vPENTEST

Sample Gray Box IPT

TECHNICAL REPORT

Demo Customer C

March 14, 2025

app.vpentest.io

vPENTEST

Copyright

Confidentiality

This document contains company confidential information of a proprietary and sensitive nature. As such, this document should be afforded the security and handling precautions that a confidential document warrants. This document should have a controlled distribution to relevant parties only and should not be copied without written permission. vPenTest Partner Demo treats the contents of a security audit as company confidential material and will not disclose the contents of this document to anyone without written permission.

Assessment Project Team

Below is a list of contacts that were involved in this engagement. Should you have any questions pertaining to the content of this document or any project and non-project-related items, please feel free to reach out to the necessary project contacts.

Primary Point of Contact	
Name:	Demo Consultant
Title:	Consultant
Office:	(844) 866-2732
Email:	support@vpentest.io

Threat Severity Rankings

To assist the organization with prioritizing findings, the findings and observations have been categorized with threat severity rankings based on the following guidelines:

SEVERITY		DESCRIPTION
	Critical	A critical threat ranking requires immediate remediation or mitigation. Exploiting these vulnerabilities require a minimal amount of effort by the adversary but poses a significant threat to the confidentiality, integrity, and/or availability of the organization's systems and data. A successful compromise of findings of this ranking leads to access to multiple systems and/or several pieces of sensitive information.
	High	A high threat ranking requires immediate remediation or mitigation. Exploiting these vulnerabilities require a minimal amount of effort by the adversary but poses a significant threat to the confidentiality, integrity, or availability of the organization's systems or data. A successful compromise of findings of this ranking leads to access to a single system or limited sensitive information.
	Medium	A medium threat ranking requires remediation or mitigation within a short and reasonable amount of time. These findings typically lead to a compromise of non-privileged user accounts on systems and/or applications or denote a denial-of-service (DoS) condition of the host, service, or application.
	Low	A low threat ranking requires remediation or mitigation once all higher prioritized findings have been remediated. These findings typically leak information to unauthorized or anonymous users and may lead to more significant attacks when combined with other attack vectors.
	Informational	An informational threat ranking does not pose a significant threat to the environment and may just be findings that could potentially disclose valuable information but do not expose the organization to any technical attacks. Findings rated as informational may be useful for an attacker performing information gathering on the organization to leverage in other attacks, such as social engineering or phishing.

Discovered Threats

DISCOVERED THREATS	THREAT SEVERITY RANKINGS	
Internal Network Penetration Test (13)		
IPMI Authentication Bypass		Critical
IPv6 DNS Spoofing		Critical
Link-Local Multicast Name Resolution (LLMNR) Spoofing		Critical
Multicast DNS (mDNS) Spoofing		Critical
NetBIOS Name Service (NBNS) Spoofing		Critical
Applications Accept Default Credentials		High
Weak Active Directory Account Password Policy		High
Anonymous FTP Enabled		Medium
Insecure Protocol - FTP		Medium
Insecure Protocol - Telnet		Medium
SMB NULL Session Authentication		Medium
SMB Signing Not Required		Medium
Egress Filtering Deficiencies		Informational

MITRE ATT&CK Mappings

This section of the report contains details about the tactics, techniques, and procedures as defined by the MITRE ATT&CK Framework. For additional details relating to these tactics, techniques, and procedures (TTPs), vPenTest Partner recommends that Demo Customer C visit the specific URLs provided within the table below. Furthermore, vPenTest Partner has also elaborated on how these TTPs were used during the penetration test in this report's Penetration Test Narrative section.

vPenTest Partner recommends Demo Customer C thoroughly leverage this report section to investigate and improve network security policies, procedures, and controls within the organization's environment. All of the attacks mentioned in this report section should have been detected and properly logged for investigation purposes by the organization.

MITRE | ATT&CK®

Time	Name	Tactic	TTPID
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Active Scanning: Scanning IP Blocks	Reconnaissance	T1595.001
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Network Service Discovery	Discovery	T1046
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Brute Force: Password Spraying	Credential-access	T1110.003
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Exploitation of Remote Services	Lateral-movement	T1210
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	System Owner/User Discovery	Discovery	T1033
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay	Credential-access	T1557.001
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Remote System Discovery	Discovery	T1018
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Gather Victim Host Information: Software	Reconnaissance	T1592.002
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	System Information Discovery	Discovery	T1082
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Network Service Discovery	Discovery	T1046
Tue, Apr 15, 2025 @ 10:23:52 AM EDT	Steal or Forge Authentication Certificates	Credential-access	T1649

Sample Gray Box IPT

Engagement Scope of Work

Through discussions with Demo Customer C's staff, the following target applications, IP addresses, and/or ranges were included as part of the engagement scope.

IP ADDRESSES & RANGES			
192.168.0.1-254			

Agent Information

To perform this assessment, vPenTest Partner used an agent consisting of the necessary tools to conduct discovery, enumeration, attacks, etc. The agent used in this assessment contained the following information:

DESCRIPTION	DETAILS
Agent Name	Dummy Agent for vPenTest-ScheduledTask
Private IP Address	192.168.0.113
Subnet Mask	255.255.255.0 (/24)
DNS Server	127.0.0.53
Default Gateway	192.168.0.1

Task Performed

To assess the targets listed above fully, vPenTest Partner performed the following tasks:

TASK PERFORMED	DEVICES/LOCATIONS ASSESSED
Performed information gathering: NSlookup, and Ping/SNMP sweeping	All targets
Performed port scans	All active targets identified
Performed vulnerability scanning	All active targets identified
Performed web application vulnerability testing	Active/Select targets
Performed vulnerability validation	All active targets identified
Performed penetration testing	Active/Select targets

Rules of Engagement

vPenTest Partner and Demo Customer C agreed to the following rules of engagements:

ACTIVITY	DEFINITION	PERMISSION
Exploitation	vPenTest Partner consultants will cautiously execute exploitation techniques to gain access to sensitive data and/or systems.	Yes
Post Exploitation	If exploitation is successful, vPenTest Partner will attempt to escalate privileges within the environment to gain further access to systems and/or data.	Yes

The following activities were either disabled or reduced as part of the penetration testing engagement to comply with the scope requirements:

ACTIVITY	CONFIGURED SETTING	RECOMMENDED
Password Guessing Limit Against Database Services	1	3
Password Guessing Limit Against Domain Accounts	1	2
Password Guessing Limit Against Other Network Services	1	3

The first step in the enumeration phase was the discovery of systems on the local subnet.

MITRE | ATT&CK®	
Name	Remote System Discovery
Tactic	Discovery
TTP ID	T1018
Note	[obfuscated] performed an arp-scan across the local network subnet to determine which systems are on the local subnet ([obfuscated]). This is also an essential task as these systems would be targets for man-in-the-middle attacks since they are on the same subnet. To facilitate this task, [obfuscated] used a tool known as *arp-scan*.

The following results demonstrate that thirty-one (31) systems exist on the same local subnet:

```
Interface: eth0, type: EN10MB, MAC: 2c:cf:67:35:af:e0, IPv4: [obfuscated]
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
[obfuscated]    94:ff:3c:63:4b:de       Fortinet, Inc.
[obfuscated]    00:ca:e5:88:af:80       Cisco Systems, Inc
[obfuscated]    00:ca:e5:0d:d9:00       Cisco Systems, Inc
[obfuscated]    50:2f:a8:c7:b3:d2       Cisco Systems, Inc
[obfuscated]    50:2f:a8:c7:b6:52       Cisco Systems, Inc
[obfuscated]    d4:76:a0:ca:d1:70       Fortinet, Inc.
[obfuscated]    00:15:5d:00:23:0a       Microsoft Corporation
[obfuscated]    00:0b:94:24:f4:bf       Digital Monitoring Products, Inc.
[obfuscated]    b0:b9:8a:58:81:83       NETGEAR
[obfuscated]    94:c6:91:1c:5c:47       EliteGroup Computer Systems Co., LTD
[obfuscated]    00:71:47:db:b2:e0       Amazon Technologies Inc.
[obfuscated]    54:bf:64:96:1d:bf       Dell Inc.
[obfuscated]    a4:bb:6d:c9:2f:0f       Dell Inc.
[obfuscated]    00:15:5d:00:23:07       Microsoft Corporation
[obfuscated]    80:ee:73:f4:1f:c2       Shuttle Inc.
[obfuscated]    00:15:5d:00:23:06       Microsoft Corporation
[obfuscated]    ac:b4:80:1a:f7:27       (Unknown)
[obfuscated]    9c:dc:71:af:b8:98       Hewlett Packard Enterprise

----- SNIPPED -----

The remainder of this output has been snipped for reporting purposes.
```

While [obfuscated] identified these systems via arp-scan on the local subnet, it was noted that (some of) these systems were not in-scope as part of this penetration test, but could potentially have exploits or vulnerabilities present. As a result, the systems identified above are only shown for informational purposes.

[obfuscated] identified one (1) FTP service within the environment. As FTP is an insecure protocol, it could potentially expose sensitive information such as user credentials or device configuration information in a man-in-the-middle attack. The following scan results display some information that was discovered as a result of these scans:

```
[+] [obfuscated]:21      - FTP Banner: '220 RICOH MP C2004e -- snipped --
```

Testing of FTP services identified one (1) system to accept anonymous FTP authentication credentials. Anonymous login credentials would allow an attacker to identify files that may exist on an FTP server. If permissions allow for write access, an attacker could also attempt to use this to store malicious code. The following output displays the results of this FTP scan:

```
Nmap scan report for [obfuscated]
Host is up, received arp-response (0.00018s latency).
Scanned at 2025-03-14 16:03:04 UTC for 0s
```

```
PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 64
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -r--r--r-- root root 200 Jan  1 01:08 help
| -r--r--r-- root root 200 Jan  1 01:08 info
| -r--r--r-- root root 200 Jan  1 01:08 prnlog
| -r--r--r-- root root 200 Jan  1 01:08 stat
|_-r--r--r-- root root 200 Jan  1 01:08 syslog
MAC Address: 00:26:73:EE:EB:2C (Ricoh Company)
```

While analyzing one of the FTP services at [obfuscated], [obfuscated] was able to enumerate the directory structure. The results of the directory structure listing are below:

```
./
./help
./info
./prnlog
./stat
./syslog
```

MITRE | ATT&CK®

Name	Network Service Discovery
Tactic	Discovery
TTP ID	T1046
Note	[obfuscated] continued testing against these services by attempting to enumerate the files stored on the affected FTP server. To facilitate this process, [obfuscated] leveraged the *lftp* tool, which can significantly expedite the time it takes to enumerate FTP services.

Based on the results of the reviewed FTP services, no sensitive information was identified.

[obfuscated] identified one (1) Telnet service within the environment. As Telnet is an insecure protocol, it could potentially expose sensitive information such as user credentials or device configuration information in a man-in-the-middle attack. The following scan results display some information that was discovered as a result of these scans:

```
[+] [obfuscated]:23      - [obfuscated]:23 TELNET RICOH Ma -- snipped --
```

[obfuscated] identified five (5) SSH services within the environment and attempted to retrieve banner information, which can be used to identify specific server versions. The following scan results display some of the obtained information:

```
[*] [obfuscated] - SSH server version: SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.11
[*] [obfuscated] - SSH server version: SSH-2.0-dropbear_2018.76
[*] [obfuscated] - SSH server version: SSH-2.0-Cisco-1.25
[*] [obfuscated] - SSH server version: SSH-2.0-Cisco-1.25
[*] [obfuscated] - SSH server version: SSH-2.0-mpSSH_0.2.1
```

[obfuscated] scanned SNMP-enabled devices to determine if any weak SNMP community strings were present. SNMP community strings act as passwords for the SNMP protocol and allow network administrators to monitor the performance of SNMP-enabled devices remotely. SNMP-enabled devices often come pre-installed with weak or default SNMP community strings. This weakness could allow a malicious attacker to enumerate information from the remote devices.

During testing, [obfuscated] could not discover any SNMP-enabled systems that used a weak and/or default SNMP community string.

Testing of LDAP services identified that two (2) systems were found to accept anonymous LDAP bind queries, which allows users to query information from within LDAP without proper authentication. This could allow an attacker to gain valuable information about the Active Directory environment, such as domain information and possibly even usernames. The following sample output was obtained while scanning for this weakness:

```
Nmap scan report for [obfuscated]
Host is up, received arp-response (0.00020s latency).
Scanned at 2025-03-14 16:01:42 UTC for 0s

PORT    STATE SERVICE REASON
389/tcp open  ldap    syn-ack ttl 128
| ldap-rootdse:
| LDAP Results
|   <ROOT>
|       domainFunctionality: 6
|       forestFunctionality: 6
|       domainControllerFunctionality: 7
|       rootDomainNamingContext: DC=[obfuscated],DC=net
|       ldapServiceName: [obfuscated]:[obfuscated]$@[obfuscated].NET
|       isGlobalCatalogReady: TRUE
|       supportedSASLMechanisms: GSSAPI
|       supportedSASLMechanisms: GSS-SPNEGO
|       supportedSASLMechanisms: EXTERNAL
|       supportedSASLMechanisms: DIGEST-MD5
|       supportedLDAPVersion: 3

----- SNIPPED -----

The remainder of this output has been snipped for reporting purposes.
```

Next, [obfuscated] identified thirteen (13) systems that exposed port 445/tcp, which is for the Server Message Block (SMB) service. This service was targeted for the enumeration of information that may be valuable. One of the first things scanned during this process is the support for SMB signing. SMB signing, when enabled, helps mitigate SMB relay attacks. SMB relay attacks are when an attacker performs a poisoning attack and tricks a vulnerable system into sending hashed authentication credentials to the attacker. The attacker then takes these hashed credentials and *relay* them to another system, pivoting off that authenticated session to perform additional attacks, such as remote code execution.

Testing identified five (5) of the thirteen (13) systems with port 445/tcp opened that did not require SMB signing, therefore being vulnerable to SMB relay attacks. The following sample output from CrackMapExec identified this weakness:

```
[obfuscated]:(signing:False)
[obfuscated]:(signing:False)
[obfuscated]:(signing:False)
[obfuscated]:(signing:False)
[obfuscated]:(signing:False)
```

MITRE | ATT&CK®

Name	System Information Discovery
Tactic	Discovery
TTP ID	T1082
Note	Additionally, scans were conducted across these systems to identify information about the operating systems, including operating system versions, service pack versions, domain membership, etc.

As part of this operating system identification process, [obfuscated] identified thirteen (13) operating systems. It's important to note that the tools and techniques used to gather information about operating system versions are not always 100% accurate.

While [obfuscated] makes several attempts to confirm the accurate operating systems through additional methods, it should be noted that some results may require additional validation from a system administrator. The following output demonstrates some of the results obtained:

```
SMB                      [obfuscated]     445    [obfuscated]   [*] Windows 10 / Server 2019 Build 17763 x64 (name:[obfusc
ated]) (domain:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]         [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]    [*] Windows 10.0 Build 26100 x64 (name:[obfuscated]) (dom
ain:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]       [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]           [*] Windows 10 / Server 2019 Build 19041 x64 (nam
e:[obfuscated]) (domain:[obfuscated]) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]       [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]        [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]         [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]         [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]     [*] Windows 10.0 Build 26100 x64 (name:[obfuscated]) (do
main:[obfuscated]) (signing:True) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]     [*] Windows 10.0 Build 26100 x64 (name:[obfuscated]) (do
main:[obfuscated]) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]   [*] Windows 10.0 Build 26100 x64 (name:[obfuscated]) (doma
in:[obfuscated]) (SMBv1:False)
SMB                      [obfuscated]     445    [obfuscated]   [*] Windows 10 / Server 2019 Build 17763 x64 (name:[obfusc
ated]) (domain:[obfuscated]) (signing:True) (SMBv1:False)
```

No outdated operating systems were identified during this enumeration process.

MITRE | ATT&CK®

Name	Gather Victim Host Information: Software
Tactic	Reconnaissance
TTP ID	T1592.002
Note	Next, in an attempt to identify some common security vulnerabilities in outdated operating systems, [obfuscated] leveraged the Metasploit Framework to perform specific checks to determine whether or not the targeted system(s) were vulnerable. These vulnerabilities are often labeled as low-hanging fruit as they can easily provide full access to the compromised system if an exploit is successful.

Three (3) systems were scanned using the auxiliary/scanner/smb/smb_ms17_010 module. This module attempts to discover systems that contain a common vulnerability named EternalBlue. When successfully exploited, this vulnerability could allow an attacker with system-level privileges on the system, allowing them to perform several post-exploitation techniques. Such post-exploitation techniques include the enumeration of local administrator password hashes, the enumeration of Active Directory infrastructure data, and more. Scans indicate that no systems were found to be vulnerable at the time of testing. The following results were obtained from this scan:

```
[-] [obfuscated]:445       - An SMB Login Error occurred while connecting to the IPC$ tree.
[-] [obfuscated]:445       - An SMB Login Error occurred while connecting to the IPC$ tree.
[-] [obfuscated]:445     - An SMB Login Error occurred while connecting to the IPC$ tree.
```

[obfuscated] then ran a custom script to check if any systems allowed for SMB NULL session authentication (i.e. without a username or password). SMB NULL sessions can allow attackers with network access to identify and possibly retrieve files that

may exist on an SMB (445/tcp) server. If permissions allow for write access, an attacker could also attempt to use this to store malicious code. The results showed that two (2) systems accepted SMB NULL session authentication:

```
[obfuscated]
[obfuscated]
```

The below sample evidence shows some of the results of this attack:

```
[[obfuscated]]
# crackmapexec smb [obfuscated] -u '' -p '' --local-auth
SMB                      [obfuscated]     445    [obfuscated]   [+] [obfuscated]\:
------------------------------------------------------------
[[obfuscated]]
# crackmapexec smb [obfuscated] -u '' -p '' --local-auth
SMB                      [obfuscated]     445    [obfuscated]   [+] [obfuscated]\:
------------------------------------------------------------
```

[obfuscated] then tried to take advantage of SMB NULL session authentication in order to enumerate the SMB shares available on the affected systems. The aim of this process was to identify any accessible shares containing potentially sensitive company data as well as shares configured with WRITE access. However, no accessible shares were identified.

Additionally, an enumeration of SMB services was performed in an attempt to identify whether usernames, password policies, or additional computer and/or domain information could be obtained. Such information could be useful for performing a password attack against the environment. A sample output of one of the results is as follows:

```
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Mar 14 16:06:03 2025

 ==========================================( Target Information )==========================================

Target ........... [obfuscated]
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ============================( Enumerating Workgroup/Domain on [obfuscated] )============================


[+] Got domain/workgroup name: [obfuscated]


 ================================( Nbtstat Information for [obfuscated] )================================

Looking up status of [obfuscated]

----- SNIPPED -----

The remainder of this output has been snipped for reporting purposes.
```

No valuable information, such as domain/local user accounts and password policies, was obtained as part of this enumeration process.

MITRE | ATT&CK®

Name	Exploitation of Remote Services
Tactic	Lateral-movement
TTP ID	T1210

Note	[obfuscated] tested 2 domain controllers for the critical vulnerability known as ZeroLogon. When exploited, ZeroLogon allows an attacker to reset the password of the domain controller's machine account. This can lead to full domain compromise. The following domain controllers were tested as part of this process: [obfuscated] ([obfuscated]) [obfuscated] ([obfuscated])

During testing, [obfuscated] was unable to identify any domain controllers that were vulnerable to ZeroLogon.

Next, [obfuscated] enumerated web services in the environment with the aim of obtaining sensitive information by exploiting default credentials, security vulnerabilities or misconfigurations. During this process, [obfuscated] identified one (1) web instance that was configured with default credentials. This could allow attackers to change configuration settings, obtain information pertaining to the network, and more. The affected service was:

```
http://[obfuscated] - Web Image Monitor
```

Next, [obfuscated]'s objective was to perform a password attack against the Active Directory environment. However, [obfuscated] needed to gather a list of potential domain user accounts to perform this process. [obfuscated] used the Kerbrute tool to assist with this process. Kerbrute is a tool that can be used to enumerate domain user accounts by interacting with Kerberos. Based on the response from a ticket-granting ticket (TGT) request to the key distribution center (KDC) server, Kerbrute is able to deduce whether or not the domain user account provided was valid or not.

The following domain was observed as part of the initial host discovery scans performed at the beginning of the assessment:

[obfuscated]

[obfuscated] used naming schemes for four different naming conventions: 1) first initial last name, 2) first name last initial, 3) first name dot last initial (e.g. First.Last), and 4) first name. A combination of common first and last names was used as part of this process, as well as publicly available resources.

MITRE | ATT&CK®

Name	System Owner/User Discovery
Tactic	Discovery
TTP ID	T1033
Note	[obfuscated] targeted the following domain controller as part of this Kerberos user enumeration attack: [obfuscated] ([obfuscated])

During this process, [obfuscated] discovered six (6) valid domain user accounts for one (1) domain. The following usernames were observed:

```
[obfuscated]
----

[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
```

During the enumeration phase of the test, [obfuscated] identified a total of six (6) domain users. [obfuscated] carried out a limited password attack, consisting of one (1) login attempt per account, against the identified users.

During this password attack, no successful login attempts were identified.

MITRE | ATT&CK®	
Name	Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay
Tactic	Credential-access
TTP ID	T1557.001
Note	As part of the exploitation phase, [obfuscated] continued to perform DNS poisoning attacks via NBNS, LLMNR and mDNS.

When enabled on Microsoft Windows systems, DNS names that cannot be resolved by a system's configured DNS server or local hosts file will be communicated in the form of NBNS and/or LLMNR broadcast packets across the network environment. Similarly, multicast DNS (mDNS) can be used within small networks to resolve a DNS name when no local DNS server exists. This is done via IP multicast query messages to the hosts on the local subnet. The problem with this configuration is that it is possible to respond to these broadcast/multicast packets and spoof the IP address of the DNS name in question. In other words, if SystemA is attempting to resolve www.helloworld.com and cannot find its IP address, an attacking system can pretend to be the IP address of www.helloworld.com. Upon a successful attack, it may be possible to capture cleartext or hashed credentials.

[obfuscated] deployed a rogue IPv6 router within the environment to determine if it'd be possible to conduct IPv6 attacks. Since IPv6 is treated with higher priority than IPv4, any time a network device sees an IPv6 router available, it will attempt to retrieve an IPv6 address. An attacker can abuse this by deploying a rogue DHCPv6 server within the environment and assigning all IPv6 clients with an IP address and DNS configurations that route traffic through the attacker's system.

While [obfuscated] was successful with capturing NBNS/LLMNR/mDNS broadcast packets across the local subnet, it was not possible to capture any credentials at the time of testing. This is primarily due to the lack of systems and/or services successfully authenticating to the penetration testing VM during these attacks. An example of these successful NBNS/LLMNR/mDNS poisoning attempts is shown below:

```
2025-03-14 16:25:02,305 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name _matterc._udp
2025-03-14 16:25:02,306 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:03,304 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name _matterc._udp
2025-03-14 16:25:03,305 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:05,307 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name _matterc._udp
2025-03-14 16:25:05,308 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:09,308 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name _matterc._udp
2025-03-14 16:25:09,308 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:17,316 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name _matterc._udp
2025-03-14 16:25:17,317 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:34,443 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File Server)
2025-03-14 16:25:34,443 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name [obfuscated].local
2025-03-14 16:25:34,444 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,444 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name [obfuscated].local
2025-03-14 16:25:34,445 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:34,445 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,447 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:34,447 - [*] [MDNS] Poisoned answer sent to [obfuscated]   for name [obfuscated].local
2025-03-14 16:25:34,447 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:34,448 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]

----- SNIPPED -----

The remainder of this output has been snipped for reporting purposes.
```

When attempting to perform IPv6 attacks, [obfuscated] successfully assigned IPv6 addresses with the attacking system set as the default DNS server. An example of this can be found below:

```
Starting mitm6 using the following configuration:
Primary adapter: eth0 [2c:cf:67:35:af:e0]
IPv4 address: [obfuscated]
IPv6 address: [obfuscated]
Warning: Not filtering on any domain, mitm6 will reply to all DNS queries.
Unless this is what you want, specify at least one domain with -d
IPv6 address [obfuscated] is now assigned to mac=b4:45:06:11:8b:46 host=[obfuscated]. ipv4=
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:07 host=connectwisesensor. ipv4=
Sent spoofed reply for [obfuscated]. to [obfuscated]
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:08 host=connectwisesensor. ipv4=
Sent spoofed reply for [obfuscated]. to [obfuscated]
Sent spoofed reply for [obfuscated]. to [obfuscated]
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:07 host=connectwisesensor. ipv4=
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:08 host=connectwisesensor. ipv4=
Sent spoofed reply for [obfuscated]. to [obfuscated]
Sent spoofed reply for [obfuscated]. to [obfuscated]
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:08 host=connectwisesensor. ipv4=
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:07 host=connectwisesensor. ipv4=
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:08 host=connectwisesensor. ipv4=
IPv6 address [obfuscated] is now assigned to mac=00:15:5d:00:23:07 host=connectwisesensor. ipv4=

----- SNIPPED -----

The remainder of this output has been snipped for reporting purposes.
```

At the time of testing, [obfuscated] was unsuccessful in capturing any valuable password hashes via NTLM relaying attacks. This is primarily due to the lack of systems and/or services successfully authenticating to the penetration testing VM during these attacks.

[obfuscated] used the previously obtained password for [obfuscated]\[obfuscated] to try and authenticate to all discovered hosts that had the SMB service (port 445/tcp) available. The aim of this attack was to check if these credentials provided local Administrator privileges on any hosts. [obfuscated] leveraged these credentials to successfully authenticate to seven (7) hosts, which provided [obfuscated] with local admin rights on the compromised systems:

```
------------------------------------------------------------------------------
[COMPROMISED HOSTS]
- IP: [obfuscated]       Privileges obtained: local administrator
- IP: [obfuscated]       Privileges obtained: local administrator
- IP: [obfuscated]       Privileges obtained: local administrator
- IP: [obfuscated]       Privileges obtained: local administrator
- IP: [obfuscated]       Privileges obtained: local administrator
- IP: [obfuscated]         Privileges obtained: local administrator
- IP: [obfuscated]         Privileges obtained: local administrator
[SAMPLE EVIDENCE]
------------------------------------------------------------------------------
SMB                     [obfuscated]   445    [obfuscated]       [+] [obfuscated]\[obfuscated]:[obfuscated] (Pwn3d!)
SMB                     [obfuscated]   445    [obfuscated]         [+] [obfuscated]\[obfuscated]:[obfuscated] (Pwn3d!)
SMB                     [obfuscated]   445    [obfuscated]    [+] [obfuscated]\[obfuscated]:[obfuscated] (Pwn3d!)
------------------------------------------------------------------------------
```

MITRE | ATT&CK®

Name	Steal or Forge Authentication Certificates
Tactic	Credential-access
TTP ID	T1649

Note	[obfuscated] attempted to leverage the client-provided cleartext credentials to find vulnerable Active Directory (AD) Certificate Services (CS) templates for exploitation. This vulnerability, when successfully exploited, could allow an attacker to escalate privileges. During this process, the following domain was targeted, along with the username used in parentheses: 1. [obfuscated] ([obfuscated])

During this enumeration process, no domain controllers were found to have vulnerable AD CS templates or the credentials used did not have permission to extract sufficient information.

Next, [obfuscated] used the CME tool together with the [obfuscated]\[obfuscated] credentials in order to obtain the password policy for the [obfuscated] domain from the domain controller at [obfuscated]:

```
Minimum password length:  14
Password history length:  10
Maximum password age:  89 days 23 hours 54 minutes

Password Complexity Flags:  001001
    Domain Refuse Password Change:  0
    Domain Password Store Cleartext:  0
    Domain Password Lockout Admins:  1
    Domain Password No Clear Change:  0
    Domain Password No Anon Change:  0
    Domain Password Complex:  1

Minimum password age:  1 day 4 minutes
Reset Account Lockout Counter:  10 minutes
Locked Account Duration:  10 minutes
Account Lockout Threshold:  5
Forced Log off Time:  Not Set
```

The results showed that the [obfuscated] domain was configured with the following weak password policy setting(s):

```
Reset Account Lockout Counter: 10
Locked Account Duration: 10
```

In order to obtain more information on the [obfuscated] domain, [obfuscated] used the password for the [obfuscated]\[obfuscated] account to enumerate LDAP using the Bloodhound-Python tool. A total of thirty-two (32) computers, twenty-seven (27) users and fifty-eight (58) groups were identified on the domain in this manner. The results also showed that the domain was configured with seven (7) domain admin accounts. The below sample evidence shows some of the active directory information that was gathered:

```
[computers]
--snipped--
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
--snipped--

[domain admins]
--snipped--
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
--snipped--

[groups]
```

```
--snipped--
enterprise key admins
key admins
storage replica administrators
[obfuscated] $ cyber operators
[obfuscated] $ cyber operators
--snipped--

[users]
--snipped--
NT AUTHORITY
[obfuscated]
[obfuscated]
[obfuscated]
[obfuscated]
--snipped--
```

The active directory information that had previously been obtained via LDAP enumeration, showed that the [obfuscated]\[obfuscated] account was part of the domain admins group. This meant that [obfuscated] had now fully compromised the [obfuscated] domain.

Next, [obfuscated] attempted to perform an attack known as Kerberoasting. This attack takes advantage of the Kerberos protocol and can be performed for any valid domain user account, regardless of privileges. When an active directory user logs in, they receive a Ticket Granting Ticket (TGT) from the Kerberos key distribution center. If the authenticated user then requests a specific resource in the domain, their TGT is used to request a Ticket Granting Service (TGS) token for that resource. Part of this TGS is encrypted with the NTLM hash of the service account for the requested resource. If an attacker obtains a TGS, they can try and crack it and obtain the user's password via brute-force methods or lists of common passwords. Obtaining a TGS requires knowledge of the existing service principal names (SPNs) that Windows uses to identify which service accounts are being used to encrypt TGS tokens.

[obfuscated] used the credentials for the [obfuscated]\[obfuscated] account together with the impacket-GetUserSPNs tool in an attempt to obtain the SPNs configured on the domain and use those to obtain TGS tokens.

However, no SPNs were obtained.

Next, [obfuscated] used the obtained domain admin credentials in order to enumerate the SMB shares available on the compromised system(s). The aim of this process was to identify any accessible shares containing potentially sensitive company data. At the time of testing, sensitive data was discovered, including financial information, credentials, tax information, email information. The below evidence shows some of the sensitive information that was observed:

```
[credentials]
\\[obfuscated]\c$\users\[obfuscated]\AppData\Roaming\NetworkDetective\Reports\[obfuscated]\Assessment-20250220-Reports3\Se
curity\Password Policies Summary.docx      A   403816  Mon Feb 24 14:59:16 2025
\\[obfuscated]\c$\users\[obfuscated]\AppData\Roaming\NetworkDetective\Reports\[obfuscated]\Assessment-20231206-Reports1\Se
curity\Password Policies Summary.docx      A   397341  Wed Dec  6 15:11:26 2023
\\[obfuscated]\c$\users\[obfuscated]\AppData\Roaming\NetworkDetective\Reports\[obfuscated]\Assessment-20240604-Reports\Sec
urity\Password Policies Summary.docx      A   414107  Wed Jun  5 19:12:57 2024
\\[obfuscated]\c$\users\[obfuscated]\AppData\Roaming\NetworkDetective\Reports\[obfuscated]\Assessment-20240815-Reports1\Se
curity\Password Policies Summary.docx      A   414324  Tue Aug 20 13:28:19 2024
\\[obfuscated]\c$\users\[obfuscated]\AppData\Roaming\NetworkDetective\Reports\[obfuscated]\Assessment-20240206-Reports\Sec
urity\Password Policies Summary.docx      A   397338  Tue Feb  6 20:33:17 2024
--snipped--

[email information]
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Customer-Email-Groups.xlsx       Asro    11672  Mon Jan 27 1
5:56:20 2025
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Downloads from Old Laptop\Email (1).pdf                    As
ro   152378  Mon Oct  7 09:12:10 2024
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Downloads from Old Laptop\Email (2).pdf                    As
ro   151667  Mon Oct  7 09:13:41 2024
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Downloads from Old Laptop\Email (3).pdf                    As
```

```
ro   152378  Mon Oct  7 13:00:31 2024
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Downloads from Old Laptop\Email.pdf                      As
ro   142746  Mon Oct  7 09:06:16 2024
--snipped--

[financial information]
\\[obfuscated]\c$\invoice-[obfuscated].docx      AH   111303  Mon Jan  6 21:06:02 2025
\\[obfuscated]\c$\invoice-[obfuscated].docx      AH   111303  Mon Jan  6 21:34:27 2025
\\[obfuscated]\c$\invoice-[obfuscated].docx      AH   111303  Mon Jan  6 18:50:56 2025
\\[obfuscated]\c$\users\[obfuscated]\Downloads\Invoice #[obfuscated].pdf               A   760740  Mon Jan  6 18:51:35 20
25
\\[obfuscated]\c$\invoice-[obfuscated].docx      AH   111303  Mon Jan  6 18:31:51 2025
--snipped--

[tax information]
\\[obfuscated]\c$\users\[obfuscated]\Downloads\[obfuscated] Tax Exempt Form - Signed 2024.pdf      A    56631  Mon Jan 20
12:44:50 2025
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Old-Laptop Migration Files\Downloads\Tax Statement_2025-01-1
9..pdf      Ar    76077  Thu Jan 23 15:29:58 2025
\\[obfuscated]\c$\users\[obfuscated]\Downloads\[obfuscated] Tax Exemption Cert [obfuscated].pdf      A  1873788  Mon Feb
3 14:17:46 2025
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\Latitude 5520 Downloads\TaxForms.pdf                    Asro
218388  Thu Jan 18 21:18:26 2024
\\[obfuscated]\c$\users\[obfuscated]\OneDrive - [obfuscated]\TEAMS\Staff Folders\[obfuscated]\Downloads from old computer
\- [obfuscated] Local Taxes.pdf          Asro    54083  Wed Nov 17 15:23:49 2021
--snipped--
```

Internal Network Environment Exposures

This phase of the security assessment focused on the security of network assets within the internal network environment. During this phase, vPenTest Partner used a comprehensive set of tools, custom scripts, and manual techniques to thoroughly identify possible threats to the environment. Like a traditional penetration test, all identified threats were tested and validated to evaluate the depth of compromise. Unlike a traditional penetration test, this evaluation of threats was not isolated or limited to a handful of threats, but rather across all threats identified.

CRITICAL — IPMI Authentication Bypass

Observation

The Intelligent Platform Management Interface (IPMI) is a critical hardware solution utilized by network administrators for centralized management of server(s). During the configuration of server(s) equipped with IPMI, certain vulnerabilities may exist that allow attackers to bypass the authentication mechanism remotely. This results in the extraction of password hashes, and in instances where default or weak hashing algorithms are employed, attackers could potentially recover the cleartext passwords.

Security Impact

The ability to extract cleartext passwords presents a significant security risk, as an attacker could leverage this information to gain unauthorized remote access to sensitive services, including Secure Shell (SSH), Telnet, or web-based interfaces. Such unauthorized access could enable configurations manipulation, negatively impacting the availability and integrity of services provided by the compromised server(s).

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.133	[obfuscated]	Undetected

Recommendation

Given the absence of a patch for this vulnerability, it is essential to implement one or more of the following mitigation strategies:

Limit IPMI access strictly to authorized system(s) that require administrative functionalities.
Disable IPMI service on server(s) that do not need it for business operations.
Change default administrator password(s) to strong, complex alternatives to enhance security.
Employ secure communication protocols, such as HTTPS and SSH, to mitigate the risk of man-in-the-middle attacks that could expose sensitive credentials.

Reproduction Steps

Leveraging the Metasploit framework, configure and run the following module against the affected service:

```
auxiliary/scanner/ipmi/ipmi_dumphashes
```

References

https://www.zenlayer.com/blog/what-is-ipmi/
https://www.tenable.com/plugins/nessus/68931
https://beyondsecurity.com/scan-pentest-network-multiple-vendor-ipmi-cipher-zero-authentication-bypass-vulnerability.html?cn-reloaded=1

Evidence

```
[+] [obfuscated]:623 - IPMI - Hash found: [+] [obfuscated]:623 - IPMI - Hash found: Administrator:4a9d5ea8ee7c7c39
74ab3d95f83e7d41[partially-obfuscated]61746f72:920d603a8dfe658bd71e7a00a6cde8a517ae6e51
```

```
[+] [obfuscated]:623 - IPMI - Hash found: Administrator:[obfuscated]:920d603a8dfe658bd71e7a00a6cde8a517ae6e51
```

CRITICAL

IPv6 DNS Spoofing

Observation

The risk of IPv6 DNS spoofing arises from the possible introduction of a rogue DHCPv6 server within the internal network infrastructure. Due to the preference of Microsoft Windows systems for IPv6 over IPv4, IPv6-capable clients are inclined to obtain their IP address configurations from any available DHCPv6 server.

Security Impact

The deployment of a rogue DHCPv6 server allows an attacker to manipulate DNS requests by redirecting IPv6-enabled clients to utilize the attacker's system as their DNS server. This capability can lead to serious consequences, such as the unauthorized capture of sensitive data, including user credentials. When all DNS queries resolve to the attacker's server, the victim's system may inadvertently communicate with malicious services operating on the attacker's infrastructure, encompassing platforms such as SMB, HTTP, RDP, and MSSQL.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.114	[obfuscated]	Undetected

Recommendation

To mitigate the risks associated with IPv6 DNS spoofing, the following strategies are recommended, with emphasis on aligning each approach with organizational operations and thorough testing prior to implementation:

Manage Rogue DHCP at the Network Layer: Implement features such as Rogue DHCP detection, DHCP snooping, and DHCP authentication on network switches and firewalls to control unauthorized DHCP servers and lessen the likelihood of DNS spoofing attacks.
Prefer IPv4 over IPv6: Utilize Group Policy Objects (GPOs) or Group Policy Preferences (GPPs) to deploy registry modifications that configure Windows systems to favor IPv4 over IPv6. It is important to note that this approach will not prevent attacks from affecting non-Windows devices.
Disable IPv6: While not generally advisable for Microsoft Windows systems, disabling IPv6 may be considered as a last resort precaution, provided thorough testing ensures there are no significant disruptions to business operations.

Reproduction Steps

Leveraging the "mitm6" tool within Kali Linux, a user is able to quickly deploy a DHCPv6 server within the local network and assign five-minute leases (by default) to IPv6-enabled clients.

References

https://blog.vonahi.io/taking-over-ipv6-networks/
https://learn.microsoft.com/en-us/troubleshoot/windows-server/networking/configure-ipv6-in-windows

Evidence

```
IPv6 address fe80::147:2 is now assigned to mac=b4:45:06:11:8b:46 host=[obfuscated].[obfuscated].net. ipv4=
IPv6 address fe80::147:3 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=
Sent spoofed reply for wpad.[obfuscated].net. to fe80::3f65:5fbd:7643:4c39
IPv6 address fe80::147:4 is now assigned to mac=00:15:5d:00:23:08 host=[obfuscated]. ipv4=
Sent spoofed reply for login.microsoftonline.com. to fe80::3f65:5fbd:7643:4c39
Sent spoofed reply for settings-win.data.microsoft.com. to fe80::3f65:5fbd:7643:4c39
IPv6 address fe80::147:6 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=
IPv6 address fe80::147:7 is now assigned to mac=00:15:5d:00:23:08 host=[obfuscated]. ipv4=
Sent spoofed reply for core.threatlocker.com. to fe80::3f65:5fbd:7643:4c39
Sent spoofed reply for api.f.threatlocker.com. to fe80::3f65:5fbd:7643:4c39
IPv6 address fe80::147:9 is now assigned to mac=00:15:5d:00:23:08 host=[obfuscated]. ipv4=
IPv6 address fe80::147:8 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=
IPv6 address fe80::147:12 is now assigned to mac=00:15:5d:00:23:08 host=[obfuscated]. ipv4=
IPv6 address fe80::147:11 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=
IPv6 address fe80::147:14 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=
IPv6 address fe80::147:15 is now assigned to mac=00:15:5d:00:23:08 host=[obfuscated]. ipv4=
IPv6 address fe80::147:16 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=
IPv6 address fe80::147:17 is now assigned to mac=00:15:5d:00:23:08 host=[obfuscated]. ipv4=
Renew reply sent to fe80::147:2
Sent spoofed reply for v10.events.data.microsoft.com. to fe80::3f65:5fbd:7643:4c39
IPv6 address fe80::9774:1 is now assigned to mac=00:15:5d:00:23:07 host=[obfuscated]. ipv4=

--snipped--
```

CRITICAL Link-Local Multicast Name Resolution (LLMNR) Spoofing

Observation

Link-Local Multicast Name Resolution (LLMNR) is a protocol designed for name resolution within internal network environments when traditional Domain Name System (DNS) services are either unavailable or ineffective. LLMNR acts as a fallback mechanism, facilitating the resolution of DNS names through multicast queries. The resolution process unfolds as follows:

1. The system first queries its local host file to find a corresponding IP address for the specified DNS name.
2. If no local entry exists, the system initiates a DNS query directed at its configured DNS server(s) to resolve the name.
3. Should the DNS server(s) fail to provide a resolution, the system broadcasts an LLMNR query across the local network, seeking responses from other hosts.

This reliance on multicast broadcasts introduces vulnerabilities, as any active system can respond to the queries, potentially misleading the requesting system.

Security Impact

The broadcasting nature of LLMNR queries allows any system on the local network to respond with its own IP address in answer to a resolution request. Malicious actors can exploit this by sending crafted responses containing the attacker's system's address. This capability opens avenues for significant security breaches, particularly if the query is tied to sensitive services such as SMB, MSSQL, or HTTP. Successful redirection can facilitate the capture of sensitive information including plaintext and hashed account credentials. It is pertinent to note that hashed credentials can be subjected to modern brute-force attacks, thereby compromising account security.

Affected Nodes

THREE (3) NODES AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.108	[obfuscated]	Undetected
192.168.0.118	[obfuscated]	Undetected
192.168.0.121	[obfuscated]	Undetected

Recommendation

To mitigate the risks associated with LLMNR spoofing, it is critical to disable LLMNR functionality across affected systems. This can be accomplished through the following methods:

Group Policy Configuration: Navigate to Computer Configuration\Administrative Templates\Network\DNS Client and set 'Turn off Multicast Name Resolution' to Enabled. For administering configurations on a Windows Server 2003 domain controller, utilize the Remote Server Administration Tools for Windows 7 available at this link.
Registry Modification for Windows Vista/7/10 Home Edition: Access the registry at HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\DNSClient and modify the 'EnableMulticast' key to 0 or remove it to disable the feature.

Reproduction Steps

On a system configured with LLMNR, attempt to interact with a DNS name that is known to be invalid (e.g. test123.local). On another system, use a network packet analyzer, such as Wireshark, to inspect the broadcasted traffic on the internal network environment.

References

https://attack.mitre.org/techniques/T1557/001/

Evidence

```
2025-03-14 16:25:34,445 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:34,447 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:34,447 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:34,448 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:40,138 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:40,140 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:40,152 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:40,152 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:50,539 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:50,540 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:50,541 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:25:50,543 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:00,641 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:00,642 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:00,643 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:00,644 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:01,932 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:01,933 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:01,934 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:01,934 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]
2025-03-14 16:38:20,127 - [*] [LLMNR]  Poisoned answer sent to [obfuscated] for name [obfuscated]

--snipped--
```

CRITICAL

Multicast DNS (mDNS) Spoofing

Observation

Multicast DNS (mDNS) serves as a name resolution protocol for local networks, facilitating the resolution of domain names when a dedicated DNS server is unavailable. The resolution process occurs in stages:

1. The system first consults its local host file for any appropriate DNS name/IP address mappings.
2. In the absence of a configured DNS server, the system resorts to mDNS, broadcasting an IP multicast query requesting identification from the host corresponding to the DNS name. This protocol behavior exposes a potential vulnerability that malicious actors can exploit, enabling them to impersonate legitimate systems by responding to these queries.

Security Impact

mDNS queries, which are transmitted across the local subnet, can be answered by any device capable of receiving them. This vulnerability allows an attacker to respond with their system's IP address, potentially misleading the querying system. Such exploitation may lead to interception of sensitive information, including unencrypted and hashed credentials, depending on the specific service the victim is trying to access (e.g., SMB, MSSQL, HTTP). It should be noted that hashed credentials can often be compromised within a relatively short timeframe using contemporary computing resources and brute-force attack methodologies.

Affected Nodes

FOUR (4) NODES AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.101	[obfuscated]	Undetected
192.168.0.108	[obfuscated]	Undetected
192.168.0.118	[obfuscated]	Undetected
192.168.0.121	[obfuscated]	Undetected

Recommendation

To mitigate the risk of mDNS spoofing, the primary recommendation is to completely disable mDNS if it is not in use. On Windows systems, this can often be done by implementing the 'Disable Multicast Name Resolution' group policy. As many applications have the potential to reintroduce mDNS functionality, an alternative strategy is to block UDP port 5353 via the Windows firewall. For non-Windows systems, disabling services such as Apple Bonjour or avahi-daemon can provide similar protection.

It is important to note that disabling mDNS may disrupt functionalities such as screen casting and certain conference room technologies. Should complete disabling not be feasible, consider isolating affected systems within a controlled network segment and mandating the use of strong, complex passwords for any accounts that access these systems.

Reproduction Steps

On a system configured with mDNS, attempt to interact with a DNS name that is known to be invalid (e.g. test123.local). On another system, use a network packet analyzer, such as Wireshark, to inspect the mDNS traffic on the internal network environment by filtering for UDP queries over port 5353.

References

http://www.multicastdns.org/

Evidence

```
2025-03-14 16:25:02,305 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:02,306 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:03,304 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:03,305 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:05,307 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:05,308 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:09,308 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:09,308 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:17,316 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:17,317 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name _matterc._udp
2025-03-14 16:25:34,443 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,444 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,444 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,445 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,447 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,450 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,452 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:34,453 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:40,137 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:40,139 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local
2025-03-14 16:25:40,150 - [*] [MDNS] Poisoned answer sent to [obfuscated] for name [obfuscated].local

--snipped--
```

CRITICAL NetBIOS Name Service (NBNS) Spoofing

Observation

The NetBIOS Name Service (NBNS) is a protocol utilized by workstations within an internal network to resolve domain names when a DNS server is unavailable or unresponsive. When a system attempts to resolve a DNS name, it follows these steps:

1. The system first checks its local host file for an entry mapping the DNS name to an IP address.
2. If no local mapping exists, the system sends a DNS query to its configured DNS server(s) in an attempt to retrieve the corresponding IP address.
3. If the DNS server(s) cannot resolve the name, the system broadcasts an NBNS query across the local network, soliciting responses from other systems.

This dependency on broadcasts makes the NBNS vulnerable to spoofing attacks, wherein an attacker can respond with a false IP address.

Security Impact

The broadcasting nature of NBNS queries means that any system on the local network can respond. This vulnerability can be exploited by malicious actors who may answer these queries with the IP address of the attacker's system, redirecting traffic intended for legitimate services. For instance, services such as SMB, MSSQL, or HTTP could inadvertently send sensitive data, including cleartext or hashed account credentials, to the attacker's system. Moreover, modern computational capabilities can facilitate the cracking of hashed credentials, potentially allowing unauthorized access to user accounts.

Affected Nodes

THREE (3) NODES AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.108	[obfuscated]	Undetected
192.168.0.118	[obfuscated]	Undetected
192.168.0.121	[obfuscated]	Undetected

Recommendation

To mitigate the risk of NBNS spoofing, it is advisable to disable the NetBIOS service across all hosts within the internal network. This can be accomplished through a variety of methods including configuration of DHCP options, adjustments to network adapter settings, or modifications to the system registry. Implementing these changes will significantly reduce the potential attack surface associated with NBNS.

Reproduction Steps

On a system configured with NBNS, attempt to interact with a DNS name that is known to be invalid (e.g. test123.local). On another system, use a network packet analyzer, such as Wireshark, to inspect the broadcasted traffic on the internal network environment.

References

http://markgamache.blogspot.com/2013/01/ntlm-challenge-response-is-100-broken.html
http://support.microsoft.com/kb/313314
http://develnet.blogspot.com/2006/10/disabling-netbios-over-tcpip-via.html
http://technet.microsoft.com/en-us/library/cc775874(v=ws.10).aspx

Evidence

```
2025-03-14 16:25:34,443 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:25:40,137 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:25:50,537 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:38:29,669 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:39:31,660 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:39:46,702 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: Workst
ation/Redirector)
2025-03-14 16:50:36,154 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:50:47,524 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:50:59,730 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:19:31,614 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:19:48,886 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:19:50,359 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:44:45,899 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:44:50,353 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:45:34,455 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:31:42,274 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:32:00,764 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:32:23,030 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:33:01,634 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:33:01,963 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)
2025-03-14 16:33:20,128 - [*] [NBT-NS] Poisoned answer sent to [obfuscated] for name [obfuscated] (service: File S
erver)

--snipped--
```

HIGH

Applications Accept Default Credentials

Observation

The penetration test identified that several web applications were configured with default or common credentials. Through a controlled password attack, it was confirmed that these applications were susceptible to exploitation due to weak authentication mechanisms. The testing was conducted with precautions to prevent any disruption to legitimate user access or service operations.

Security Impact

The presence of default credentials within applications poses a significant security risk, as it may allow unauthorized users to gain access to sensitive functionalities. If exploited, attackers could manipulate critical network resources, leading to potential data breaches or system compromises that undermine organizational security.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.132	[obfuscated]	Undetected

Recommendation

It is essential to log into the affected web applications and systematically update any default credentials to ensure compliance with robust password policies. This should include implementing complex passwords that meet organizational standards and regularly reviewing user accounts to prevent unauthorized access.

Reproduction Steps

Navigate to the affected web application URLs and attempt to log in using default credentials.

References

https://www.cisa.gov/news-events/alerts/2013/06/24/risks-default-passwords-internet,
https://cwe.mitre.org/data/definitions/1392.html

Evidence

```
http://[obfuscated] - Web Image Monitor
```

HIGH

Weak Active Directory Account Password Policy

Observation

An Active Directory Domain Password Policy is extremely critical as it is the security settings that many domain user accounts will use when having their accounts configured. These policies include lockout thresholds, lockout durations, minimum characters required, password complexity requirements, and more. During post-exploitation, it was discovered that the password policy configured does not meet security best practices.

Security Impact

A weak password policy can be disastrous for a company in that it allows attackers to exploit the weaknesses of domain user accounts. For example, the lack of a strict account lockout threshold allows malicious attackers to perform numerous login attempts to domain user accounts prior to being locked out. Here are some of the security impacts that can be associated with domain password policies:

Minimum password length: An attacker can take advantage of this by trying weak passwords that exist in the dictionary, such as Apple, Car, Dog, etc. By increasing the minimum password length, an attacker's chances of successfully guessing and/or even cracking (through password cracking techniques) a password is much lower.
Lockout threshold: If the lockout threshold value is too low, an attacker can perform numerous login attempts to the user accounts before locking out an account, which then depends on the lockout duration for unlocking the domain user account.
Lockout duration (minutes): If the account does not remain locked out for a long period of time, then attackers can continuously perform login attempts every X amount of minutes that the account gets unlocked. A small number increases the chances of a successful attack as the disruption to user accounts will be minimum.
Lockout observation window (minutes): By default, Microsoft Windows sets this to 30. This setting indicates how many times someone can perform a login attempt before it subtracts from the lockout threshold. For example, if this setting is set to 30, then this means an attacker can perform one login attempt per 30 minutes, and the lockout threshold will never exceed the value of 1 because the observation window *resets* the counter every 30 minutes.

Recommendation

Use the references to reconfigure your domain's password policy to adhere to security best practices. It is crucial to enforce complex passwords. In addition, the following minimum configurations are recommended:

```
- Minimum password length: no lower than 8
- Reset Account Lockout Counter / Lockout observation window (minutes): at least 30 minutes
- Locked Account Duration / Lockout duration (minutes): at least 30 minutes
- Lockout threshold / Account Lockout Threshold: no higher than 5
```

Note: the Lockout threshold / Account Lockout Threshold configuration should not be set to 0, because that disables the threshold and can allow malicious actors to perform bruteforce password attacks without the risk of locking out users.

Reproduction Steps

Using the Microsoft Windows command line interface (CLI), use the following command to query the domain's password policy:

```
net accounts "domain" /domain
```

References

https://blog.devolutions.net/2018/02/top-10-password-policies-and-best-practices-for-system-administrators
https://www.microsoft.com/en-us/microsoft-365/blog/2018/03/05/azure-ad-and-adfs-best-practices-defending-against-password-spray-attacks/

Evidence

The following weak Active Directory password policy settings were identified for the [obfuscated] domain:

```
Reset Account Lockout Counter: 10
Locked Account Duration: 10
```

MEDIUM

Anonymous FTP Enabled

Observation

The presence of an Anonymous File Transfer Protocol (FTP) service permits unrestricted access to a remote FTP server, enabling users to transfer files without authentication. This configuration typically establishes user credentials, which may consist of complex usernames and passwords. However, the penetration test identified that anonymous FTP was enabled, allowing any user to log in without credentials and browse files stored on the server.

Security Impact

The availability of anonymous FTP poses significant security risks, as it grants any individual, including potential attackers, unrestricted access to the FTP server's contents. This level of access increases the risk of unauthorized actions, such as the retrieval of sensitive files and, depending on configured permissions, even the ability to upload malicious files. Such exposure compromises the confidentiality and integrity of sensitive data that should only be accessible to authorized users.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.132	[obfuscated]	Undetected

Recommendation

To mitigate the risks associated with anonymous FTP, if the service is not essential for business operations, it is recommended to disable it and revise the organization's configuration baseline to disable unnecessary services by default. If the continuation of the service is necessary, anonymous authentication should be turned off, and robust authentication mechanisms, incorporating complex passwords, should be implemented to control access effectively.

Reproduction Steps

Using the operating system's built in FTP client, Metasploit, or Nmap, connect to the affected FTP server(s) using "anonymous/anonymous" (username and password).

References

https://www.cobalt.io/blog/anonymous-ftp-servers-overview,
https://security.stackexchange.com/questions/13828/anonymous-ftp-risks

Evidence

```
Nmap scan report for [obfuscated]
Host is up, received arp-response (0.00018s latency).
Scanned at 2025-03-14 16:03:04 UTC for 0s

PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 64
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -r--r--r-- root root 200 Jan  1 01:08 help
| -r--r--r-- root root 200 Jan  1 01:08 info
| -r--r--r-- root root 200 Jan  1 01:08 prnlog
| -r--r--r-- root root 200 Jan  1 01:08 stat
|_-r--r--r-- root root 200 Jan  1 01:08 syslog
MAC Address: 00:26:73:EE:EB:2C (Ricoh Company)
```

MEDIUM

Insecure Protocol - FTP

Observation

The File Transfer Protocol (FTP) is commonly employed by client systems for the purpose of connecting to, storing, and retrieving files on server(s). A significant vulnerability inherent to FTP is its lack of encryption for the data exchanged between the client and server, which renders all transmitted information visible in cleartext format. Although FTP has the capability to establish a secure connection through TLS, the current configuration of the affected server(s) does not initiate or negotiate this secure protocol. Consequently, any sensitive data transferred over FTP is susceptible to interception.

Security Impact

The absence of encryption in FTP exposes all communications between client(s) and server(s) in an unprotected manner. This vulnerability creates opportunities for attackers to execute man-in-the-middle attacks, where they can capture sensitive user credentials and the contents of files being transmitted. The retrieved information could subsequently facilitate further attacks within the network environment, thereby increasing the overall risk of data breaches and unauthorized access.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.132	[obfuscated]	Undetected

Recommendation

If the FTP service is not essential for business operations, disabling it is strongly recommended to minimize risk exposure. For environments where file transfers are necessary, transitioning to Secure FTP (SFTP) is advised. SFTP employs encryption protocols to secure communication channels between client(s) and server(s), thereby enhancing the security posture and protecting sensitive data during transmission.

Reproduction Steps

Use an FTP client to connect to one of the affected servers on port 21/tcp. The following syntax can be used to attempt connecting to an FTP server:

```
ftp <server_ip_address>
```

Furthermore, if an FTP client does not exist and the available operating system leverages the native telnet command, connectivity can be tested against an FTP server using the following syntax and leveraging the Telnet command:

```
telnet <server_ip_address> 21
```

If the command above works, then the remote server is listening on port 21/tcp.

References

https://www.ipa.go.jp/security/rfc/RFC2577EN.html

Evidence

```
[+] [obfuscated]:21      - FTP Banner: '220 RICOH MP C2004e -- snipped --
```

MEDIUM

Insecure Protocol - Telnet

Observation

The Telnet service is commonly utilized by network administrators for remote administration of network devices. However, this service lacks encryption, resulting in the transmission of all data in cleartext. This vulnerability poses significant risks, especially in environments where sensitive information is transmitted, as it can be intercepted by unauthorized actors.

Security Impact

Due to the unencrypted nature of Telnet communications, an attacker can execute a man-in-the-middle attack, facilitating the interception of sensitive data, including user credentials and command outputs. The compromise of such information can empower further attacks within the environment, potentially leading to a broader security breach.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
192.168.0.132	[obfuscated]	Undetected

Recommendation

It is highly recommended to disable the Telnet service unless it is essential for specific business operations. In cases where remote administration is necessary, adopting an alternative protocol, such as Secure Shell (SSH), is advisable. SSH provides robust encryption that secures the data in transit, significantly mitigating the risk of interception.

Reproduction Steps

Use a telnet client to connect to a telnet server. Using a network packet analyzer, such as Wireshark, observe the packets originating from the telnet client to discover the cleartext communications.

References

https://isc.sans.edu/diary/Computer+Security+Awareness+Month+-+Day+18+-+Telnet+an+oldie+but+a+goodie/7393

Evidence

```
[+] [obfuscated]:23      - [obfuscated]:23 TELNET RICOH Ma -- snipped --
```

MEDIUM

SMB NULL Session Authentication

Observation

The Server Message Block (SMB) protocol provides functionality for shared access to files and printers across networks. However, it is vulnerable to NULL session authentication, which permits unauthenticated access without requiring a username or password. This flaw allows any user, including malicious actors, to connect to SMB shares and potentially navigate through uploaded files.

Security Impact

The presence of SMB NULL session authentication poses a significant security risk as it enables unauthorized individuals to gain remote access to SMB shares, thereby exposing the contents to potential threats. Should this weak authentication lead to write permissions, an attacker could exploit this vulnerability to upload or execute malicious software. Furthermore, unauthorized exposure of sensitive files housed within these SMB shares can undermine the confidentiality and integrity of information that is intended for restricted access only.

Recommendation

To mitigate the risks associated with SMB NULL session authentication, it is advisable to disable the SMB service if it is not essential for business operations, while also updating the organizational configuration baseline to ensure unnecessary services are deactivated prior to system deployment. If the SMB service is necessary, disable NULL session authentication and enforce a robust authentication mechanism that utilizes complex passwords to strengthen access controls.

Reproduction Steps

Connect to the affected SMB server(s) using a blank username and a blank password. For the built-in Unix utility smbclient, the syntax is shown below:

```
smbclient -L <IP> --no-pass
```

If the operation succeeds without any errors and smbclient prints information about the configured shares and/or workgroups, the SMB server is affected.
The same checks can also be performed using dedicated scripts that are part of the Metasploit framework or the Nmap portscanning tool.

References

https://www.beyondsecurity.com/resources/vulnerabilities/null-session-availablesmb,
https://techcommunity.microsoft.com/blog/filecab/smb-and-null-sessions-why-your-pen-test-is-probably-wrong/1185365

Evidence

```
[[obfuscated]]
# crackmapexec smb [obfuscated] -u '' -p '' --local-auth
SMB                      [obfuscated]     445    [obfuscated]   [+] [obfuscated]\:
-----------------------------------------------------------
[[obfuscated]]
# crackmapexec smb [obfuscated] -u '' -p '' --local-auth
SMB                      [obfuscated]     445    [obfuscated]   [+] [obfuscated]\:
-----------------------------------------------------------
```

MEDIUM

SMB Signing Not Required

Observation

The penetration test identified configuration vulnerabilities in Microsoft Windows that could elevate the risk of attacks against operating systems within the targeted environment. By default, Microsoft Windows includes various settings that must be explicitly adjusted by network administrators to strengthen security postures. Failing to modify these configurations can leave systems open to multiple forms of exploitation.

Notably, the test revealed that the SMB signing feature was not enforced. SMB signing is a critical security mechanism designed to prevent SMB relay attacks, which occur when an attacker deceives a victim system into authenticating to the attacker, who then relays those valid credentials to another system. Without required SMB signing, systems may become susceptible to these types of attacks.

Security Impact

The prevalence of Microsoft Windows and Active Directory environments for user management means that a compromised Windows system may expose organizations to broader security risks, such as privilege escalation and lateral movement. Moreover, the uniformity of configurations across many systems due to Group Policy can amplify the consequences of a single misconfiguration.

In the context of SMB signing, a successful SMB relay attack could enable attackers to gain access to targeted systems determined by the permissions associated with the relayed credentials. This vulnerability may facilitate unauthorized remote command execution, resource access, and additional consequences that compromise system integrity.

Recommendation

To mitigate this vulnerability, it is essential to enforce SMB signing across all systems within the organization through Group Policy configuration. This proactive measure will enhance security by ensuring that SMB communications are authenticated, thereby reducing the risk of relay attacks.

Reproduction Steps

Leverage the "smb-security-mode" script within Nmap to scan a system for SMB signing. The following command can be run from a Linux system with Nmap installed:

```
nmap <ip> -p 445 -sS -Pn --script smb-security-mode -v -n
```

References

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory
https://www.microsoft.com/security/blog/2018/12/05/step-1-identify-users-top-10-actions-to-secure-your-environment/

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-security-baselines
https://support.microsoft.com/en-us/help/887429/overview-of-server-message-block-signing

Evidence

```
SMB                      [obfuscated]   445    [obfuscated]           [*] Windows 10 / Server 2019 Build 19041 x64
(name:[obfuscated]) (domain:[obfuscated]) (signing:False) (SMBv1:False)
SMB                      [obfuscated]   445    [obfuscated]   [*] Windows 10.0 Build 26100 x64 (name:[obfuscated])
(domain:[obfuscated]) (signing:False) (SMBv1:False)
SMB                      [obfuscated]   445    [obfuscated]         [*] Windows 10.0 Build 26100 x64 (name:[obfusc
ated]) (domain:[obfuscated]) (signing:False) (SMBv1:False)
SMB                      [obfuscated]   445    [obfuscated]       [*] Windows 10.0 Build 26100 x64 (name:[obfuscat
ed]) (domain:[obfuscated]) (signing:False) (SMBv1:False)
SMB                      [obfuscated]   445    [obfuscated]    [*] Windows 10.0 Build 26100 x64 (name:[obfuscate
d]) (domain:[obfuscated]) (signing:False) (SMBv1:False)
SMB                      [obfuscated]   445    [obfuscated]    [*] Windows 10.0 Build 26100 x64 (name:[obfuscate
d]) (domain:[obfuscated]) (signing:False) (SMBv1:False)
```

INFORMATIONAL Egress Filtering Deficiencies

Observation

An egress filtering evaluation was conducted during the internal penetration test to assess the internal network's access to the public Internet, highlighting potential risks associated with data exfiltration. This evaluation did not focus solely on a specific in-scope target; instead, it broadly examined egress permissions against the public Internet at large.

During the evaluation, it was identified that access to an extensive array of ports on the public Internet was permitted. The assessment specifically targeted scanme.nmap.org, a resource meant for organizations to verify the extent of their outward-facing network access.

Security Impact

Providing end-users with unrestricted access to numerous services, such as SSH and Telnet, poses significant security risks. Such access can enable an attacker or unauthorized user to circumvent existing security measures, leading to potential data exfiltration through non-standard communication channels. Furthermore, attackers could exploit this broad access to set up a command-and-control (C2) infrastructure, facilitating bidirectional communication with compromised systems.

Recommendation

It is crucial to disable access to all non-essential services that are not aligned with business needs. By implementing strict access controls to limit egress traffic to only those services imperative for business operations, organizations can enhance oversight of their communication channels. This allows for the detection of indicators of compromise (IoC) and mitigates the risks associated with unauthorized data exfiltration attempts.

Reproduction Steps

With permission, perform a scan against an Internet-facing service that has an excessive amount of ports opened. Analyze the results of the results to determine where services may be visible from the internal network environment.

References

https://insights.sei.cmu.edu/blog/best-practices-and-considerations-in-egress-filtering/,
https://www.packetlabs.net/posts/egress-filtering/

Evidence

```
Nmap scan report for scanme.nmap.org ([external-ip])
Host is up (0.096s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 984 closed tcp ports (reset)
PORT      STATE    SERVICE
21/tcp    open     ftp
```

```
22/tcp    open     ssh
25/tcp    open     smtp
53/tcp    open     domain
80/tcp    open     http
110/tcp   open     pop3
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   open     imap
443/tcp   open     https
445/tcp   filtered microsoft-ds
2000/tcp  open     cisco-sccp
5060/tcp  open     sip
8010/tcp  open     xmpp
9929/tcp  open     nping-echo
31337/tcp open     Elite

--snipped--
```

Appendix A: Host Discovery (Operating Systems)

Internal Network Penetration Test

The following table shows the operating systems that were discovered as part of this assessment. It should be noted that the operating system discovery techniques are only able to identify the specific OS versions based on the way the targets respond to various fingerprinting methods. In some cases, all operating systems may not be identifiable at the time of testing.

IP Address	DNS Name	Operating System	Domain
[obfuscated]	[obfuscated]	Windows 10 / Server 2019 Build 17763 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 10 / Server 2019 Build 17763 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 10 / Server 2019 Build 19041 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]
[obfuscated]	[obfuscated]	Windows 11 Build 26100 x64	[obfuscated]

Appendix B: Host Discovery (Opened Ports)

Internal Network Penetration Test

IP Address	DNS Name	Port	Protocol
[obfuscated]		443	tcp
[obfuscated]		123	udp
[obfuscated]		443	tcp
[obfuscated]		22	tcp
[obfuscated]		161	udp
[obfuscated]		443	tcp
[obfuscated]		123	udp
[obfuscated]		161	udp
[obfuscated]		22	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	389	udp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	123	udp
[obfuscated]	[obfuscated]	88	udp
[obfuscated]	[obfuscated]	53	udp
[obfuscated]	[obfuscated]	5357	tcp
[obfuscated]	[obfuscated]	3269	tcp
[obfuscated]	[obfuscated]	3268	tcp
[obfuscated]	[obfuscated]	636	tcp
[obfuscated]	[obfuscated]	593	tcp
[obfuscated]	[obfuscated]	464	tcp
[obfuscated]	[obfuscated]	389	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	88	tcp
[obfuscated]	[obfuscated]	53	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	88	tcp
[obfuscated]	[obfuscated]	53	tcp
[obfuscated]	[obfuscated]	88	udp
[obfuscated]	[obfuscated]	389	udp

[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	123	udp
[obfuscated]	[obfuscated]	53	udp
[obfuscated]	[obfuscated]	3269	tcp
[obfuscated]	[obfuscated]	3268	tcp
[obfuscated]	[obfuscated]	636	tcp
[obfuscated]	[obfuscated]	593	tcp
[obfuscated]	[obfuscated]	464	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	389	tcp
[obfuscated]		1080	udp
[obfuscated]		8888	tcp
[obfuscated]		1080	tcp
[obfuscated]		5353	udp
[obfuscated]		1701	udp
[obfuscated]		53	udp
[obfuscated]		22	tcp
[obfuscated]		443	tcp
[obfuscated]		53	tcp
[obfuscated]		2002	tcp
[obfuscated]		2001	tcp
[obfuscated]		80	tcp
[obfuscated]		137	udp
[obfuscated]		139	tcp
[obfuscated]		135	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	139	tcp

[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]		514	tcp
[obfuscated]		22	tcp
[obfuscated]		80	tcp
[obfuscated]		7001	tcp
[obfuscated]		135	tcp
[obfuscated]		137	udp
[obfuscated]		139	tcp
[obfuscated]		2179	tcp
[obfuscated]		139	tcp
[obfuscated]		137	udp
[obfuscated]		81	tcp
[obfuscated]		135	tcp
[obfuscated]		62078	tcp
[obfuscated]		49152	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	137	udp

[obfuscated]	[obfuscated]	445	tcp
[obfuscated]		23	tcp
[obfuscated]		80	tcp
[obfuscated]		139	tcp
[obfuscated]		443	tcp
[obfuscated]		514	tcp
[obfuscated]		65000	tcp
[obfuscated]		137	udp
[obfuscated]		161	udp
[obfuscated]		1900	udp
[obfuscated]		5353	udp
[obfuscated]		21	tcp
[obfuscated]		515	tcp
[obfuscated]		631	tcp
[obfuscated]		7443	tcp
[obfuscated]		8080	tcp
[obfuscated]		9100	tcp
[obfuscated]		623	udp
[obfuscated]		22	tcp
[obfuscated]		80	tcp
[obfuscated]		443	tcp
[obfuscated]		17988	tcp
[obfuscated]		161	udp
[obfuscated]		80	tcp
[obfuscated]		443	tcp
[obfuscated]		2068	tcp
[obfuscated]		161	udp
[obfuscated]		2068	tcp
[obfuscated]		161	udp
[obfuscated]		443	tcp
[obfuscated]		80	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	445	tcp

[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	139	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	135	tcp
[obfuscated]	[obfuscated]	137	udp
[obfuscated]	[obfuscated]	445	tcp
[obfuscated]	[obfuscated]	139	tcp