Penetration Testing Report for NewBizz Ltd

Author: Leen El Mir

Date: 20-05-25

Contents

I. Executive Summary 4

Assessment Overview 4

Key Findings 4

Remediation 4

II. Introduction 5

III. Scope & Methodology 5

Test Environment 5

Test Scope 5

Frameworks and Standards Followed 6

A. PTES (Penetration Testing Execution Standard) 6

B. OWASP Testing Guide v4 6

C. MITRE ATT&CK Framework 6

Tools Used 7

Testing Phases 7

Enumeration 7

Scanning 7

Vulnerability Analysis 8

Exploitation & Post-Exploitation 8

IV. Risk Assessment 8

V. Findings and Evidence 11

MyHobbyServer 11

Finding #1: OpenSSH 6.7 - User Enumeration 11

Finding #2: WordPress 3.8.1 – Weak Admin Credentials 12

Finding #3: WordPress 3.8.1 – Stored XSS & Reverse Shell 13

Finding #4: Privilege Escalation – Dirty COW Exploit (Linux 3.16.0) 15

Finding #5: WordPress 3.8.1 – Insecure Storage of Database Credentials 16

Finding #6: WordPress 3.8.1 – Unencrypted Transmission of Login Credentials 16

Finding #7: WordPress 3.8.1 – Lack of Dual Control on Admin Actions 16

DISGUISE Server 17

Finding #1: Hidden Subdomain – dark.disguise.hmv 17

Finding #2: Insecure Session Management 18

Finding #3: Remote Code execution via Reverse Shell Upload 19
Finding #4: Reverse Shell Upload -Privilege Escalation 22
Finding #5: Root Privilege Escalation 23
MSEdge – Win10 24
Finding #1: SMB Signing Not Required 24
Finding #2: RDP Exposed (BlueKeep Risk) 25
Finding #3: HTTPAPI Exposed Over HTTP (WinRM) 25
Finding #4: Outdated OS – Windows 10 Build 17763 25
Win Server 2019 26
Finding #1: IIS 10.0 – Trace Method enabled 26
Finding #2: Unencrypted WinRM Over HTTP 27
Finding #3: Unsecured Active Directory Ports 27
Finding #4: Information Leakage via IIS Misconfigurations 27
DevServer 28
Finding #1: Insecure FileZilla FTP Server Configuration 28
Finding #2: Apache Web Server Information Disclosure 29
Finding #3: Unsecured Mercury/32 Mail Server Exposure 29
Risk Level: Medium 29
Finding #4: Vulnerable MariaDB Database Server Version 29
Risk Level: Low 29
Finding #5: Expired SSL/TLS Certificate on Port 443 30
Risk Level: Medium 30
VI. Recommendations 30
VII. Conclusion 33
VIII. Appendices 34
Appendix A: Detailed Evidence Log & Methodology 34
IX. References 41

I. Executive Summary

Table 1.Summary of risk assessment.

SEVERITY	NUMBER OF VULNERABILITIES	URGENCY OF REMEDIATION
CRITICAL/HIGH	10	Immediate
MEDIUM	11	Short-term
LOW	4	Long-term

Assessment Overview

To strengthen the cybersecurity posture of NewBizz Ltd, our team conducted an in-depth pentest on five critical virtual machines within the company's network, covering Linux and Windows-based servers. Testing was conducted out of office hours, with full authorization to perform real-world exploitation scenarios to demonstrate the impact of a breach. This assessment aimed at uncovering exploitable vulnerabilities in the company's infrastructure and web applications.

Key Findings

Through this assessment, we identified **25 security vulnerabilities**, categorized as follows: **10 high-risk, 11 medium-risk, and 4 low-risk** (Table 1). Exploitation of high-risk issues led to unauthorized administrative access, remote code execution, and session hijacking. Particularly concerning were the use of weak credentials, outdated WordPress instances, and misconfigured server scripts that allowed privilege escalation to root-level access. Medium-risk vulnerabilities, such as poor password storage, present opportunities for lateral movement and internal compromise. Low risk issues, while less urgent, could be exploited during a chained attack and should not be overlooked.

Remediation

All vulnerabilities are provided with detailed recommendations in the 'Recommendations' section, prioritizing the critical issues. Addressing these issues allows NewBizz Ltd to significantly reduce cyber threats exposure.

II. Introduction

This report presents the findings of a simulated pentest conducted for NewBizz Ltd, a company undergoing its first formal security assessment. The organization has limited cybersecurity experience, so the goal of this assessment is to provide clear insights into its current security posture.

The pentest was commissioned by the management team—who are the only ones aware of the exercise— and was designed to assess the security of five machines running within the company's network. Testing took place outside regular working hours, with no interaction with end users. The assessment focused exclusively on technical exploitation to reflect the potential impact of a real-world cyberattack.

The primary objectives of this pentest include conducting a full vulnerability assessment of internal and external assets. The assessment also classifies risks based on severity and likelihood. Several exploitation techniques are demonstrated, and the potential impact of such attacks is assessed. Finally, remediation guidance and security recommendations are provided to improve the company's security. All steps and evidence have been documented for replication and reference.

This report is intended for internal use by senior management, software developers, SOC analysts, and the IT manager. It outlines security gaps and recommendations for reducing organizational risk.

III. Scope & Methodology

Test Environment

The testing environment was set up within a controlled private network, using the IP range 172.16.1.0/24. The testing machine was assigned the IP address 172.16.1.4. Testing was performed out of office hours to avoid disrupting activities.

Test Scope

The assessment scope included infrastructure-level and web application penetration testing over five servers. These systems represent critical assets in the company NewBizz's network architecture. The test was performed using a black-box method. Full system exploitation was permitted, exploiting identified vulnerabilities, gaining shell/system access, and/or exfiltrating sensitive files.

In-Scope Targets

The in-scope targets are the five servers – MyHobbyServer (172.16.1.5), Disguise (172.16.1.6), MSEdge – Windows 10 (172.16.1.7), DevServer (172.16.1.7), and Windows Server 2019 (172.16.1.8).

Excluded Assets

Since the testing was conducted out of office hours, no user interaction was permitted, eliminating social engineering. The scope excludes physical access to the servers or infrastructure.

Frameworks and Standards Followed

To ensure an ethical and structured assessment, several frameworks/standards were followed.

A. PTES (Penetration Testing Execution Standard)

PTES outlines a methodology for conducting a pentest; this was used to ensure a structured and thorough approach (PTES, 2014). The relevant PTES phases are - Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post-Exploitation, and Reporting. These phases were followed for all five virtual machines.

B. OWASP Testing Guide v4

This framework was used for evaluating web applications security, particularly the WordPress instances (OWASP, 2021). It guided the manual/automated testing of vulnerabilities such as authentication flaws. The guide's checklist helped assess file upload vulnerabilities, session hijacking, and more for the web servers.

C. MITRE ATT&CK Framework

This framework classified the vulnerabilities/exploits observed during exploitation (MITRE, 2023). Most attack paths found—such as session hijacking—were mapped to documented ATT&CK techniques, standardizing how attacks were classified.

Tools Used

TOOL NAME	PURPOSE	SCOPE OF USE	OUTCOME
NMAP	Network discovery and port scanning	Identifying open ports and running services	Discovered active hosts and services (e.g., HTTP, SSH, MySQL)
WPSCAN	WordPress vulnerability scanning	Enumerated plugins, themes, and user data	Detected outdated plugins, weak admin credentials
METASPLOIT	Exploitation framework	Exploited known vulnerabilities and gained shells	Achieved reverse shell and privilege escalation
FFUF	Fuzzing hidden directories and endpoints	Web enumeration of hidden subdomains and files	Discovered dark.disguise.hmv and upload endpoints
SEARCHSPLOIT	Local exploit database	Mapped discovered services to public exploits	Identified matching exploits for outdated software
BURP SUITE	Interception and web exploitation	Manual testing of input validation, cookies, sessions	Detected insecure session handling, cookie reuse, and SQL injection
NIKTO	Web server vulnerability scanning	Quick scan for known server misconfigurations	Revealed outdated Apache versions, misconfigured headers

Table 2. Summary of tools used for penentration test.

Testing Phases

Enumeration

During this phase, tools like Nmap were used to enumerate valid usernames, server names, or other targets. This used active connections to the system.

Scanning

During this phase, the network infrastructure was mapped. Passive recon was conducted, involving reviewing WHOIS records and subdomain data to uncover insecure entry points. Active recon was conducted using Nmap, identifying open ports,

OSs, running services and versions. The default Nmap scripts for each port were run to detect misconfigurations or potential vulnerabilities.

Vulnerability Analysis

Discovered services were analyzed using several tools. Nikto flagged misconfigured headers and insecure HTTP methods. WPScan identified vulnerabilities in WordPress plugins, outdated core files, etc... The tool SearchSploit mapped service versions to known exploits/CVEs.

Exploitation & Post-Exploitation

Several tools were used to test the real-world impact of identified exploits. Metasploit ran automated exploits, such as a reverse shell exploit using WordPress. BurpSuite was used to intercept requests (like logins) and modify/inject the payload, triggering errors. Malicious files were uploaded to vulnerable endpoints, triggering privilege escalations.

IV. Risk Assessment

Table 3. Risk Assessment of discovered vulnerabilities.

VULNERABILITY NAME	SERVERITY	AFFECTED ASSETS	POTENTIAL IMPACT
WEAK AUTHENTICATION AND CREDENTIAL EXPOSURE	Critical	MyHobbyServer, Disguise VM	Unauthorized admin access through default passwords and session hijacking; enables full system control and remote code execution
UNENCRYPTED CREDENTIAL TRANSMISSION (OVER HTTP)	Critical	MyHobbyServer, Disguise VM	Credentials and session tokens are exposed over network, allowing attackers to intercept logins and impersonate users or admins.
PRIVILEGE ESCALATION (LOCAL AND/OR ROOT)	Critical	MyHobbyServer, Disguise VM	Attackers escalate low-privilege shell to root, gaining full control over the system; They could install rootkits, backdoors, modify system files, etc...
REMOTE CODE ESCALATION	High	MyHobbyServer, Disguise VM	Attacker can gain shell access to enumerate

THROUGH FILE UPLOAD			users, read sensitive files, or pivot further.
INSECURE CREDENTIAL STORAGE	High	MyHobbyServer, Disguise VM	Plaintext credentials exposed to attackers with shell access, enabling backend database compromise.
INSECURE SESSION MANAGEMENT & COOKIE REUSE	High	Disguise VM	Reuse of stolen session tokens allows attacker to fully impersonate users, including admins, without needing credentials.
SMB SIGNING NOT REQUIRED	High	MSEDGE – WIN10	The host is susceptible to SMB relay attacks. Hackers can steal NTLM hashes and target other systems.
LACK OF DUAL CONTROL & ADMIN OVERSIGHT	Medium	MyHobbyServer – Disguise VM	A single admin can make unrestricted changes without review; this allows attackers to silently take over the site or upload backdoors.
UNSECURED SUBDOMAINS	Medium	Disguise VM	Hidden subdomain unintentionally exposes an unmonitored attack surface. This could lead to discovery of critical flaws and access to backend systems.
RDP EXPOSED WITHOUT NLA (BLUEKEEP RISK)	Medium	MSEDGE – WIN10	Vulnerability to CVE-2019-0708 (BlueKeep). Attackers can brute-force credentials or gain complete GUI access.
HTTPAPI EXPOSED OVER HTTP (WINRM)	Medium	MSEDGE – WIN10	Allows MITM and command injection attacks.
OUTDATED OS – WINDOWS 10 BUILD 17763	Medium	MSEDGE – WIN10	Host is missing patches for local privilege escalation, so attackers

			can gain SYSTEM-level access.
TRACE METHOD ENABLED	Medium	WIN SERVER 2019	RACE method allows Cross-Site Tracing, so session tokens can be exfiltrated.
UNENCRYPTED WINRM OVER HTTP	Medium	WIN SERVER 2019	Admin commands and credentials can be intercepted; full command injection and session hijacking could occur.
INSECURE FILEZILLA FTP SERVER CONFIGURATION	Medium	DEVSERVER	FTP traffic is unencrypted. Attackers can steal credentials or capture transferred data.
APACHE WEB SERVER INFORMATION DISCLOSURE	Medium	DEVSERVER	Apache headers disclose server version and PHP 8.0.30 (EOL); attackers can exploit known vulnerabilities in outdated modules.
UNSECURED MAIL SERVER EXPOSURE	Medium	DEVSERVER	Multiple ports leak administrative login info. CVE-2005-1523 may be used to abuse mail relay features.
EXPIRED SSL/TLS CERTIFICATE ON PORT 443	Medium	DEVSERVER	Expired, self-signed certificate weakens HTTPS security. Attackers can intercept or spoof traffic.
USER ENUMERATION VIA OPENSSH	Low	MyHobbyServer	Disclosure of valid usernames can support password guessing or brute-forcing, increasing likelihood of credential compromise.
INFORMATION LEAKAGE VIA IIS MISCONFIGURATIONS	Low	WIN SERVER 2019	HTTP reveals IIS version, paths, and debug info — aiding recon and targeted exploits.
UNSECURED ACTIVE DIRECTORY PORTS	Low	WIN SERVER 2019	Lack of LDAP signing or channel binding allows

		spoofing and AD enumeration.

V. Findings and Evidence

MyHobbyServer

This machine's specifications are summarized in Table 4. It was assigned the IP address 172.16.1.5, running on Linux. Figure A.1 displays the full Nmap scan.

MyHobbyServer.Machine.Specifications

IP Address	172.16.1.5
Hostname	blog.mycompany.ex
Operating System	Linux
SSH Service	OpenSSH 6.7p1 Debian 5 (Port 22)
Web Server	Apache 2.4.10 (Port 80)
CMS Detected	WordPress 3.8.1
Open Ports	22 (SSH), 80 (HTTP), 111 (RPCBind), 39205 (RPC status)
Services Detected	SSH, HTTP, RPCBind, RPC Status
SSH Host Keys	DSA (1024), RSA (2048), ECDSA (256), ED25519 (256)
Service Info	OS: Linux; CPE: cpe:/o:linux:linux_kernel
Kernel Version	Linux 3.16.0-4-amd64 (Debian 3.16.7-ckt9-2)

Table 4. MyHobbyServer: System Specifications and Active Services

Finding #1: OpenSSH 6.7 - User Enumeration

Risk Level: low

Evidence: The version of OpenSSH (v6.7p1) running on the server is vulnerable to user enumeration (CVE-2018-15473) (Figure 1), due to differences in server responses when valid/invalid usernames login. A Metasploit exploit identified valid system accounts (Figure 2).

```
┌──(kali㉿kali)-[~]
└─$ searchsploit SSH 6.7
 Exploit Title                                     |  Path
OpenSSH 2.3 < 7.7 - Username Enumeration           | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (         | linux/remote/45210.py
OpenSSH < 7.4 - UsePrivilegeSeparation Di          | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary L         | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)               | linux/remote/45939.py
```

Figure 1. List of Exploitable Vulnerabilities for Open SSH v6.7.

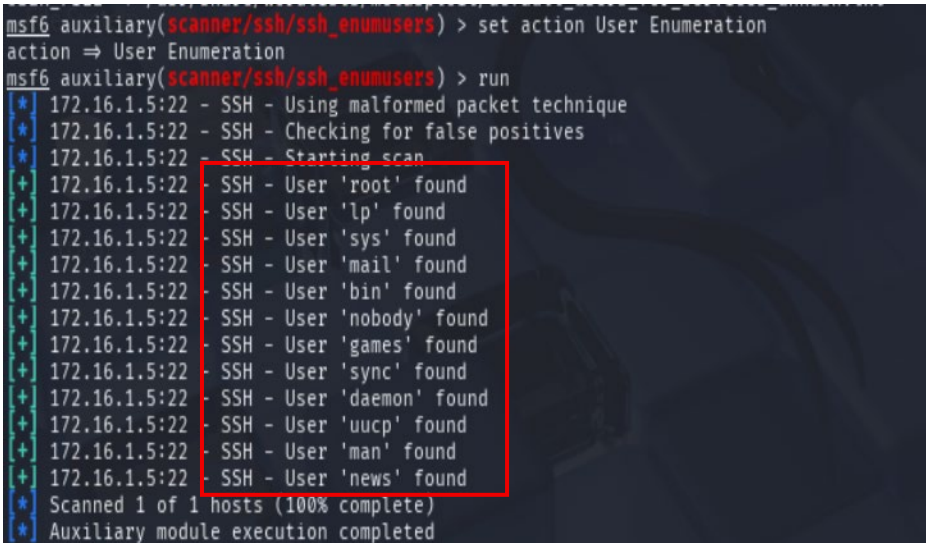


Figure 2. Successful user enumeration completed for Open SSH v6.7.

Impact:

Enumerating the usernames can cause a targeted brute-force attacks for user passwords. Attackers can customize password lists using usernames as key words, increasing the efficacy of password guesses.

Finding #2: WordPress 3.8.1 – Weak Admin Credentials

Risk Level: Critical

Evidence: The version of WordPress (v3.8.1) is extremely outdated version and highly exploitable. The credentials can be brute forced, especially default ones. Figure 3 portrays a successful login to the WordPress admin dashboard with default credentials **admin:12345678,** discovered using Nmap.

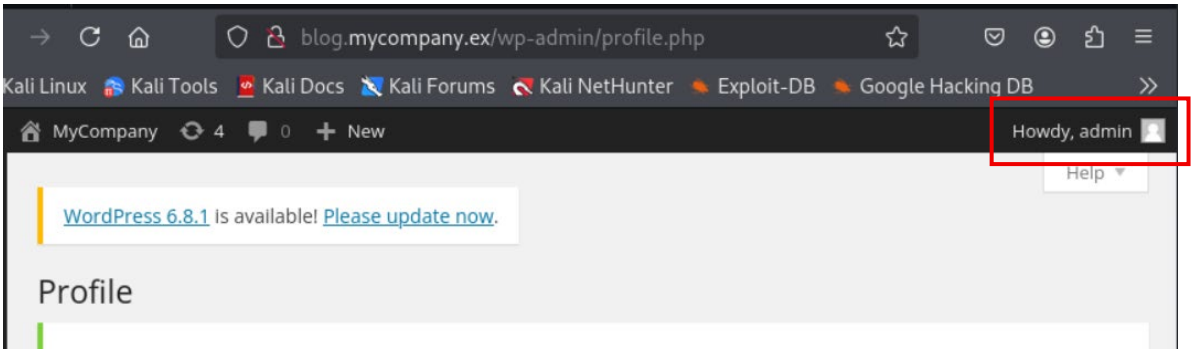


Figure 3. WordPress 3.8.1: Logged in admin user.

Impact:

Attackers impersonating admins could perform critical changes. They can upload/modify PHPs, posts, existing user accounts, etc... (Figure 4). Modifying the PHP

files leads to RCE, and defacing the website pages leads to malicious code injection or phishing attempts. Attackers can change the admin's current password, denying the company access to the admin. Figure 5 shows a sample exploit of an attacker modifying the blog's pages, leaving a 'hacked message'.

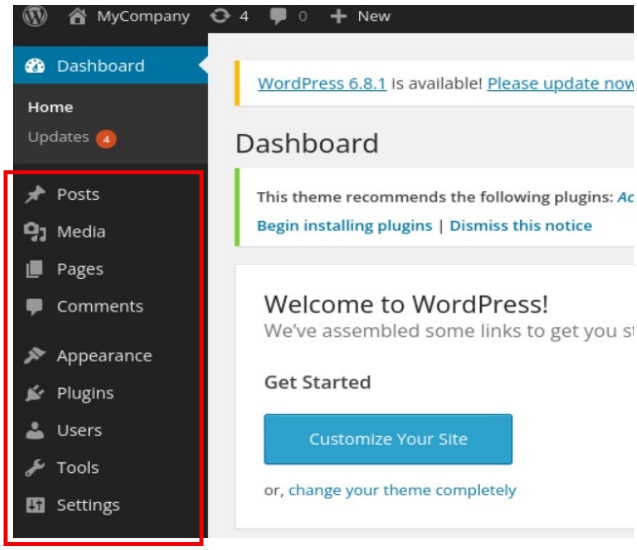


Figure 4. WordPress 3.8.1 - List of items that an admin user can modify.

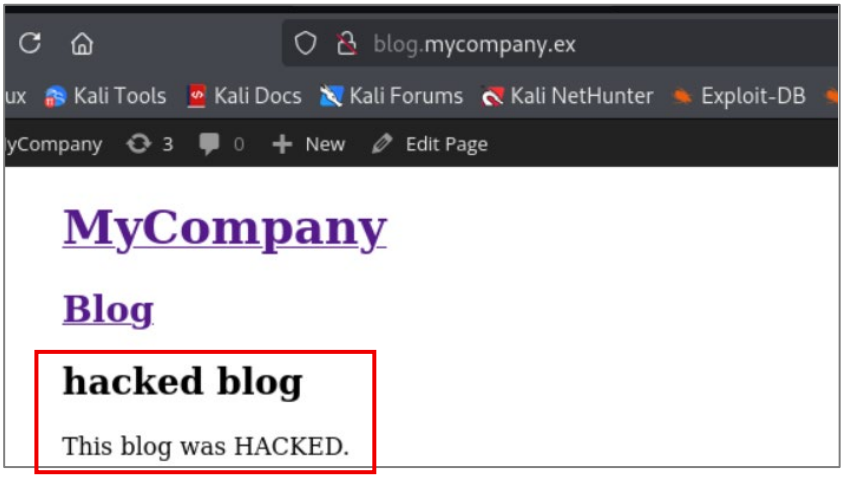


Figure 5. WordPress 3.8.1 - Admin user modifying the website's main page.

Finding #3: WordPress 3.8.1 – Stored XSS & Reverse Shell

Risk Level: Critical

Evidence:

Attackers impersonating admin could inject a reverse shell payload into the WordPress theme file 404.php. Users visiting a non-existent page are redirected to the /404.php page, triggering a reverse shell connection from the target back to the attacker's server (Figure 6). An automated reverse shell exploit, run by Metasploit, could also successfully trigger a reverse shell without requiring a user to access a non-existent page (Figure 7).

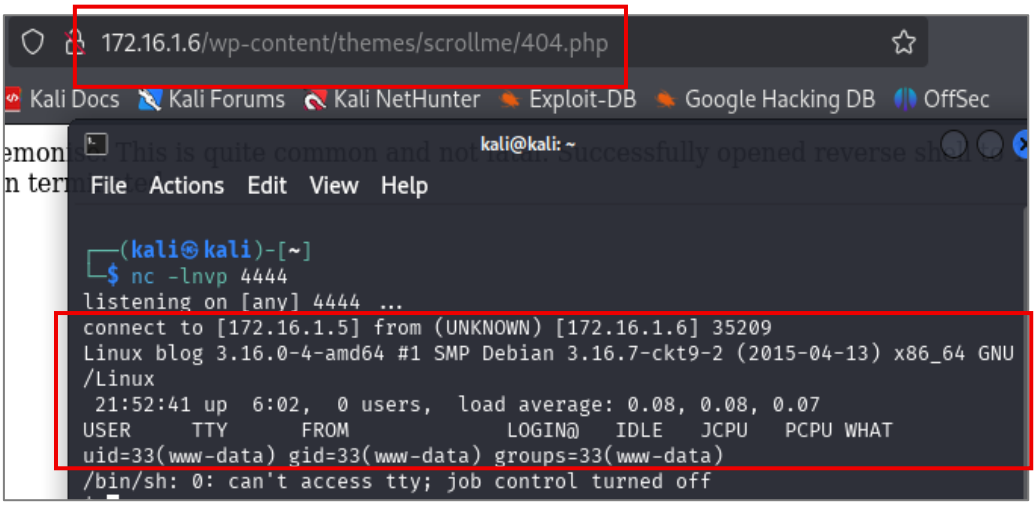


Figure 6. WordPress 3.8.1 - Successful reverse shell connection triggered by manual PHP upload.

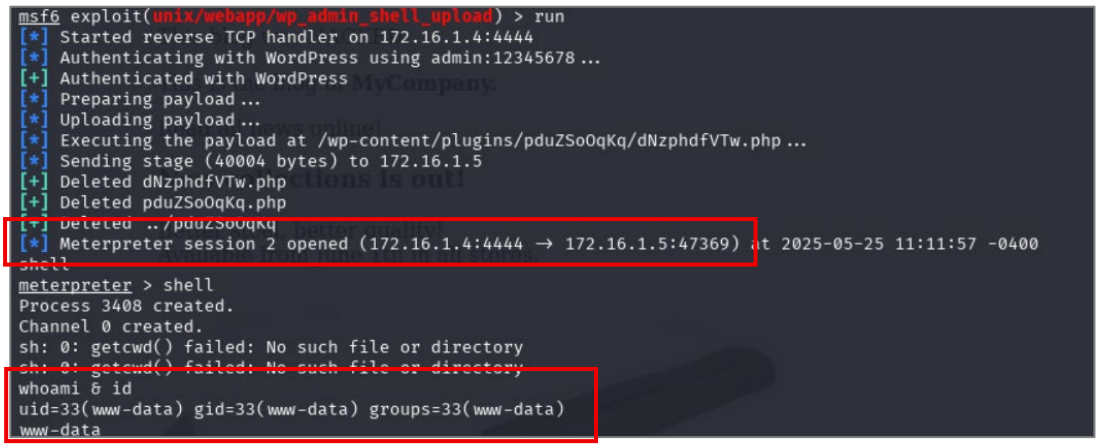


Figure 7. WordPress 3.8.1 - Successful reverse shell connection triggered by automated PHP upload.

Impact:

This vulnerability allows for RCE. Attackers can gain a shell as www-data, the default web user. They may exfiltrate sensitive system files, such as configuration files (Figure 9). The configuration file, wp-config.php, is accessible through this shell and contains the DB credentials in plaintext. As the user www-data, the attacker can use the DB credentials to view existing users (Figure 8).

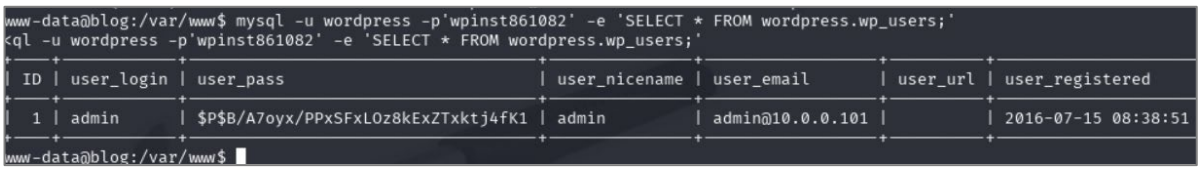


Figure 8. Successful Connection to the WordPress 3.8.1 Database Showing Existing User Accounts.

```
www-data@blog:/var/www$ ls
ls
index.php         wp-blog-header.php    wp-cron.php         wp-mail.php
license.txt       wp-comments-post.php  wp-includes         wp-settings.php
readme.html       wp-config-sample.php  wp-links-opml.php   wp-signup.php
wp-activate.php   wp-config.php         wp-load.php         wp-trackback.php
wp-admin          wp-content            wp-login.php        xmlrpc.php
```

Figure 9. WordPress 3.8.1 - Root Directory Listing (/var/www) Showing PHP Files to Exfiltrate.

Finding #4: Privilege Escalation – Dirty COW Exploit (Linux 3.16.0)

Risk Level: Critical

Evidence:

Using the shell, the Linux machine's version was exposed (Linux 3.16.0-64 bit) (Figure 10). This version is vulnerable to the Dirty Cow Privilege Escalation exploit (CVE-2016-5195), allowing an unprivileged user to overwrite memory, enabling root privilege escalation (Figure 10).

The exploit was hosted on the testing machine's http server. On the reverse shell, the file was installed from the testing machine's server and run, creating a root-level user (firefart) with a password (Figure 10). Running another shell in parallel, attackers can login to the user (firefart) using the new password and execute root commands momentarily (Figure 10).

```
www-data@blog:$ su firefart
su firefart
Password: leen

shell-init: error retrieving current directory: getcwd: cannot access parent directories: No
 such file or directory
firefart@blog:# whoami
whoami
firefart
firefart@blog:# id
id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@blog:# uname -a
uname -a
Linux blog 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt9-2 (2015-04-13) x86_64 GNU/Linux
```

Figure 10. Successful login to root user firefart.

Impact:

The root access is temporary, bypassing all user/process-level restrictions. Attackers could install rootkits, modify/exfiltrate system files, or create backdoors. The shell soon causes the server to crash (Figure 11), causing a complete denial of service.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sS 172.16.1.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-25 15:47 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 1.56 seconds
```

Figure 11. MyHobbyServer crashes after root access is granted.

Finding #5: WordPress 3.8.1 – Insecure Storage of Database Credentials

Risk Level: High

Evidence: The database credentials are stored in plaintext in the wp-config.php file (Figure A.2). Since this file is readable by the web server user (www-data), reverse shells expose these credentials.

Impact: Web server-level access allows access to the backend database using the credentials,allowing full read/write access to user data, site content, and sensitive configurations.

Finding #6: WordPress 3.8.1 – Unencrypted Transmission of Login Credentials

Risk Level: Critical

Evidence: Login attempts to wp-login.php are sent over HTTP. Using Burpsuite, traffic can be captured when a user (like admin) logs in, revealing plaintext credentials (Figure 12).

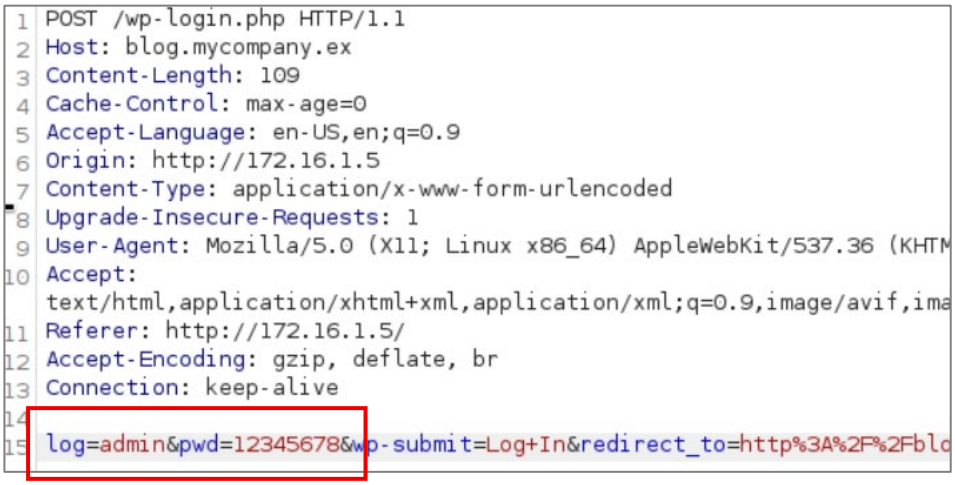


Figure 12. Intercepted login request reveals admin credentials in plain text (admin:12345678).

Impact: Attackers on the same network could intercept login credentials,granting unauthorized access to the WordPress admin panel. Risks include unauthorized admin control, site defacement, and full database/file access.

Finding #7: WordPress 3.8.1 – Lack of Dual Control on Admin Actions

Risk Level: Medium

Evidence: An admin can edit content, upload PHP files, create new users, and change credentials without requiring approval or generating alerts. The lack of separation of duties or the need for dual account approval allows a single admin (or attacker) control.

Impact: Attackers can take complete control of the WordPress site silently, uploading malicious payloads, changing site appearance, or creating persistent backdoors. No review or secondary approval is triggered, increasing long-term compromise.

DISGUISE Server

This machine's specifications are summarized in Table 5. It was assigned the IP address 172.16.1.6, running on Linux. Figure A.3 displays the full scan.

Disguise.Machine.Specifications	
Attribute	Details
IP.Address	172.16.1.6
Hostname	disguise.hmv
Operating.System	Linux
SSH.Version	OpenSSH 7.9p1 Debian 10+deb10u4 (Port 22)
Web.Server.Version	Apache 2.4.59 (Port 80)
CMS.Version	WordPress 6.8.1
Open.Ports	22 (SSH), 80 (HTTP)
Running.Services	SSH, HTTP
SSH.Host.Keys	RSA (2048), ECDSA (256), ED25519 (256)
OS.CPE.Info	OS: Linux; CPE: cpe:/o:linux:linux_kernel

Table 5. Disguise Server: System Specifications and Active Services

Finding #1: Hidden Subdomain – dark.disguise.hmv

Risk Level: Medium

Evidence: A subdomain enumeration was performed on the main domain disguise.hmv using ffuf (Figure A.4), showing an unlisted subdomain: dark.disguise.hmv (a sales platform) (Figure 13). Since the subdomain was not accessible through standard browsing, it was likely intended to be hidden from public exposure.

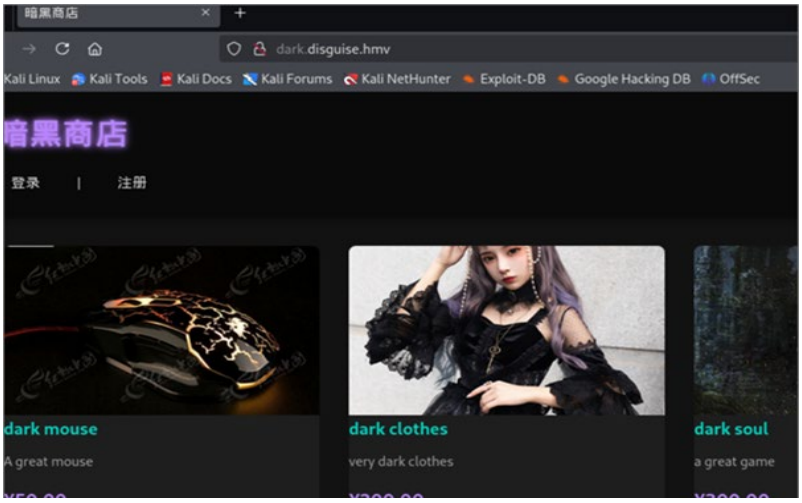


Figure 13. Home page of the subdomain dark.disguive.hmv on Disguise Server (172.16.1.6).

Impact:

This subdomain may allow for staging environments, admin panels, etc... If unsecure, it exposes sensitive services or outdated applications. This subdomain could be an entry point to gain shell access, escalating to root.

Finding #2: Insecure Session Management

Risk level: High

Evidence:

Logging into dark.disguise.hmv, the request is transmitted over HTTP, displaying plaintext credentials (Figure 14). Attackers on the same network can intercept and steal the credentials. Similarly, the response from the server returns a dark session cookie, unique to each user, in plaintext. The cookie is used in later requests to authenticate users while site-navigating.

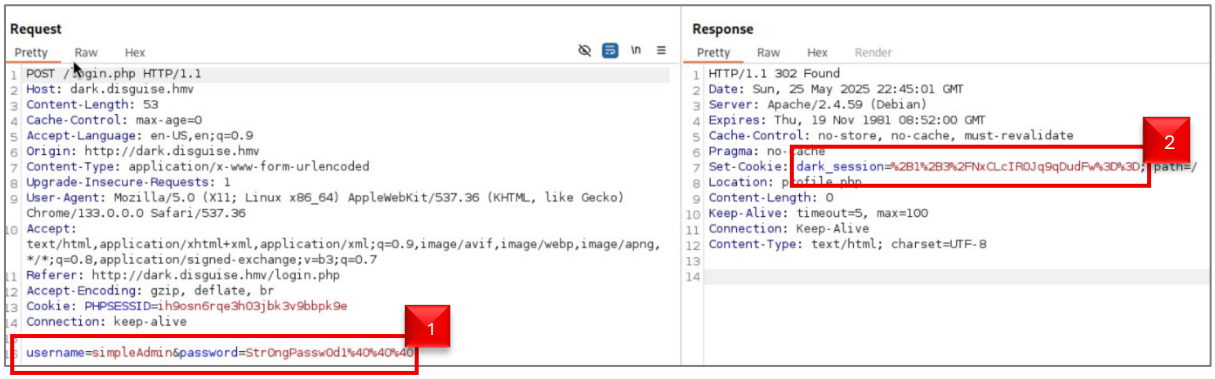


Figure 14. Intercepted login request to dark.disguise.hmv, displaying plaintext credentials (1) and the unencrypted dark session cookie (2) for username simpleAdmin.

Impact:

The lack of credential encryption allows easy theft of credentials, allowing attackers to login as authenticated users (like admin). In fact, intercepting and reusing the dark session cookie is enough to impersonate a user.

This was demonstrated using two users logging in, *simpleAdmin* and *Leen (Figures 14/15-Table 3)*. The user *simpleAdmin* is a pre-existing account with admin privileges. The user's credentials were determined by intercepting an admin login. The user *Leen* was created to simulate a real site user.

Existing User Accounts

Username	Password	Admin	Pre-existing	Dark session ID
SimpleAdmin	Str0ngPassw0d1@@@	Yes	Yes	%2B1%2B3%2FNxCLcIROJq9qDudFw%3D%3D
Leen	Leen	No	No	4kzT7IP%2Bq%2F5Df674dVq%2Fcw%3D%3D

Table 3. Specifications of discovered/created user accounts on dark.disguise.hmv.

The admin's dark session cookie was intercepted using Burpsuite during login. The user *Leen* could then send a GET request for the /profile.php endpoint, injecting the admin's

dark cookie, to access to the admin's profile (Figure 16). The user *Leen* can view the admin's page and make admin-level changes. So, the dark session cookie alone allows attackers to impersonate admin users. This is especially risky if the dark session cookie's algorithm is easy to crack, allowing attackers to login to any user.

Figure 16. Intercepted login request to dark.disguise.hmv, displaying plaintext credentials (1) and the unencrypted dark session cookie (2) for username leen.

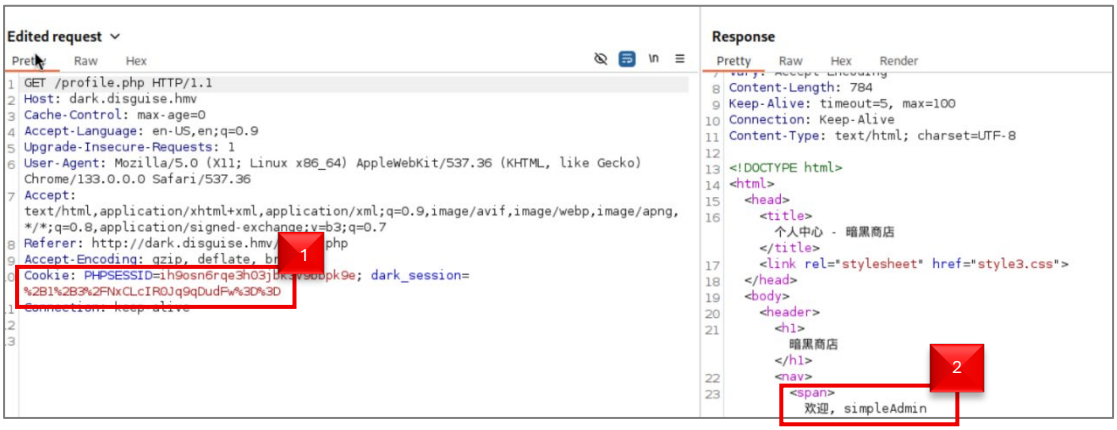


Figure 17. Attacker (user 'Leen') hijacks admin session by injecting a stolen dark cookie via GET request to /profile.php, gaining access to the admin's profile.

Finding #3: Remote Code execution via Reverse Shell Upload

Risk level: High

Evidence: The application allows admins to upload product images (with a name, description and price) but lacks proper file-type validation. Impersonating an admin, attackers can change the file filter to "All-Files" (Figure 17) and upload a PHP reverse

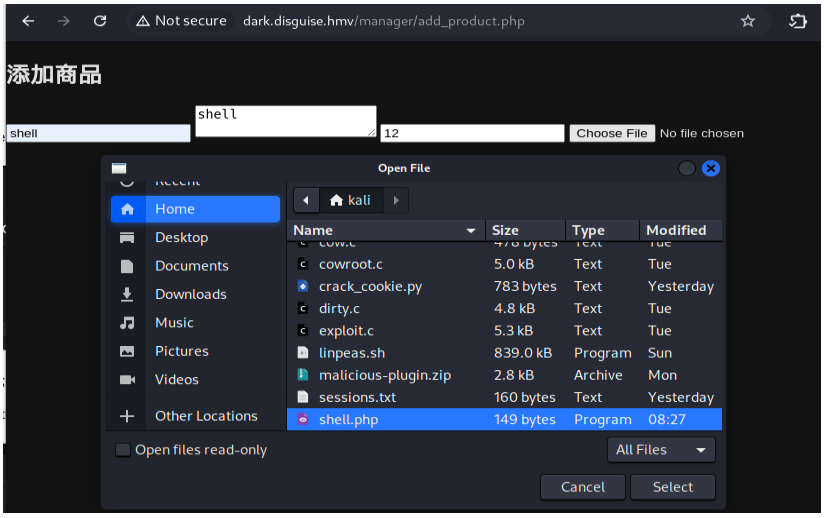


Figure 18. Uploading malicious shell.php file onto dark.disguise.hmv.

shell file (Figure A.5). When executed, this file connects back to the attacker's machine (172.16.1.4:4444), granting shell access. To locate the uploaded file's path, the price input was modified using Burp Suite to trigger an SQL syntax error (e.g., submitting 6') (figure 19).

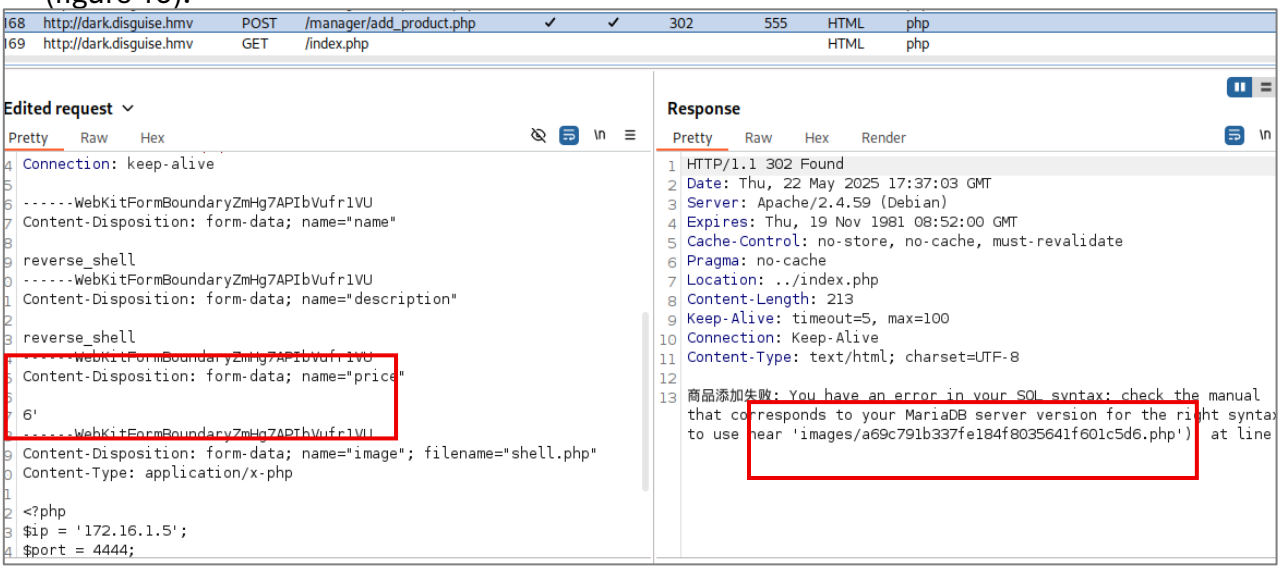


Figure 19. Modifying the price value to trigger an SQL error while posting a new product.

The resulting error reveals the full URL of the uploaded file, allowing direct access to trigger the reverse shell (Figure 19). When the file is accessed, a listener on the attacker's machine will establish the connection (Figure A.6).

Once connected, attackers have a reverse shell, with user www-data, which enumerates system details (Figure 20), including users root and darksoul (Figure 22) and files like config.ini containing database credentials (Figure 21/24). Figure 20 lists system files along with their privileges, highlighting the current user can read config.ini but not user.txt.

```
python3 -c 'import pty; pty.spawn("/bin/bash");'
www-data@disguise:/var/www/dark/images$ cd /
cd /
www-data@disguise:/$ whoami
whoami
www-data
www-data@disguise:/$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@disguise:/$ uname -a
uname -a
Linux disguise 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64 GNU/
Linux
www-data@disguise:/$ hostname
hostname
disguise
www-data@disguise:/$ sudo -l
sudo -l
sudo: unable to resolve host disguise: Temporary failure in name resolution

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for www-data:
```

Figure 20. Enumerating system details using the reverse shell established on Disguise

```
www-data@disguise:/$ ls -la /home/darksoul
ls -la /home/darksoul
total 40
drwxr-xr-x 4 darksoul darksoul 4096 Apr  2 04:19 .
drwxr-xr-x 3 root     root     4096 Mar 31 11:19 ..
lrwxrwxrwx 1 root     root        9 Apr  2 00:16 .bash_history → /dev/null
-rw-r--r-- 1 darksoul darksoul  220 Mar 31 11:19 .bash_logout
-rw-r--r-- 1 darksoul darksoul 3526 Mar 31 11:19 .bashrc
drwx------ 3 darksoul darksoul 4096 Apr  1 10:03 .gnupg
drwxr-xr-x 3 darksoul darksoul 4096 Apr  1 10:04 .local
-rw-r--r-- 1 darksoul darksoul  807 Mar 31 11:19 .profile
-rw-r--r-- 1 root     root      114 Apr  2 04:03 config.ini
-rw-r--r-- 1 root     root       31 May 22 08:42 darkshopcount
-rw------- 1 darksoul darksoul   68 Apr  2 04:22 user.txt
```

Figure 21. Listing system files and their respective privileges on Disguise server.

```
www-data@disguise:/$ cat /etc/passwd | grep /bin/bash
cat /etc/passwd | grep /bin/bash
root:x:0:0:root:/root:/bin/bash
darksoul:x:1000:1000:,,,:/home/darksoul:/bin/bash
```

Figure 22. Enumerating system users on Disguise server.

Impact:

Using credentials stored in config.ini, attackers can access the web server's DB, called dark_shop (Figure A.7), viewing users and products listed. The users are vulnerable since passwords are stored simply base64-encoded (figure 23). Attackers can even modify users, making them admins (figure 23).

```
MariaDB [dark_shop]> UPDATE users SET isAdmin = 1 WHERE username = 'leen';
UPDATE users SET isAdmin = 1 WHERE username = 'leen';
Query OK, 0 rows affected (0.000 sec)
Rows matched: 1  Changed: 0  Warnings: 0

MariaDB [dark_shop]> select * from users;
select * from users;
+----+-------------+----------------------------+---------+---------------------+
| id | username    | password                   | isAdmin | created_at          |
+----+-------------+----------------------------+---------+---------------------+
|  1 | simpleAdmin | U3RyMG5nUGFzc3cwZDFAQEA=   |       1 | 2025-04-01 04:33:22 |
| 30 | leen        | bGVlbg==                   |       1 | 2025-05-21 15:43:57 |
| 31 | leen1       | bGVlbg==                   |       0 | 2025-05-21 16:33:52 |
| 32 | leen12      | bGVlbg==                   |       0 | 2025-05-21 16:38:49 |
| 33 | leen123     | bGVlbg==                   |       0 | 2025-05-21 16:40:51 |
| 34 | leen1234    | bGVlbg==                   |       0 | 2025-05-21 16:41:57 |
+----+-------------+----------------------------+---------+---------------------+
```

Figure 23. Modifying users in dark_shop DB using reverse shell on Disguise server.

Finding #4: Reverse Shell Upload -Privilege Escalation

Risk level: High

Evidence: Attackers can brute force the user darksoul's password easily. Two discovered passwords have been preluded with 'Str0ngPassw0d1' followed by symbols. The brute force is successful, revealing the credentials 'darksoul: Str0ngPassw0d1???' (Figure A.8). Reusing a password pattern is a vulnerability that can escalate privileges by logging into darksoul.

```
darksoul@disguise:~$ cat config.ini
cat config.ini
[client]
user = dark_db_admin
password = Str0ngPassw0d1**
host = localhost
database = dark_shop
port = int(3306)
```

Figure 24. Contents of config.ini file.

Impact:

As darksoul, the attacker's privileges are escalated (Figure 24). Darksoul has uid=1000, a human user with more privileges than www-data. Darksoul can read the file user.txt, containing a hidden flag (Figure 25).

```
www-data@disguise:/$ su darksoul
su darksoul
Password: Str0ngPassw0d1???

darksoul@disguise:/$ whoami & id
whoami & id
[1] 7952
darksoul
uid=1000(darksoul) gid=1000(darksoul) groups=1000(darksoul)
[1]+  Done                    whoami
darksoul@disguise:/$
```

Figure 25. Enumerating user id and name on Disguise server.

```
darksoul@disguise:~$ cat user.txt -A
cat user.txt -A
Good good study & Day day up,but where is the flag?$
hmv{hiddenflag}^Mdarksoul@disguise:~$ xxd user.txt
xxd user.txt
00000000: 476f 6f64 2067 6f6f 6420 7374 7564 7920  Good good study
00000010: 2620 4461 7920 6461 7920 7570 2c62 7574  & Day day up,but
00000020: 2077 6865 7265 2069 7320 7468 6520 666c   where is the fl
00000030: 6167 3f0a 686d 767b 6869 6464 656e 666c  ag?.hmv{hiddenfl
00000040: 6167 7d0d                                ag}.
darksoul@disguise:~$
```

Figure 26. Contents of user.txt displaying a hidden flag on Disguise server.

Finding #5: Root Privilege Escalation

Risk level: Critical

Evidence:

As darksoul, attackers can install tools to identity attack routes. The tool 'pspy64' can be installed to locate background processes that are regularly run by root. The file query.py is regularly executed by root and uses config.ini for configurations (Figure 26).

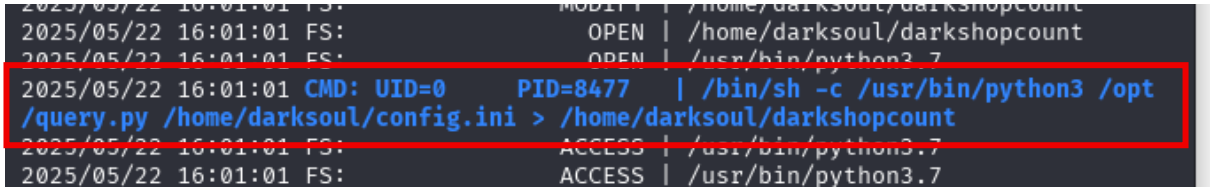


Figure 26. Process query.py being run by root, using config.ini.

Attackers can inject shell code into the file config.ini, causing root to run it. The file query.py uses MySQL Connector v8.0.33, vulnerable to code injection via the allow_local_infile parameter. The original config.ini can be appended with a line to allow for shell connection (Figure 27).

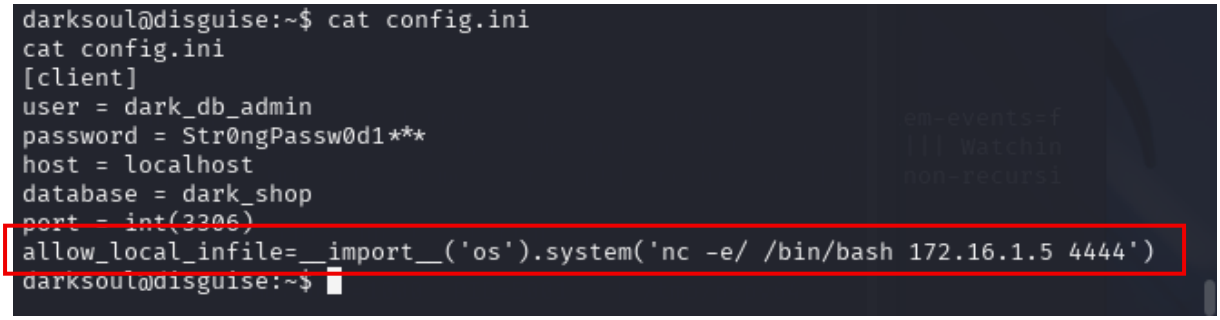


Figure 27. Modifying config.ini to add a shell injection on Disguise server.

Once the root executed query.py, a root-level reverse shell was established (Figure 28).

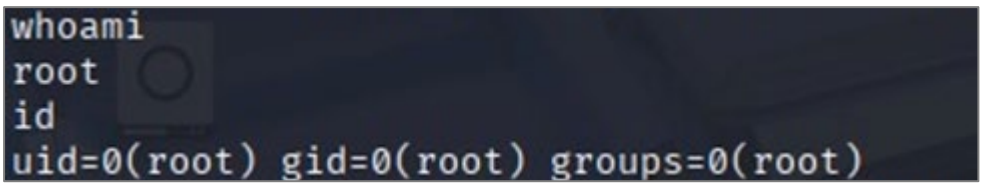


Figure 28. Establishing a root shell when query.py is run by root on Disguise server.

Impact:

As root, attackers have full control, altering system files. For example, attackers can add ssh keys for persistent server access or read the file root.txt, containing a hidden flag (Figure 29).

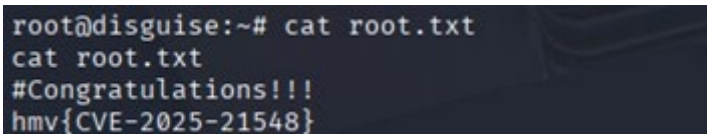


Figure 29. Accessing root.txt using root shell on Disguise server.

MSEdge – Win10

This machine's specifications are summarized in Table 6. It was assigned the IP address 172.16.1.7, running on windows. Figure A.9 displays the full scan.

MSEdge.- .Win.76.Machine.Specifications

Attribute	Details
IP.Address	172.16.1.7
Hostname	MSEDGEWIN10
Operating.System	Windows 10 (Build 10.0.17763) – Likely Windows 10 Enterprise LTSB
NetBIOS.Name	MSEDGEWIN10
Open.Ports	135 (RPC), 139 (NetBIOS), 445 (SMB), 3389 (RDP), 5985 (HTTP), 7680, 49686
Running.Services	Microsoft RPC, SMBv1/NetBIOS, Remote Desktop, HTTP (HTTPAPI 2.0), Pando-pub
SSL.Cert.Common.Name	MSEDGEWIN10
MAC.Address	08:00:27:E6:E5:59 (PCS Systemtechnik / Oracle VirtualBox)
OS.CPE.Info	OS: Windows; CPE: cpe:/o:microsoft:windows
Attribute	Details

Table 6.MSEdge -Win 10 Server: System Specifications and Active Services

Finding #1: SMB Signing Not Required

Risk Level: High

Evidence:

The Nmap scan (Figure A.10) shows ports 139 (NetBIOS-ssn) and 445 (SMB) are open. Nmap's script smb2-security-mode highlights the current configurations allow SMB connections without mandatory signature validation.

Impact:

The host is vulnerable to SMB relay attacks, whereby hackers intercept NTLM authentication hashes when SMB signing is not enforced. Without the user's

knowledge, attackers can use authentication attempts to obtain unauthorized access to shared files or resources. Unauthorized data access, credential theft, and lateral network movement are possible.

Finding #2: RDP Exposed (BlueKeep Risk)

Risk Level: Medium

Evidence:

The Nmap scan (Figure A.10) shows port 3389 is open, hosting Microsoft Terminal Services.

Network Level Authentication (NLA) does not seem to be enforced. If NLA is not enforced, the host may be vulnerable to CVE-2019-0708 (BlueKeep) – a critical pre-auth RCE flaw in RDP.

Impact:

An unauthenticated attacker could execute code remotely or brute-force RDP login credentials. In some cases, this could result in full remote system compromise, ransomware infection, or persistence via GUI access.

Finding #3: HTTPAPI Exposed Over HTTP (WinRM)

Risk Level: Medium

Evidence:

The Nmap scan shows port 5985 open, running Microsoft HTTPAPI httpd 2.0 (Figure A.11). No SSL/TLS certificate is shown, confirming use of unencrypted HTTP. Port 5985 is the default for Windows Remote Management (WinRM) using HTTP, allowing remote PowerShell and administrative actions.

Impact:

These unencrypted WinRM sessions can be intercepted over the network. Once intercepted, attackers can view administrative actions and PowerShell remoting to MITM attacks. This would allow for session hijacking, command injection, and credential exposure.

Finding #4: Outdated OS – Windows 10 Build 17763

Risk Level: Medium

Evidence:

The Nmap scan shows Windows 10 Version 1809, which is End-of-Life for most editions. This build is missing patches for local privilege escalation vulnerabilities, such as CVE-2021-36934 (HiveNightmare) and CVE-2022-21919 (User Profile Service Escalation).

Impact:

Local attackers (like through a reverse shell) can possibly escalate privileges to system-level, giving control over the machine. Attackers can then install persistence mechanisms and kernel-level malware.

Win Server 2019

This machine's specifications are summarized in Table 7. It was assigned the IP address 172.16.1.8, running on windows. Figure A.10 displays the full Nmap scan.

Win.Server.867❺Machine.Specifications

Attribute	Details
IP.Address	172.16.1.8
Hostname	NY2016SERVER
Operating.System	Windows Server 2019 (Based on service versions + NetBIOS naming)
Open.Ports	53, 80, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 5985, 5986, 9389, 47001
Running.Services	DNS, Kerberos, LDAP, RPC, SMB, LDAPS, WinRM, HTTP(S), NetBIOS, Microsoft IIS
Web.Servers	Microsoft IIS 10.0, Microsoft HTTPAPI 2.0 (SSDP/UPnP)
Certificate.Subject	NY2016SERVER.wngpma.local
MAC.Address	08:00:27:4e:a7:a2 (PCS Systemtechnik / Oracle VirtualBox)
OS.CPE.Info	OS: Windows; CPE: cpe:/o:microsoft:windows

Table 7. Win Server 2019: System Specifications and Active Services

Finding #1: IIS 10.0 – Trace Method enabled

Risk Level: Medium

Evidence:

The Nmap scan (Figure A.10) shows the enabled TRACE method on the Server Microsoft-IIS/10.0. This was further validated by manual testing, where a TRACE request to the server successfully returned a complete echo of the submitted HTTP headers, demonstrating the method's active status.

Impact:

The enabled TRACE method has potential for Cross-Site Tracing (XST) attacks. Attackers could exploit this in order to bypass HTTPOnly cookie protections by retrieving session

tokens through reflected TRACE responses. They could even chain this vulnerability with an XSS attack to automate session hijacking without needing to interact with users.

Finding #2: Unencrypted WinRM Over HTTP

Risk Level: Medium

Evidence:

Network scanning (Figure A.10) confirmed port 5985 (WinRM/HTTP) is accessible. Service fingerprinting shows no TLS/SSL certificate presented.

Impact:

This vulnerability allows for credentials exposure and compliance violations. All PowerShell Remoting, WMI queries, and automation tasks via WinRM are transmitted in plaintext. Attackers can intercept admin commands or credentials via MITM and replay captured sessions to execute unauthorized actions.

Finding #3: Unsecured Active Directory Ports

Risk Level: Low

Evidence:

Active Directory service ports were exposed: LDAP (389/tcp) with unencrypted directory access, LDAPS (636/tcp), with certificate issued to NY2016SERVER.wngpma.local, and Global Catalog (3268/3269), with Forest-wide directory queries (Figure A.10). No display of LDAP signing or channel binding and the misconfiguration of certificate validation shows potential for spoofing.

Impact:

If LDAP signing or channel binding is not enforced, attackers can man-in-the-middle LDAP authentication. The Open LDAP/GC ports can even be used for Active Directory enumeration and brute-force attacks.

Finding #4: Information Leakage via IIS Misconfigurations

Risk Level: Low

Evidence:

Nmap scan shows ports 80,8080 and 47001 have inconsistent HTTP titles, like “Not Found” (Figure A.10).

Impact:

Attackers can use these vulnerabilities to conduct recon. Default banners and error messages expose server versions (IIS/10.0), framework details (ASP.NET), and file system paths. This allows for version-specific exploits, informs attack vector selection and enables local file inclusion (LFI).

DevServer

This machine's specifications are summarized in Table 8. It was assigned the IP address 172.16.1.7, running on windows. It was noted that this machine shares the same IP and MAC address as the MSEdge – Win10 Server. Figure A.11 displays the full Nmap scan.

DevServer.Machine.Specifications

Attribute	Details
IP.Address	172.16.1.9
Hostname	localhost
Operating.System	Windows (Based on Apache Win64 header and Mercury/32 services)
MAC.Address	08:00:27:E6:E5:59 (Oracle VirtualBox Virtual NIC)
Open.Ports	21, 25, 79, 80, 105, 106, 110, 143, 443, 2224, 3306
Web.Server	Apache 2.4.58 (Win64) with PHP 8.0.30, OpenSSL 3.1.3
FTP.Server	FileZilla 0.9.41 beta (UNIX emulation)
Mail.Services	Mercury/32: SMTP (25), POP3 (110), IMAP (143), Addressbook (105, 106)
Database.Server	MariaDB 10.3.23 or earlier (port 3306)
CPE.Info	cpe:/o:microsoft:windows

Table 8.DevServer: System Specifications and Active Services

Finding #1: Insecure FileZilla FTP Server Configuration

Risk Level: Medium

Evidence:

Nmap detected FileZilla FTP Server (0.9.41 Beta) running on port 21/tcp (Figure A.11). This outdated and unsupported version allows for anonymous login capability if settings are misconfigured. It also transmits credentials and data in plaintext and discloses the version in the banner (ftpd 0.9.41 beta).

Impact:

These misconfigurations allow attackers to steal credentials by sniffing unencrypted logins through MITM attacks. If authentication is enabled, credentials may be brute forced. Interceptors can capture transferred data, leading to data exfiltration.

Finding #2: Apache Web Server Information Disclosure

Risk Level: Medium

Evidence:

Nmap scan of ports 80/443 shows “Server: Apache/2.4.58 PHP/8.0.30”, discloses the version clearly. The PHP’s version has reached end-of-life and has not been supported since Nov 2023.

Impact:

Attackers can use version data to exploit known Apache/PHP CVEs. PHP 8.0.x is vulnerable to RCE via auto_prepend_file manipulation. This can expose paths/plugins inform attack chains for attackers.

Finding #3: Unsecured Mercury/32 Mail Server Exposure

Risk Level: Medium

Evidence:

Mercury/32 is mentioned across ports: smptd, imap4, pop3, http admin (Figure A.11). In fact, port 79 leaks admin information, mentioning “login: admin | Mail System Administrator”, exposing an admin username.

Impact:

The admin username exposure enables attackers to enforce targeted brute-forcing. CVE-2005-1523, related to Mercury/32 mail relay abuse, may be exploited.

Finding #4: Vulnerable MariaDB Database Server Version

Risk Level: Low

Evidence:

MariaDB version 10.3.23 is detected on port 3306/tcp (Figure A.11). This version is unsupported since 2023, and the unauthorized version disclosure is also a vulnerability.

Impact:

The version exposure can lead to version-specific DB exploits. It can be exploited to dump or alter sensitive backend data, possibly leading to data breaches and privilege escalation.

Finding #5: Expired SSL/TLS Certificate on Port 443

Risk Level: Medium

Evidence:

Nmap shows port 443/tcp is serving HTTPS (Figure A.11). The SSL certificate displayed is self-signed, expired since 2019, and has an end date before its start date. Clients will reject this certificate or connect insecurely, and some may switch to HTTP.

Impact:

When users who access the site through HTTPs encounter trust failures, attackers may launch Man-in-the-Middle (MITM) attacks. This gets unsuccessful encryption attempts or authentication bypasses. Expired, self-signed certificates compromise TLS integrity and result in session hijacking attacks.

VI. Recommendations

Table 9 lists remediation strategies for each identified vulnerability.

Table 9. Remediation Strategies for Identified Vulnerabilities

RISK ID	VULNERABILITY	AFFECTED VM	RISK LEVEL	RECOMMENDED REMEDIATION
1	OpenSSH 6.7 – User Enumeration	MyHobbyServer	Low	Upgrade OpenSSH to a patched version (OpenSSH 7.7+). Enable Fail2Ban to block Brute-Force Attempts on usernames.
2	WordPress 3.8.1 - Weak WordPress Admin Credentials	Disguise	Critical	Enforce strong password policies and implement two-factor authentication. Upgrade WordPress to the latest secure version.
3	Stored XSS & Reverse Shell via Theme Editor	MyHobbyServer	Critical	Add *define('DISALLOW_FILE_EDIT', true);* to wp-config.php to prevent PHP file modifications via the WordPress admin dashboard. Legit admins can make changes via version-controlled deployments. Ensure WordPress is upgraded to the latest stable version to patch known vulnerabilities.

4	Dirty COW Privilege Escalation (Linux 3.16.0)	MyHobbyServer	Critical	Upgrade the Linux kernel to a patched version (4.8.3+). Since Dirty COW manipulates memory mappings (/proc/self/mem), enable kernel hardening to Block Memory Exploits.
5	Plaintext DB Credentials in wp-config.php	MyHobbyServer	High	Change file ownership to root:root and set permissions to chmod 640 (restricting access to root and web server users). Move database credentials to environment variables (.env files).
6	Unencrypted WordPress Login (HTTP)	MyHobbyServer	Critical	Install an SSL certificate to encrypts all traffic between users and your server (HTTPS). Disable HTTP entirely by removing "listen 80;" from the ports.conf file.
7	Lack of Dual Control for Admin Actions	MyHobbyServer	Medium	Implement role-based access control (RBAC) with distinct admin/editor roles. Use plugins like Activity Log to track admin actions. For critical actions, like uploading a file, implement approval workflows using hooks or plugin-based logic.
8	Hidden Subdomain – dark.disguise.hmv	Disguise VM	Medium	Configure the DNS provider to alert on unintended subdomain creation. Use tools like Subfinder during audits to discover hidden subdomains.
9	Insecure Session Management	Disguise VM	High	Enforce HTTPS site-wide. Rotate session IDs post-login to prevent session fixation. Limit cookie lifetime with short TTLs (15 mins).
10	RCE via Reverse Shell File Upload	Disguise VM	High	Implement server-side file extension whitelisting (.jpg, .png only), validate MIME types, and check magic bytes.

				Store uploads outside the document root and disable PHP execution via .htaccess or php_admin_flag engine off.
11	Password Pattern Reuse (darksoul)	Disguise VM	High	Implement period password changes. Enforce strong and unique passwords use in the system.
12	Root Privilege Escalation via config.ini Injection	Disguise VM	Critical	Restrict write access to config.ini to root-only (chmod 600). Run scripts as non-root wherever possible. Use tools like pspy64 regularly to audit scheduled processes.
13	Message Signing Not Enforced	MSEdge Win 10	Medium	Configure Group Policy or registry to require SMB message signing on both clients and servers.
14	RDP Exposed (BlueKeep Risk)	MSEdge Win 10	Medium	Monitor 3389 traffic, limit RDP access using a firewall or VPN, and confirm and enforce NLA.
15	HTTPAPI Exposed Over HTTP (WinRM)	MSEdge Win 10	Medium	Enforce HTTPS-only WinRM via port 5986 with the appropriate TLS certificates and disable HTTP on port 5985.
16	Outdated OS – Windows 10 Build 17763	MSEdge Win 10	Medium	Upgrade to a supported version of Windows 10 or 11.
17	IIS 10.0 – Trace Method enabled	Win Server 2019	Medium	Disable the TRACE method by modifying the line “<add verb “TRACE” allowed = “true” />” to false in the site’s web.config.
18	WinRM Over HTTP	Win Server 2019	Medium	Enforce HTTPS on WinRM (using port 5986), disable Port 5985, and restrict remote management to specific admin subnets.
19	Unsecured Active Directory Ports (LDAP/LDAPS/GC)	Win Server 2019	Low	Enforce LDAP signing and channel binding to harden the communications. Restrict access to LDAP/GC ports from trusted IPs only.

				Replace self-signed certificates with PKI-issued certificate.
20	Information Leakage via IIS Misconfigs	Win Server 2019	Low	Harden HTTP headers (in web.config). Disable directory browsing in the IIS Manager. Sanitize ASP.NET error messages to hide system pathways.
21	Insecure FileZilla FTP Server Configuration	DevServer	Medium	In the short term, migrate to SFTP and enforce TLS 1.2+ encryption in the FileZilla settings. In the long term, replace FileZilla with Cerberus FTP on windows.
22	Apache Web Server Information Disclosure	DevServer	Medium	Harden headers in Apache config. (turn off ServerSignature) Upgrade to Apache 2.4.x latest and PHP 8.3.x In the short term, disable dangerous functions like shell_exec in php.ini.
23	Unsecured Mercury/32 Mail Server Exposure	DevServer	Medium	Restrict SMTP/POP3/IMAP to strictly the internal VLANs. Disable Finger protocol through mercury.ini.
24	Vulnerable MariaDB Database Server Version	DevServer	Low	Upgrade MariaDB to 10.6 or higher.
25	Expired SSL/TLS Certificate on Port 443	DevServer	Medium	Install a valid (up-to-date) TLS certificate from a reliable CA. Disable HTTP fallback and enforce TLS 1.2 or higher.

VII. Conclusion

This assessment revealed a range of vulnerabilities across the five machines assessed, spanning critical system misconfigurations, insecure authentication mechanisms, etc.... Several findings pose significant risks to the confidentiality, integrity, and availability of the organization's digital assets.

Some vulnerabilities require immediate remediation due to their exploitability and potential impact, while others represent security weaknesses that could be leveraged in multi-stage attacks. Addressing these risks promptly will improve the organization's security and align it with best practices.

VIII. Appendices

Appendix A: Detailed Evidence Log & Methodology

Figure A.1: Full Nmap Scan enumerating services, versions, and scripts running on MyHobbyServer (172.16.1.5).

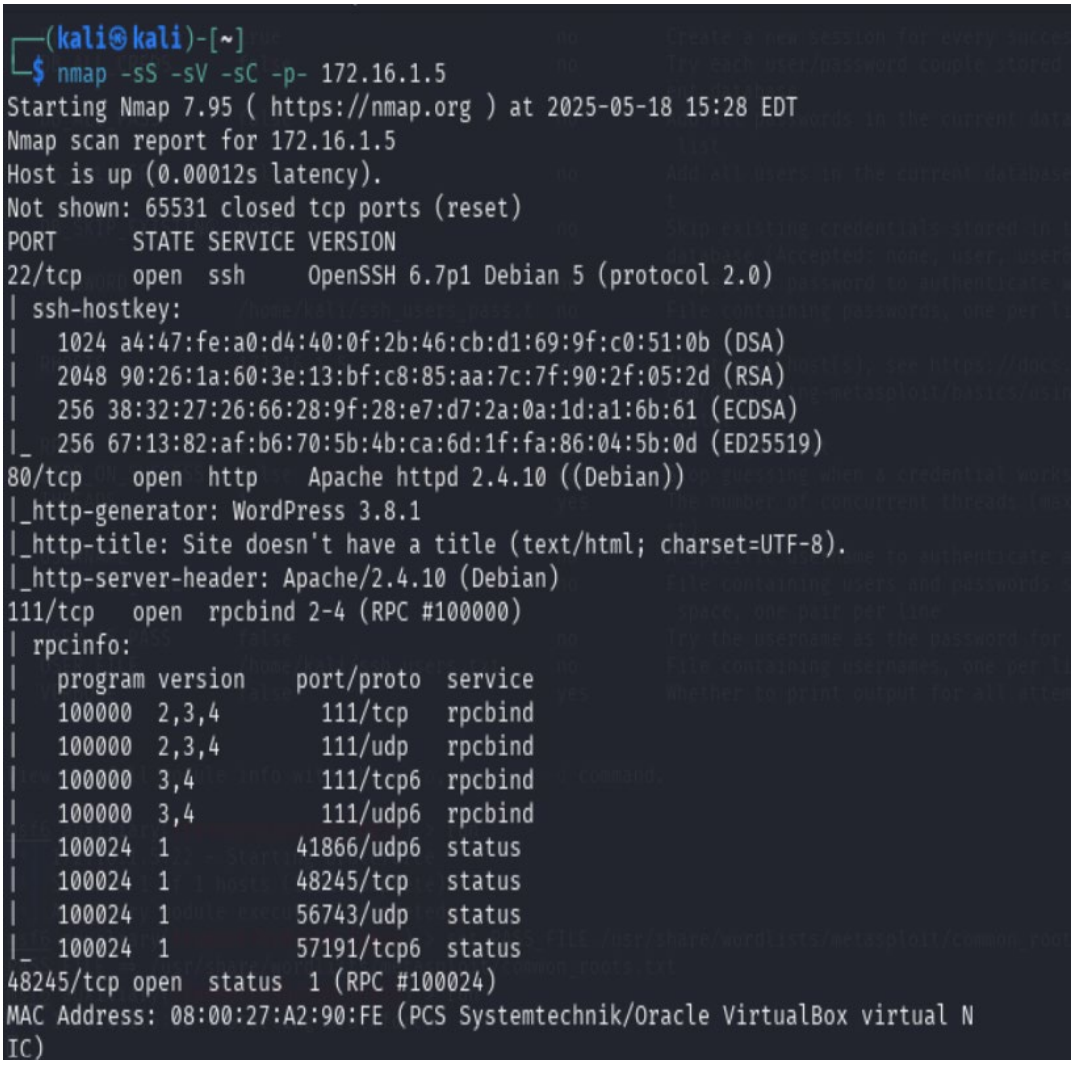


Figure A.2: Reverse Shell Access to MyHobbyServer (172.16.1.5) Exposing Plaintext Database Credentials in wp_config¡php

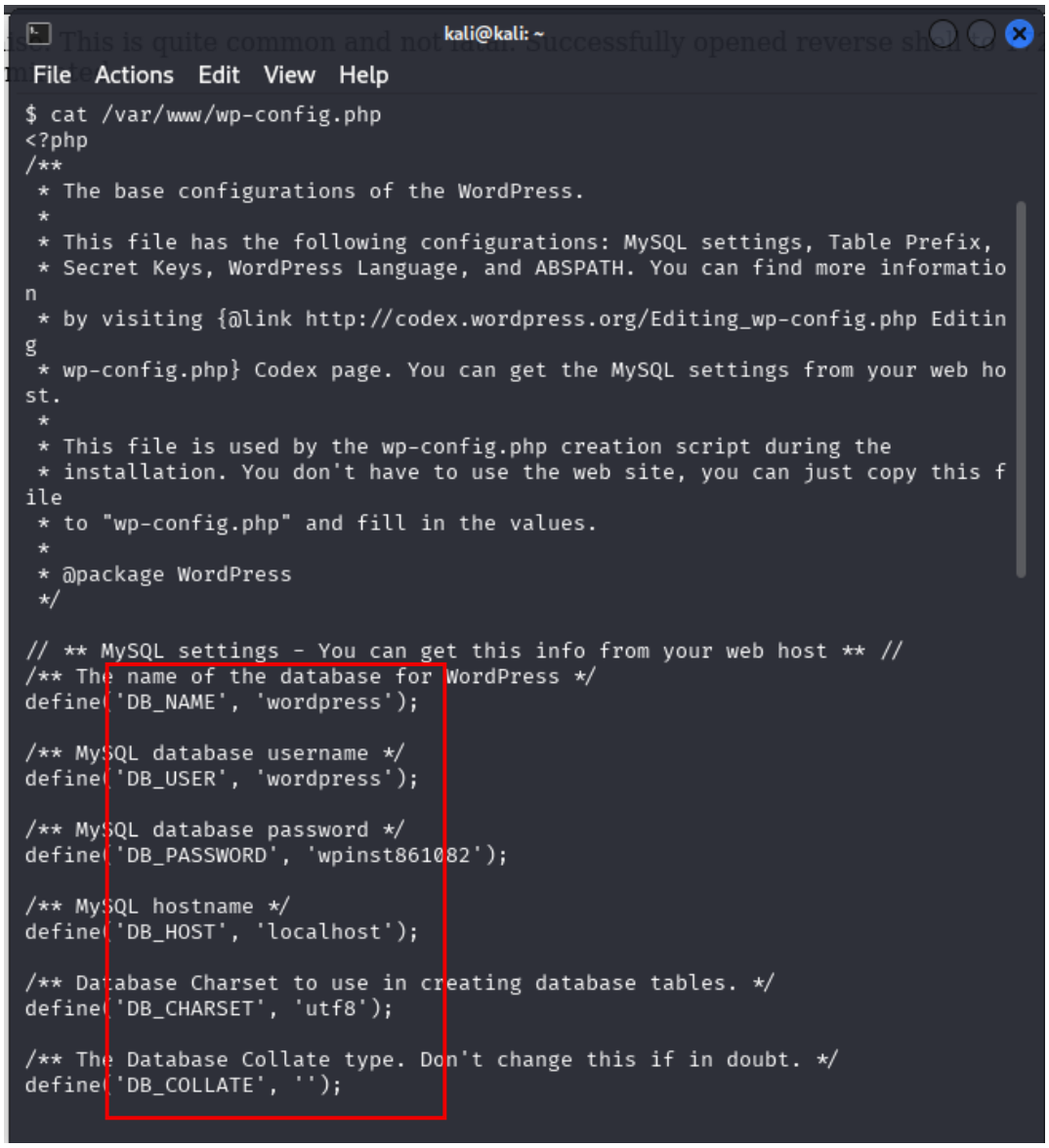


Figure A.3: Full Nmap Scan enumerating services, versions, and scripts running on Disguise Server (172.16.1.6).

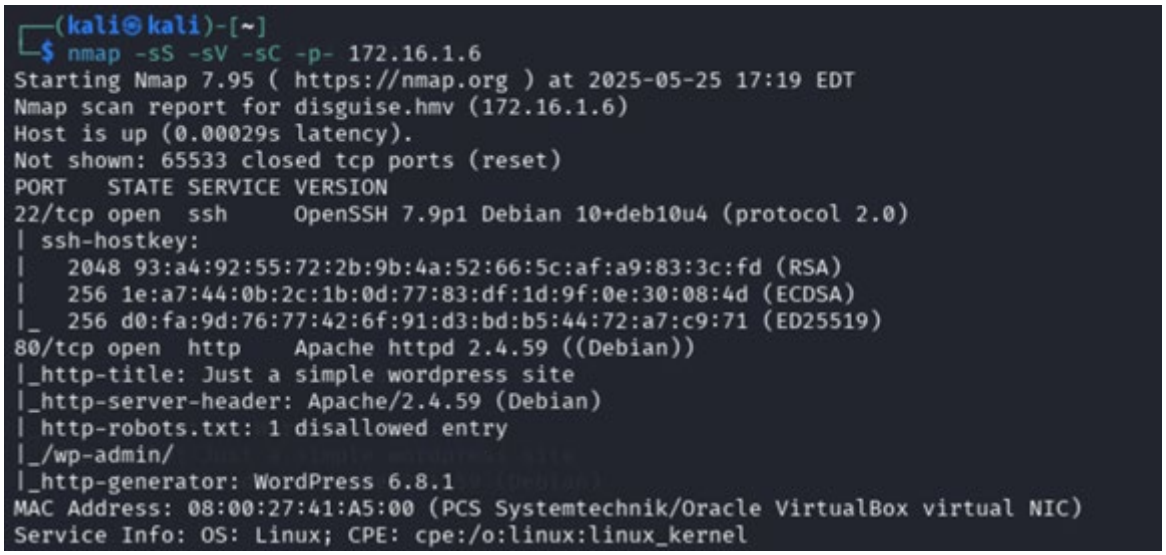


Figure A.4: Subdomain enumeration of disguise.hmv using FFUF.

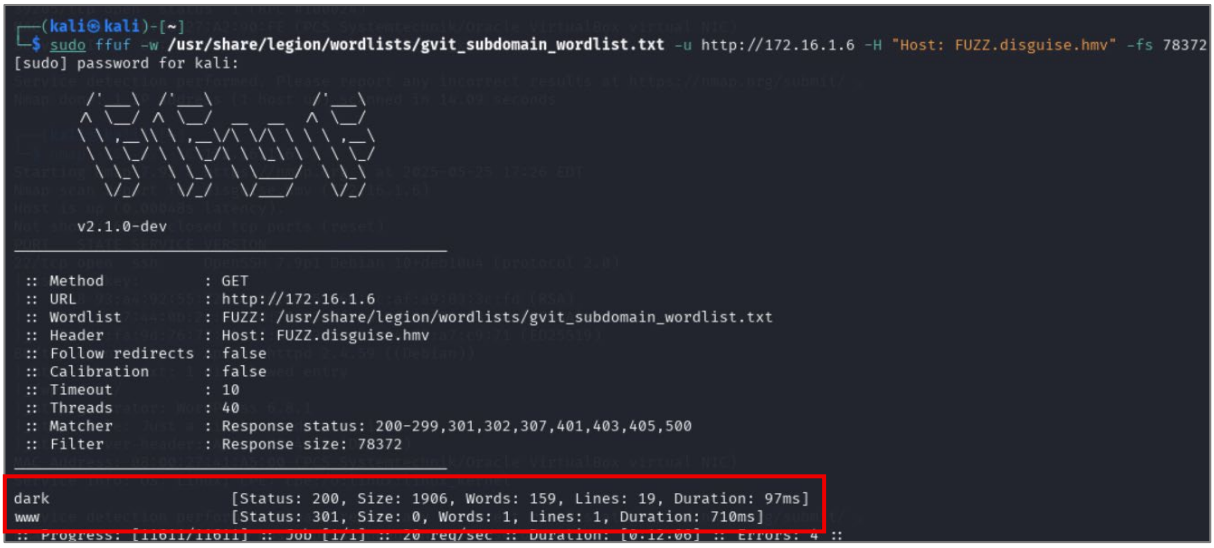


Figure A.5 Contents of the shell.php file

Figure A.6 Successful connection to the Kali machine from the target machine when accessing the shell.php's url.

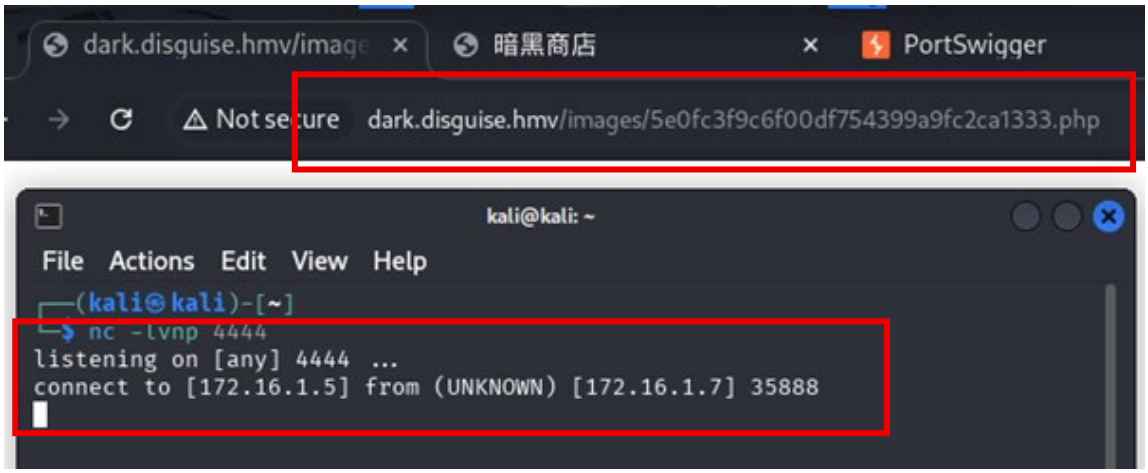


Figure A.7 Successful login to the MySQL DB on Disguise server, using config.ini.

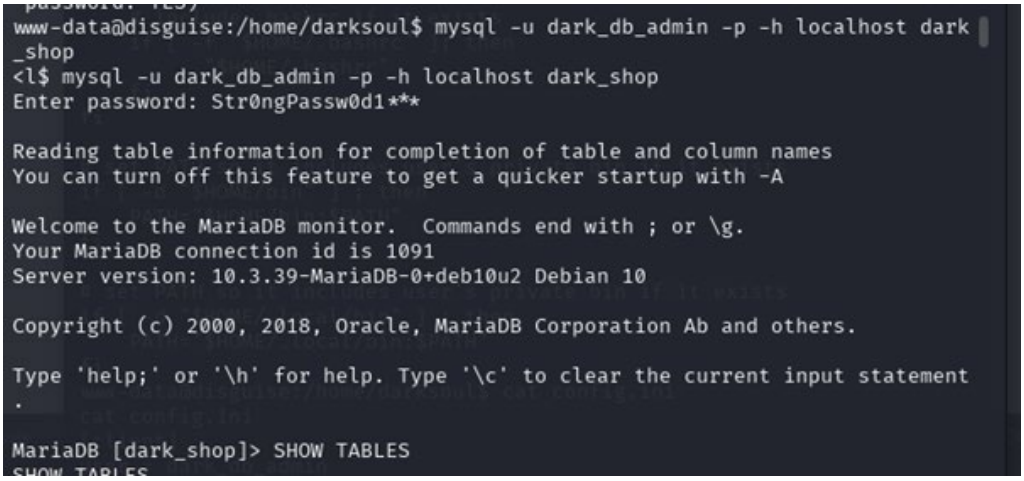


Figure A.8 Successfully brute forcing the user darksoul's password on Disguise Server.

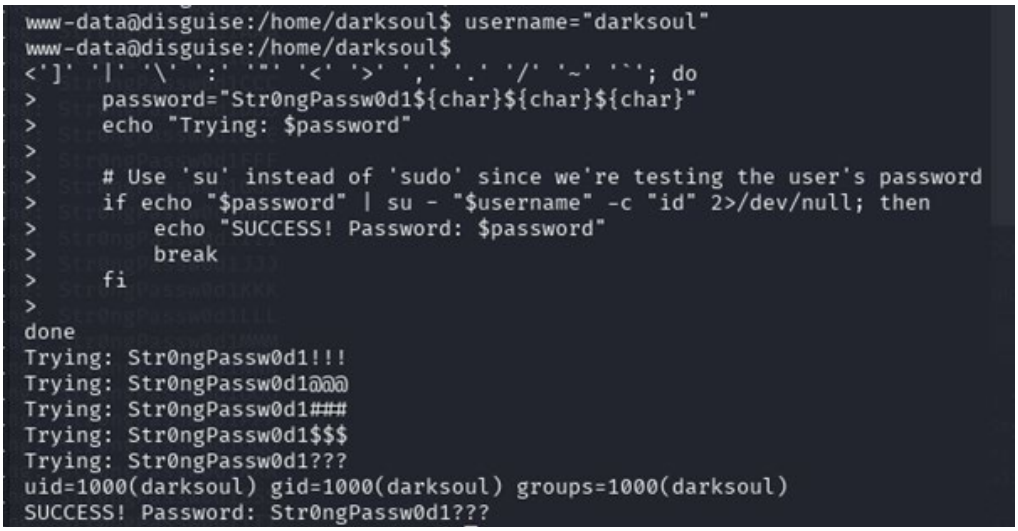


Figure A.9. Full Nmap Scan enumerating services, versions, and scripts running on MSEdge – Win 10 (172.16.1.7).

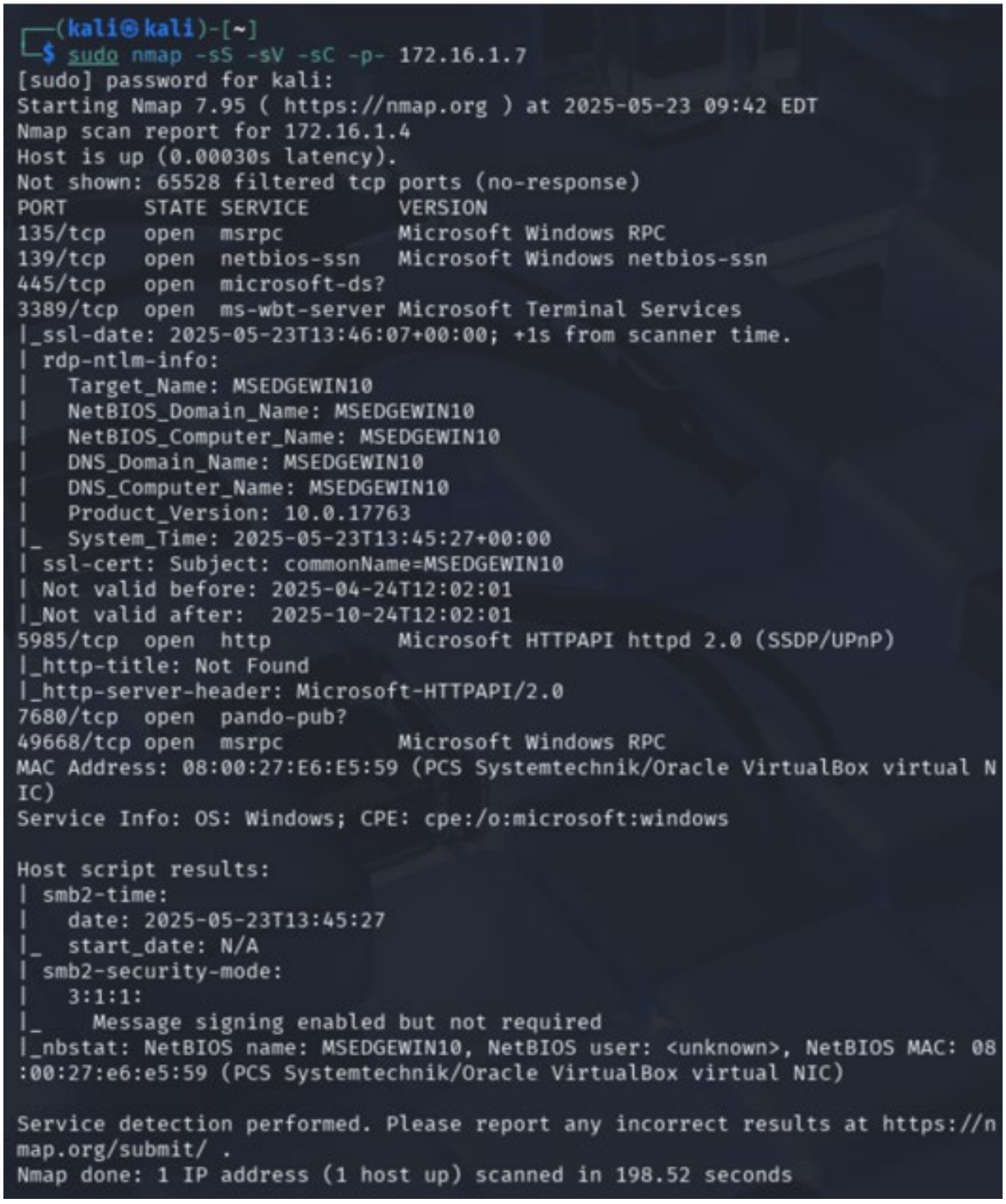


Figure A.10. Full Nmap Scan enumerating services, versions, and scripts running on Win Server 2019 (172.16.1.8).

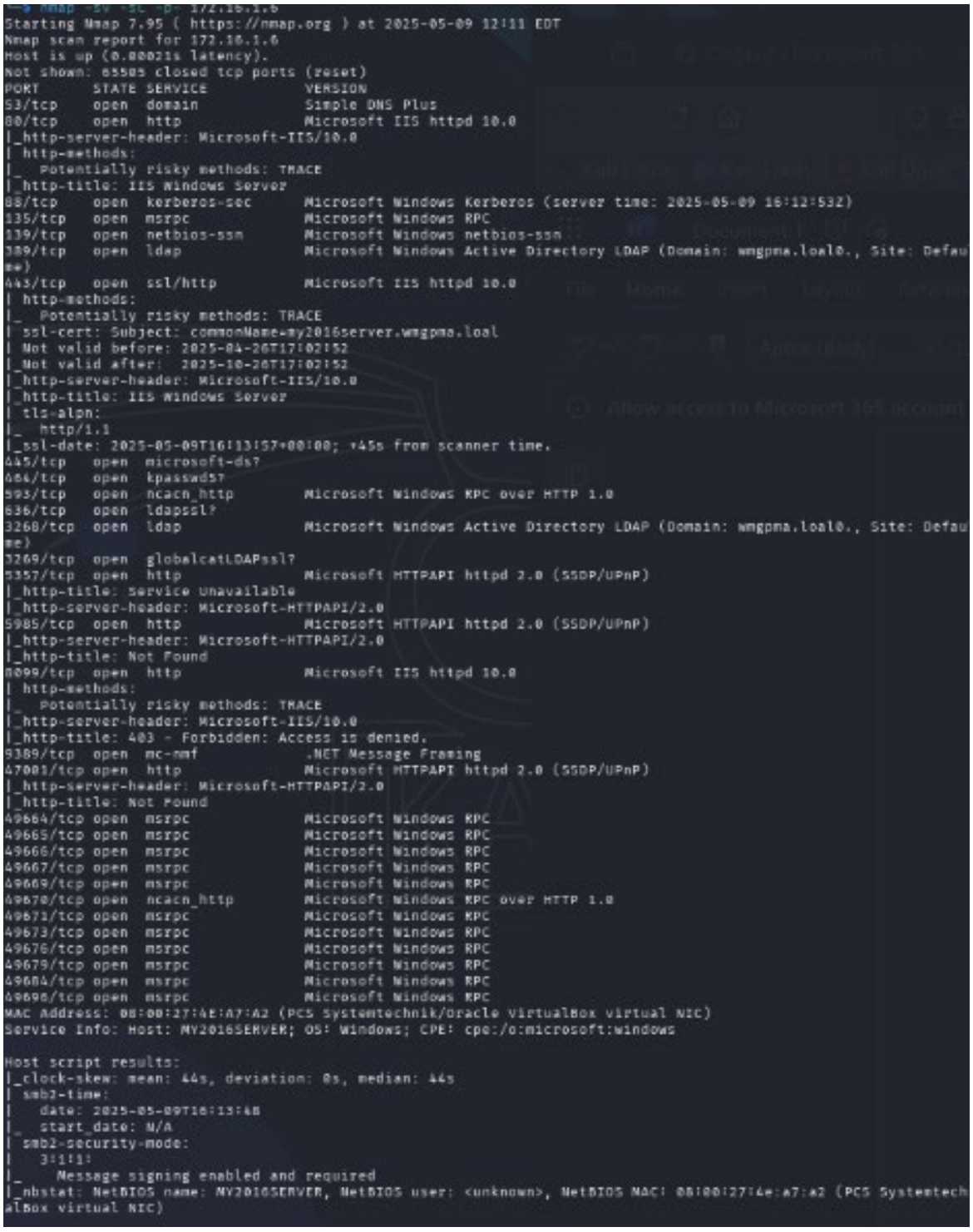
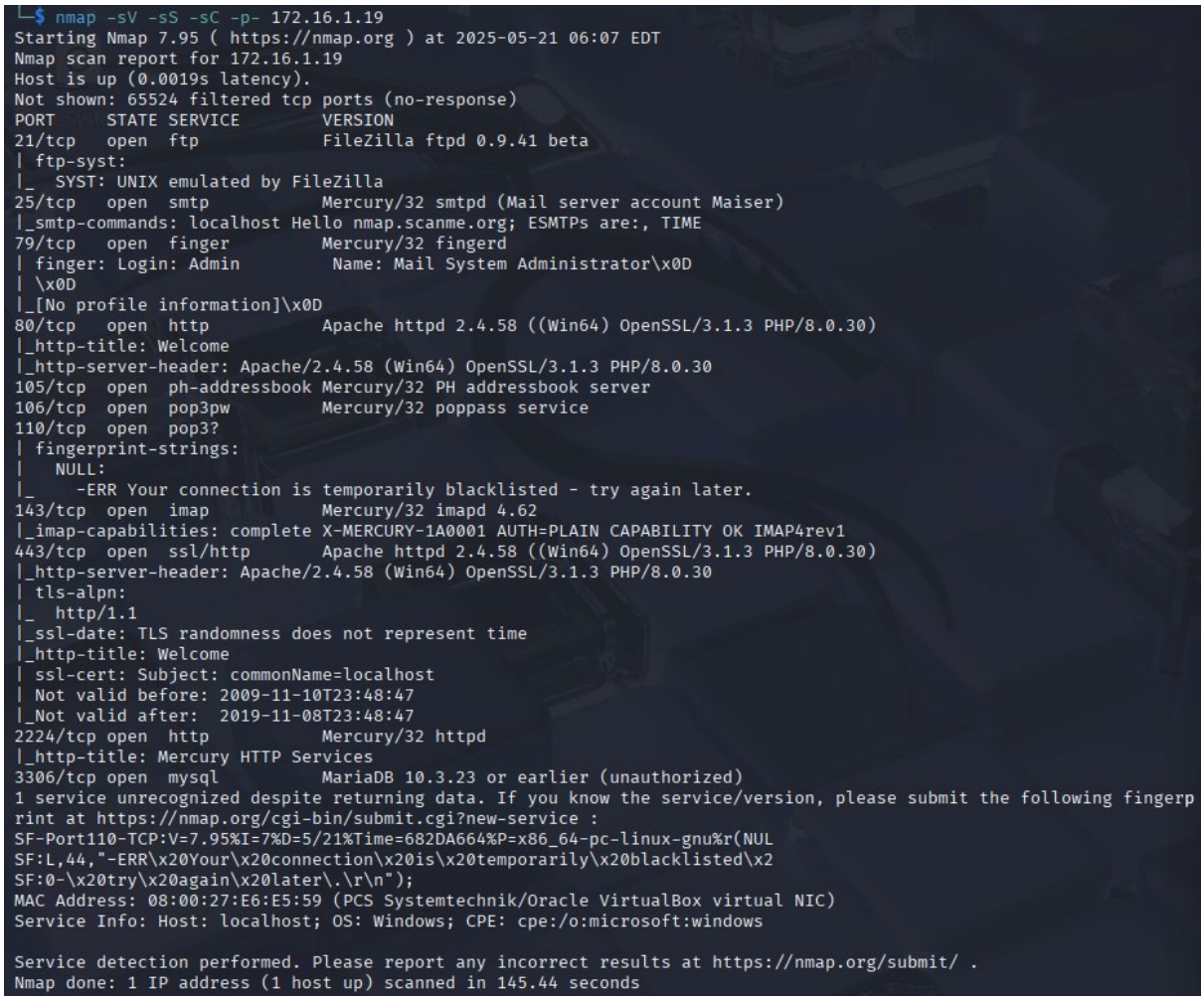


Figure A.11. Full Nmap Scan enumerating services, versions, and scripts running on Dev Server (172.16.1.7).



* Image displays 172.16.1.19 since the VM was reassigned an IP when reinstalled

IX. References

The Penetration Testing Execution Standard (2014) *PTES: Penetration Testing Execution Standard*. Available at: http://www.pentest-standard.org/index.php/Main_Page.

OWASP (2021) OWASP Web Security Testing Guide, version 4.2. Available at: https://owasp.org/www-project-web-security-testing-guide/v42/.

MITRE (2023) *MITRE ATT&CK: Adversarial Tactics, Techniques, and Common Knowledge*. Available at: https://attack.mitre.org/.