IT Security Principles:

Windows Exploitation

IT 444 – Network Security

Understanding LLMNR and NBNS

- Windows systems go through several different steps to resolve a hostname to an IP address for us.
- Windows will search the *hosts* or *LMHosts* file on the system to see if there's an entry in that file.
- If there isn't, then the next step is to query **DNS**. Windows will send a DNS query to the default nameserver to see if it can find an entry.
- In most cases, this will return an answer, and we'll see the web page or target host we're trying to connect to.
- In situations where DNS fails, modern Windows systems use two protocols to try to resolve. **LLMNR** and **NetBios**

Understanding LLMNR and NBNS

- **LLMNR**: this protocol uses multicast in order to try to find the host on the network. Other Windows systems will subscribe to this multicast address, and when a request is sent out by a host, if anyone listening owns that name and can turn it into an IP address, a response is generated. Once the response is received, the system will take us to the host
- If the host can’t be found using LLMNR, Windows use the **NetBIOS** protocol to try to discover the IP. It does this by sending out a broadcast request for the host to the local subnet, and then it waits for someone to respond to that request. If a host exists with that name, it can respond directly, and then our system knows that to get to that resource, it needs to go to that location

Understanding LLMNR and NBNS

- Both LLMNR and NBNS rely on *trust*
- As a malicious actor, though, we can respond to any request sent out to LLMNR or NBNS and say that the host being searched for is owned by us.
- Then when the system goes to that address, it will try to negotiate a connection to our host, and we can gain information about the account that is trying to connect to us

Understanding Windows NTLMv1 and NTLMv2 Authentication

- There are a number of ways in which systems can authenticate, such as via Kerberos, certificates, and NetNTLM
- NetNTLM provides a safer way of sending Windows NT LAN Manager (NTLM) hashes across the network
- Before Windows NT, LAN Manager (LM) hashes were used for network-based authentication
- One of the weaknesses of the LM hash was that it was actually two separate hashes combined together
- A password would be converted to uppercase and padded with null characters until it reached 14 characters, and then the first and second halves of the password would be used to create the two portions of the hash

NTLM

- With NTLM, the RC4 algorithm was used for generating the hash
- This is vastly more secure for host-based authentication, but there’s an issue with network-based authentication
- If someone is listening and we’re just passing raw NTLM hashes around
- The NetNTLMv1 and NetNTLMv2 challenge/response hashes were created to give additional randomness to the hashes and make them slower to crack

NTLM

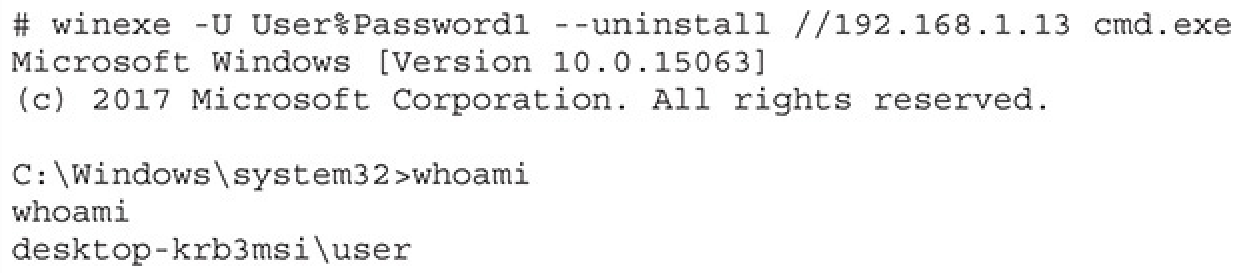
- NTLMv1 uses a server-based nonce to add to the randomness.
  - When we connect to a host using NTLMv1, we first ask for a nonce.
  - Next, we take our NTLM hash and re-hash it with that nonce.
  - Then we send that to the server for authentication. If the server knows the NT hash, it can re-create the challenge hash using the challenge that was sent.
  - If the two match, then the password is correct
- The problem with NTLM is that a malicious attacker could trick someone into connecting to their server and provide a static nonce

NTLMv2

- NTLMv2 provides two different nonces in the challenge hash creation.
  - The first is specified by the server, and the second by the client.
  - Regardless of whether the server is compromised and has a static nonce, the client will still add complexity through its nonce, thus ensuring that these credentials crack more slowly

Remote administration tool for Windows systems

- Using **Winexe** is a common way for attackers to access remote systems.
  - It uses named pipes through the hidden IPC share on the target system to create a management service.
  - Once that service is created, we can connect to it and call commands as the service
  - Winexe for remote access



```
# winexe -U User%Password1 --uninstall //192.168.1.13 cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
desktop-krb3msi\user
```