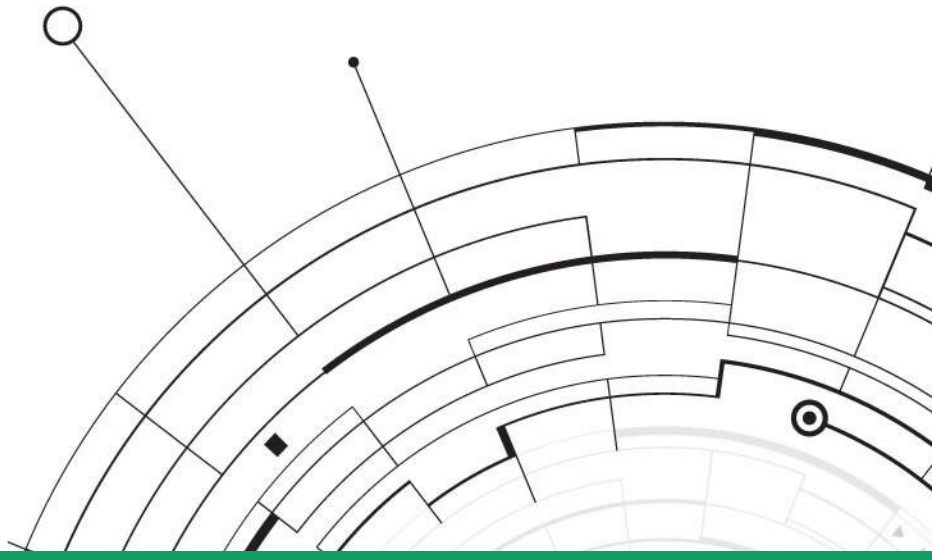
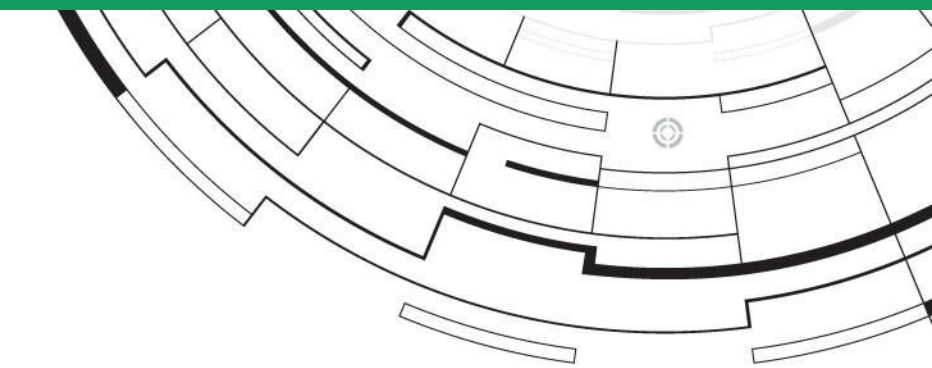


Internal Pentest + Vulnerability Assessment

TECHNICAL REPORT



Company

Feb 07, 2024

RSITEX.COM

Copyright

Confidentiality

This document contains confidential company 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. RSI treats the contents of a security audit as confidential company 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:	David Burgeson
Title:	CEO
Office:	(254) 222-2222
Email:	gojenne@rsitex.com
Primary Point of Contact	
Name:	Neil Borne
Title:	VP-Technology/Security Engineering
Office:	(512) 333-3333
Email:	gojene@rsitex.com

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 Security Assessment (6)		
IPv6 DNS Spoofing		Critical
Link-Local Multicast Name Resolution (LLMNR) Spoofing		Critical
Multicast DNS (mDNS) Spoofing		Critical
NetBIOS Name Service (NBNS) Spoofing		Critical
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), Renaissance Systems Inc. recommends that COMPANY visit the specific URLs provided within the table below. Furthermore, Renaissance Systems Inc. has also elaborated on how these TTPs were used during the penetration test in this report's Penetration Test Narrative section.

Renaissance Systems Inc. recommends COMPANY 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, Feb 07, 2024 @ 07:15:40 AM CST	Active Scanning: Scanning IP Blocks	Reconnaissance	xxxxxx
Tue, Feb 07, 2024 @ 07:15:40 AM CST	Network Service Discovery	Discovery	xxxxxx
Tue, Feb 07, 2024 @ 07:18:04 AM CST	Remote System Discovery	Discovery	xxxxxx
Tue, Feb 07, 2024 @ 08:49:46 AM CST	System Information Discovery	Discovery	xxxxxx
Tue, Feb 07, 2024 @ 08:54:58 AM CST	Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay	Credential-access	xxxxxx

Fayetteville Internal Pentest + Vulnerability Assessment

Engagement Scope of Work

Through discussions with COMPANY staff, the following target applications, IP addresses, and/or ranges were included as part of the engagement scope.

IP ADDRESSES & RANGES			
xxx.xxx.x.xx	xxx.xxx.x.xx		

Agent Information

To perform this assessment, Renaissance Systems Inc. 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	AGENT
Private IP Address	xxx.xxx.x.xx
Subnet Mask	xxx.xxx.x.xx (/24)
DNS Server	xxx.xxx.x.xx
Default Gateway	xxx.xxx.x.xx

Task Performed

To assess the targets listed above fully, Renaissance Systems Inc. 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

Renaissance Systems Inc. and COMPANY agreed to the following rules of engagements:

ACTIVITY	DEFINITION	PERMISSION
Exploitation	Renaissance Systems Inc. consultants will cautiously execute exploitation techniques to gain access to sensitive data and/or systems.	Yes
Post Exploitation	If exploitation is successful, Renaissance Systems Inc. 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

Penetration Test Narrative

This phase of the internal network penetration test describes some of the action performed as part of the penetration test, including host discovery, enumeration, exploitation, and post-exploitation (if opportunities were identified). It should be noted that this portion of the report does not represent the entire list of activities that were performed as part of this assessment, primarily just those that led to some level of access, significant exposure to information, and other activities relevant to the goal of the assessment. It should also be noted that this portion of the test heavily focused on the network layer within the environment.

Host Discovery

The first process that was performed during the penetration test was host discovery. Host discovery includes several tasks, including port scanning and ping sweeps, to identify the active systems within the environment. This is a crucial step in the penetration test as it allows attackers to determine what systems are active within the targeted IP addresses and/or ranges.

Of the two (2) IP addresses/ranges that were provided as part of the scope, Renaissance Systems Inc. was able to identify a total of two (2) systems to be active within the targeted environment.

MITRE | ATT&CK®	
Name	Active Scanning: Scanning IP Blocks
Tactic	Reconnaissance
TTP ID	xxxxxx
Note	Renaissance Systems Inc. also performed a port scan against two (2) targets to identify opened ports and running services. Port scanning is also important in that it allows one to identify which ports are opened and visible from the tested system. By discovering open ports within the environment, it is then possible to determine which services are running and if any of the running services are vulnerable.

Of the two (2) addresses/ranges that were scanned, Renaissance Systems Inc. found ten (10) ports opened.

Enumeration

After identifying the available hosts within the network, the next phase is to conduct enumeration. Enumeration consists of scanning the identified ports to determine what services are running. Additional scans are performed based on the running services to attempt to enumerate information from the running services (if possible). Such information may be useful for identifying additional vulnerabilities or knowledge for performing an attack against the service.

To help understand the operating systems and ports that were found to be most common within the environment, the following tables display the top 10 operating systems and top 10 ports.

OPERATING SYSTEM	COUNT
Windows 11 Build 22621 x64	1
Undetected	1

PORT/PROTOCOL	COUNT
7547/tcp	1
80/tcp	1
22/tcp	1

5040/tcp	1
3389/tcp	1
445/tcp	1
139/tcp	1
135/tcp	1
7680/tcp	1
5357/tcp	1

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	xxxxx
Note	Renaissance Systems Inc. performed an arp-scan across the local network subnet to determine which systems are on the local subnet (xxx.xxx.x.xx /24). 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, Renaissance Systems Inc. used a tool known as *arp-scan*.

The following results demonstrate that ten (10) systems exist on the same local subnet:

```
Interface: enp2s0, type: EN10MB, MAC: 84:00:00:0f:7b:be, IPv4: xxx.xxx.x.xx
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/xxxxx/arp-scan)
XXX.XXX.X.XX      xxxxx                   Cisco Systems, Inc
XXX.XXX.X.XX      xxxxx                   Dell Inc.
XXX.XXX.X.XX      xxxxx                   Raspberry Pi Foundation
XXX.XXX.X.XX      xxxxx                   Ubiquiti Networks Inc.


14 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.022 seconds (126.61 hosts/sec). 10 responded
```

While Renaissance Systems Inc. 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.

Renaissance Systems Inc. identified one (1) SSH service 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:

```
[*] XXX.XXX.X.XX - SSH server version: SSH-2.0-dropbear_2022.25
```

Next, Renaissance Systems Inc. identified one (1) system that exposed the Remote Desktop Protocol (RDP) service on port 3389/tcp. The following scan results display (some of) the identified services:

```
[*] XXX.XXX.X.XX            - Detected RDP on XXX.XXX.X.XX:3389       (name:CCCC-XX-XXX) (domain: CCCC-XX-XXX) (domain_XX
dn:ccccc -- snipped --
```

The identified operating system for this host is not compatible with common vulnerabilities affecting the RDP service. As a result, the RDP service was not scanned for vulnerabilities.

Next, Renaissance Systems Inc. identified one (1) system 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 *relays* them to another system, pivoting off that authenticated session to perform additional attacks, such as remote code execution.

Testing identified one (1) of the one (1) system 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:

```
XXX.XX.X.XX:(signing:False)
```

MITRE | ATT&CK®	
Name	System Information Discovery
Tactic	Discovery
TTP ID	xxxxx
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, Renaissance Systems Inc. identified one (1) operating system. It's important to note that the tools and techniques used to gather information about operating system versions are not always 100% accurate. While Renaissance Systems Inc. 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                      xxx.xxx.xx.xx    445     CCC-CCC-CCCC      [*] Windows 11 Build 22621 x64 (name:CCC-CCC-CCCC) (dom
ain:ccc-ccc-cccc) (SMBv1:False)
```

No outdated operating systems were identified during this enumeration process.

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 Tue Feb  7 14:53:00 2024

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

Target ........... xxx.xxx.x.xx
RID Range ........ 500-550,1000-1050
Username.....''
Password.....''
Known Usernames .. administrator, guest, domain admins, root, bin, none


 ============================( Enumerating Workgroup/Domain on xxx.xxx.x.xx )============================


[+] Got domain/workgroup name: GROUP


 ================================( Nbtstat Information for xxx.xxx.x.xx )================================
```

```
Looking up status of xxx.xxx.x.xx

---------- 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.

Next, Renaissance Systems Inc.'s objective was to perform a password attack against the Active Directory environment. However, Renaissance Systems Inc. needed to gather a list of potential domain user accounts to perform this process. Renaissance Systems Inc. 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.

At the time of testing, Renaissance Systems Inc. was unable to identify any in-scope domain controllers with the Kerberos service (88/tcp) available. It was therefore not possible to perform domain username guessing.

MITRE | ATT&CK®	
Name	Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay
Tactic	Credential-access
TTP ID	xxxxx
Note	As part of the exploitation phase, Renaissance Systems Inc. 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.

Renaissance Systems Inc. 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 Renaissance Systems Inc. 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:

```
2024-02-07 15:07:56,768 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J33J3J3J._dosvc
 2024-2-07  15:07:56,773 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J3J3J3J(1)._dosvc
2024-02-07 15:07:56,776 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J3J3J3J(2)._dosvc
```

```
      SNIPPED

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

When attempting to perform IPv6 attacks, Renaissance Systems Inc. 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: enp2s0 [84:00:00:0f:0b:be]
IPv4 address: xxx.xxx.x.xx
IPv6 address: fe80::00:00c8:e0df:000c
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 fe80::000:1 is now assigned to mac=68:00:ca:e0:00:00 host=COMPE-1VD0PK0. ipv4=
Sent spoofed reply for dns.google. to fe80::0000:1
Sent spoofed reply for dns.google. to fe80::0000:1
Sent spoofed reply for instance1.rfc868server.com. to fe80::0000:1
Renew reply sent to fe80::0000:1
Sent spoofed reply for dns.google. to fe80::0000:1
Sent spoofed reply for instance1.rfc868server.com. to fe80::0000:1

Shutting down packet capture after next packet...

      SNIPPED

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


----------          ----------
```

At the time of testing, Renaissance Systems Inc. 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. The following output is a snippet of the unsuccessful results:

```
Impacket v0.12.0.dev1 - Copyright 2023 Fortra


[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Client requested path: /wpad.dat
[*] HTTPD(80): Connection from COMPE-1VD0PK3/ADMIN@::ffff:xxx.xxx.x.xx controlled, attacking target smb://xxx.xxx.x.xx
[*] HTTPD(80): Client requested path: /lpq0wde0ub
[*] HTTPD(80): Client requested path: /lpq0wde0ub
[*] HTTPD(80): Client requested path: /lpq0wde0ub
[-] Authenticating against smb://xxx.xxx.x.xx as WAVE-1VD0PK0/ADMIN FAILED

SNIPPED

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

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, Renaissance Systems Inc. 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 — IPv6 DNS Spoofing

Observation

IPv6 DNS spoofing is possible due to the possibility of deploying a rogue DHCPv6 server on the internal network. Since Microsoft Windows systems prefer IPv6 over IPv4, IPv6-enabled clients will prefer to obtain IP address configurations from a DHCPv6 server when one is available.

During an attack such as the one performed during this assessment, an IPv6 DNS server was assigned to IPv6-enabled clients; however, the IPv6-enabled clients retained their pre-existing IPv4 address configurations - IP address, default gateway, and subnet mask.

Security Impact

By deploying a rogue DHCPv6 server, an attacker is able to intercept DNS requests by reconfiguring IPv6-enabled clients to use the attacker's system as the DNS server. Such an attack could potentially lead to the successful capture of sensitive information, including user credentials and other information. Resolving all DNS names to an attacker's system results in the victim's system communicating with services such as SMB, HTTP, RDP, MSSQL, etc. all hosted on the attacker's system.

Recommendation

Disable IPv6 unless it is required for business operations. As disabling IPv6 could potentially cause an interruption in network services, it is strongly advised to test this configuration prior to mass deployment. An alternative solution would be to implement DHCPv6 guard on network switches. Essentially, DHCPv6 guard ensures that only an authorized list of DHCP servers are allowed to assign leases to clients.

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.fox-it.com/2018/01/11/mitm6-compromising-ipv4-networks-via-ipv6/

Evidence

```
IPv6 address fe80::3318:1 is now assigned to mac=68:05:ca:e9:55:99 host=WAVE-1VD9PK3. ipv4=
Sent spoofed reply for dns.google. to fe00::0000:1
Sent spoofed reply for dns.google. to fe00::33000018:1
Sent spoofed reply for updates.vmsproxy.com. to fe00::0000:1
Sent spoofed reply for mobile.events.data.microsoft.com. to fe00::0000:1
Sent spoofed reply for instance1.rfc868server.com. to 0e80::0000:1
Sent spoofed reply for dns.google. to fe00::0000:1 0

--snipped--
```

CRITICAL

Link-Local Multicast Name Resolution (LLMNR) Spoofing

Observation

Link-Local Multicast Name Resolution (LLMNR) is a protocol used amongst workstations within an internal network environment to resolve a domain name system (DNS) name when a DNS server does not exist or cannot be helpful.

When a system attempts to resolve a DNS name, the system proceeds with the following steps:

1. The system checks its local host file to determine if an entry exists to match the DNS name in question with an IP address.
2. If the system does not have an entry in its local host's file, the system then sends a DNS query to its configured DNS server(s) to attempt to retrieve an IP address that matches the DNS name in question.
3. If the configured DNS server(s) cannot resolve the DNS name to an IP address, the system then sends an LLMNR broadcast packet on the local network to seek assistance from other systems.

Security Impact

Since the LLMNR queries are broadcasted across the network, any system can respond to these queries with the IP address of the DNS name in question. This can be abused by malicious attackers since an attacker can respond to all of these queries with the IP address of the attacker's system. Depending on the service that the victim was attempting to communicate with (e.g. SMB, MSSQL, HTTP, etc.), an attacker may be able to capture sensitive cleartext and/or hashed account credentials. Hashed credentials can, many times, be recovered in a matter of time using computing modern-day computing power and brute-force techniques.

Recommendation

The most effective method for preventing exploitation is to configure the Multicast Name Resolution registry key in order to prevent systems from using LLMNR queries.

- **Using Group Policy:** Computer Configuration\Administrative Templates\Network\DNS Client \Turn off Multicast Name Resolution = Enabled (To administer a Windows 2003 DC, use the Remote Server Administration Tools for Windows 7 - http://www.microsoft.com/en-us/download/details.aspx?id=7887)
- **Using the Registry for Windows Vista/7/10 Home Edition only:** HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\DNSClient \EnableMulticast

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

```
2024-02-07 15:14:12,169 - [*] [LLMNR]  Poisoned answer sent to Fe00::cfdf:0000:a000:000f for name  COMPE-1VD0PK0
2024-02-07 15:14:12,173 - [*] [LLMNR]  Poisoned answer sent to xxx.xxx.x.xx for name WAVE-1VD0PK0
2024-02-07 14:55:06,108 - [*] [LLMNR]  Poisoned answer sent to Fe00::cfdf:0000:a000:000f for name  CCPE-1VD0PK0
2024-02-07 14:55:06,109 - [*] [LLMNR]  Poisoned answer sent to xxx.xxx.x.xx for name WAVE-1VD0PK0
2024-02-07 14:55:06,608 - [*] [LLMNR]  Poisoned answer sent to Fe00::cfdf:0000:a000:000f:a5000:00 CCPE-1VD0PK0
                                                               9f for name
```

CRITICAL

Multicast DNS (mDNS) Spoofing

Observation

Multicast DNS (mDNS) is a protocol used within small networks to resolve a domain name system (DNS) name when a local DNS server does not exist.

When a system attempts to resolve a DNS name, the system proceeds with the following steps:

1. The system checks its local host file to determine if an entry exists to match the DNS name in question with an IP address.
2. On small networks where no DNS Server is configured, the system then uses mDNS to send an IP multicast query message to the systems on the local subnet that asks the host having that name to identify itself. Attackers can take advantage of this by answering this request and impersonating a system on the network.

Security Impact

Since the mDNS queries are sent to systems on the local subnet, any system can respond to these queries with the IP address of the DNS name in question. This can be abused by malicious attackers since an attacker can respond to all of these queries with the IP address of the attacker's system. Depending on the service that the victim was attempting to communicate with (e.g. SMB, MSSQL, HTTP, etc.), an attacker may be able to capture sensitive cleartext and/or hashed account credentials. Hashed credentials can, many times, be recovered in a matter of time using computing modern-day computing power and brute-force techniques.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
xxx.xxx.x.xx	CMPE-XXX-XX	Windows 11 Build 22621 x64

Recommendation

The most effective method for preventing exploitation is to disable mDNS altogether if it is not being used. Depending on the implementation, this can be achieved by disabling the Apple Bonjour or avahi-daemon service.

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

```
2024-02-07 15:07:56,791 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX(8).□_dosvc
2024-02-07 15:07:56,794 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX (9).□_dosvc
2024-02-07 15:07:56,797 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX (10).□_dosvc
2024-02-07 15:07:56,799 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX (11).□_dosvc
2024-02-07 15:07:56,801 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX (12).□_dosvc
2024-02-07 15:07:56,804 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX (13).□_dosvc
2024-02-07 15:07:56,806 - [*] [MDNS] Poisoned answer sent to xxx.xxx.x.xx     for name DESKTOP-J0C30CJX (14).□_dosvc
```

CRITICAL

NetBIOS Name Service (NBNS) Spoofing

Observation

NetBIOS Name Service (NBNS) is a protocol used amongst workstations within an internal network environment to resolve a domain name system (DNS) name when a DNS server does not exist or cannot be helpful.

When a system attempts to resolve a DNS name, the system proceeds with the following steps:

1. The system checks its local host file to determine if an entry exists to match the DNS name in question with an IP address.
2. If the system does not have an entry in its local hosts file, the system then sends a DNS query to its configured DNS server(s) to attempt retrieving an IP address that matches the DNS name in question.
3. If the configured DNS server(s) cannot resolve the DNS name to an IP address, the system then sends an NBNS broadcast packet on the local network to seek assistance from other systems.

Security Impact

Since the NBNS queries are broadcasted across the network, any system can respond to these queries with the IP address of the DNS name in question. This can be abused by malicious attackers since an attacker can respond to all of these queries with the IP address of the attacker's system. Depending on the service that the victim was attempting to communicate with (e.g. SMB, MSSQL, HTTP, etc.), an attacker may be able to capture sensitive cleartext and/or hashed account credentials. Hashed credentials can, many times, be recovered in a matter of time using computing modern-day computing power and brute-force techniques.

Recommendation

Disable the NetBIOS service for all hosts in the internal network. This can be performed via DHCP options, network adapter settings, or a registry key. For additional information, refer to the references section.

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

```
2024-02-07 14:55:08,126 - [*] [NBT-NS] Poisoned answer sent to xxx.xxx.x.xx.3.50 for name WPAD (service:
Workstation/Re director)
```

MEDIUM

SMB Signing Not Required

Observation

Testing identified Microsoft Windows configuration concerns that could potentially result in an increased risk of an attack against Microsoft operating systems within the targeted environment. By default, Microsoft Windows comes pre-installed with several configuration issues that require network administrators to explicitly disable or enable to enhance security. If these options are not modified, then these systems could remain vulnerable to several attacks.

More specifically, the SMB signing feature was not found to be required at the time of testing. SMB signing is a security feature implemented by Microsoft to combat SMB relay attacks. An SMB relay attack occurs when an attacker tricks the victim system into authenticating to the attacker, and the attacker relays those credentials to another system.

Security Impact

Since many organizations use Microsoft Windows and Active Directory environments to manage users, a successful attack against a Microsoft Windows system could potentially expose the organization to other attacks, including privilege escalation and lateral movement. Furthermore, many Microsoft Windows systems share similar configurations due to Group Policy's ability to configure settings on a global scale. A single misconfiguration within Group Policy could present significant threats.

As it relates to SMB signing, a successful SMB relay attack could provide an attacker with access to a system of the attacker's choosing, depending on the permission levels of the authentication credentials being relayed. This could result in remote command execution, access to resources, and more.

Affected Nodes

ONE (1) NODE AFFECTED		
IP Address	**Host Name**	**Operating System**
xxx.xxx.x.xx	CMPE-XXX-XXXX	Windows 11 Build 22621 x64

Recommendation

Enforce SMB signing by configuring this across the organization's systems via Group Policy.

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                      xxx.xxx.x.xx    445    CMPE-XXX-XXXX    [*] Windows 11 Build 22621 x64 (name:CMPE-XXX-XXX
TH) (domain:Arkss-fay-beth) (signing:False) (SMBv1:False)
xxx.xxx.x.xx:(signing:False)
```

INFORMATIONAL

Egress Filtering Deficiencies

Observation

An egress filtering check was performed as part of the internal network penetration test. This check aims to determine if the internal environment allows excessive access to the public Internet, which could increase the risk of data exfiltration. This check was not performed against a specific in-scope target, but on the public Internet in general to evaluate this risk.

During this check, it was possible to identify access to an excessive number of ports residing on the public Internet. This particular check targeted scanme.nmap.org, which is designed for organizations to check whether or not they have access to servers on the public Internet.

Security Impact

Allowing end-users access to excessive services, such as SSH, Telnet, etc. allows for an attacker or end-user to bypass security controls by exfiltrating information through other communication channels. During an attack, an attacker may also leverage this excessive access to establish a command-and-control (C2) server to communicate commands and data back and forth between a compromised system.

Recommendation

Disable access to services that are not required for business operations. Restricting access to only services that are required for business operations allows the organizations to establish more control over communication channels, allowing for inspection of indicators of compromise (IoC) as well as malicious 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.

Evidence

```
# Nmap 7.95 scan initiated Tue Feb  7 13:18:05 2024 as: nmap -sS -Pn -v -n -oA /root/pentest/192345/discovery/scan
me scanme.nmap.org
Nmap scan report for scanme.nmap.org ([external-ip])
Host is up (0.054s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f00c:00ff:fe00:bb0f
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Read data files from: /usr/local/bin/../share/nmap
# Nmap done at Tue Feb  7 13:18:06 2024 -- 1 IP address (1 host up) scanned in 1.15 seconds
```

Appendix A: Host Discovery (Operating Systems)

Internal Network Security Assessment

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
xxx.xxx.x.xx	CMPE-XXX-XXXX	Windows 11 Build 22621 x64	

Appendix B: Host Discovery (Opened Ports)

Internal Network Security Assessment

IP Address	DNS Name	Port	Protocol
xxx.xxx.x.xx	CMPE-XXX-XXXX	5357	tcp
xxx.xxx.x.xx	CMPE-XXX-XXXX	7680	tcp
xxx.xxx.x.xx	CMPE-XXX-XXXX	135	tcp
xxx.xxx.x.xx	CMPE-XXX-XXXX	139	tcp
xxx.xxx.x.xx	CMPE-XXX-XXXX	445	tcp
xxx.xxx.x.xx	CMPE-XXX-XXXX	3389	tcp
xxx.xxx.x.xx	CMPE-XXX-XXXX	5040	tcp
xxx.xxx.x.xx		22	tcp
xxx.xxx.x.xx		80	tcp
xxx.xxx.x.xx		7547	tcp