Adversary-in-the-Middle: Poisoning Windows Name Resolution Services for SMB-Relay

School of Cybersecurity
Bachelors in Cybersecurity
Davis, Jacen A.
01218081
Old Dominion University

ABSTRACT

Windows devices support alternative means of name resolution outside of DNS called the Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBNS). These peer-to-peer services exist for backward compatibility and secondary use in networks that do not use DNS. Without post-deployment modification, Windows systems, PC and Server support these services by default. Both services are vulnerable to poison attacks where an attacker responds to one or more queries that do not have an authoritative host to claim them. Using a tool like Responder/Multi-Relay, an attacker or pentester can exploit this vulnerability to poison name caches for these services and redirect victims to their system. Redirected victims can then authenticate to an attacker's rouge authentication services. Once a victim begins to authenticate, their credentials can be captured for offline cracking or used in SMB-Relay as an adversary-in-the-middle to relay authentication to an attacker/pen tester's preferred target. If successful, access to the target is obtained, compromising the host, and giving the attacker/pen tester complete control over the target system. The mitigations for these exploits are to disable LLMNR/NBNS on all Windows systems and prevent SMB-Relay by requiring SMB signing on all clients and servers.

Keywords: *Adversary-in-the-Middle; LLMNR; NBNS; Responder/Multi-Relay; SMB-Relay.*

INTRODUCTION

Windows devices are very flexible in backward compatibility and support alternative means of communications and discovery when other methods fail [1, p. 68]. In Windows peer-to-peer networking or Workgroups, other Windows devices can aid in discovering one another when a peer needs to find a device but only knows the name. In specific scenarios, the Domain Name System (DNS) is not desirable or is unavailable. However, Windows networks have their means of resolving names that cannot leave the link-local (the local area network or LAN) called the Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBNS).

LLMNR and NBNS are alternatives to DNS found near-exclusively on Windows networks and serve a similar purpose, restricted only by their link-local scope. Like DNS, a Windows device can query LLMNR/NBNS for name resolution/discovery and obtain the IPv4 or IPv6 address of the device with that hostname. Unlike DNS, where all workstations on a corporate network or personal devices on a home network have a DNS server to send all their queries to as a client, LLMNR/NBNS is peer-to-peer. When a peer queries LLMNR/NBNS over the network for name resolution, the reply is from another Windows peer in the link-local. Under normal circumstances, the peer that responds to the LLMNR/NBNS query is the device authoritative for that name. However, if the hostname does not exist, or the user misspelled the name to an existing application/service when attempting to discover and connect to it [1, p. 80], an opportunity to poison the name resolution process and direct authentication to an alternate system exists.

A malicious device on the link-local network can poison LLMNR/NBNS and redirect Windows hosts to theirs when they attempt name discovery for devices and services that do not exist on the network. When they connect using whatever service they intended, the malicious device can run the same services the victim expects and ask for their credentials. If the user manually supplies it or their Windows device automatically forwards their credentials, they have just become the victim of credential harvesting.

Depending on the service they are targeting, like gaining administrative access to a Windows server on the network, and the lack thereof other services on the network where they can obtain credentials clear text (FTP shares, HTTP services), an attacker can use SMB for such access. For current versions of SMB, if security signing (SMB signing) of each packet is not required for the connection of the client or server, it is open to a relay attack called SMB-Relay. The goal of SMB-Relay is to relay an SMB connection between an attacker and their preferred target by using the victim's credentials. Once the relay is started, without any form of tampering detection that SMB signing provides, an attacker can successfully relay the connection, including authentication. Once the victim is no longer required, they are disconnected soon after, leaving the attacker authenticated and accessing the server.

This paper will first examine the ubiquity of LLMNR and NBNS enabled by default on typical Windows installations, how they both work, and how they allow poisoning attacks.

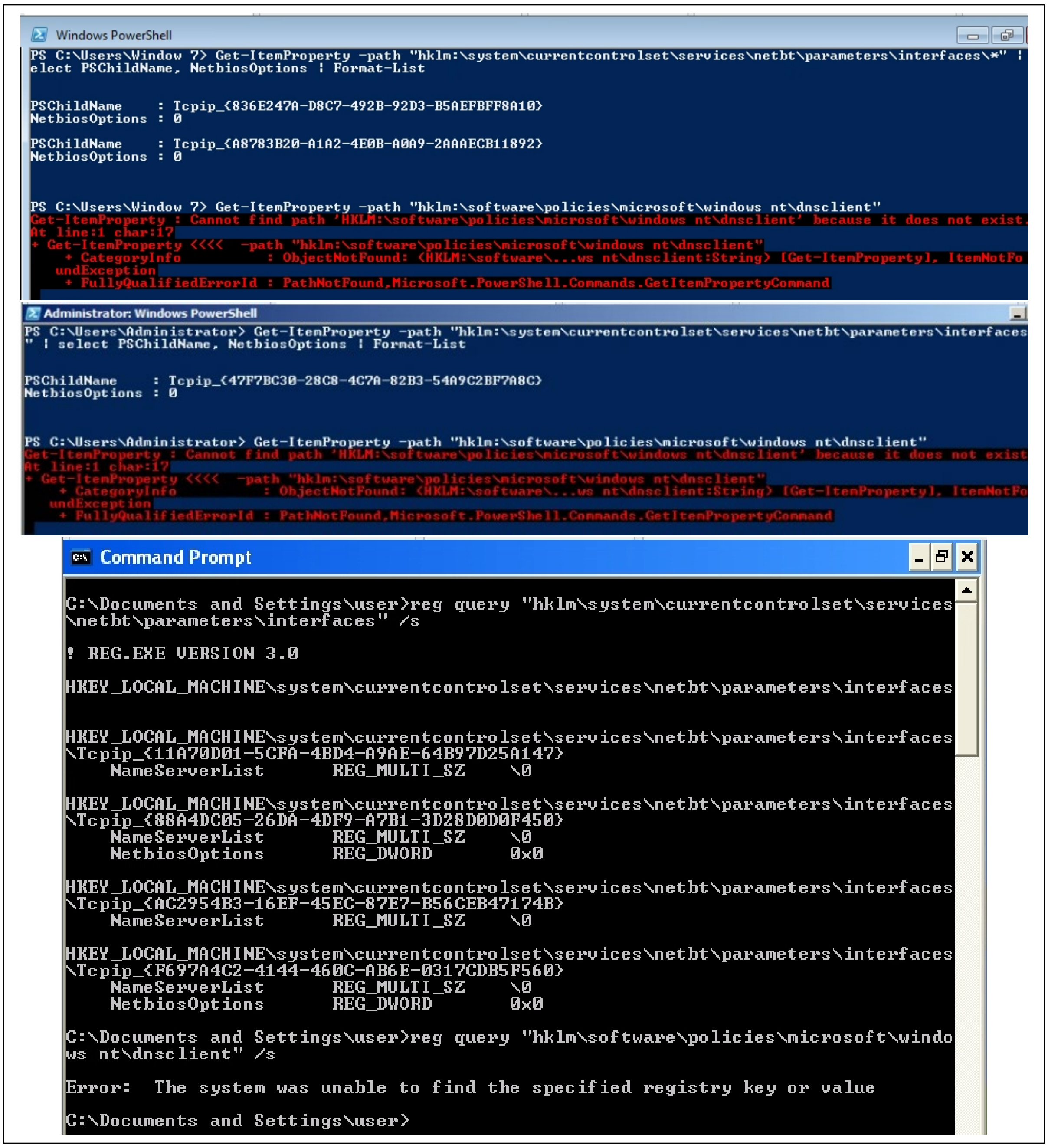
Next, SMB-Relay will be discussed, and a penetration testing tool that can perform the poison attack and the relay attack called Responder/MultiRelay. Finally, mitigations are proposed to eliminate LLMNR/NBNS poisoning and prevent the SMB-Relay attack from occurring at all.

LLMNR AND NBNS: UBIQUITY ON WINDOWS INSTALLATION AND DETECTION

The LLMNR and NBNS services are legacy name resolution protocols meant for backward compatibility and support when DNS is not implemented or required. By default, all versions of Windows, both PC and Server versions, have both services enabled by default [1]–[3]. Using the Windows registry, an administrator or cybersecurity analyst can confirm the actual state of both services on any given Windows system.

LLMNR's service status may or may not be found in its registry path *HKLM\SOFTWARE\Policies\Microsoft\Windows NT* inside the *DNSClient* key, instead of requiring the key to be created manually or by enabling it in Group Policy [2]. If the key is created, checking for a change in *EnableMulitcast* from **0** to **1** indicates LLMNR is enabled. If the *DNSClient* key or *EnableMulticast* is not present, this too indicates LLMNR is enabled (see Figure 1). A value of **0** set in *EnableMulticast* indicates LLMNR is disabled on Windows.

For NBNS, checking the *Tcpip_{UUID}* subkeys under *HKLM\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces* will confirm NBNS status. Under the Tcpip interface subkeys, if *NetBiosOptions* is **0**, control over NetBIOS is given to the DHCP server [4]. If the *NetBiosOptions* is **1**, all enabled NetBIOS extensions, including Name Service, are enabled for that interface. Finally, if *NetBiosOptions* is **2**, all NetBIOS extensions, including Name Service, are disabled. If *NetBiosOptions* is **0**, the DHCP server on the network must be reviewed for confirmation if it is disabling NetBIOS or not. On the DHCP server, if the DHCP option "001 Microsoft Disable Netbios Option" (in Windows DHCP Server Role) is unset or is **1**, NetBIOS services, including NBNS, are enabled for Windows devices [2]. Finally, if a static IP address is on the interface, NetBIOS and all extensions are enabled by default.



```
PS C:\Users\Window 7> Get-ItemProperty -path "hklm:\system\currentcontrolset\services\netbt\parameters\interfaces\*" |
elect PSChildName, NetbiosOptions | Format-List


PSChildName    : Tcpip_{836E247A-D8C7-492B-92D3-B5AEFBFF8A10}
NetbiosOptions : 0

PSChildName    : Tcpip_{A8783B20-A1A2-4E0B-A0A9-2AAAECB11892}
NetbiosOptions : 0



PS C:\Users\Window 7> Get-ItemProperty -path "hklm:\software\policies\microsoft\windows nt\dnsclient"
Get-ItemProperty : Cannot find path 'HKLM:\software\policies\microsoft\windows nt\dnsclient' because it does not exist.
At line:1 char:17
+ Get-ItemProperty <<<<  -path "hklm:\software\policies\microsoft\windows nt\dnsclient"
    + CategoryInfo          : ObjectNotFound: (HKLM:\software\...ws nt\dnsclient:String) [Get-ItemProperty], ItemNotFo
   undException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemPropertyCommand
```

```
PS C:\Users\Administrator> Get-ItemProperty -path "hklm:\system\currentcontrolset\services\netbt\parameters\interfaces
" | select PSChildName, NetbiosOptions | Format-List


PSChildName    : Tcpip_{47F7BC30-28C8-4C7A-82B3-54A9C2BF7A8C}
NetbiosOptions : 0



PS C:\Users\Administrator> Get-ItemProperty -path "hklm:\software\policies\microsoft\windows nt\dnsclient"
Get-ItemProperty : Cannot find path 'HKLM:\software\policies\microsoft\windows nt\dnsclient' because it does not exist
At line:1 char:17
+ Get-ItemProperty <<<<  -path "hklm:\software\policies\microsoft\windows nt\dnsclient"
    + CategoryInfo          : ObjectNotFound: (HKLM:\software\...ws nt\dnsclient:String) [Get-ItemProperty], ItemNotFo
   undException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetItemPropertyCommand
```

```
C:\Documents and Settings\user>reg query "hklm\system\currentcontrolset\services
\netbt\parameters\interfaces" /s

! REG.EXE VERSION 3.0

HKEY_LOCAL_MACHINE\system\currentcontrolset\services\netbt\parameters\interfaces


HKEY_LOCAL_MACHINE\system\currentcontrolset\services\netbt\parameters\interfaces
\Tcpip_{11A70D01-5CFA-4BD4-A9AE-64B97D25A147}
    NameServerList      REG_MULTI_SZ    \0

HKEY_LOCAL_MACHINE\system\currentcontrolset\services\netbt\parameters\interfaces
\Tcpip_{88A4DC05-26DA-4DF9-A7B1-3D28D0D0F450}
    NameServerList      REG_MULTI_SZ    \0
    NetbiosOptions      REG_DWORD       0x0

HKEY_LOCAL_MACHINE\system\currentcontrolset\services\netbt\parameters\interfaces
\Tcpip_{AC2954B3-16EF-45EC-87E7-B56CEB47174B}
    NameServerList      REG_MULTI_SZ    \0

HKEY_LOCAL_MACHINE\system\currentcontrolset\services\netbt\parameters\interfaces
\Tcpip_{F697A4C2-4144-460C-AB6E-0317CDB5F560}
    NameServerList      REG_MULTI_SZ    \0
    NetbiosOptions      REG_DWORD       0x0

C:\Documents and Settings\user>reg query "hklm\software\policies\microsoft\windo
ws nt\dnsclient" /s

Error:  The system was unable to find the specified registry key or value

C:\Documents and Settings\user>
```

FIGURE 1: Registry Values of NBNS and LLMNR configurations in Windows 7 Enterprise, Windows Server 2008 R2 Standard, and Windows XP Professional SP3 (in sequence).

LINK-LOCAL MULTICAST NAME RESOLUTION OVERVIEW

The goal of Link-Local Multicast Name Resolution is to "enable name resolution in scenarios in which conventional DNS name resolution is not possible" [5]. LLMNR runs on TCP/UDP port 5355 and uses the multicast address 224.0.0.252 and FF02::1:3 for IPv4 and IPv6, respectively, to propagate messages on the link-local. LLMNR supports all DNS current and future formats, types, and classes for propagating queries and responses to them. LLMNR is a peer-to-peer name resolution service: peers act as senders transmitting queries for host identification and responders replying to senders if they are authoritative over that name. LLMNR responders can self-allocate a single-label name, defined in RFC 1001, without the need for registration [5, p. 26]. LLMNR senders should only send queries over LLMNR for single-label names to avoid unnecessary DNS queries [5, p. 15]. **Figure 2.** shows the typical LLMNR usage in a small Windows network.

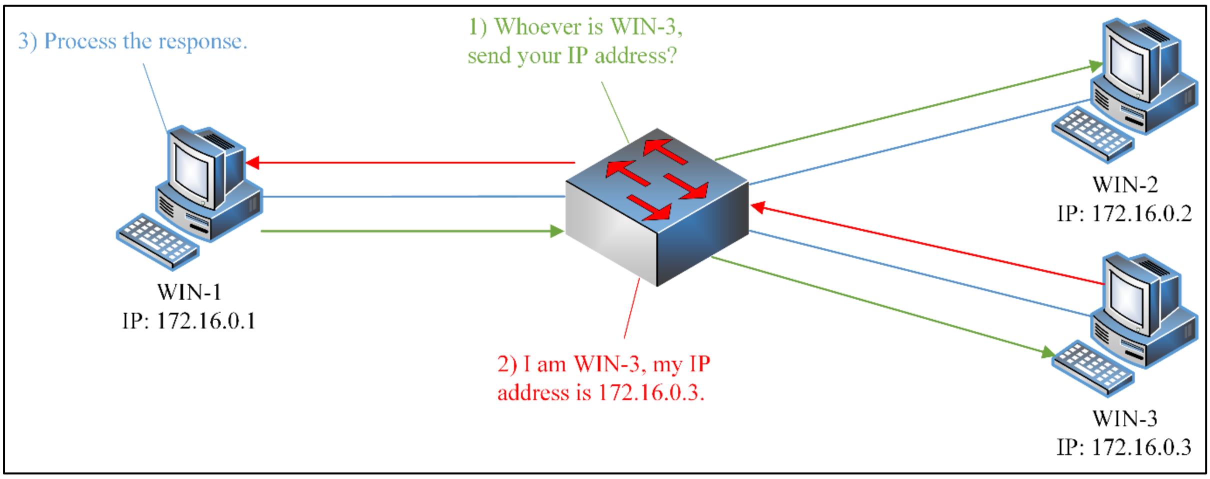

3) Process the response.
1) Whoever is WIN-3, send your IP address?
WIN-2
IP: 172.16.0.2
WIN-1
IP: 172.16.0.1
2) I am WIN-3, my IP address is 172.16.0.3.
WIN-3
IP: 172.16.0.3

FIGURE 2: LLMNR name resolution process overview. The sender, WIN-1, forwards the LLMNR query to port 5355 using the multicast address 224.0.0.252 as the destination. The multicast floods across the network and only the authoritative responder for WIN-3 should reply.

NETBIOS NAME SERVICE OVERVIEW

The second name resolution alternative used by Windows systems is NetBIOS Name Service (NBNS). NBNS runs over TCP/UDP port 137 and is one of three services of the NetBIOS[1] over TCP/IP (NetBT or NBT) protocol developed in RFCs 1001 and 1002 [3]. The goal of NBNS is to resolve names of resources, computers, and services to an IPv4 address for NBT communications on a TCP/IP network. NBT typically operates on networks using legacy Microsoft and Windows applications and services, are not operating as a domain, and do not have access to a DNS server [3]. NBT allows Windows computers to "browse" the network for other computers, services, file sharing, and provide name resolution across the network

[1] NetBIOS was a vendor-neutral application programming interface (API), not a network communication protocol [3]. As such, NetBIOS was implemented to work on top of different proprietary networking stacks such as IBM and Microsoft's legacy NetBIOS Extended User Interface (NetBEUI) protocol [3], [4, p. 92] to function. NBT was designed to preserve preexisting applications developed to use the NetBIOS API to work on the TCP/IP stack and allowed it to have the widest compatibility with as many systems as possible [3].

[3]. Name resolution through NBNS is an essential function of NetBIOS and for NBT as NetBIOS resources are referenced by name [6] that are a part of a flat namespace [3].

NBT is built around the concept of an end-node for applications and services. There are three types of end-nodes in NBT: broadcast ("B"), point-to-point ("P"), and mixed-mode ("M") nodes [3, p. 16]. Strictly speaking, the default behavior of NBT on Windows systems, if left unaltered by administrators, is to act as a B-node in the absence of configuration via DHCP or through WINS [5]. B-nodes perform peer-to-peer name discovery over the broadcast to find the hosts and services they are looking for and are therefore limited to the maximum extent of the broadcast domain [3, p. 21]. NBNS name resolution and discovery for B-Nodes are similar to LLMNR name resolution. The requesting node sends their query over IPv4 broadcast and waits for a response from a node claiming to be authoritative [6, p. 40] (see **Figure 2**).

LLMNR/NBNS POISONING OVERVIEW

LLMNR is susceptible to spoofing, and as a peer-to-peer name resolution protocol, no trust model is assumed between hosts that can prevent poisoning of the LLMNR cache [5, pp. 24–25]. An attacker has three options for the LLMNR poisoning attack [5, p. 24]:

1. An attacker performs a denial of services against DNS and sends spoofed LLMNR responses to LLMNR queries with falsified information.
2. An on-link attacker can spoof LLMNR responses to legitimate hosts with falsified information faster than the real responder.
3. An on-link attacker can spoof LLMNR responses to queries on names not on the link and shall never receive a legitimate response (see Figure 3).

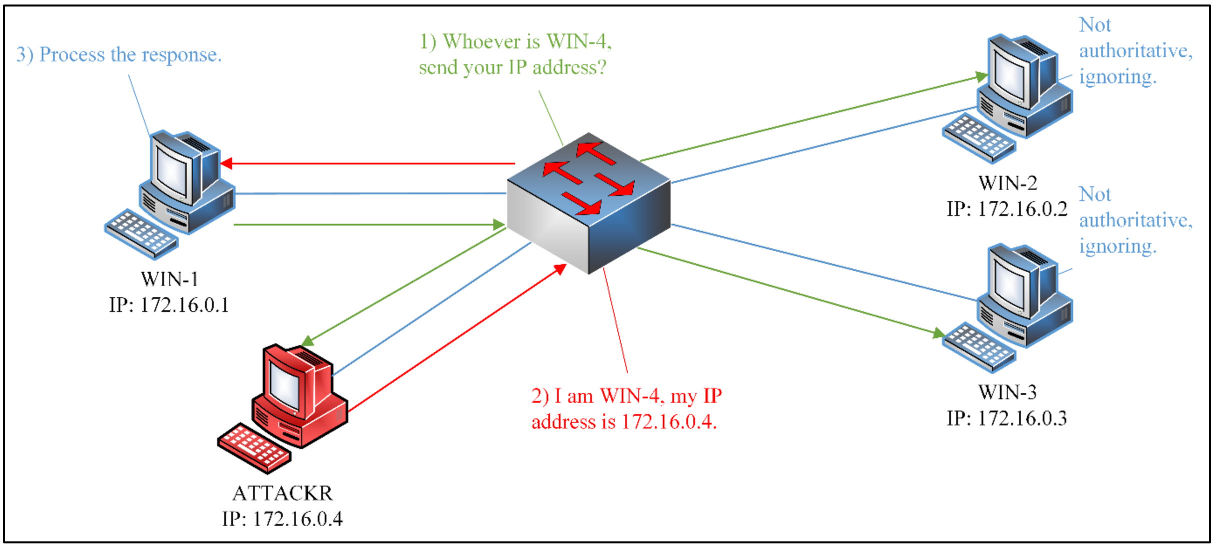

3) Process the response.
1) Whoever is WIN-4, send your IP address?
Not authoritative, ignoring.
WIN-2
IP: 172.16.0.2
Not authoritative, ignoring.
WIN-1
IP: 172.16.0.1
WIN-3
IP: 172.16.0.3
2) I am WIN-4, my IP address is 172.16.0.4.
ATTACKR
IP: 172.16.0.4

FIGURE 3: On-link attacker receives a copy of the LLMNR query and spoof themselves as the responder for an off-link or non-existent name.

LLMNR, as opposed to NBNS, does come with options to configure pre-arranged security between peer-to-peer systems [5, p. 25]. These security features include using TSIG security mechanisms and IPsec Encapsulating Security Payload with NULL encryption for authentication of unicast responses from responders by being configured as part of a group. Both do not protect responders from forgery if the attacker is a member of the group or has access to the pre-shared key of the group [5]. LLMNR may support a limited deployment of DNSSEC if the LLMNR implementation is DNSSEC aware. Unlike being registered as part of a group to prove authenticity, DNSSEC permits a responder to demonstrate ownership, enabling protection against forgeries [5]. However, as this is dependent on implementation and configuration, none of these features can be present if LLMNR is operating as an afterthought or the implementation (Microsoft, for example) omits DNSSEC.

NBNS, like LLMNR, is susceptible to poisoning attacks. The NetBIOS Working Group, when releasing RFCs 1001 and 1002 in 1987, did not conceive nor investigate the potential for poisoning vulnerabilities in NBNS. Regardless, NBNS functions similarly to LLMNR as it uses broadcasts and relies on other peer devices within link-local scope to respond (see sections LLMNR and NBNS Overviews), allowing the forging of replies in the name resolution process similar to the scenario depicted in **Figure 3** [1, p. 78], [7].

Once an attacker poisons the Windows victim's name cache, the victim may communicate with the attacker's system. If the host attempts to communicate with a requested resource or service that requires authentication, username and NTLMv1/2 hash are sent by Windows victim [7]. An attacker performing the LLMNR/NBNS poisoning attack at this point has two options: collect user credentials for offline cracking or use the poison attack in conjunction with another that gives the attacker access to another system. One method available is through the SMB-Relay exploit.

SMB-RELAY OVERVIEW

SMB-Relay is an Adversary-in-the-Middle exploit to use a victim's authentication material (username, password hash) to access a particular target. Once relayed and connected, the attacker can perform remote code execution on the target system. If successful, the attacker can execute commands, launch a remote shell, and install malware, effectively gaining control over the target system. To work, an attacker must (1) redirect the victim client into connecting to the attacker's system and (2) successfully relay the authentication to gain access to the target. In **Figure 4**, LLMNR/NBNS poisoning redirects a victim to the attacker, where they connect and are required to authenticate. In **Figure 5**, the attacker relays the user's connection and authentication material to connect to the target successfully.

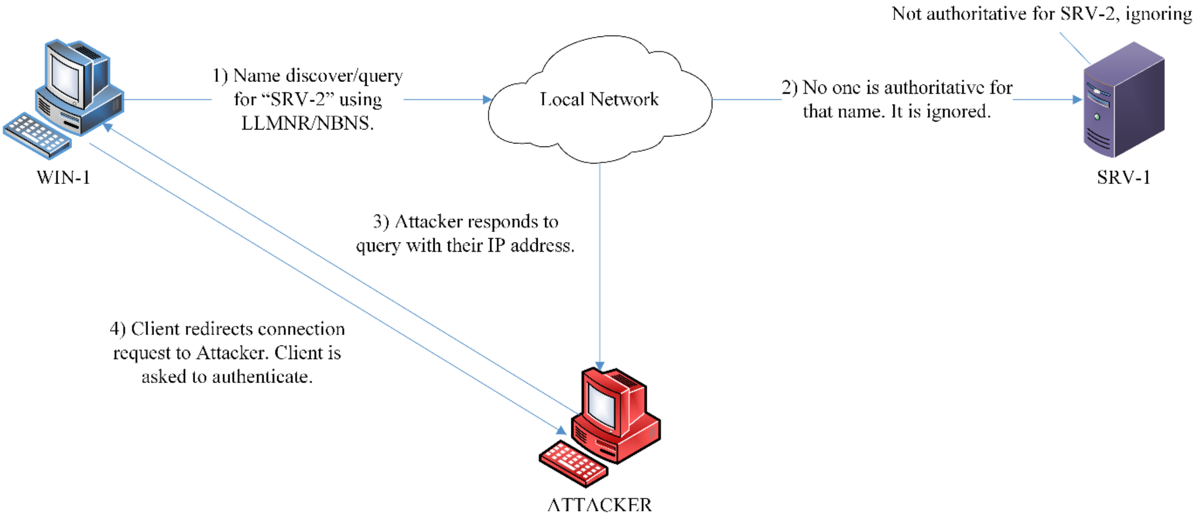

Not authoritative for SRV-2, ignoring
1) Name discover/query for "SRV-2" using LLMNR/NBNS.
Local Network
2) No one is authoritative for that name. It is ignored.
WIN-1
SRV-1
3) Attacker responds to query with their IP address.
4) Client redirects connection request to Attacker. Client is asked to authenticate.
ATTACKER

FIGURE 4: A client attempts to resolve a name to a computer or service that does not exist. The attacker hijacks the name resolution process; the client is redirected to the attacker's system for authentication.

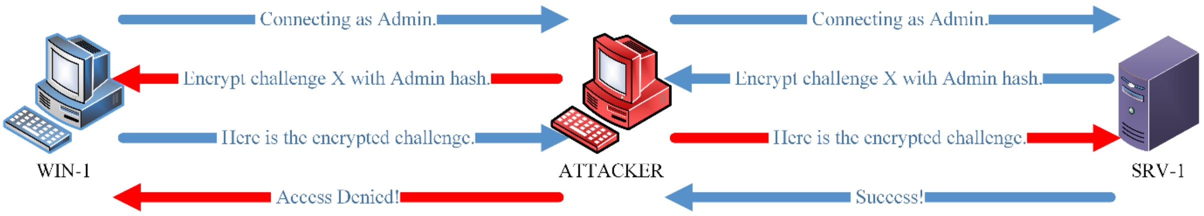

Connecting as Admin.
Connecting as Admin.
Encrypt challenge X with Admin hash.
Encrypt challenge X with Admin hash.
Here is the encrypted challenge.
Here is the encrypted challenge.
WIN-1
ATTACKER
SRV-1
Access Denied!
Success!

FIGURE 5: The attacker relays authentication between the victim and the target. Once the attacker successfully connects to the target server, the attacker disconnects the victim from the connection [8].

RESPONDER/MULTI-RELAY: PYTHON PENETRATION TESTING TOOL

Responder/Multi-Relay (Responder) is a Python-based pentest tool for poisoning LLMNR, NBNS, and MDNS with rouge authentication support for capturing Windows credentials [9]. The pentest tool comes pre-packaged in Kali Linux and is a fork of SpiderLabs' Responder tool. The main script used to listen and respond to LLMNR/NBNS requests is *Responder.py*, located in the */usr/share/responder* directory in Kali. Responder can perform poison attacks for credential harvesting or facilitate SMB-Relay. The major components that define and control Responder are:

a. *Responder.py*: the main script responsible for running the LLMNR/NBNS/MDNS service listeners, poisoners, and enabled rouge authentication servers to handle redirected client connections and authentication attempts.
b. *Responder.conf*: configures what rouge authentication services are enabled, log file locations, and finetuning options for the tool.
c. *Responder.db*: SQLite3 database file for storing poisoning attempts (Poisoned Table) and captured user credentials, both cleartext and hashed (Responder Table).
d. Poisoner scripts *LLMNR.py*, *MDNS.py*, and *NBTNS.py*: they are imported into *Responder.py* and calculate and send a response to requests received by one of Responder's LLMNR, NBNS, or MDNS service listeners.
e. Server scripts (*FTP.py, HTTP.py, SMB.py*): used for emulation of services and facilitating credential harvesting. Captured credentials are recorded to the *Responder.db* database.
f. The *tools/MultiRelay.py* SMB-Relay script: used to perform SMB-Relay. Using *MultiRelay.py* in conjunction with *Responder.py* requires the Responder's SMB server to be set to *OFF* in *Responder.conf* [9].
g. The *tools/MultiRelay/bin* executables: uncompiled executables *Runas, Syssvc,* and *Mimikatz*/*Mimikatzx86*; they should be compiled before running *MultiRelay.py*.

MultiRelay.py, the second central script in the Responder tool, facilitates SMB-Relay after LLMNR/NBNS poisoning. If successful, a remote shell opens, and the executables placed in the *tools/MultiRelay/bin* directory are uploaded to the target. *MultiRelay.py,* used in conjunction with the Responder toolset, facilitates everything a pentester needs to demonstrate and exploit LLMNR/NBNS and gain access to a target through SMB-Relay. The tool can easily be adapted for more nefarious uses by threat actors as any tool can. Next, mitigations are proposed to render both attacks impossible to deploy and use using Responder/Multi-Relay or any other custom attack tool.

MITIGATIONS: DISABLE LLMNR/NBNS AND REQUIRE SMB SIGNING

The possibility of LLMNR/NBNS poisoning and SMB-Relay can both be minimized by disabling the legacy name resolution services and forcing all SMB connections to require SMB signing. Below, **Table 1.** shows recommended options to modify each service and eliminate the vulnerabilities related to the attack.

TOPIC: AITM Poison Windows Name Resolution Services for SMB-Relay

Vulnerability	***Solution***	***Option***	***Settings***	***Description***
LLMNR Service Enabled	Enable Group Policy Admin Template	*Computer Configuration > Administrative Templates > Network > DNS Client > Turn off multicast name resolution*	Not Configured Enabled (**0**) Disabled (**1**)	Creates registry key *DNSClient* and adds value *EnableMulticast* when configured.
NetBIOS Services (NBNS) Enabled	Turn off manually via Interface Properties	*Internet Protocol Version 4 (TCP/IPv4) > Advanced > WINS > NetBIOS setting*	Default (**0**) Enable (**1**) Disable (**2**)	Explicitly control the use of NetBIOS and extensions (see LLMNR and NBNS Ubiquity section).
	Turn off via Windows DHCP Server Role	*DHCP > [Server's Name] > IPv4 > Server Options > Configure Options > Advance > Windows Options (Vendor Class) > 001 Microsoft Disable Netbios Option*	**0x1** (Enabled) **0x2** (Disabled)	Control use of NetBIOS from DHCP server.
SMB2 Security Signing Not Required	Enable Group Policy Security Setting for SMB Client	*Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options > Microsoft network client: Digitally sign communications (always)*	Enabled (**1**) Disabled (**0**)	Enforce security signing on client.
	Enable Group Policy Security Setting for SMB Server	*Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options > Microsoft network server: Digitally sign communications (always)*	Enabled (**1**) Disabled (**0**)	Enforce security signing on the server.

TABLE 1: Windows vulnerabilities for LLMNR/NBNS poisoning and SMB-Relay and their recommended mitigations [2], [7], [10].

CONCLUSION

Windows computers that use the alternative legacy name services LLMNR and NBNS are vulnerable to LLMNR/NBNS poisoning attacks when attempting name resolution to non-existent or off-link hosts. If exploited, an attacker can redirect connections to their system and facilitate an attack on Windows systems called SMB-Relay as an adversary-in-the-middle (AitM). An attacker uses the victim by running a rouge SMB authentication service and relaying a victim's authentication attempt to a target system. If successful, the attacker can disconnect from the victim and gain complete control over the target system.

By default, Windows has LLMNR/NBNS enabled for legacy and backward compatible name resolution if other Windows peers cannot use DNS. Both services are based on multicast/broadcast communications to flood the link-local with name queries to find hosts on the local network. Any host that claims authority over a name can respond. The vulnerability in both services is a lack of authentication and verification of responders. This vulnerability means any device on the link-local can respond. Mainly, if a name queried for does not exist, the opportunity for a malicious system to respond as authoritative exists.

Because tools like Responder/Multi-Relay exist to exploit this and the possibility for other attack tools to be developed by attackers, disabling LLMNR/NBNS services is necessary if they are not needed, and enabling SMB signing eliminates the possibility for SMB-Relay.

REFERENCE

[1] P. Bramwell, Hands-on penetration testing on Windows: Unleash Kali Linux, PowerShell, and Windows debugging tools for security testing and analysis. 2018. Accessed: Jan. 31, 2022. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1860845

[2] S. M. in M. Security, S. Recommendation, and T. Reference, "Securing Windows Workstations: Developing a Secure Baseline," Active Directory Security, Oct. 21, 2016. https://adsecurity.org/?p=3299 (accessed Feb. 06, 2022).

[3] J. L. Carrell, E. Tittel, and J. Pyles, Eds., Guide to TCP/IP: IPv6 and IPv4, Fifth edition. Boston, Massachusetts: Cengage Learning, 2016.

[4] windows-driver-content, “NetbiosOptions.” https://docs.microsoft.com/en-us/windows-hardware/customize/desktop/unattend/microsoft-windows-netbt-interfaces-interface-netbiosoptions (accessed Feb. 16, 2022).

[5] L. Esibov, D. Thaler, and B. D. Aboba, "Link-local Multicast Name Resolution (LLMNR)," Internet Engineering Task Force, Request for Comments RFC 4795, Jan. 2007. doi: 10.17487/RFC4795.

[6] "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods," Internet Engineering Task Force, Request for Comments RFC 1001, Mar. 1987. doi: 10.17487/RFC1001.

[7] "Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay, Sub-technique T1557.001 - Enterprise | MITRE ATT&CK®." https://attack.mitre.org/techniques/T1557/001/ (accessed Jan. 31, 2022).

[8] "SANS Penetration Testing | SMB Relay Demystified and NTLMv2 Pwnage with Python | SANS Institute." https://www.sans.org/blog/smb-relay-demystified-and-ntlmv2-pwnage-with-python/ (accessed Feb. 06, 2022).

[9] lgandx, Responder/MultiRelay. 2022. Accessed: Feb. 04, 2022. [Online]. Available: https://github.com/lgandx/Responder

[10] Deland-Han, "Overview of Server Message Block signing - Windows Server." https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/overview-server-message-block-signing (accessed Feb. 06, 2022).