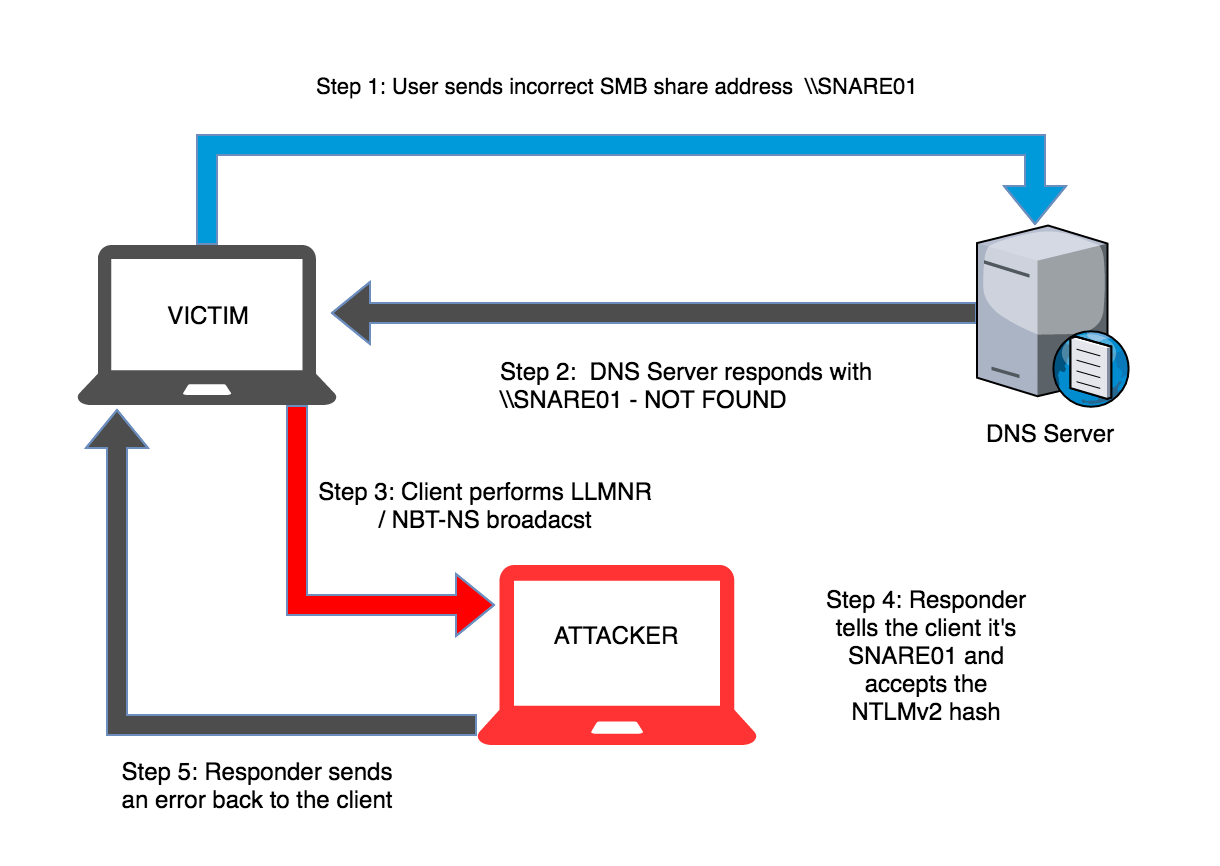
LAB – Exploiting Active Directory Using LLMNR/NBT-NS Poisoning

**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 Window’s components are used for host identification/name resolution.

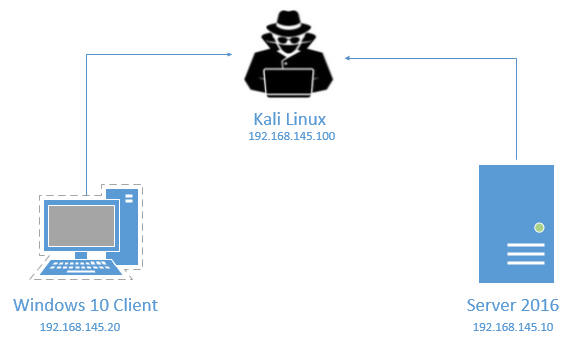


1. The victim machine wants to connect to a share at \\SHARE01 but mistakenly types in \\SNARE01.
2. The DNS server responds to the victim, saying that it does not know that host.
3. The victim sends out a broadcast asking if anyone on the local network knows the location of \\SNARE01
4. The attacker 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**

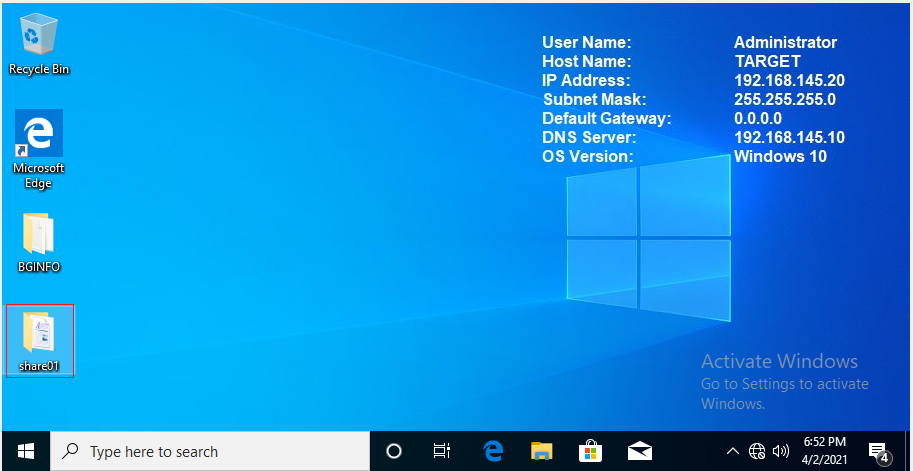
1. Virtual install of Kali Linux.
2. One virtual install of Windows 10 Pro or Enterprise.
3. One virtual install of Server 2012, 2016, or 2019.
4. Server must be a domain controller running Active Directory.
5. Windows 10 must be a member of the domain.

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.



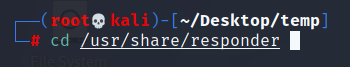
**Begin the lab!**

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

****

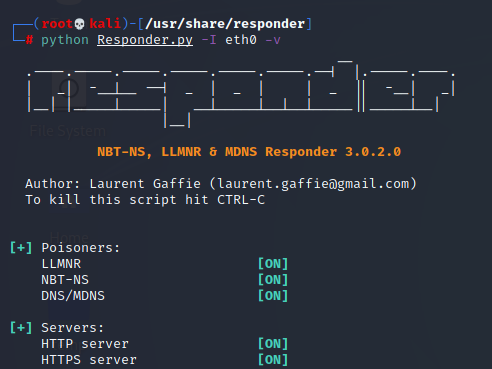
**Launch Responder**

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**

****

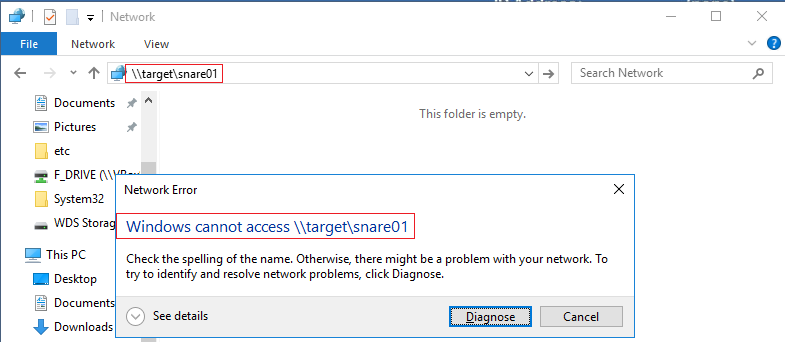
Once inside the working directory, I launch the application using the following command.

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

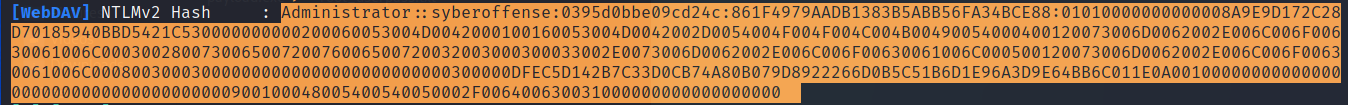


The -v is for verbose and allows me to see the captured hash repeatedly.

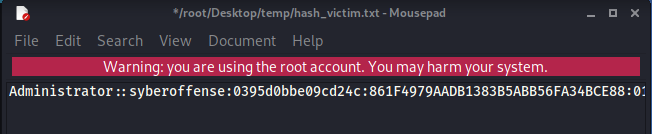
From my Windows Server 2016 DC1, I attempt to connect to the share located on the Windows 10 machine, but I type in the share name wrong. This casing the connection to fail, and Server 2106 sends out a broadcast asking if anyone knows where this share is located.



When the Server 2016 machine sends its credentials, responder is listening and captures the sent username and password.



We next copy everything highlighted in orange into a text file using any text editor we choose. In this example, I used Mousepad. I saved the file as hash\_victim.txt.

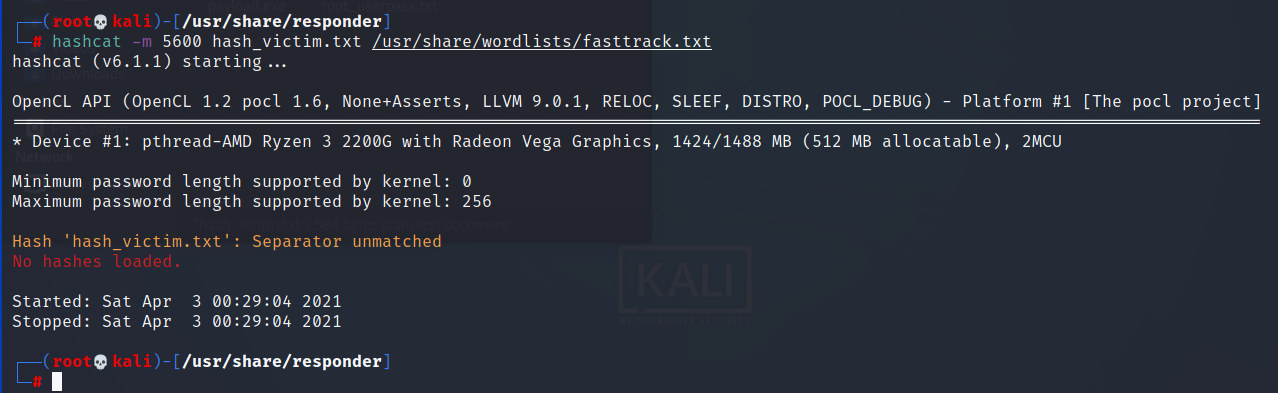


**Using John to Crack the Hash**

Once we have captured the hashed credentials, we need to decipher or crack the hash. For this part of the demonstration, I will be using John.

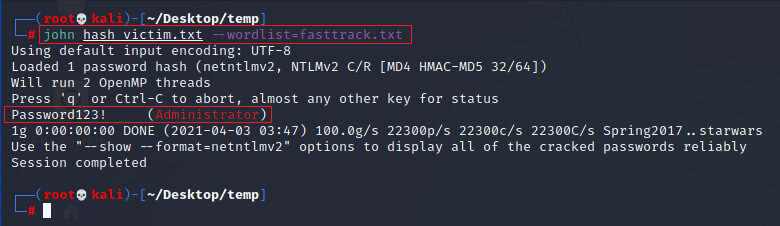
**Caveat**

I tried using Hashcat, but there is a known issue with the program that prevents it from being able to encode the hash.



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

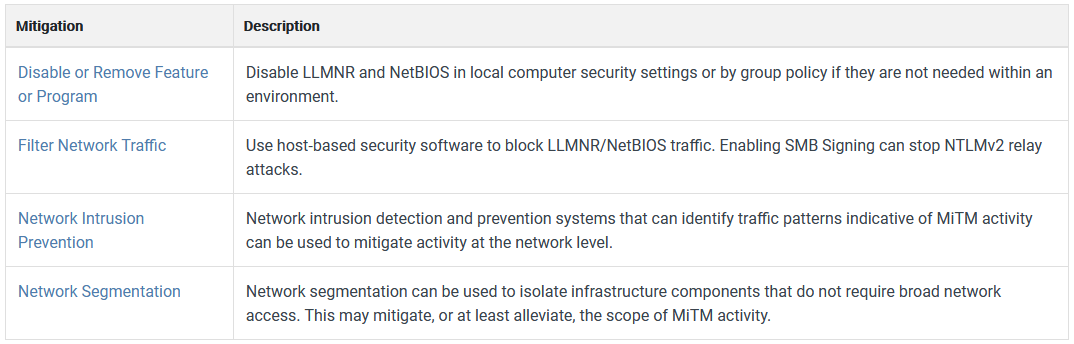
I switched over to John and was able to crack the hash with ease.



**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:



End of Lab!