What should we do?

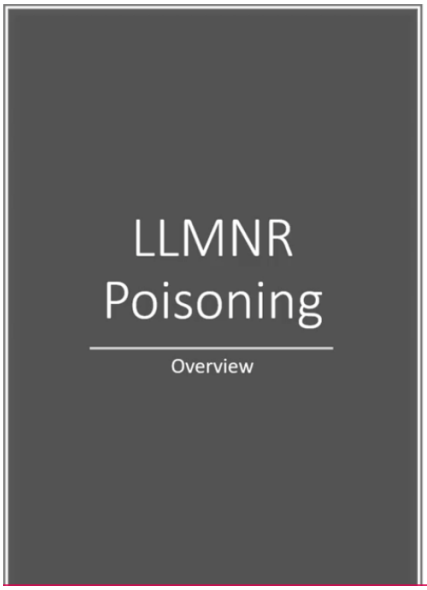
Here, We are ready with everything as setting up lab and etc

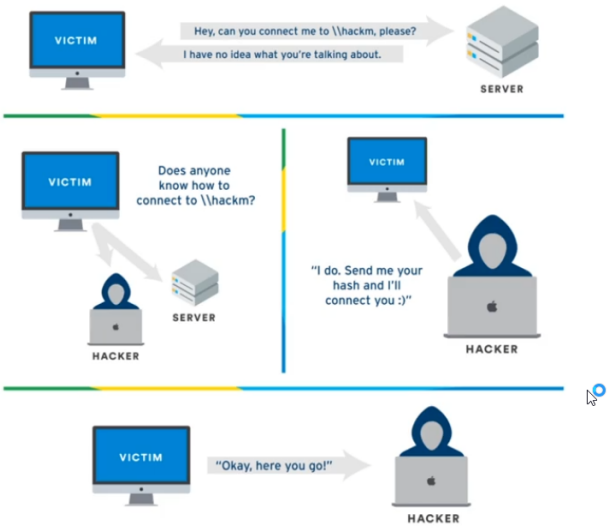
Guide: https://medium.com/@adam.toscher/top-five-ways-i-got-domain-admin-on-your-internal-network-before-lunch-2018-edition-82259ab73aaa

LLMNR Poisoning

What is LLMNR?

Link Local Multicast Name resolution it identifies hosts when DNS fails and its previoisly known as NBT-NS (Netbios)
When the server responds to us it responds with a username and NTLM password hash





When a user try to connect to a wrong server . Since its wrong DNS cant get it and LLMNR takes it into action and then we will be sniffing and be the MITM and then we respond like" I have that server so just send me your username and password hash I will connect to you(to user)"" so then user says okay take my username and password hash. And then we decode the hash to know the password

Tools Used:

Responder.py

Its is a part of Impacket tool kit and also we use this to capture the ntlmv2 hashes
Usage: python Responder.py -I <interface> -rdw

Hashcat

Usage: .\hashcat.exe -m 5600 hash_file wordlist

'-m' → Method of hahing used

hash_file → File where hash was stored

wordlist_file → All guessing passwords in a Wordlist

Proctection

Disabling LLMNR/NBT-NS

LLMNR: Just select "Turn OFF Multicast Name Resolution" under Local Computer Policy> Computer Configuration > Administrative Templates >

Network > DNS Client in the group policy ediitor. NBT-NS: Navigate to Network Connections > Network Adapter properties > TCP/IPv4 Properties> Advanced Tab > WINS tab and select " Disable NetBIOS over TCP/IP".

Without Disabling LLMNR/NBT-NS

Require Network Access Control This helps to allow a device to access your network and its MAC address gets displayed Require strong user passwords :
Rules :

* More than 14 characters of password
* no common words in password
* Dont re use the passwords
* Dont keep same password for a long time

These help because it takes more time and even some times impoossible to crack the password

SMB Relay

What is SMB Relay?
Instead of cracking hashes gathered with Responder, we can instead relay those hashes to specific machines and potentially gain access

Requirements

- SMB signing must be disabled on the target

- Relayed user credentials must be admin on machine

→ Config file

Now it looks like

.

→ Command when running

Tools

ntlmrelayx.py

```
root@kali:/opt/impacket/examples# python ntlmrelayx.py -tf targets.txt -smb2support
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server
```

Usage:

```
python ntlmrelayx.py -tf targets.txt -smb2support
```

nmap

Usage:

```
nmap --script=smb2-security-mode.nse -p445
<ip_address/CIDR>
```

'-p' - Port numbers
--script - can mention any nmap scripts (frm nmap_script_engine)

Verification

If u can see *message signing enabled and required* it was not vulnerable (if not it is mostly vulnerable)

Defense:

Enable SMB Signing on all devices

- Pro: Completely stops the attack
- Con: Can cause performance issues with file copies

Disable NTLM Authentication on network

- Pro: Completely stops the attack
- Con: If Kerberos stops working, Windows defaults back to NTLM

Account Tiering:

- Pro: Limits domain admins to specific tasks (e.g. only log onto severs with need for DA)
- Con: Enforcing the policy may be difficult

Local admin restriction:

- Pro: Can prevent a lot of lateral movement
- Con: Potential increase in the amount of service desk tickets

IPV6 DNS Take Over via mitm6 LDAP !

Tools:

mitm6

Usage:
`mitm6 -d domain.local`
'-d' for Domain Controller

ntlmrelayx.py

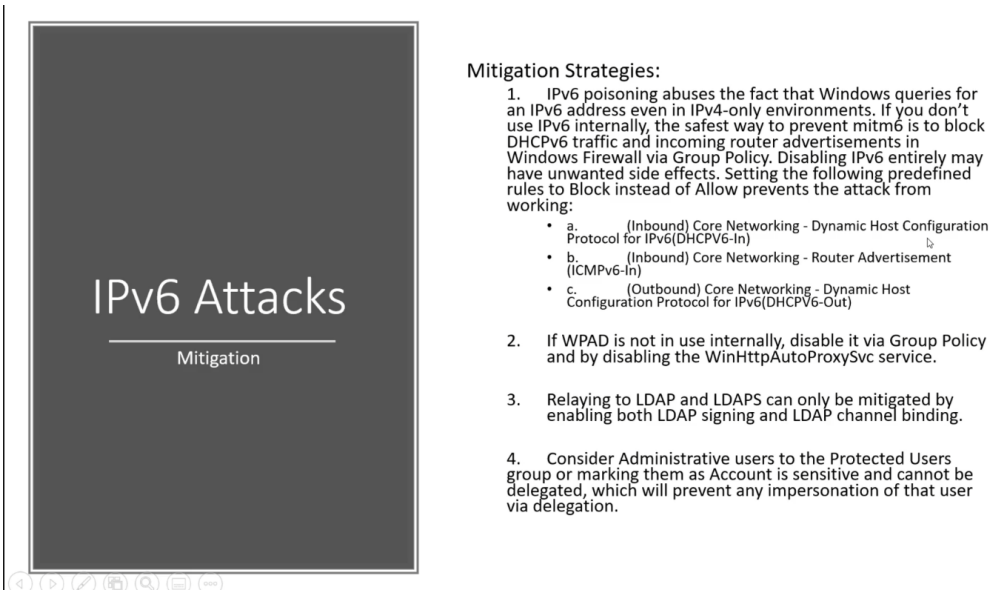
Usage:
`ntlmrelayx.py -6 -t ldaps://<ip_of_domain_controller> -wh fakewpad.domain.local -l lootme`
'-6' for IPv6
-t for target
ldaps - ldap secured
-wh for wpad_host

- wpad is Web Proxy Auto Discovery Protocol used by clients to locate the URL configuration file usiing DHCP or DNS discovery methods
  -l for loot_dir where looted SAM files are stored
  Now we can get domain controllers all info like description and all other users stuff into lootme directory

https://blog.fox-it.com/2018/01/11/mitm6-compromising-ipv4-networks-via-ipv6/
https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/

Defending From IPv6 Attacks

- Turn of IPv6 in domain controller if you are personally not using it by blocking DHCPv6 traffic , blockin gincoming router advertisements in WIndows Firewall via Group Policy. Disabling IPv6 directly may have unwanted side effects. So setting the predefined rules to block instead of Allow prevents the attack from working:
  a. Core Networking IPv6 protocol - Dynamic Host COnfiguration Protocol for IPv6
  b. Core Networking - Router Advertisment
  c. Core Networking - Dynamic Host Configuration Protocol for IPv6
- If WPAD id not in use internally, disable it via Group Policy and by disabling the WinHttpAutoProxySvc Service
- Relaying to LDAP and LDAPS can only be migated by enabling both LDAP signing and LDAP channel binding
- Consider Administrative users to the Protected Users group or marking them as Account is sensitive and cannot be delegated, which will prevent any impersonation of that user via delelgation.



Other Attack Vectors and Strategies

Strategies:

- Begin with mitm6 or responder at 8am

- Run Scans to generate traffic
- If scans are taking too long, look for websites in scope
  (http_version in msfconsole)
  Look for default credentials on web logins
- Printers and check whether it has admin privilege and smb running for a scan
- Jenkins
- Etc