John David Watts

COMP 6970

Final Project

12/12/2019

Responder Report

**Overview**

Responder is a tool that allows us to use LLMNR, NBT-NS, and MDNS poisoning. What this means is that we can use an LLMNR and NBT-NS Spoofing attack against a network. This sort of attack takes advantage of default Windows configurations in order to achieve its end goal.

It is important to understand what a LLMNR and NBT-NS server broadcast is in order to understand how this kind of attack works. When a DNS server request fails, Microsoft Windows systems use Link-Local Multicast Name Resolution (LLMNR) and the Net-BIOS Name Service (NBT-NS) for a “fallback” name resolution. This poses a huge threat as if the DNS name is not resolved, then the client (aka the victim in this scenario) performs and unauthenticated UDP broadcast to the network asking all other systems if it has the name that it is looking for. We can see now why this is a problem as this entire process is unauthenticated and broadcasted to the entire network. This allows any machine on the network to respond and claim to be the target machine.

Now that we understand the background of this process, we will proceed to dive deeper on just how an LLMNR and NBT-NS Poisoning Attack works. First, the attacker must be actively listening for LLMNR and NetBIOS broadcasts and if this is the case, then it can hide itself on the network to proceed onto pretending as the machine that the victim wants to connect to. Once the attacker accepts the connection from the spoofed machine, we can then use this spoofed machine (our attacking machine pretending to be who the victim is looking for) to run the Responder tool and forward on the request to a rouge service that performs the authentication process. While this authentication is taking place, the client will send the spoofed machine a NTLMv2 hash for the user that it is trying to authenticate. If we can capture this hash, it can be cracked offline of the network with a few of the tools that we have learned this semester such as: Hashcat or John the Ripper. A figure of this entire process is shown below to aid your understanding of what kind of attack we are going to perform with the responder tool.

A screenshot of a cell phone

Description automatically generated

Basic attack where a user mistypes the server name

**Basic Demonstration**

We will now show a basic attack with the Responder tool using the Kali Machine (.10) against the Windows Machine (.201). For this demonstration, we assume that you have the version that is already installed on the Kali Machine(.10).

The first step in our process is to go ahead and get Responder running on our attack machine. We can do this by running the command: *responder -I eth0 -wrFb*

A screenshot of this first step working properly is shown below.

A screenshot of a cell phone

Description automatically generated

Once we have Responder up and running on our attack machine, we can navigate over to our Windows 7 victim machine (.201) and open up the File Explorer. Once here we can click on the top toolbar and enter in ‘\\abc’ to simulate a user tying the wrong SMB server name. Once the user types in the wrong server name, the DNS lookup fails and therefore our attack begins. One we have pressed the ‘Enter’ key after typing this command in the toolbar we can see our Kali Machine with Responder running in the background begins to execute its attack and the only thing we are prompted to do is enter in a username and password on the Windows 7 machine but it does not matter if we do or not because our attack has already taken place.

A screenshot of a cell phone

Description automatically generated

A screen shot of a computer

Description automatically generated

Navigating over back over to our Kali Machine and into the ‘/usr/share/responder/logs/ directory we can see that we have generated a new file called ‘*SMBv2-NTLMv2-SSP-192.168.150.201.txt’*. Looking at this file using the cat command, we can see that it contains a long hash. By using either hashcat or john the ripper, we can crack this hash to therefore obtain the username and password to the system.

A screenshot of a cell phone

Description automatically generated

**Detecting an Attack**

For this process we are going to use the lab machines to show that an attack against the Windows 7 machine (.200) from the Kali Machine (.10). As stated previously we are going to use the responder tool in Kali Linux to perform an ‘Man-in-the-Middle’ attack by intercepting the traffic flow from a bad DNS server call from the Windows 7 machine. Once we send an LLMNR or a NETBIOS broadcast from the Kali Machine, the Windows 7 machine will accept this broadcast. Once this broadcast has been accepted, our attacker will grab a file named ‘*SMBv2-NTLMv2-SSP-192.168.150.201.txt’* in which we can decrypt in order to see the username and passwords.

Now that we know what exactly will happen on the network, we can easily see how our IDS needs to be implemented in order to help prevent this attack. To prevent this attack, all we need to check for is if the source IP address is not the source IP addresses of the DNS server or the Windows 7 machine which we will already know as we are familiar with what network we are on. If this source IP address is not the Windows 7 machine (192.168.x.201) or the DNS server we are trying to connect to, then we need to check what source is sending packets. For this instance, our behavioral IDS checks to see that the source equals 192.168.x.201, if it does not match, then we send a message to the user saying that there is an issue. These checks are done on both NBNS protocols and LLMNR protocols as shown below.

A screenshot of a cell phone

Description automatically generated

IDS code for Responder attacks

A screenshot of a social media post

Description automatically generated

NBNS protocol traffic

A screenshot of a cell phone

Description automatically generated

LLMNR protocol traffic

As you can clearly see, we have detected all of attempted ‘Man-in-the-Middle’ attacks from our Kali Machine. It is important to note that we must check both of these protocols as if LLMNR is disabled, the next default Windows systems server naming convention is the NBNS protocol.

**Final Recommendation for Prevention**

Our final recommendation to prevent these kinds of attacks is fairly easy. We recommend disabling both LLMNR and NETBIOS Name Service on your Microsoft Windows system. Do note that if you only disable LLMNR, then Windows by default will use NETBIOS Name Server for resolution and therefore still have your machine at risks. Examples of this process is shown below.

A screenshot of a social media post

Description automatically generated

LLMR disabling

A screenshot of a cell phone

Description automatically generated

NBT-NS disabling

**Sources Used**

* <https://www.notsosecure.com/pwning-with-responder-a-pentesters-guide/>
* <https://tools.kali.org/sniffingspoofing/responder>
* <https://forums.kali.org/showthread.php?36036-Penetration-Testing-How-to-use-Responder-py-to-Steal-Credentials>
* <https://www.4armed.com/blog/llmnr-nbtns-poisoning-using-responder/>