organizational Security Maturity

7. Security awareness and training:
    - Implement an organization-wide security awareness program.
    - Conduct regular phishing simulation exercises.
    - Provide role-based security training.
    - Establish a security champions program.
8. Vendor and third-party risk management:
    - Implement a vendor security assessment program.
    - Establish security requirements for third-party software.
    - Conduct regular supply chain security reviews.

Technology investment

9. Security tool consolidation:
    - Evaluate unified security platform solutions.
    - Implement centralized security management.
    - Reduce security tool sprawl through consolidation.
    - Ensure tool integration and automation.
10. Continuous security validation:
    - Implement continuous penetration testing.
    - Deploy breach and attack simulation (BAS) tools.
    - Establish red-team/blue-team exercises.
    - Conduct regular security control effectiveness testing.

Priority Matrix

Priority	Timeframe	Focus Area	Key actions
Critical	0 - 7 days	Vulnerability remediation	- Patch EternalBlue, - Reset credentials, - Disable SMBv1.
High	7 - 30 days	Security controls	- Implement MFA, - Deploy EDR, - Configure monitoring.
Medium	1 - 3 months	Architecture	- Migrate EOL systems, - Implement segmentation.
Long-term	3 - 6 months	Maturity	- Establish program, - Implement frameworks, - Continuous testing.

Success Metrics

Immediate (30 days):

- 100% critical vulnerabilities patched.
- No successful SMB brute-force attacks.
- All compromised credentials reset and secured.

Short-term(3 months):

- 90% reduction in attack surface (exposed services).
- MFA implemented for all administrative accounts.
- EDR deployed on all critical systems.

Long-term (6 months):

- Zero Windows 7 systems in production.
- SIEM providing actionable security intelligence.
- Incident response time reduced by 50%

SCOPE

Project Scope

All testing was conducted in accordance with the agreed scope outlined in the Request for Proposal (RFP) and subsequent written communications. The assessment covered multiple layers of the internal environment, simulating real-world attack scenarios across network, application, credential, and forensic domains

In-Scope components

- Internal Network infrastructure - corporate LAN environment.
- Web application - Damn Vulnerable Web Application (DVWA).
- Windows endpoints and servers - Windows 7, Windows Server 2008 R2.
- Forensic disk image - 8-jpeg-search.dd (161MG image for post-incidents analysis).
- Authentication and Credential stores - Local user accounts, SAM database.

Network Information

Network range	Environment / Purpose
192.168.57.0/24	Internal lab / assessment Network
192.168.57.20	Windows 7 Professional (primary target)
192.168.57.30	DVWA Web Application Server
192.168.57.10	Kali Linux attacker system

Assessment objectives

Internal Network assessment

- Objective:
  - Identify and exploit vulnerabilities in internal network systems.
- Scope:
  - IP Range:
    - 192.168.57.0/24
  - Primary target:
    - Windows server 2008 R2

        - Windows 7 - 192.168.57.20
- Assessment type:
    - Black-box and Gray-box testing
- Activities:
    - Network scanning.
    - Vulnerability exploitation.
    - Service enumeration.

Web application assessment

- Objective:
    - Identify and exploit common web application vulnerabilities.
- Scope:
    - Target:
        - DVWA - 192.168.57.30
    - Assessment type:
        - Authenticated gray-box testing.
    - Activities:
        - SQL injection.
        - XSS attack.
        - File upload - php script.
        - Command injection.
        - Webshell execution.

Credential security assessment

- Objective:
    - Evaluate password strength and authentication security.
- Scope:
    - Target:
        - Windows 7 Professional - 192.168.57.20
    - Assessment type:
        - Post-exploitation credential testing.
    - Activities:
        - User enumeration.

        - SMB brute-force.
        - Hash extraction .
        - Cracking passwords.

Digital Forensic investigation

- Objective:
    - Conduct evidence collection
    - Recovery from a compromised system
- Scope:
    - Target image:
        - 8-jpeg-search.dd
        - Forensic copy
    - Tools:
        - Autopsy
        - MD5 hashing utilities
    - Activities:
        - Image integrity verification.
        - Hidden file recovery.
        - NTFS metadata analysis.

Overall engagement goals

1. Identify vulnerabilities

   Discover security weaknesses across network, web, and host layers.
2. Demonstrate exploitability

   Validate risks through controlled exploitation.
3. Assess credential security

   Evaluate password policies and authentication controls.
4. Conduct forensic analysis

   Recover hidden data and establish a chain of custody.
5. Provide actionable remediation

   Deliver prioritized recommendations to mitigate identified risks.
6. Enhance security posture

Support improved incident response and security maturity.

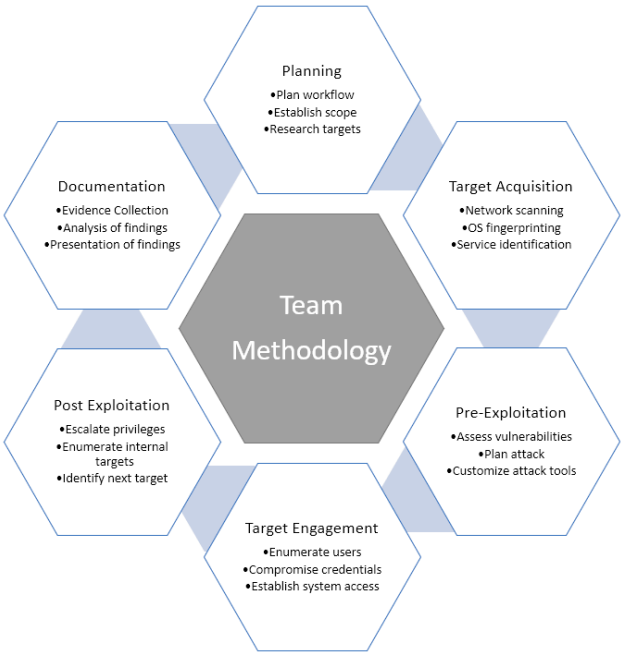
Out-of-scope items

- Social engineering attacks.
- Denial-of-service (DoS) attacks outside agreed parameters.
- Physical security assessments.
- Wireless network testing
- Third-party system and cloud environments not explicitly authorized.

TESTING METHODOLOGY

The testing methodology followed a structured, multi-phase approach aligned with industry-standard frameworks, including the **Penetration Testing Execution Standard** (**PTES**) and **NIST SP 800-115**. The engagement was conducted across four distinct phases, each simulating a different stage of a real-world attack chain-from initial network compromise and web application exploitation to credential theft and digital forensic investigation.

The following image is a graphical representation of this methodology:



Step 1: Reconnaissance and enumeration

During reconnaissance, passive and active information-gathering techniques were used to map the target environment, identify live hosts, and enumerate services, users, and applications.

- Network discovery:
  - nmap scans (`-sS`, `-sV`, `-O`) across `192.168.57.0/24`
- Service enumeration:
  - SMB
  - HTTP
  - FTP
  - RDP
- Web application mapping:
  - Manual exploration of
    - DVWA
    - Input field identification
    - Functional analysis
- User enumeration:
  - Local user extraction *via* `net users` and Metasploit post-modules.

Step 2: Vulnerability assessment

Identified vulnerabilities were analysed and validated to determine exploitability and potential impact. Both automated tools and manual testing were used.

- Automated scanning:
  - `nmap` NSE scripts for SMB vulnerabilities such as `smb-vuln-ms17-010`.
- Manual verification:
  - Confirmation of EternalBlue (`MS17-010`).
  - SQL injection.
  - XSS attack, and
  - File upload flaws.
- Credential testing:
  - Weak password identification *via* custom wordlist and policy analysis.

Step 3: Exploitation

Controlled exploitation of validated vulnerabilities was performed to demonstrate real-world risk and establish access.

- Network exploitation:
    - EternalBlue exploitation *via* Metasploit

        `exploit/windows/smb/ms17_010_eternalblue`
- Web Application attacks:
    - SQL injection for database extractions.
    - Stored XSS *via* guestbook functionality.
    - Malicious file upload and webshell execution.
- Credential attacks:
    - SMB brute-force using Hydra with custom wordlists.

Step 4: Post-Exploitation

After initial compromise, post-exploitation activities were conducted to simulate attacker persistence, privilege escalation, and internal movement.

- Meterpreter sessions:
    - Reverse shell establishments and system reconnaissance.
- Hash extractions:
    - Dumping of SAM database via `hashdump`.
- Credential reuse:
    - Testing of cracked passwords across other services.
- Evidence collection:
    - File extraction.
    - ASCII reports.
    - Hex reports.
    - Metadata export.

Step 5: Digital forensics and evidence handling

A forensic investigation was conducted on a compromised disk image to recover hidden data, verify integrity, and establish chain of custody.

- Integrity verification:

    - MD5 hash generation and comparison for 8-jpeg-search.dd
- Case management:
    - Autopsy case creation with proper metadata and chain of custody.
- File recovery:
    - Keyword searches.
    - File carving.
    - Extra of hidden JPGs.
- Artifact analysis:
    - NTFS metadata examination $MFT, $LogFile,
    - Obfuscation detection, and
    - Timeline reconstruction.

Tools used:

Tool / Framework	Environment / Purpose
nmap	Network scanning, service and OS detection
Metasploit Framework	Exploitation, post-exploitation, payloads
Hydra	SMB brute force
Autopsy	Digital forensic analysis and evidence management
Hashcat	Password hash cracking (demonstration)
md5sum	File integrity verification
Manual testing / curl	Web application exploitation and validation
DVWA	Vulnerable web application target

Ethical and operational considerations

All testing was conducted in a controlled, isolated lab environment to prevent disruption to production systems. Evidence was preserved, and chain of custody was maintained throughout the forensic phase. Findings were fully documented in real-time, and all exploited systems were restored to their original state post-assessment.

CLASSIFICATION DEFINITIONS

Risk Classifications

Level	Score	Description
Critical	10	The vulnerability poses an immediate threat to the organization. Successful exploitation may permanently affect the organization. Remediation should be immediately performed.
High	7-9	The vulnerability poses an urgent threat to the organization, and remediation should be prioritized.
Medium	4-6	Successful exploitation is possible and may result in notable disruption of business functionality. This vulnerability should be remediated when feasible.
Low	1-3	The vulnerability poses a negligible/minimal threat to the organization. The presence of this vulnerability should be noted and remediated if possible.
Informational	0	These findings have no clear threat to the organization, but may cause business processes to function differently than desired or reveal sensitive information about the company.

Exploitation Likelihood Classifications

Likelihood	Description
Likely	Exploitation methods are well-known and can be performed using publicly available tools. Low-skilled attackers and automated tools could successfully exploit the vulnerability with minimal difficulty.
Possible	Exploitation methods are well-known, may be performed using public tools, but require configuration. Understanding of the underlying system is required for successful exploitation.
Unlikely	Exploitation requires deep understanding of the underlying systems or advanced technical skills. Precise conditions may be required for successful exploitation.

Business Impact Classifications

Impact	Description
Major	Successful exploitation may result in large disruptions of critical business functions across the organization and significant financial damage.
Moderate	Successful exploitation may cause significant disruptions to non-critical business functions.
Minor	Successful exploitation may affect few users, without causing much disruption to routine business functions.

Remediation Difficulty Classifications

Difficulty	Description
Hard	Remediation may require extensive reconfiguration of underlying systems that is time consuming. Remediation may require disruption of normal business functions.
Moderate	Remediation may require minor reconfigurations or additions that may be time-intensive or expensive.
Easy	Remediation can be accomplished in a short amount of time, with little difficulty.

ASSESSMENT FINDINGS

#	Finding	Risk Score	Risk
1	MS17-010 EternalBlue SMB remote code execution	**10**	**Critical**
2	SQL injection in Web Application	**10**	**Critical**
3	Unpatched Windows 7 Operating System (EOL)	**9**	**High**
4	Systematic data concealment via obfuscation and compression	**8**	**High**
5	Stored Cross-Site scripting (XSS)	**9**	**High**
6	Unrestricted file upload to Web Server	**8**	**High**
7	Weak password policy and SMB brute-force vulnerability	**7**	**High**
8	Hidden JPG files recovered from deleted and compressed space	**7**	**High**
9	SAM database hash extraction	**7**	**High**
10	Unauthenticated SMB Service Exposure	**7**	**High**
11	Remote Code Execution *via* Meterpreter (Post-exploitation)	**7**	**High**
12	File signature mismatch and extension obfuscation	**5**	**Medium**
13	Microsoft RPC service exposure	**6**	**Medium**
14	Legacy RTSP service exposure	**5**	**Medium**
15	HTTP-API services with default configurations	**4**	**Medium**
16	Insecure default and common passwords	**6**	**Medium**
17	Image integrity verification (informational)	**0**	**Informational**
18	Proper Chain of Custody established (informational)	**0**	**Informational**

1- MS17-10 EternalBlue SMB remote code execution

CRITICAL RISK (10/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Major**
Remediation Difficulty	**Easy**

Synopsis

A critical remote code execution vulnerability in Microsoft SMBv1 servers allows unauthenticated attackers to gain **system**-level control over affected Windows system, leading to complete network compromise, ransomware deployment, and data exfiltration

Analysis

The EternalBlue vulnerability (CVE-2017-0143) was identified on the Windows7 system (192.168.57.20) during a network vulnerability scan using nmap with (smb-vuln-ms17-010) script. The host was found to be running SMBv1 with no patches applied for MS17-010, making it susceptible to a publicly available exploit that requires no authentication or user interaction.

The vulnerability was successfully exploited using the Metasploit Framework, resulting in Meterpreter session with SYSTEM privileges. This level of access enables an attacker to:

- Install persistent backdoors.
- Dump password hashes from the SAM database.
- Move laterally across the network.
- Deploy ransomware or other malware.

```
┌──(root㉿attacker)-[~]
└─# nmap --script vuln 192.168.57.20 -oA vulnerability_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2026-01-20 12:14 EST
Nmap scan report for 192.168.57.20
Host is up (0.00028s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
2869/tcp  open  icslap
5357/tcp  open  wsdapi
10243/tcp open  unknown
MAC Address: 00:50:56:8E:22:BC (VMware)

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Nmap done: 1 IP address (1 host up) scanned in 43.66 seconds
```

***Figure 2.3.1**: refer to 01-1-evidence-net-discovery.pdf p.6*

Impact

- Complete system compromise:
  - Attackers gain SYSTEM-level control over the affected host.
- Credential theft:
  - Ability to extract password hashes and clear text credentials.
- Lateral movement:
  - Use of a compromised host as a pivot point to infiltrate other systems.
- Ransomware and data exfiltration:
  - Deployment of encrypting malware or theft of sensitive data.
- Operational disruption:
  - Potential for total system downtime and recovery costs.

Recommendations

- Immediate actions MS17-010:
    - Download and install the official Microsoft patch from the Security Update Guide.
    - Validate patch installation *via* systeminfo or Windows Update history
- Disable SMBv1 Protocol:
    - Run the following Powershell command as Administrator:

```
Set-SmbServerConfiguration -EnableSMB1Protocol $false -Force
```

    - Verify SMBv1 is disable *via*:

```
Get-SmbServerConfiguration | Select EnableSMB1Protocol
```

- Network segmentation:
    - Isolate unpatched system in a restricted VLAN
    - Implement firewall rules to block SMB (TCP/445) traffic from untrusted networks.

Short-term improvements

1. Enable SMB signing:
    - Configure Group Policy to require SMB packet signing.
2. Implement endpoint detection and response:
    - Deployment EDR solutions to detect exploitation attempts and anomalous SMB activity.
    - Enable real-time alerting for EternalBlue exploit patterns.
3. Vulnerability management:
    - Schedule regular vulnerability scans focusing on SMB-related CVEs.
    - Establish a patch management policy for critical vulnerabilities policy within 72 hours of release.

Long-term improvements

1. Upgrade End-of-Life systems:
    - Replace Windows 7 and other unsupported Operating Systems with modern, supported versions.
2. Network hardening:

- Implement application whitelisting to block unauthorized executables.
- Deploy intrusion prevention system (IPS) with rules to block EternalBlue exploits.

3. Security awareness training:
    - Train IT staff on identifying and responding to SMB-based attacks.
    - Conduct a tabletop exercises simulating ransomware outbreaks stemming from EternalBlue.

Verification steps

After remediation, verify the fix by:

1. Running the nmap script again

```
nmap --script smb-vuln-ms17-010 192.168.57.20
```

Output should show State: NOT VULNERABLE.

2. Conducting a manual exploitation attempt with Metasploit to confirm the system is no longer exploitable.
3. Reviewing Windows Event Logs for SMB-related errors or warning post-patch.

2- SQL injection in Web Application

CRITICAL RISK (10/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Major**
Remediation Difficulty	**Easy**

Synopsis

A SQL injection vulnerability was identified in the Damn Vulnerable Web Application (DVWA), allowing attackers to execute arbitrary SQL commands on the backend database. This can lead to unauthorized data access, data manipulation, authentication bypass, and full database.

Analysis

During authenticated testing of DVWA (192.168.57.30), a SQL injection flaw was discovered in the user input field of the SQL injection module. By injecting malicious SQL payloads, it was possible to:

- Extract database schema, table names, and column data.
- Retrieve sensitive information such as usernames and passwords.
- Perform authentication bypass.
- Execute database administration commands (depending on database user privileges).

The vulnerability is due to lack of input validation and parameterized queries, allowing user controlled input to be directly concatenated into SQL statements.

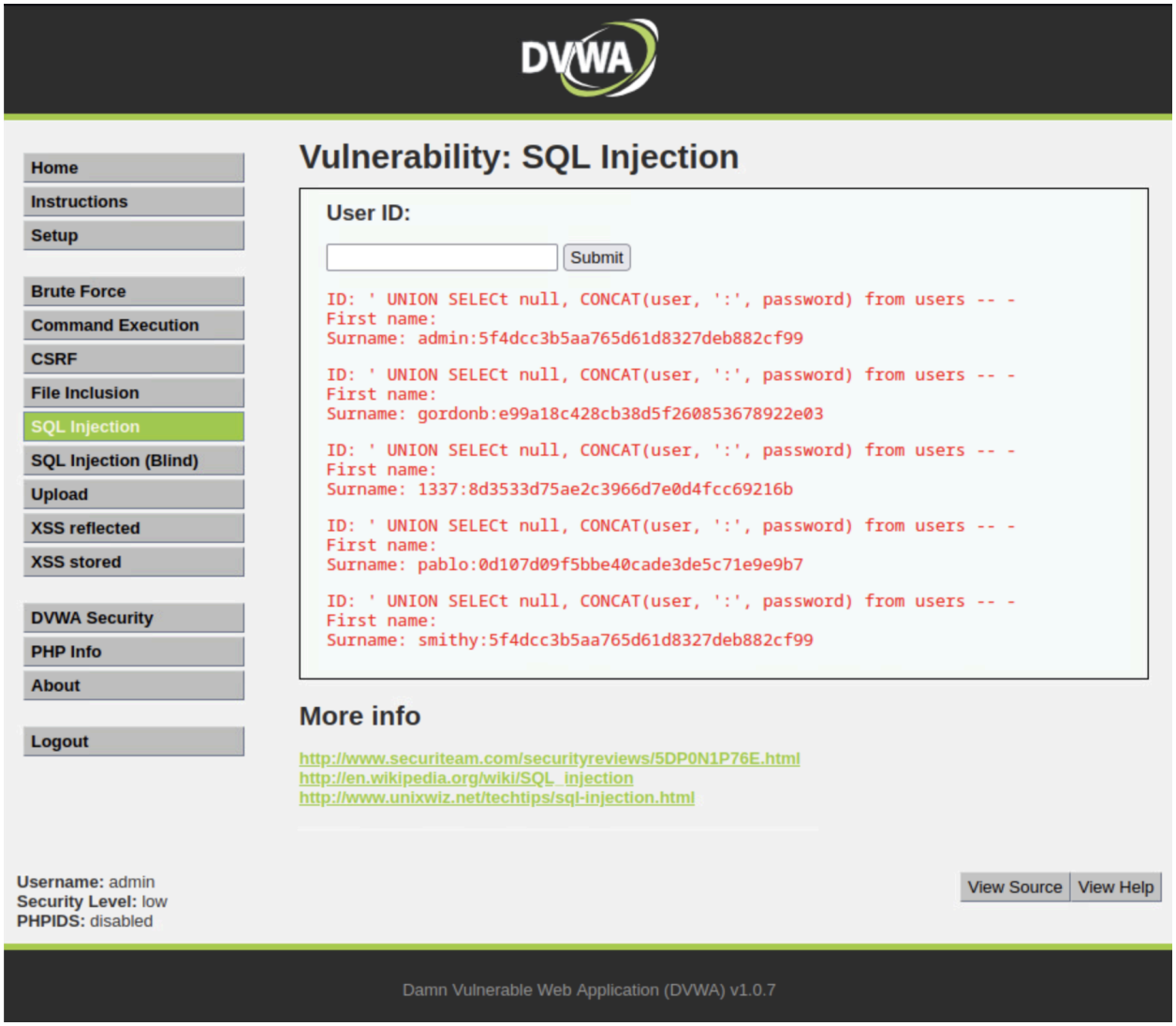


***Figure 2.3.1**: refer to 02-DVWA-evidence.pdf p.1-13*

Result

- Extracted database user credentials (MD5 hashes).
- Retrieve database version and schema information.
- Demonstrated complete database enumeration capability.

Impact

- Data Breach:
    - Unauthorized access to sensitive information stored in the database.
- Authentication bypass:

    - Ability to log in without valid credentials.
- Data manipulation:
    - Modify, delete, or insert records in the database.
- Privilege escalation:
    - Potential to execute commands on the underlying server.
- Reputational damage:
    - Loss of customer trust and potential regulatory penalties.
- Financial loss:
    - Costs associated with breach notification, forensic investigation, and remediation.

Recommendations

1. Implement parameterized queries/prepared statements:
    - Rewrite all SQL queries to use parameterized statements instead of string concatenation
2. Input validation and sanitization:
    - Implement strict input validation (whitelist approach) for all user inputs.
    - Use built-in sanitization functions specific to the company's programming language.
3. Web application firewall (WAF):
    - Deploy or configure WAF rules to block SQL injections patterns.
    - Enable logging and alerting for SQL injection attempts.

Short-term improvements

1. Database permission hardening:
    - Apply the principle of least privilege to database accounts.
    - Restrict application database users to only necessary operations (`SELECT`, `INSERT`, `UPDATE` as needed)
    - Revoke administrative privileges (`DROP`, `CREATE`, `ALTER`, etc.) from application accounts.
2. Error handling configuration:
    - Disable detailed error messages in production environments.
    - Implement custom error pages that don't reveal database structure or query details.

3. Regular security testing:
    - Implement automated SQL injection scanning in CI/CD pipelines.
    - Conduct regular manual penetration testing focusing on injection vulnerabilities.

Long-term improvements

1. Developer security training:
    - Conduct secure coding training focusing on injection prevention.
    - Implement code review checklists that include SQL injections checks.
    - Use static application security testing (SAST) tools.
2. Database encryption:
    - Implement transparent data encryption (TDE) for sensitive data at rest.
    - Consider column-level encryption for highly sensitive information.
3. Incident response planning:
    - Develop specific playbooks for responding to SQL injection incidents.
    - Implement database activity monitoring (DAM) solutions.

Verification steps

After remediation, verify the fix by:

1. Manual testing:
    - Attempt SQL injection using previously successful payloads.
2. Automated scanning:
    - Run SQL injection scanners (SQLMap, OWASP ZAP, Burp Suite)
3. Code review:
    - Validate all database interactions use parameterized queries.
4. Log monitoring:
    - Review application and database logs for injection attempts.
5. Running verification command:

```
sqlmap -u "http://192.168.57.30/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="se
curity=low; PHPSESSID=..." --batch
```

Output should show State: NOT VULNERABLE.

3- Unpatched Windows 7 Operating System

HIGH RISK (9/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Major**
Remediation Difficulty	**Moderate**

Synopsis

The targeted system (192.168.57.20) is running Windows 7 Professional, an operating system that reached End of Life (EOL) on January 14, 2020. Running EOF software exposes the organization to unpatched vulnerabilities, lack of security updates, and increased risk of compromise from known exploits.

Analysis

Nmap OS fingerprinting identified the target as running Microsoft Windows 7 Professional. Windows 7 is no longer supported by Microsoft, meaning:

- No security updates or patches are released.
- No technical support is available.
- Known vulnerabilities remain unpatched indefinitely.
- Compliance violations may occur in regulated industries.

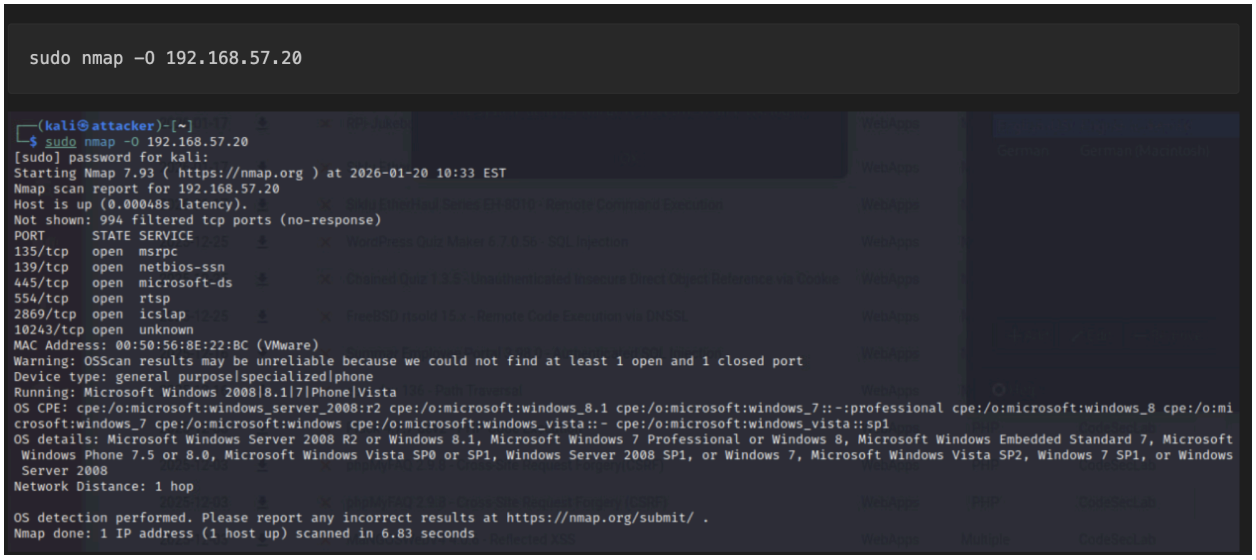


***Figure 2.3.1**: refer to 01-1-evidence-net-discovery.pdf p2*

Additional verification

- `systeminfo` command *via* Meterpreter session confirmed Windows 7 Professional.
- Lack of recent Windows Update history.
- Presence of multiple unpatched vulnerabilities.

Impact

- Unpatched vulnerabilities:
  - Exposure to hundreds of known CVEs with available public exploits.
- Compliance violations:
  - Failure to meet PCI DDSS, HIPAA, GDPR, or other regulatory requirements.
- Increased attack surface:
  - Modern security features not available (Windows Defender ATP, Credential Guard, etc.)
- Ransomware risk:
  - High susceptibility to ransomware attacks targeting EOL systems.
- Business continuity risk:
  - System failures without vendor support.
- Lateral movement:
  - Compromised EOL system as entry point to entire network.

Recommendations

1. Immediate isolation:
   - Move Windows 7 systems to a segregated VLAN with restricted access.
   - Implement firewall rules to limit inbound/output traffic.
   - Monitor these systems closely for suspicious activities.
2. Compensating controls:
   - Deploy additional endpoint protection (third-party EDR/XDR).
   - Implement application whitelisting.
   - Enable enhanced logging and monitoring.
3. Risk assessments documentation:
   - Document the business justification for keeping EOL systems.
   - Obtain formal risk acceptance from management.

- Update risk register with mitigation control.

Short-term improvements

1. Migration planning:
    - Inventory all EOL Windows systems in the environment.
    - Develop migration plan to Windows 10/11 or Windows Server 2022.
    - Prioritize systems based on criticality and exposure.
2. Extended security updates (ESU):
    - If migration is not immediately possible, purchase Microsoft ESU licenses.
    - ESUS for Windows 7 ended January 2023; consider third-party security solutions.
3. Enhanced monitoring:
    - Implement Security Information and Event Management (SIEM) rules for EOL systems.
    - Configure alerts for any exploitation attempts.
    - Regular vulnerability scanning with focus on EOL software.

Long-term strategy

1. Complete migration:
    - Execute phased migration to supported operating systems.
    - Test application for compatibility with newer Windows versions.
    - Update documentation and procedures.
2. Lifecycle management policy:
    - Establish formal software lifecycle management policy.
    - Implement automated discovery of EOL software.
    - Create a dashboard for tracking software end-of-life dates.
3. Modern security architecture:
    - Implement Zero Trust architecture.
    - Deploy modern endpoint protection platforms.
    - Utilize cloud-based security solutions.
4. Staff training:
    - Train IT staff on risks of EOL software.
    - Develop procedures for identifying and reporting EOL systems.

- Regular review of software inventories.

Verification steps

After remediation, verify:

1. Inventory verification:
    - No Windows 7 system in production without approved exceptions.
2. Patch status:
    - All systems receiving regular security updates.
3. Monitoring:
    - SIEM alerts configured for EOL system detection.
4. Documentation:
    - Risk acceptance forms completed for any remaining EOL systems.
5. Verify Windows version remotely:

```
nmap -sV --script smb-os-discovery 192.168.57.20
```

Expected result: Should not detect Windows 7 in production environment.

4- Systematic data concealment *via* obfuscation and compression

HIGH RISK (8/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Major**
Remediation Difficulty	**Moderate**

Synopsis

Forensic analysis revealed sophisticated data concealment techniques used by thread actors to hide sensitive JPG files through file extension obfuscation, compression in multiple archive formats, and manipulation of NTFS metadata. This indicates intentional efforts to evade detection and maintain persistence while exfiltration or hiding data.

Analysis

During forensic examination of the disk image `8-jpeg-search.dd`, the following concealment methods were discovered:

1. File extension obfuscation:
    - JPG files renamed with misleading extensions:
        - `.dat`
        - `.hmm`
        - `.boo`
2. Compression and archiving:
    - Hidden within multiple archive formats:
        - `.zip`
        - `.tar.gz`
    - Nested compression to evade basic scanning
        - Example: `file8.jpg` found within `C:\archive\file8.zip`
3. NTFS metadata Manipulation:
    - References to hidden files found in `$LogFile` and `$MFT`.
    - Timestamps manipulated to obscure file creation/modification times.

- Files marked as “invalid” or placed in misleading directories
    - C:\invalid\
    - C:\misc\

4. Malicious document integration:
    - file12.doc containing references to hidden JPG files.
    - Potential macro-based malware for staged payload delivery

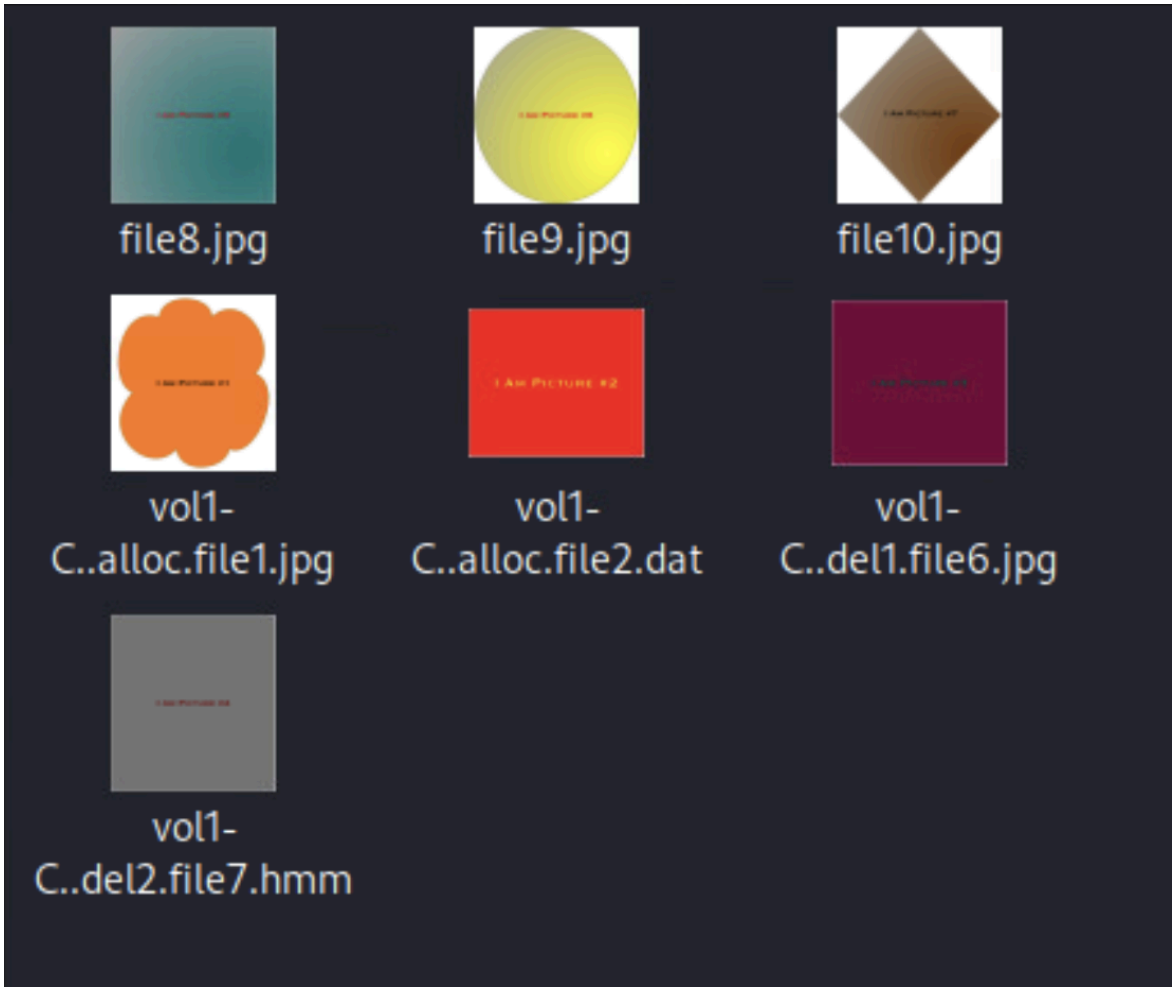


***Figure 2.3.1**: refer to recovered files in 04-forensics-report-evidence.pdf p.22*

Evidence - forensic findings:

- 5 hidden JPG files recovered from:
    - Deleted space
        - C:\del1\file6.jpg
    - Compressed archives

        - file8.zip
        - file9.boo
        - file10.tar.gz
    - Obfuscated locations
        - C:\\alloc\file2.dat
- NTFS artifacts:
    - Entries in $MFT
        - file1.jpg
        - file3.jpg
        - file4.jpg
        - file6.jpg
- Suspicious files:
    - In C:\misc\
        - file13.dll.here

Findings summary

Files	Origina location	State
`file1.jpg`	`C:\alloc\`	Recovered from assigned space
`file2.dat`	`C:\alloc\`	Recovered (extension `dat`) - possible `jpg` renamed
`file3.jpg`	`C:\invalid\`	Marked as "invalid" - damaged/corrupted format
`file4.jpg`	Not found	Unexistent, overwritten or obfuscated file
`file5.trf`	`C:\invalid\`	Extension `.trf` - unknown format
`file6.jpg`	`C:\del1\`	Recovered from deleted space
`file7.hmm`	`C:\del2\`	Recovered (extension `hmm`) - unknown format
`file8.zip`	`C:\archive\`	Contains `file8.jpg`
`file9.boo`	`C:\archive\`	Contains `file9.jpg` (obfuscated extension)
`file10.tar.gz`	`C:\archive\`	Contains `file10.jpg`
`file13.dll.here`	`C:\misc\`	Suspicious `dll` file - suspicious executable file

***Figure 2.3.1**: refer to recovered files in 04-forensics-report-evidence.pdf p.25*

Impact

- Data exfiltration:
    - Hidden files may contain stolen sensitive information.
- Persistence mechanism:

    - Concealed files could serve as backdoors or secondary payloads.
- Evasion of security controls:
    - Bypasses traditional AV and DLP solutions.
- Incident response complexity:
    - Increases time and effort for forensic investigation.
- Regulatory non-compliance:
    - Hidden sensitive data may violate data protection regulations.
- Reputational damage:
    - Evidence of successful data concealment indicates security failure.

Recommendations

1. Enhanced file monitoring:
    - Implement File Integrity Monitoring (FIM) for critical directories.
    - Monitor for file extension changes and compression activities.
    - Alert on creation of archives in user directories.
2. Forensic investigation expansion:
    - Conduct through investigation of all systems for similar concealment patterns.
    - Use forensic tools to scan for hidden data in unallocated space.
    - Examine NTFS metadata across all endpoints.
3. File analysis automation:
    - Implement automated file signature verification.
    - Deploy tools for automatic archive inspection.
    - Create scripts to detect extension-header mismatches.

Short-term improvements

1. Advanced threat detection:
    - Deploy EDR solutions with file analysis capabilities.
    - Implement behavior-based detection for data hiding activities.
    - Configure alerts for:
        - Multiple file renames in short timeframes.
        - Compression tool usage outside normal patterns.
        - NTFS metadata modification attempts.
2. Forensic readiness program:

- Establish baseline forensic imaging procedures.
- Train IT staff on data concealment detection techniques.
- Create incident response playbooks for data hiding incidents.

3. File analysis automation:
    - Implement automated file signature verification.
    - Deploy tools for automatic archive inspections.
    - Create scripts to detect extension-header mismatches.

Long-term strategy

1. Comprehensive data protection strategy:
    - Implement Data Classification and tagging.
    - Deploy Data Loss Prevention (DLP) with advanced content inspection.
    - Consider encryption for sensitive data at rest and in transit.
2. Threat intelligence integration:
    - Subscribe to threat intelligence feeds focusing on data exfiltration techniques.
    - Update security controls based on emerging concealment methods.
    - Participate in information sharing communities.
3. Security architecture review:
    - Implement Zero Trust Architecture to limit data movement.
    - Deploy Network Segmentation to contain potential exfiltration.
    - Consider Cloud Access Security Broker (CASB) for cloud data protection.
4. Staff training and awareness:
    - Train security team on advanced data concealment techniques.
    - Conduct tabletop exercises simulating data exfiltration scenarios.
    - Develop a forensic investigation certification program.

Verification Steps

1. Regular forensic drills:
    - Conduct quarterly forensic investigations on test systems.
2. Detection testing:
    - Validate detection rules with simulated concealment attempts.
3. Compliance audits:
    - Include data concealment checks in security audits.

4. Tool validation:
    - Regularly update and test forensic and detection tools.
5. Verify file integrity monitoring:

```
# Check FIM alerts for last 7 days
Get-WinEvent -FilterHashtable @{LogName='Security'; ID='4660'} | Where-Object {$_.TimeCreat
ed -gt (Get-Date).AddDays(-7)}
```

5- Stored Cross-Site Scripting (XSS)

HIGH RISK (9/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Moderate**
Remediation Difficulty	**Easy**

Synopsis

A Stored Cross-Site scripting (XSS) vulnerability was identified in the Damn Vulnerable Web Application (DVWA) guestbook functionality. This vulnerability allows attackers to inject malicious JavaScript code that is permanently stored on the server and executed automatically when other users view the affected page, leading to session hijacking, credential theft, and complete account compromise.

Analysis

During authenticated testing of DVWA (192.168.57.30), a Stored XSS vulnerability was discovered in the guestbook/comment functionality. The application fails to properly sanitize user input before storing and displaying it, allowing persistent script injection.

Attack scenario:

1. Attacker injects malicious JavaScript into guestbook entry.
2. Entry is saved to the database without sanitization.
3. All users viewing the guestbook page execute the malicious script.
4. Attacker can:
    - Steal session cookies.
    - Redirect users to phishing or malicious sites.
    - Perform actions on behalf of authenticated users.
    - Install malware *via* drive-by downloads.

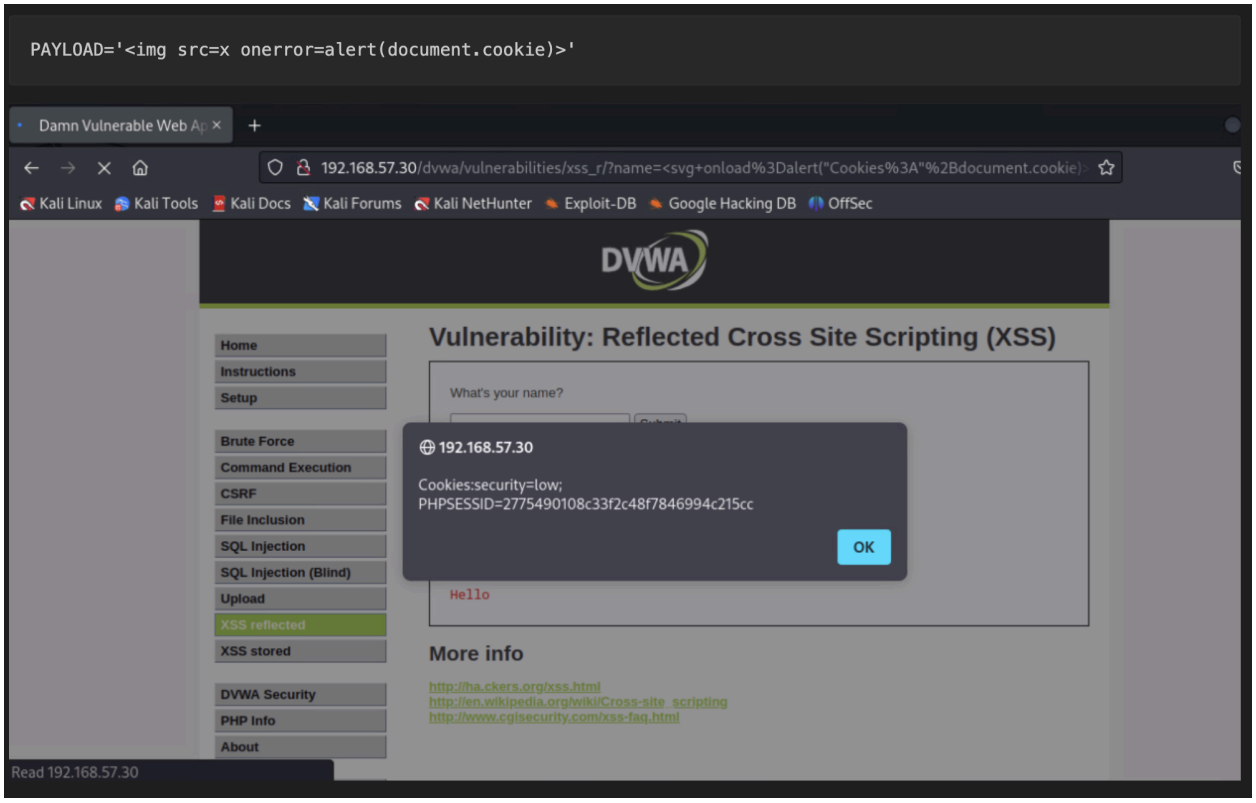


***Figure 2.3.1**: refer to 02-DVWA-evidence.pdf p.13-20*

Impact

- Session hijacking:
  - Steal authenticated session cookies to impersonate users.
- Credential theft:
  - Capture login credentials *via* fake login forms.
- Account takeover:
  - Complete compromise of user accounts.
- Malware distribution:
  - Redirect users to malicious sites or deliver malware.
- Defacements:
  - Modify website appearance/content.
- Data theft:
  - Exfiltrate sensitive data displayed on the page
- Reputational damage:
  - Loss of user trust and brand reputation

Recommendations

1. Input validation and output encoding:
    - Implement context-aware output encoding for all user-supplied data.
    - Use established libraries (OWASP Java Encoder, PHP html-specialchars)
2. Content Security Policy (CSP)
    - Implement strict CSP headers to mitigate XSS impact.
    - Report-only mode initially: `Content-Security-Policy-Report-Only`
3. HTTP-only and Secure cookies:
    - Ensure all session cookies have `HttpOnly` and Secure flags.

Short-term improvements

1. Automated security testing:
    - Integrate SAST tools into CI/CD pipeline to detect XSS vulnerabilities.
    - Implement DAST scanning for production applications.
    - Use OWASP ZAP or Burp Suite for regular scanning.
2. Framework security features:
    - Utilize built-in XSS protection features of modern frameworks:
        - Angular: Automatic sanitation.
        - React: JSX escaping.
        - Vue.js: Text interpolation.
    - Avoid `innerHTML` and similar dangerous methods
3. Web Application Firewall (WAF):
    - Configure WAF rules to block XSS payloads.
    - Enable loggings of blocked attempts.
    - Regular rule updates.

Long-term strategy

1. Secure Development Lifecycle:
    - Mandatory secure coding training for all developers.
    - Implement security requirements in-design phase.
    - Code review checklists including XSS prevention.
2. Client-Side Protection:
    - Implement Subresource Integrity (SRI) for external scripts.

- Use Trusted types API (modern browsers).
- Consider using libraries like DOMPurify for HTML sanitization.

3. Monitoring and incident response:
    - Monitor for XSS attempts *via* web server logs.
    - Implement alerting for successful XSS exploitation.
    - Develop incident response playbook for XSS incidents.

Verification Steps

After remediation, verify the fix by:

1. Manual testing:
    - Attempt to inject XSS payloads previously successful.
2. Automated scanning:
    - Run XSS scanners (XSStrike, XSSer, OWASP ZAP).
3. Code review:
    - Validate all output using proper encoding.
4. CSP Validation:
    - Check CSP headers using securityheaders.com
5. Cookie audit:
    - Verify HttpOnly and Secure flags on all session cookies.
6. Testing commands:

```
# Test with curl for CSP headers
curl -I http://192.168.57.30/dvwa/ | grep -i "content-security-policy"

# Test XSS with simple payload
curl -X POST http://192.168.57.30/dvwa/vulnerabilities/xss_s/ \
  -d "txtName=<script>alert(1)</script>&mtxMessage=test&btnSign=Sign+Guestbook"
```

Expected result: Script tags should be properly encoded in response.

False Positive/Negative considerations:

- Modern frameworks may have built-in protection.
- WAFs might block exploitation but not fix the root cause.
- Consider DOM-based XSS which requires client-side remediation.

6- Unrestricted file upload to Web Server

HIGH RISK (8/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Moderate**
Remediation Difficulty	**Easy**

Synopsis

An unrestricted file upload vulnerability was identified in the DAMN Vulnerable Web Application (DVWA), allowing attackers to upload malicious files (including PHP webshells) to the web server. This vulnerability enables remote code execution, server compromise, and complete control over the affected web application and underlying system.

Analysis

During testing of the DVWA file upload functionality (192.168.57.30/dvwa/vulnerabilties/upload/), it was discovered that the application accepts file uploads without proper validation of:

- File types/extensions.
- File content/magic bytes.
- File size limits.
- Dangerous file name/paths

Attack execution

1. Webshell creation:
   Created a PHP webshell (webshell.php) with backdoor functionality.
2. File upload:
   Successfully uploaded the malicious file to the server.
3. Execution:
   Accessed the uploaded file via web browser, achieving remote code execution.
4. Privilege escalation:
   Used the webshell to execute system commands and establish a Meterpreter session.

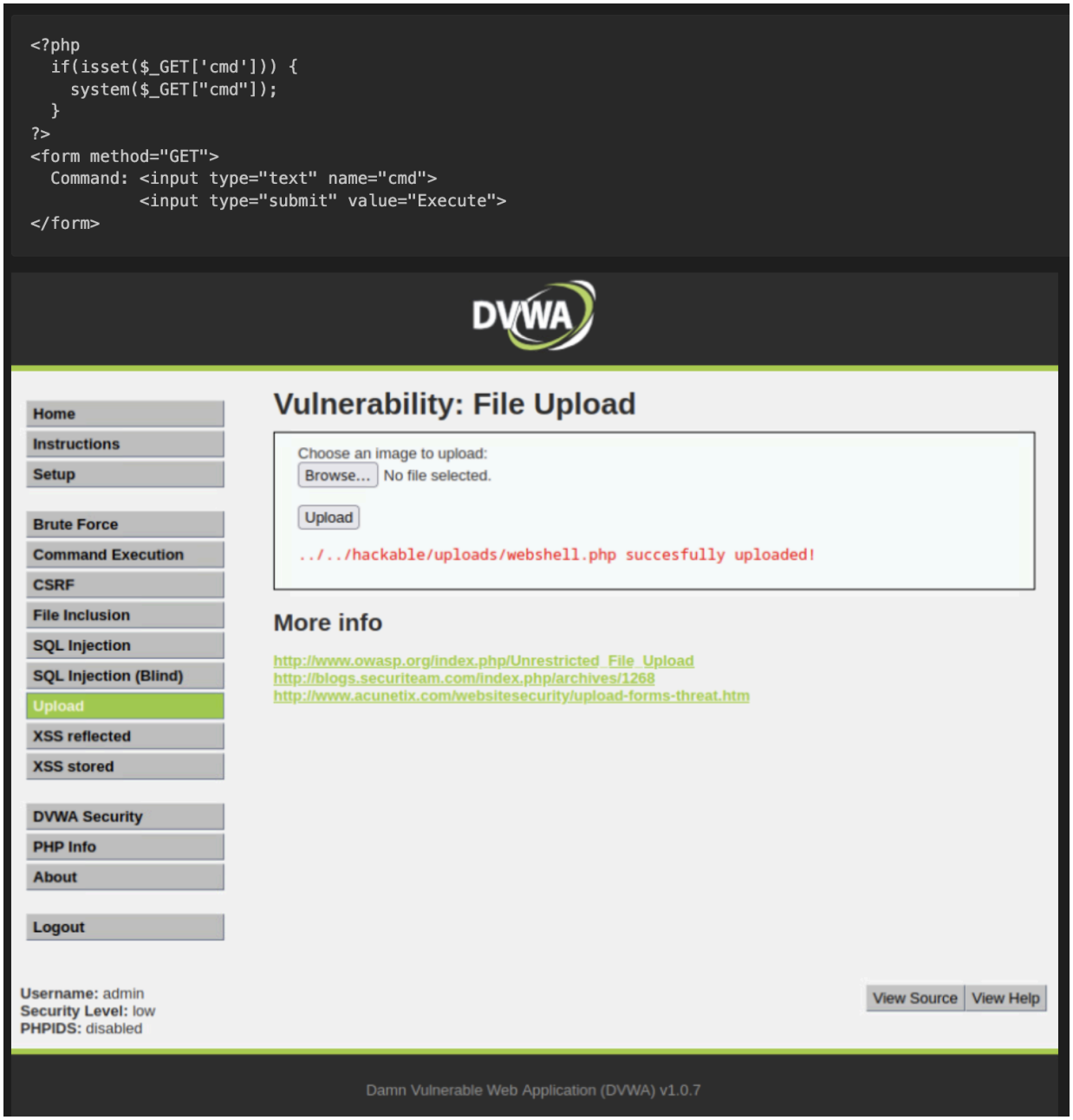


***Figure 2.3.1**: refer to 02-DVWA-evidence.pdf p. 20-23*

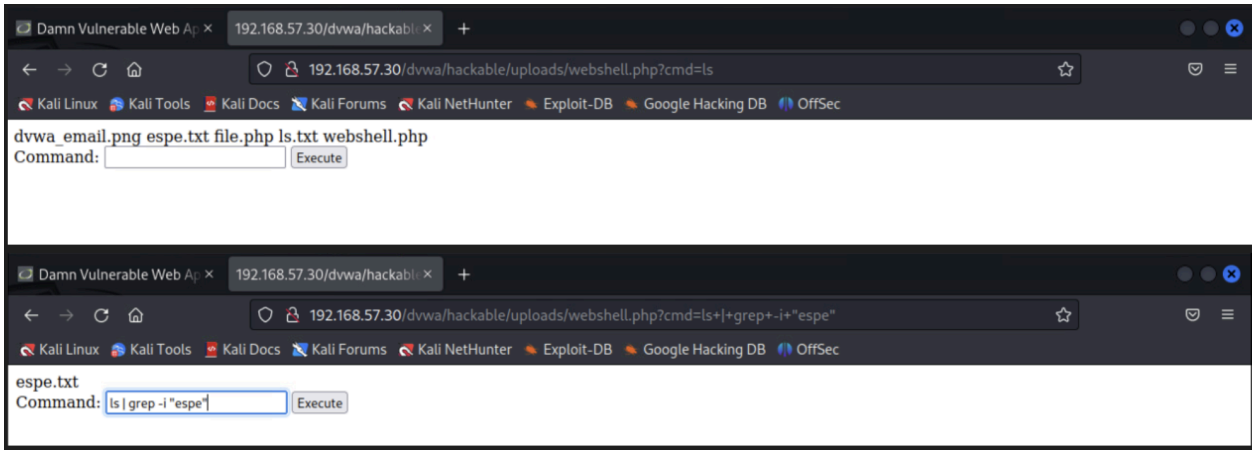


***Figure 2.3.1**: refer to 02-DVWA-evidence.pdf p. 20-23*

Impact

- Remote code execution:
  - Complete control over the Web Server.
- Data breach:
  - Access to databases,
  - Configuration files, and
  - Sensitive data.
- Server compromise:
  - Install persistent backdoors,
  - Ramsonware, or
  - Mining software.
- Lateral movement:
  - Use compromised server to attack internal network
- Website defacement:
  - Modify or delete website content.
- Denial of Service:
  - Overwrite critical system files.
- Reputational damage:
  - Loss of customers' trust and business credibility.

Recommendations

1. Implement strict file type validation:
    - Whitelist approach: only allow specific, safe file extensions.
    - Server-side validation: not relies on client-side checks.
2. File content verification:
    - Check file magic bytes (MIME type) not just extensions.
    - Use finfo_file() or getimagesize() for image validation
3. Secure file storage:
    - Store uploaded files outside the web root directory.
    - Use random filenames to prevent direct access.
    - Implement proper permissions (no execute permissions).

Short-term improvements

1. File uploaded security controls:
    - Implement file size limits.
    - Scan uploaded files with antivirus/antimalware.
    - Use secure file upload libraries/framework.
2. Web application firewall (WAF):
    - Configure rules to block malicious file uploads.
    - Implement file type filtering at WAF level.
    - Enable logging and alerting for upload attempts.
3. Server hardening:
    - Configure PHP to disable dangerous functions
    - Use open_basedir restriction.
    - Implement Suhosin or other PHP hardening solutions.

Long-term strategy

1. Secure architecture design:
    - Implement separate file storage servers.
    - Use content delivery networks (CDN) for static files.
    - Consider cloud storage service with built-in security.
2. Developer training:
    - Secure file handling training for all developers.

- Code review checklists for file upload functionality.
- Regular security testing of upload features.

3. Monitoring and incident response:
    - Monitor for suspicious file uploads.
    - Implement file integrity monitoring.
    - Develop incident response playbook for upload compromises.

Verification Steps

After remediation, verify the fix by:

1. Manual testing:
    - Attempt to upload malicious files with various extensions.
2. Automated scanning:
    - Use tools like Burp Suit, OWASP ZAP file upload scanners.
3. Code review:
    - Validate all file upload endpoints and implement proper validation.
4. Server configurations:
    - Check web server and PHP configurations
5. Penetration testing:
    - Conduct focused file upload security testing
6. Testing commands

```
# Test with curl
curl -X POST http://192.168.57.30/dvwa/vulnerabilities/upload/ \
  -F "uploaded=@shell.php" \
  -F "Upload=Upload" \
  -b "security=low; PHPSESSID=..."

# Check file permissions
ls -la /var/www/uploads/
```

Expected results:

- Malicious file uploads should be rejected.
- Uploaded files should have safe permissions (no execute).
- Files should be stored outside web root or with proper access controls.

Other considerations

- File parsing vulnerabilities:
    - Some file types (.pdf, .docx) may contain embedded malicious content.
- ZIP bombs:
    - Protect against decompression bombs.
- Race conditions:
    - Ensure atomic operations when processing uploads.
- Metadata stripping:
    - Remove EXIF and other metadata from uploaded files.

7- Weak Password Policy and SMB brute-force vulnerability

HIGH RISK (7/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Major**
Remediation Difficulty	**Moderate**

Synopsis

A weak password policy combined with exposed SMB services enabled successful brute-force attacks against Windows 7 systems. Attackers may be able to compromise multiple user accounts using common passwords, demonstrating inadequate passwords complexity requirements and lack of account lockout protections.

Analysis

Following initial exploitation *via* EternalBlue, post-exploitation activities revealed critical authentications

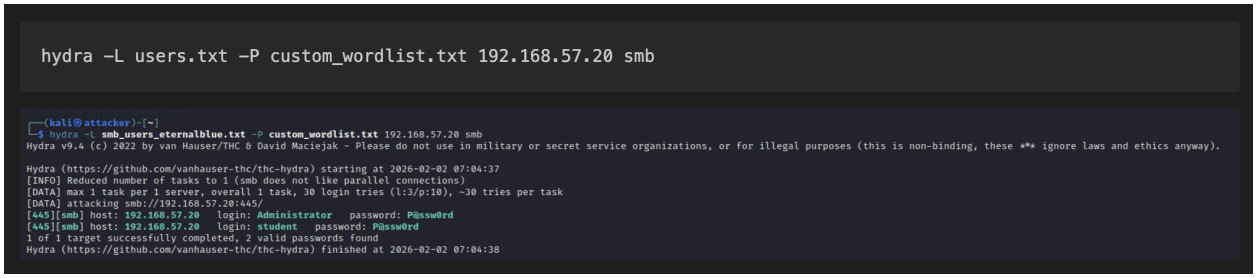


***Figure 2.3.1**: refer to 03-password-security-evidence.pdf p.2-5*

Impact

- Unauthorized access:
    - Complete system access *via* compromised credentials.
- Privilege escalation:
    - Administrator account compromise provides SYSTEM-level control.

- Lateral movement:
    - Use credential to access other systems (credential reuse).
- Persistence:
    - Establish persistent access independent of vulnerabilities.
- Data breach:
    - Access to sensitive files and data.
- Compliance violations:
    - Failure to meet password policy requirements (PCI DSS, HIPAA, etc.)
- Increased attack surface:
    - Multiple entry points for attackers.

Recommendations

1. Enforce strong password policy:
    - Minimum length policy: n-characters
    - Complexity requirements: upper, lower, number, special characters.
    - Password history: prevent reuse of last m-passwords.
    - Maximum age: k-days
2. Implement account lockout policy:
    - Threshold over failed attempts.
    - Lockout duration: x-minutes.
    - Reset counter: after x-minutes.
3. Reset compromise credentials:
    - Immediately change passwords for all compromised accounts.
    - Enforce password change at next logon.
    - Monitor for suspicious activity from compromised accounts.

Short-term improvements

1. Multi-Factor Authentication (MFA):
    - Implement MFA for all administrative accounts.
    - Consider MFA for all user accounts accessing sensitive resources.
    - Use Windows Hello for Business of third-party MFA solutions.
2. SMB Service Hardening:
    - Disable SMBv1 protocol.

- Require SMB access *via* firewall rules.
- Implement SMB encryption (like SMB 3.0+)

3. Password Auditing and Monitoring:
    - Regular password strength audits.
    - Implement password breach detection (HaveIBeenPwoned API)

Long-term strategy

1. Passwordless authentication:
    - Implement Windows Hello for Business.
    - Deploy FIDO2 security keys.
    - Consider certificate-based authentication.
2. Privileged Access Management (PAM):
    - Implement just-in-time administrative access.
    - Use privileged identity management solutions.
    - Separate administrative and standard user accounts.
3. Security awareness training:
    - Regular password security training for all users.
    - Phishing awareness to prevent credentials theft.
    - Password manager implementation and training.

Additional recommendations

1. Passwords managers:

   Provide enterprise password managers to users.
2. Regular audits:

   Quarterly password policy compliance audits.
3. Incident response:

   Develop playbooks for credentials compromise incidents.
4. Backup authentication:

   Ensure backup administrative accounts with strong passwords.
5. Monitoring:

   Real-time alerting for authentication anomalies.

Verification Steps

After remediation, verify the fix by:

1. Password Policy Enforcement:

```
net accounts
```

Expected: Minimum password length, lockout threshold.

2. SMB configuration:

```
Get-SmbServerConfiguration | Select EnableSMB1Protocol, RequireSecuritySignature, Encrypt
Data
```

Expected: SMBv1 disabled, signing required, encryption enabled.

3. Account logout testing:
    - Attempt k-failed logins.
    - Verify account locks for x-minutes.
    - Confirm automatic unlock after lockout period.
4. Brute-force protection testing:

```
hydra -L test_users.txt -P test_passwords.txt 192.168.57.20 smb
```

Expected: account lock after n-attempts.

Compliance considerations

- NIST SP 800-63B: Digital identity guidelines.
- PCI DSS requirements 8: Identify and authenticate access to system components.
- ISO 27001 A.9.4: System and application access control.
- CDPR: Appropriate technical measures for security.

8- Hidden JPG files recovered from *deleted* and *compressed* space

HIGH RISK (7/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Major**
Remediation Difficulty	**Hard**

Synopsis

This finding focuses specifically on the recovery of JPG files from deleted storage space and compressed archives, demonstrating successful data recovery despite intentional deletion and compression by threat actors. While related to systematic concealment, this finding highlights forensic recovery capabilities and data persistence risks.
This finding complements **#4** by demonstrating that *while* concealment occurs, forensic recovery is possible with proper tools and procedures

Analysis

Specific recovery achievements:

1. From deleted space:
    - `file6.jpg` successfully recovered from `C:\del1\` despite deletion.
    - `file7.jpg` successfully recovered from `C:\del2\` despite deletion.
2. From compressed archives:
    - `file8.jpg` from `C:\archive\file8.zip`
    - `file9.jpg` from `C:\archive\file9.boo`
    - `file10.jpg` from `C:\archive\file10.tar.gz`
3. Forensic significance:
    - Data remains recoverable even after deletion and compression.

Impact

- Data persistence risk:

    - Deleted/compressed files remain recoverable.
- Evidence preservation:
    - Critical for incident investigation.
- Compliance requirements:
    - Data recovery capabilities needed for legal/regulatory compliance.
- Incident response value:
    - Demonstrated forensic recovery effectiveness.

Recommendations

1. Implement secure deletion:
   Use wiping tools for sensitive data destruction.
2. Encrypt sensitive archives:
   Require encryption for compressed sensitive data.
3. Forensic tool validation
   Ensure recovery tools remain effective.

Short-term improvements

1. Data Lifecycle management:
   Formal policies for secure deletion.
2. Forensic training:
   Staff training on data recovery techniques.
3. Encryption enforcement:
   Mandate encryption for archiving sensitive data.

Long-term strategy

1. Forensic readiness:
   Maintain forensic investigation capabilities.
2. Secure archiving:
   Implement secure archival solutions.
3. Regular testing:
   Test data recovery procedure periodically.

Verification

- Test data recovery from *deleted* space quarterly.
- Validate forensic tool effectiveness.
- Conduct secure deletion audit.

9- SAM Database hash extraction

HIGH RISK (7/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Major**
Remediation Difficulty	**Hard**

Synopsis

Successful extraction of password hashes from the Security Account Manager (SAM) database following EternalBlue exploitation, demonstrated the ability to obtain credential hashes that can be cracked offline or used in pass-the-hash attacks, leading to lateral movement and privilege escalation across the network.

Analysis

Hash extraction process:

1. Initial compromise:

   System access gained via EternalBlue exploit.
2. Privilege escalation:

   Archived SYSTEM-level privileges via Meterpreter.
3. Hash extraction:

   Used hashdump Meterpreter module to extract NTLM hashes:

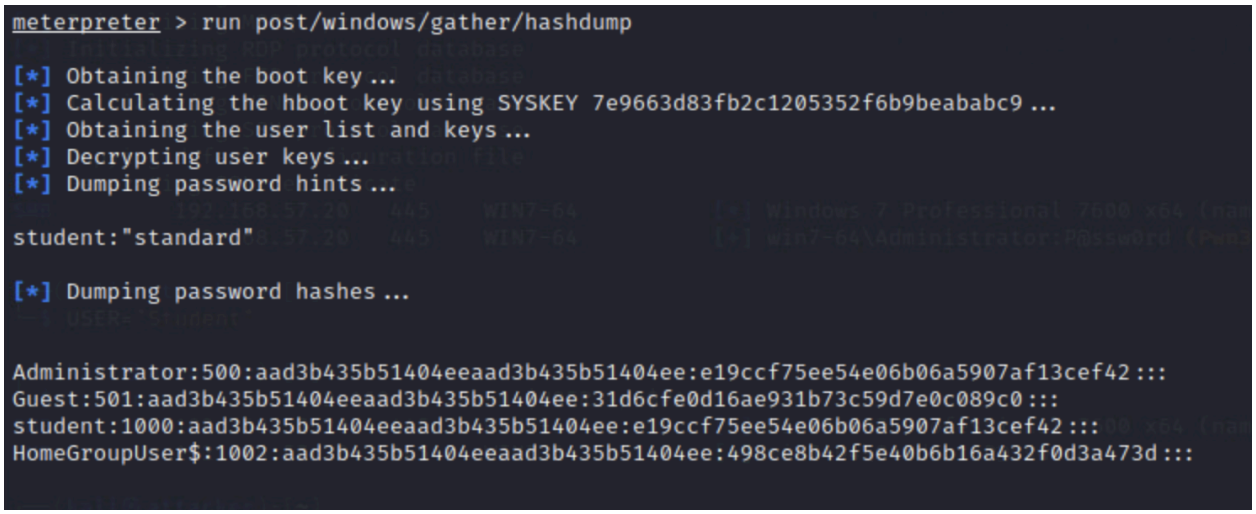


***Figure 2.3.1**: refer to 03-password-security-evidence.pdf p.5*

Hash details:

- Format:

  NTLM hashes (LM:NTLM format)

- Cracking demonstration:

  Hashes successfully cracked using Hashcat:

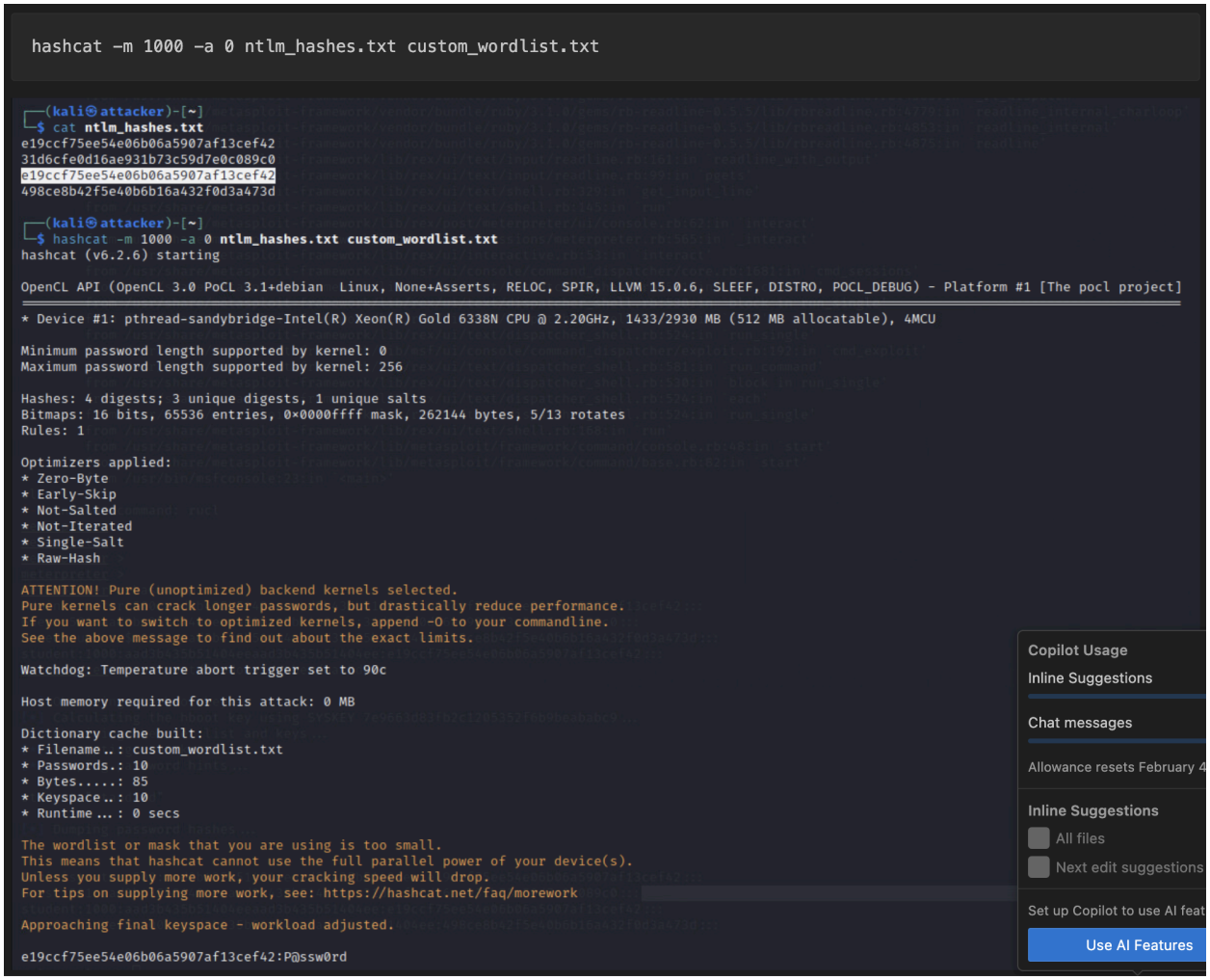


***Figure 2.3.1**: refer to 03-password-security-evidence.pdf p.6*

Impact

- Credential theft:
  Obtained password hashes for all accounts.
- Pass-the-hash attacks:
  Hashes can be used directly for authentication.
- Lateral movement:
  Compromised credentials enable network-wide access.
- Persistence:
  Hash access provides long-term persistence.
- Privilege escalation:
  Administrative hash access == full system control.
- Offline cracking:
  Hashes can be cracked without alerting the security system.

Recommendations

1. Reset all compromised hashes:
   - Force password changes for all extracted accounts.
   - Implement immediate credential rotation.
   - Monitor for hash reuse attempts.
2. Enable LSA protection:
   - Prevents unauthorized SAM database access.
   - Requires reboot to take effect.
3. Implement credential guard (Windows 10/11):
   - Virtualization-based security for credentials.
   - Isolates secrets from OS.

Short-term improvements

1. Restrict SAM Access:
   - Configure SAM access permissions *via* Group Policy.
   - Limit local account usage where possible.
   - Implement Protected Users group.
2. Monitor hash extraction attempts:

- Enable Audit Process creation.
- Monitor for hash dumping tools.
- Implement real-time alerts for SAM access.

3. Implement Microsoft LAPS:
    - Local Administrator Password solution.
    - Randomizes local admin passwords.
    - Centralized management and retrieval.

Long-term strategy

1. Privileged Access Management:
    - Implement just-in-time administrative access.
    - Use privileged identity management solutions.
    - Eliminate shared/local admin accounts
2. Advanced Credential protection:
    - Deploy Windows Defender Credential Guard.
    - Implement Device Guard and AppLocker.
    - Use Windows Hello for Business.
3. Network Segmentation:
    - Isolate high-value systems.
    - Implement microsegmentation.
    - Restrict lateral movement pathways.

Verification Steps

After remediation, verify the fix by:

1. Test LSA Protection.
2. Verify Credential Guard.
3. Test LAPS functionality.

Compliance Alignment:

- NIST 800-53: Authenticator Management (IA-5).
- CIS Controls: Account management, Access Control
- PCI DSS: Identify and authenticate access.

10- Unauthenticated SMB service exposure

HIGH RISK (7/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Moderate**
Remediation Difficulty	**Easy**

Synopsis

The Server Message Block (SMB) service on Windows systems was found to be exposed and accessible without authentication requirements, enabling unauthorized information disclosure, user enumeration, and serving as an initial attack vector for brute-force and pass-the-hash attacks.

Analysis

1. Service Detection: SMB service (TCP/445) identified as open and accessible.

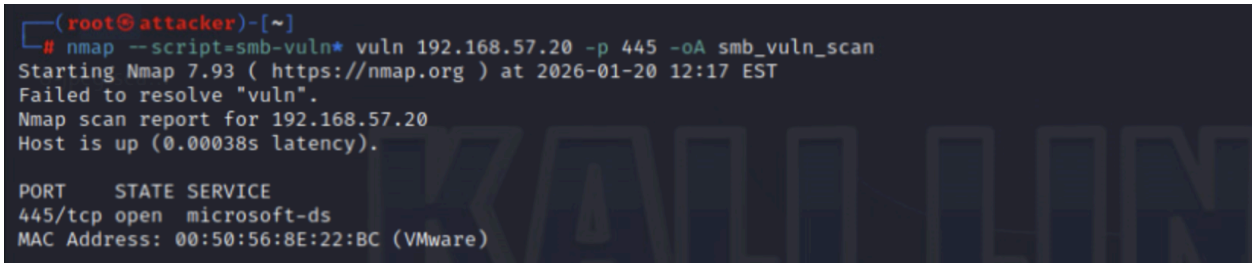


***Figure 2.3.1**: refer to 01-nmap-discovery-pdf p.7*

2. Unauthenticated access: Null session connection allowed:
   - Successfully listed shares without credentials.
   - Enumerated system information.
   - Identified available resources.
3. Risk combination:

   This exposure combined with weak password **#7** and hash extraction **#9** created a critical chain attack

Impact

- Information disclosure:
  Unauthorized share and user enumeration.
- Initial Attack Vector:
  Gateway for credential attacks.
- Lateral Movement entry point:
  Access to network file systems.
- Compliance violation:
  Violates least privilege principle.
- Amplification risk:
  Enumerated data used for targeted attacks.

Recommendations

1. Disable SMBv1.
2. Restrict NULL sessions.
3. Network segmentation:
    - Firewall rules to restrict SMB access.
    - Allow only from authorized management networks.
    - Implement VLAN segmentation.

Short-term improvements

1. Enable SMB security features.
2. Implement Access controls:
    - Share-level permissions review.
    - NTFS permissions hardening.
    - Least privilege principle application.
3. Monitoring and alerting:
    - Monitor for SMB enumeration attempts.
    - Alert on failed authentication attempts.
    - Log all SMB access attempts.

Long-term strategy

1. SMB replacement considerations:
    - Evaluate alternative file sharing protocols.
    - Consider cloud-based file sharing solutions.
    - Implement secure FTP or HTTPS alternatives.
2. Zero Trust implementation:
    - Microsegmentation for file servers.
    - Identify-based access controls.
    - Continuous authentication verification.

Verification Steps

After remediation, verify the fix by:

1. Test SMBv1 status >> Expected: false.
2. Test null sessions restriction >> Expected: Connection failure or empty share list.
3. Verify SMB signing >> Expected: True for SMB Server and SMB Client configuration.

Compliance Alignment

- CIS Benchmark: SMB security settings.
- NIST 800-53: Access enforcement, boundary protection, transmission confidentiality.
- PCI DSS: Network security controls.

11- Remote code execution *via* Meterpreter (Post-explotation)

HIGH RISK (8/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Moderate**
Remediation Difficulty	**Moderate**

Synopsis

Successful establishment of a Meterpreter reverse shell session following web application exploitation, demonstrating persistent remote code execution capabilities that enable complete system control, data exfiltration, and lateral movement from compromised web servers.

Analysis

Attack chain:

1. Initial compromise:
   Web application vulnerability exploitation (File Upload / SQL injection)
2. Payload delivery:
   Malicious PHP webshell upload to DVWA server.
3. Meterpreter session:
   Reverse TCP shell established to attacker-controlled system.
4. Post-Exploitation:
   System reconnaissance and control achieved.

- Session establishment:

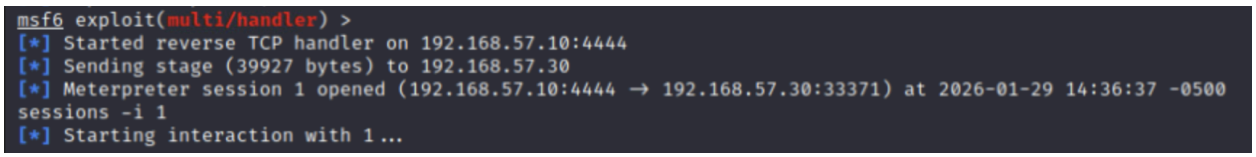


***Figure 2.3.1**: refer to 02-DVWA-evidence.pdf p.28*

5. Post-Exploitation capabilities demonstrated:
   - System information enumeration:

***Figure 2.3.1**: refer to 02-DVWA-evidence.pdf p.28*

- File system navigation and exfiltration.
- Process listing and manipulation.
- Privilege escalation attempts.
- Network reconnaissance.
- Persistence mechanisms establishment.

Impact

- Complete system control:
  Full command execution on compromised server
- Data breach:
  Access to web application data, databases, and configuration files.
- Lateral movement:
  Pivot point to internal network systems.
- Persistance:
  Backdoor installation for long-term access.
- Credential Theft:
  Harvest credentials from server memory and files.
- Service disruption:
  Ability to modify or stop critical services.
- Compliance violation:
  Breach of data protection and privacy regulations.

Recommendations

1. Web Server isolation:
   - Immediately isolate compromised webserver (192.168.57.30)
   - Block outbound connections to unauthorized IPs.
   - Implement network segmentation.
2. Malware detection and removal:

- Scan for webshells and backdoors.
- Check for unauthorized cron jobs and startup scripts.
- Review Web Server access and error logs.

3. Web application security hardening:
    - Implement Web Application Firewall (WAF).
    - Disable dangerous PHP functions.

```
disable_functions = exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source
```

    - Enable open_basedir restrictions.

Short-term improvements

1. Endpoint Detection and Response (EDR):
    - Deploy EDR on all Web Servers.
    - Configure alerting for reverse shell connections.
    - Implement behavioral detection for post-exploitation activities.
2. Network Security controls:
    - Implement egress filtering.
    - Monitor for unusual outbound connections.
    - Deploy Intrusion Prevention Systems (IPS).
3. Web Server hardening:
    - Run Web Servers with least privilege accounts.
    - Implement file integrity monitoring.
    - Regular security patch management.

Long-term strategy

1. Secure Development Lifecycle:
    - Implement DevSecOps practices.
    - Regular security code reviews.
    - Automated security testing in CI/CD.
2. Advanced Threat Protection:
    - Deploy Runtime Application Self-Protection (RASP).
    - Implement deception technologies (honeytokens).
    - Use threat intelligence feeds.

3. Incident Response Enhancement:
    - Develop web application incident response playbook.
    - Regular incident response drills.
    - Forensic investigation capability development.

Verification Steps

1. Test webshell detection.
2. Verify PHP hardening.
3. Test network Monitoring:
    - Simulate reverse shell connections.
    - Verify Suricata/IDS alerts triggers.
    - Check SIEM for correlation alerts.

Incident Response Checklist:

- ☐ Isolate the affected system.
- ☐ Preserve logs and evidence.
- ☐ Identify the initial compromise vector.
- ☐ Remove malicious files.
- ☐ Reset credentials.
- ☐ Patch vulnerabilities.
- ☐ Restore from clean backup if necessary.
- ☐ Implement additional monitoring.
- ☐ Document lessons learned.

12- File signature mismatch and extension obfuscation

MEDIUM RISK (6/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Moderate**
Remediation Difficulty	**Easy**

Synopsis

Detection of files with intentionally mismatched extensions used to hide JPG content (s.t. .dat, .hmm, .boo files containing valid JPG headers), demonstrating evasion techniques that bypass basic file type filtering while enabling data concealment or malicious payload delivery.
This finding represents a specific technical evasion method within the broader data concealment strategy documented in findings **#4** and **#8**.

Analysis

- Mismatched files:
  - file2.dat contained a valid JPG signature, but non-standard extension.
- Obfuscation purpose:
  - Avoid detection by extension-based security controls.
- Part of systematic data concealment strategy in #4, and #8.

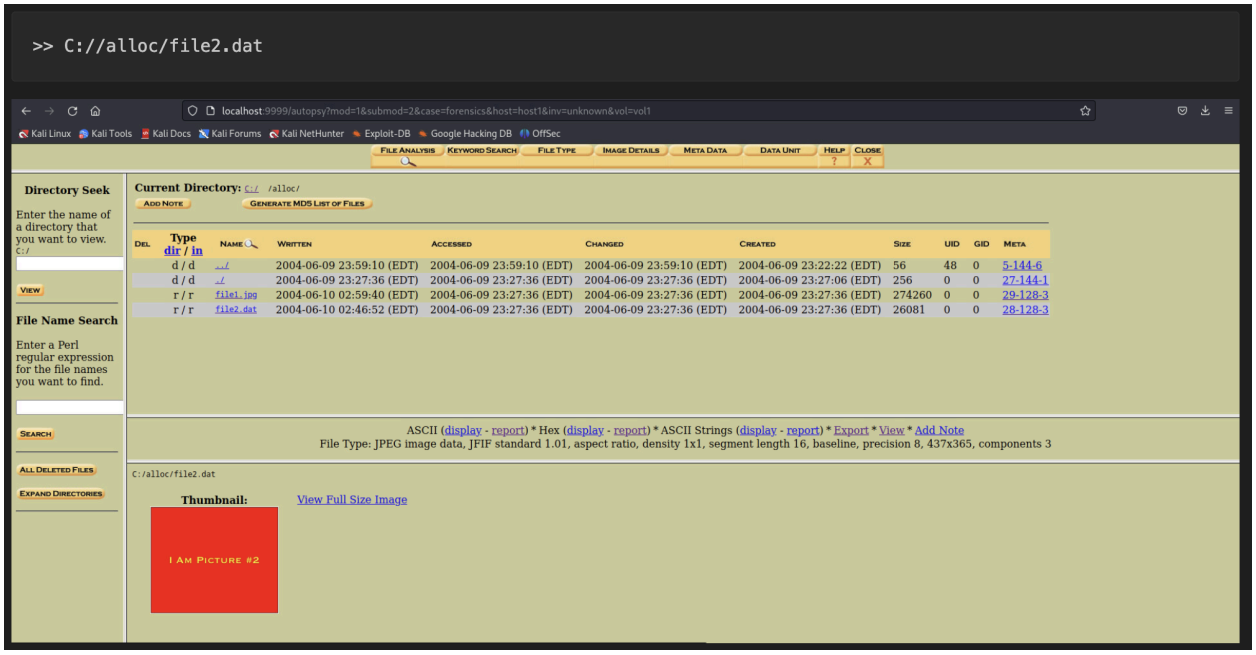


***Figure 2.3.1**: refer to 04-forensics-reports-evidence.pdf in p.19*

Impact

- Security control evasion:

  Bypasses extension-based filtering and AV scanning.
- Data concealment:

  Enables hiding of sensitive files in plain sight.
- Malware delivery:

  Potential vector for disguised malicious executables.
- Forensic complexity:

  Increases investigation time and effort.

Recommendations

Many recommendations were already written in Finding #4.

1. Implement file signature verification.
2. Update security controls:
    - Configure AV/EDR to inspect file signatures, not just extensions.
    - Implement DLP rules for content-type mismatches.

Short-term improvements

1. Automated scanning:
    - Schedule regular file signature audits.
    - Implement real-time file analysis.
2. User awareness:
    - Train staff on file type recognition.
    - Report suspicious file naming patterns.

Long-term strategy

1. Advanced threat protection:
    - Deploy file content analysis solutions.
    - Implement machine learning-based file classification.

Verification Steps

After remediation, verify the fix by:

- Run file signature verification scripts weekly.
- Test detection with known mismatched files.
- Monitor security logs for evasion attempts.

13- Microsoft RPC service exposure

MEDIUM RISK (6/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Minor**
Remediation Difficulty	**Easy**

Synopsis

Unnecessary exposure of Microsoft Remote Procedure Call (RPC) service on TCP port 135, providing potential attack surface for enumeration, information disclosure, and exploitation of associated services without proper access restrictions or network segmentation.
This finding represents a common network service exposure that, while not critical alone, contributes to overall attack surface and should be addressed as part of comprehensive network hardening.

Analysis

Detection with nmap scan identified open RPC service on port TCP/135 with Microsoft RPC Endpoint Mapper service.

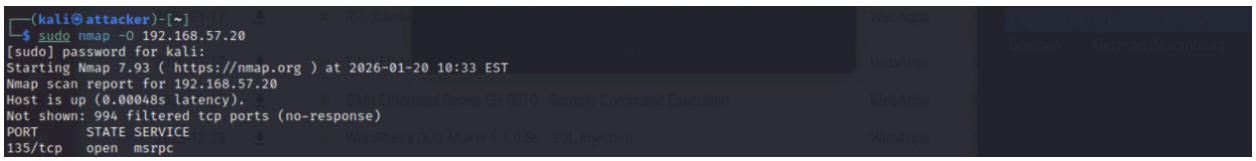


***Figure 2.3.1**: refer to 01-nmap-discovery.pdf p.2*

Associated risks:

- Service enumeration *via* RPC endpoints.
- Potential information disclosure about system services.
- Attack vector for DCOM and other RPC-based exploits.
- Often used in conjunction with other vulnerabilities for lateral movement.

Impact

- Information disclosure:
  Enumeration of RPC endpoints and services.
- Attack surface expansion:
  Additional entry point for attackers.
- Lateral movement:
  Potential pivot point in network attacks.
- Service exploitation:
  Vulnerabilities in RPC-implemented services.

Recommendations

1. Network access restrictions thought NetFirewall Rules:
    - Restrict remote access to management subnet on protocol TCP/135.
    - Block RPC from untrusted inbound on protocol TCP/135.
2. Service hardening:
    - Disable unnecessary RPC services.
    - Configure RPC filtering whenever it is required.
    - Implement RPC over HTTPS where it is possible.

Short-term improvements

1. Network segmentation:
    - Place systems with RPC requirement in restricted VLANs.
    - Implement microsegmentation.
    - Use jump hosts for RPC management access.
2. Monitoring and alerting

Long-term strategy

1. Architecture Review:
    - Evaluate RPC dependency for applications.
    - Migrate to modern, secure alternatives.
    - Implement Zero Trust network architecture.

Verification Steps

1. Test RPC access restriction >> Expected: Connection blocked or limited information.
2. Verify firewall rules.
3. Monitor RPC traffic:
    - Review security logs for RPC events.
    - Verify alerts for unauthorized access attempts.

14- Legacy RTSP service exposure

MEDIUM RISK (5/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Moderate**
Remediation Difficulty	**Easy**

Synopsis

Unnecessary exposure of legacy Real Time Streaming Protocol (RTSP) service on TCP/554, representing outdated network service with potential vulnerabilities that increases attack surface without clear business justification in corporate environment.

This finding represents unnecessary legacy service exposure contributing to overall attack surface. Similar to finding **#13**, it should be addressed as part of comprehensive service hardening and network surface reduction strategy.

Analysis

Detection with nmap over port TCP/554 which identified open RTSP port.

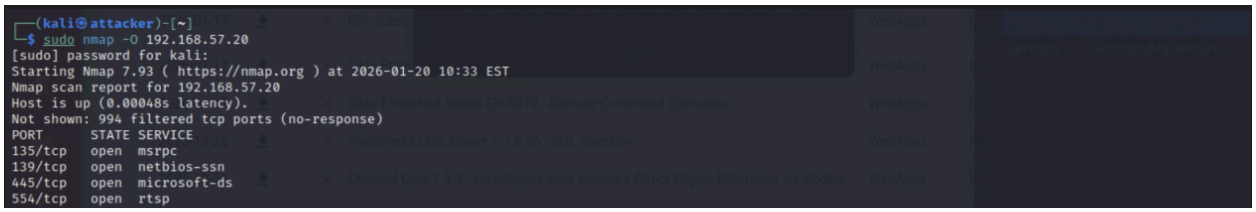


***Figure 2.3.1**: refer to 01-nmap-discovery.pdf p.2*

Associated risks:

- Legacy service with known historical vulnerabilities.
- Potential buffer overflow and code execution vulnerabilities.
- Information disclosure through service enumeration.
- Unnecessary network service increasing attack surface.
- Often unpatched and unmonitored.

Impact

- Attack surface expansion:
  Additional entry point for network attacks.
- Potential exploitation:
  Historical vulnerabilities: CVE-2009-5033, CVE-2014-0568, etc.
- Information disclosure:
  Service fingerprinting and enumeration.
- Resource consumption:
  Potential DoS attacks against streaming services.

Recommendations

1. Service disablement (if unused).
2. Network access restrictions
   Block RTSP over protocol TCP/554

Short-term improvements

1. Service inventory and justifications:
   - Document business need for RTSP service.
   - Identify responsible application/owner.
   - Evaluate alternative modern solutions.
2. Monitoring and alerting.

Long-term strategy

1. Modern solution migration:
   - Evaluate WebRTC or modern streaming alternatives.
   - Implement secure streaming solutions if required.
   - Deployment in DMZ with proper security controls.

Verification Steps

1. Test RTSP access >> Expected: Port filtered or no-response.
2. Verify service Status.
3. Check firewall configuration.

15- HTTPAPI services with default configurations

MEDIUM RISK (4/10)	
Exploitation Likelihood	**Possible**
Business Impact	**Minor**
Remediation Difficulty	**Easy**

Synopsis

Exposure of HTTPAPI services (HTTP Service API) with default configurations on Windows systems potentially revealing system information and providing unnecessary attack surface without proper security hardening or access restrictions.

Analysis

Nmap scan identified open HTTPAPI service over Web Services on Devices API on TCP/10243

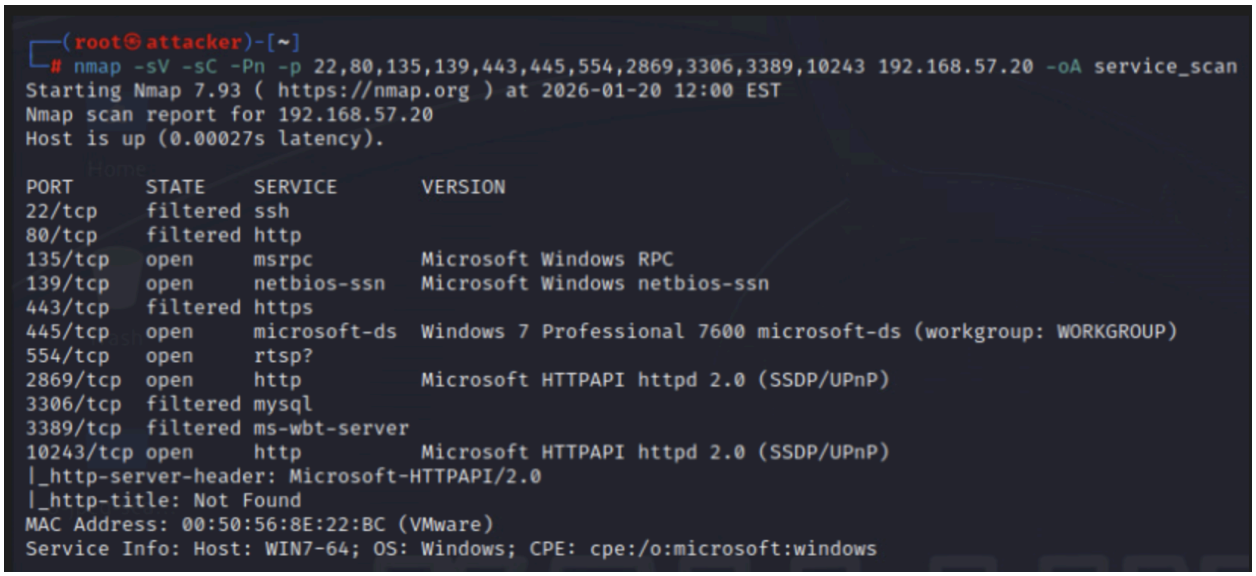


***Figure 2.3.1**: A php webshell uploaded to XYZ Application*

- Server header disclosure: Microsoft-HTTPAPI/2.0
- Default page response.
- No authentication or access controls observed.

Associated risks:

- Server information disclosure *via* headers.
- Potential enumeration of HTTPAPI features.
- Historical vulnerabilities in HTTPAPI.
- Unnecessary service exposure in a corporate environment.
- Default configuration may lack security hardening.

Impact

- Information disclosure:
  Server version and technology stack exposure.
- Attack surface expansion:
  Additional HTTP endpoint for attackers.
- Service fingerprinting:
  Easy identification of Microsoft HTTP stack.
- Potential exploitation:
  Vulnerabilities HTTPAPI implementation.

Recommendations

1. Server header obfuscation:
   Configure HTTPAPI to hide server version.
2. Network access restriction:
   - Restrict port 10243 access for specific network only.
   - Block HTTPAPI on port TCP/10243

Short-term improvements

1. Service assessment:
   - Determine if port 10243 is required for business functions.
   - Identify applications using this service.
   - Document service dependencies.
2. Configuring hardening:
   - Enabling Error Loggings.
   - Url segment maximum length.

Long-term strategy

1. Service minimization:
    - Disable unnecessary HTTPAPI features.
    - Consider migrating to properly configured IIS if web services required.
    - Regular review of exposed HTTP service.

Verification Steps

1. Test Server header disclosure >> Expected: no “Server” header or generic header.
2. Verify firewall configuration.
3. Check registry settings.

16- Insecure default and common passwords

MEDIUM RISK (6/10)	
Exploitation Likelihood	**Likely**
Business Impact	**Moderate**
Remediation Difficulty	**Easy**

Synopsis

Use of insecure default and common passwords across multiple systems and services, enabling easy credential compromise through dictionary attacks and credential stuffing. This finding directly relates to and amplifies risks from finding **#7** (Weak passwords policy) and **#9** (SAM hash extraction).

This finding addresses the fundamental authentication weakness that enabled multiple attack vectors documented throughout the assessment.

Analysis

Compromised credentials:

- Administrator
- Student

Related evidence from previous findings:

- **#7**: successful SMB brute-force with simple wordlists.
- **#9**: extracted NTLM hashes cracked to reveal weak passwords.
- Custom wordlist content:
  Includes predictable passwords.

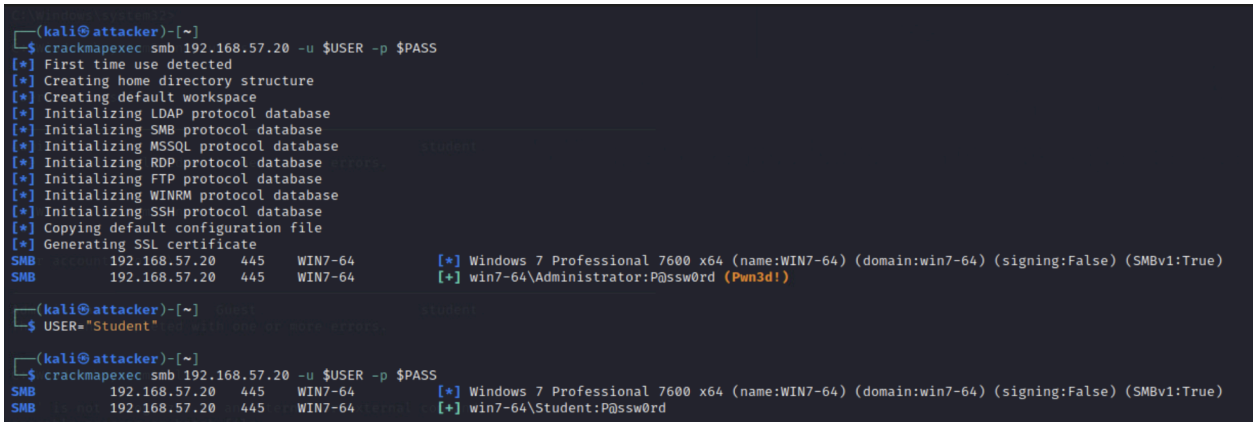


***Figure 2.3.1**: refer to 03-password-security-evidence.pdf on p.4-5*

Attacked success rate:

- 2/2 tested accounts compromised.
- Rapid cracking.
- No account lockout triggered.

Impact

- Easy credential compromise:
  Low-skill attackers can gain access.
- Lateral movement:
  Compromised credentials usable across systems.
- Privilege escalation:
  Administrative account compromise.
- Persistance:
  Long-term access through weak credentials.
- Compliance violation:
  Violates multiple security frameworks.

Recommendations

1. Password reset enforcement:
   - Force password change for all users.
   - Check password last set date.
2. Command password blocking:

- Implement custom password filter DLL, or
- Evaluate to use third-party password protection solutions

Short-term improvements

1. Password policy enhancement:

   Enforce complexity
2. Password auditing:
   - Regular password strength audits.
   - Integrate with HaveIBeenPwned API.
   - Implement breach password detection.

Long-term strategy

1. Multi-Factor authentication:
   - Implement MFA for all administrative accounts.
   - Consider MFA for all user accounts.
   - Use Windows Hello for Business.
2. Passwordless authentication:
   - Implement FIDO2 security keys.
   - Deploy certificate-based authentication.
   - Consider biometric authentication.

Verification Steps

1. Test password policy >> Expected: result over password complexity.
2. Attempt weak password >> Expected: Access denied - password changed.
3. Audit password strength:
   - Use password auditing tools.
   - Check for password reuse.
   - Verify MFA implementation.

17- Image integrity verification

INFORMATIONAL (0/10)	
Exploitation Likelihood	**N/A**
Business Impact	**N/A**
Remediation Difficulty	**Easy**

Synopsis

Successful verification of forensic disk image integrity through MD5 hash matching, establishing proper chain of custody and evidence preservation procedures. This procedural finding supports the validity of forensic evidence used in findings **#4**, **#8**, and **#12**.
This informational finding demonstrated proper forensic procedures that support the technical

Analysis

1. Verification process:
    - Original image hash.
    - Forensic copy creation.
    - Hash comparison:
      Both matches exactly.

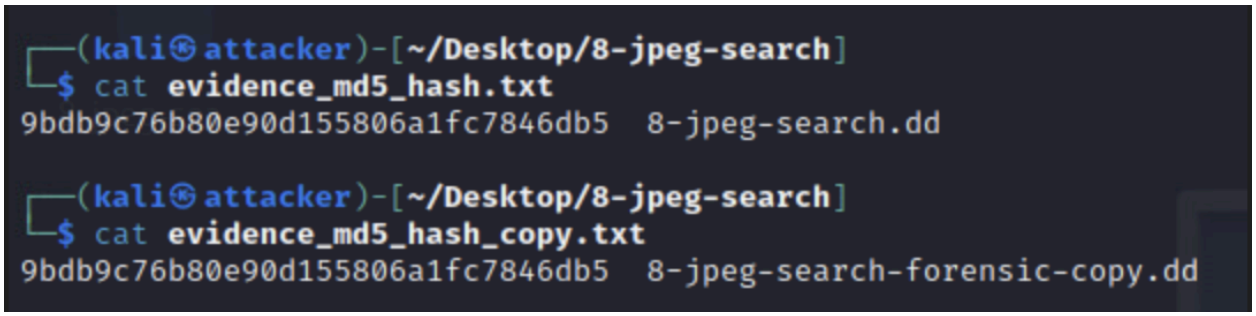


***Figure 2.3.1**: refer to 04-forensics-report-verification p.3-4*

- Autopsy verification:
  Hash confirmed within forensic toolkit.

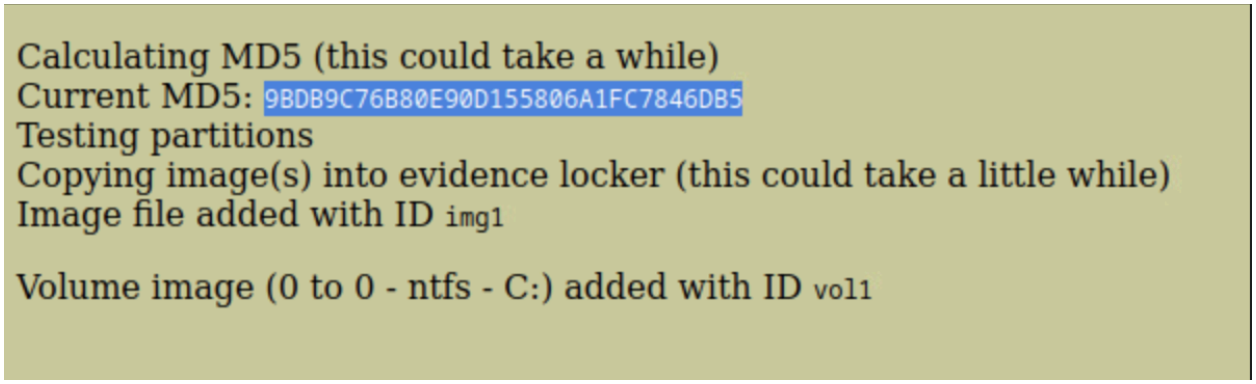


***Figure 2.3.1**: refer to 04-forensics-report-verification p.3-4*

- Evidence preservation:

  Hashes documented in evidence_md5_hash.txt

2. Forensic significance:
   - Ensures evidence has not been altered or tampered with.
   - Establishes legal admissibility foundation.
   - Demonstrates proper forensic procedures.
   - Supports findings from subsequent analysis phases.

Impact

- Legal compliance:

  Meets evidence handling standards.
- Investigation validity:

  Ensures forensic findings are based on unaltered evidence.
- Procedural demonstration:

  Shows adherence to forensic best practices.
- Chain of Custody:

  Proper documentation maintained.

Recommendations

1. Procedural improvement rather than security fixes.

Standard operating procedures:

1. Documented hash procedures.
2. Chain of custody forms:
   Implement digital chain of custody tracking.
3. Automated verification:
   Script-based hash verification at each handling stage.

Tool standardization:

1. Approve tool list:
   Standardize on specific forensic tools.
2. Version control:
   Maintain tool version documentation.
3. Validation testing:
   Regular tool functionality verification.

Training and documentation:

1. Forensic procedure training:
   Regular staff training.
2. Checklist:
   Step-by-step evidence handling checklists.
3. Audit trails:
   Comprehensive logging of all evidence handling.

Verification Steps

1. Regular procedure audits:
   - Quarterly review of evidence handling procedures.
   - Random hash verification tests.
   - Chain of custody documentation checks.
2. Tool validation.

Integration with other findings:

- Supports findings **#4**, **#8**, **#12**:
  Validates forensic evidence used for data concealment analysis.
- Complements Chain of Custody:
  Part of comprehensive evidence handling.
- Foundation for legal admissibility:
  Essential for investigations with potential legal proceedings.

18- Proper Chain of Custody established

MEDIUM RISK (6/10)	
Exploitation Likelihood	**N/A**
Business Impact	**N/A**
Remediation Difficulty	**Moderate**

Synopsis

Establishment and maintenance of proper chain of custody throughout the forensic investigation, including comprehensive case documentation, metadata recording, and evidence handling procedures that support the validity of all technical findings in this assessment.
In fact, this informational finding represents a security strength and procedural competency that underpins the entire forensic investigation process documented in this assessment.

Analysis

1. Chain of custody components documented.
    a. Case documentation: name, investigator, description, etc.
    b. Evidence documentation: host, image details and storage location.
    c. Procedural compliance:
        - Forensic copy creation before analysis.
        - Hash verification at multiple stages.
        - Screenshot documentation of critical steps.
        - Systematic file recovery and logging.
2. Integration with other findings:
    a. Support finding **#17**: Hash verification procedures.
    b. Validated findings **#4**, **#8**, **#12**: Forensic evidence handling for data concealment analysis.
    c. Strengthens all technical findings: Legal and procedural foundation.

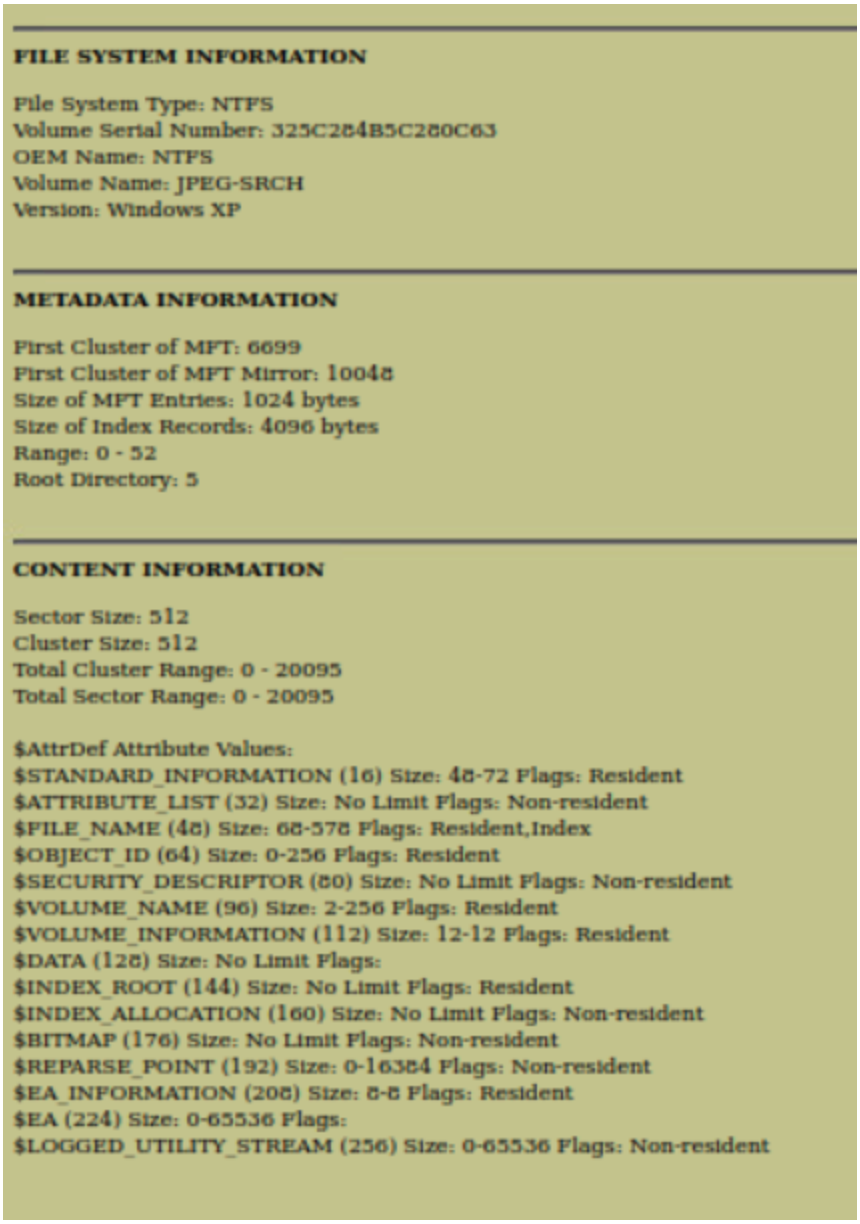


***Figure 2.3.1**: General file system details*

Impact

- Legal admissibility:

  Meets standards for evidence in legal proceedings.
- Investigation integrity:

  Ensures findings are based on properly handled evidence.
- Reproducibility:

  Allows independent verification of results.

- Professional standards:
  Demonstrates compliance with forensic best practices.
- Audits trail:
  Comprehensive documentation for review and validations.

Recommendations

Procedural enhancements for ongoing forensics readiness.

1. Use of standardized templates.
2. Digital chain of custody system:
    - Implement blockchain or cryptographic ledger for evidence tracking.
    - Automated timestamping and signing.
    - Access control and audit logging.

Process enhancements:

1. Evidence handling procedures:
    - Two-person verification for critical evidence.
    - Secure evidence storage with access logging.
    - Regular integrity re-verification.
2. Training program:
    - Regular forensic procedure training.
    - Legal requirements education.
    - Tool-specific certification.

Technology integration

1. Forensic platform standardization:
    - Centralized evidence management system.
    - Automated metadata extraction.
    - Integration with case management tools.
2. Monitoring and auditing:
    - Regular chain of custody audits.
    - Automated alerting for evidence access.
    - Compliance reporting.

Integration with assessments findings

This procedural finding provides the foundation that validates all technical findings in this report.

Proper chain of custody:

1. Ensures evidence integrity:
   For findings involving data recovery **#4**, **#8**, **#12**.
2. Supports legal validity:
   For potential legal proceedings.
3. Enables reproducibility:
   For verification of all technical results.
4. Demonstrates professionalism:
   In forensic investigation standards.

APPENDIX A - TOOLS USED

TOOL	Version	DESCRIPTION	USED IN PHASE
nmap	7.93	- Network reconnaissance. - Port scanning. - Service enumeration. - Vulnerability detection.	1, 2, 3
Metasploit	6.3	- Vulnerability exploitation. - Payload generation. - Post-exploitation. - Hash extraction	1, 2, 3
Autopsy digital forensics platform	2.24	- Forensic image analysis. - File recovery. - Metadata examination. - Evidence management	4
Hydra	9.4	- SMB brute-force attacks. - Credential cracking. - Service authentication testing	3
md5sum	Linux built-in	- File integrity verification. - Hash generation. - Evidence integrity checking.	4
Hashcat	6.2.6	- Password hash cracking. - Credential recovery demonstration	3
smbclient	4.15.12	- SMB protocol interaction. - Share enumeration. - Credential verification.	3
SQLMap	1.7.2	- SQL injection detection and exploitation. - Database enumeration.	2
OWASP ZAP	2.14.0	- Web application security testing. - Automated vulnerability scanning	1
TCPDump	4.99.1	- Command-line network packet analyzer. - Traffic recording	1
John-the-Ripper	1.9.0	- Password hash cracking. - Wordlist attacks	3
7-Zip	23.01	- Archive extraction.	4

		- Compressed file analysis.	
Hex Editors (xxd, HxD)	-	- Binary file analysis. - File signature verification.	4
Kali Linux	2023.4	- Penetration testing distribution containing all tools.	1, 2, 3, 4
VirtualBox	7.0.12	- Virtualization platform for isolated testing environment.	1, 3
Windows 7 Professional	SP1	- Target operating system for vulnerability assessment.	1, 3
Windows Server 2008 R2	-	- Additional target system for network assessment.	1
DVWA (Damn Vulnerable Web App)	1.10	- Web application target for security testing.	2
Custom scripts (python/PHP/Java Script/bash)	-	- Automation of repetitive tasks. - Custom exploitation	1, 2, 3, 4

Table A.1: *Tools used during assessment*

APPENDIX B - ENGAGEMENT INFORMATION

Client Information

Client	Neuefische
Primary Contact	Esperanza Buitrago Díaz
Approvers	The following people are authorized to change the scope of engagement and modify the terms of the engagement ● Puthrivi SriRaj Kuppagiri

Version Information

Version	Date	Description
1.0	February 6th 2026	Initial report to client

Contact Information

Name	Esperanza Buitrago Díaz
Github	www.github.com/ebuitragod
Email	ebuitragod@gmail.com