Pentesting Lab

Active Directory

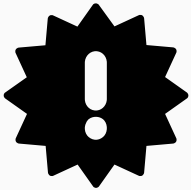
Felix, Possegger, Pongratz, Prodinger, Schauklies, Schwarzl

24.03.2025

Summer 2025, www.isec.tugraz.at/ptl

Table of contents

Introduction



- Since this is a pentesting course, we will focus on attack points on Active Directory
- Therefore, many concepts and technical details will be wildly simplified and/or omitted!
- However, you should know what you attacked and why it worked, especially in the submission reviews.
- If you want to learn more, check out this Microsoft Learn Path:
- https://learn.microsoft.com/en-us/training/paths/active-directory-domain-services/

What is Active Directory?

- Active Directory is a directory service for (Windows) domain networks.
- It is a collection of ”Roles” that can be assigned to one or many Windows Servers:
  - Active Directory Domain Services
  - Active Directory Federation Services
  - Active Directory Certificate Services
  - There are more!
- It is based on standard technologies
  - LDAP (Lightweight Directory Access Protocol)
  - Kerberos
  - DNS (Domain Name System)
  - SMB (Server Message Block)

A Brief History

- Released with Windows 2000 Server edition
- Support retrofitted back to Windows 95
- Features and security have been greatly enhanced since then
- Still needs to be backwards compatible
- That's were the problems start:
  - Old Operating Systems do not support modern encryption
  - They only support insecure protocols
  - They cannot work with modern Hashing-Algorithms
  - etc.

Why (pentest) Active Directory?

- About 90% of Fortune 1000 companies use Active Directory[1]
- You are almost guaranteed to encounter it in an internal pentest
- Active Directory can do everything an administrator needs
- But does everyone know how to configure everything properly?
- Very hard to do everything right
  - A single mistake can lead to disaster
  - There are checkboxes that (if checked) lead to instant domain compromise for every user!

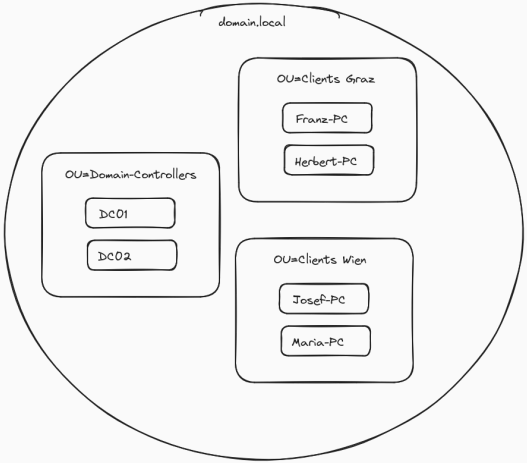
[1] https://www.frost.com/frost-perspectives/active-directory-holds-the-keys-to-your-kingdom-but-is-it-secure/

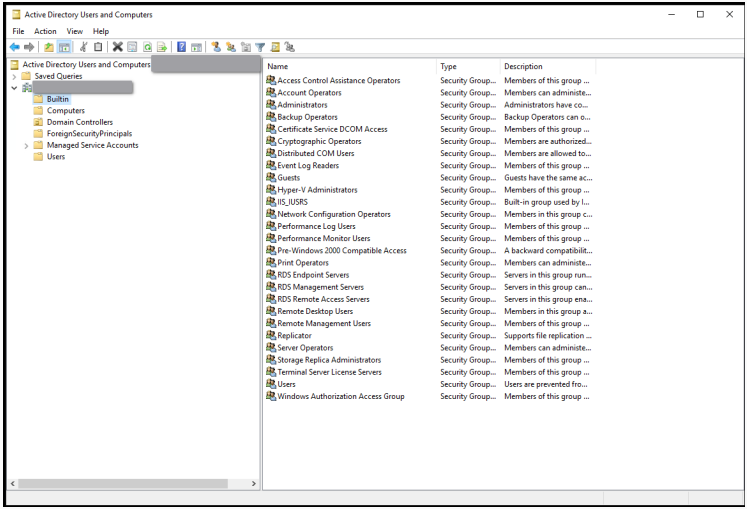
The Core Structure

Domains and Forests

- Domain(s)
  - A logical grouping of network objects (users, computers, groups)
  - Establishes boundaries and ACLs
  - Organizational Units (OUs)
    - Hierarchically managed containers
    - Grouping similar assets together (e.g. Client-Workstations)
- Forest(s)
  - Group of Domains
  - Sharing a common schema and configuration

Domains and Forests




Active Directory Users and Computers
File Action View Help
Active Directory Users and Computers
Saved Queries
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Managed Service Accounts
Users
Name
Type
Description
Access Control Assistance Operators
Security Group... Members of this group ...
Account Operators
Security Group... Members can administe...
Administrators
Security Group... Administrators have co...
Backup Operators
Security Group... Backup Operators can o...
Certificate Service DCOM Access
Security Group... Members of this group ...
Cryptographic Operators
Security Group... Members are authorized...
Distributed COM Users
Security Group... Members are allowed to...
Event Log Readers
Security Group... Members of this group ...
Guests
Security Group... Guests have the same ac...
Hyper-V Administrators
Security Group... Members of this group ...
IIS_IUSRS
Security Group... Built-in group used by I...
Network Configuration Operators
Security Group... Members in this group c...
Performance Log Users
Security Group... Members of this group ...
Performance Monitor Users
Security Group... Members of this group ...
Pre-Windows 2000 Compatible Access
Security Group... A backward compatibilit...
Print Operators
Security Group... Members can administe...
RDS Endpoint Servers
Security Group... Servers in this group run...
RDS Management Servers
Security Group... Servers in this group can...
RDS Remote Access Servers
Security Group... Servers in this group ena...
Remote Desktop Users
Security Group... Members in this group a...
Remote Management Users
Security Group... Members of this group ...
Replicator
Security Group... Supports file replication ...
Server Operators
Security Group... Members can administe...
Storage Replica Administrators
Security Group... Members of this group ...
Terminal Server License Servers
Security Group... Members of this group ...
Users
Security Group... Users are prevented fro...
Windows Authorization Access Group
Security Group... Members of this group ...

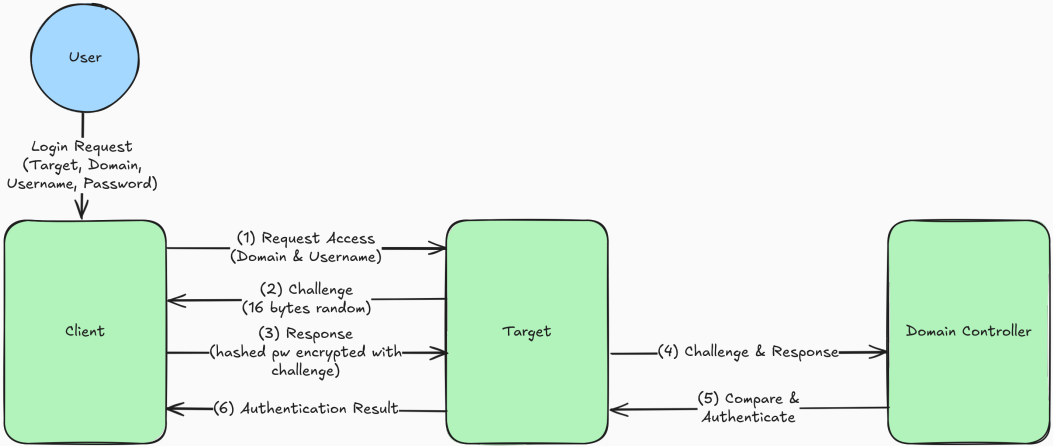
Roles

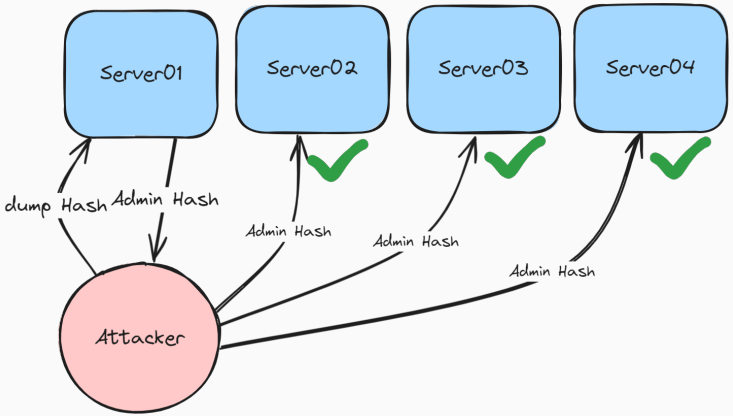
- Active Directory Domain Services
  - Hosted on a Domain Controller
  - Users & Groups
  - Organizational Units
  - Group Policies
  - Access Control
  - etc.
- Active Directory Certificate Services
  - Should be a different server (sometimes it's hosted on a DC, bad!)
  - Certificate Management
  - Issuing certs based on templates
  - Certificates are used for Encryption, Signing and Authentication

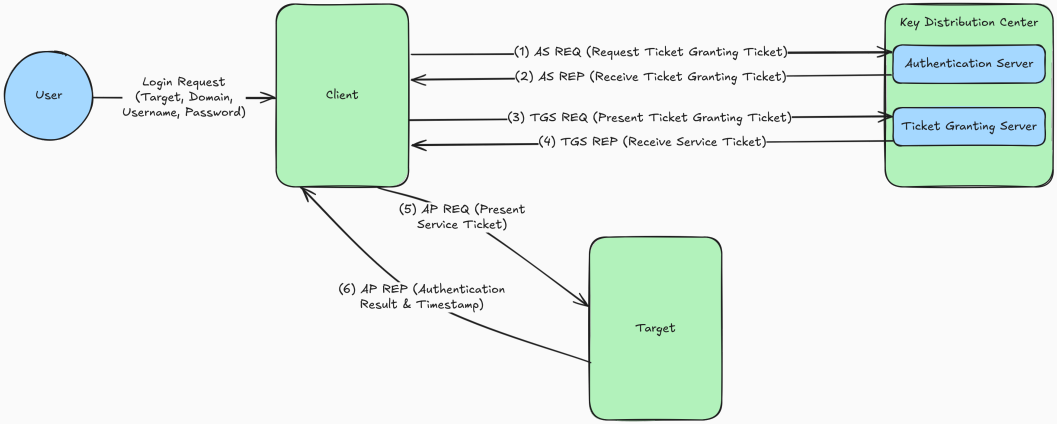
Core Services

- LDAP
  - Accessing the Directory Services
  - Common interface for all relevant Active Directory queries
  - Ports: 389/tcp (LDAP) or 636/tcp (LDAPS)
- DNS
  - Resolving Domain Names to IP Addresses
  - Crucial for a working environment (Kerberos, Certificates etc.)
  - But there is a fallback mechanism...
  - Ports: 53/udp
- SMB
  - Used for fileshares and remote administration
  - Tightly connected and required for Group Policies and Startup Scripts
  - Ports: 137-139/tcp & 445/tcp

- Either via NTLM (legacy)
- Or Kerberos (modern)
- NTLM is considered insecure and allows lots of attacks
- However, it is still widely in use today.
- Kerberos is more modern but has some problems too


User
Login Request
(Target, Domain,
Username, Password)
Client
(1) Request Access
(Domain & Username)
(2) Challenge
(16 bytes random)
(3) Response
(hashed pw encrypted with
challenge)
(6) Authentication Result
Target
(4) Challenge & Response
(5) Compare &
Authenticate
Domain Controller


Server01
Server02
Server03
Server04
dump Hash
Admin Hash
Admin Hash
Admin Hash
Admin Hash
Attacker


User
Login Request
(Target, Domain,
Username, Password)
Client
(1) AS REQ (Request Ticket Granting Ticket)
(2) AS REP (Receive Ticket Granting Ticket)
(3) TGS REQ (Present Ticket Granting Ticket)
(4) TGS REP (Receive Service Ticket)
Key Distribution Center
Authentication Server
Ticket Granting Server
(5) AP REQ (Present
Service Ticket)
(6) AP REP (Authentication
Result & Timestamp)
Target

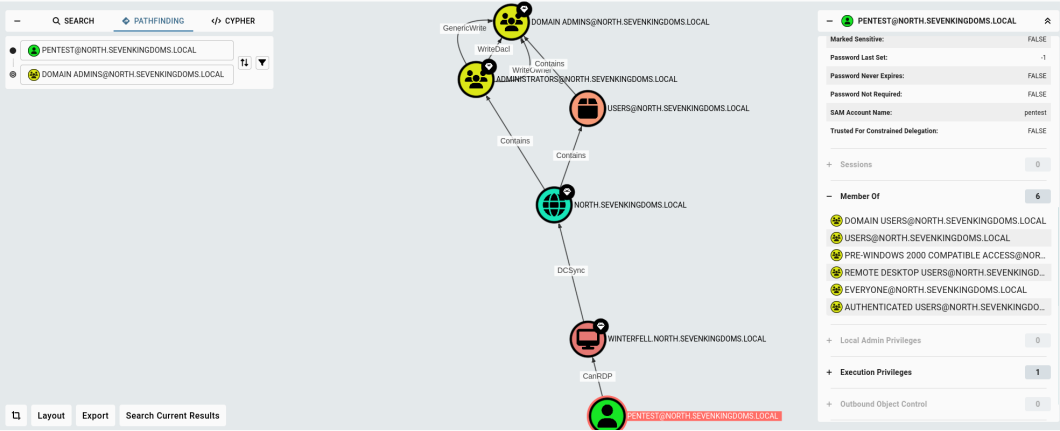
Tooling

- You can do a lot of attacks and recon with built-in Microsoft Tools
- This can be time-consuming however
- In a real pentest, you are probably going to be too slow
- There are some tools you should be familiar with when offensively working with AD
- Know how they work, their output and their limitations

- NetExec is the swiss army knife for pentesting Active Directory Environments
- It allows authentication, information gathering and code execution over multiple channels
- It can run vulnerability scans, enumerate targets, dump credentials and deploy your C2
- https://github.com/Pennyw0rth/NetExec
- https://www.netexec.wiki/getting-started/installation/installation-on-unix

- It's the de-facto standard for (offensively) enumerating AD relationships
- Writes all necessary information into a neo4j DB which makes it extremely easy to search
- https://github.com/SpecterOps/BloodHound
- https://github.com/SpecterOps/SharpHound/releases

BloodhoundAD: A Graph



- Active-Directory Auditing Tool
- Shows you a health score of the general AD Environment
- Very useful for identifying major misconfigurations
- https://github.com/netwrix/pingcastle

Indicators

Domain Risk Level: 65 / 100

It is the maximum score of the 4 indicators and one score cannot be higher than 100. The lower the better

Compare with statistics

Privacy notice

Stale Object : 31 /100 — 6 rules matched

It is about operations related to user or computer objects

Trusts : 1 /100 — 1 rules matched

It is about connections between two Active Directories

Privileged Accounts : 40 /100 — 4 rules matched

It is about administrators of the Active Directory

Anomalies : 65 /100 — 14 rules matched

It is about specific security control points

Reconnaissance

- Goal: Find as much information as possible
- Tooling
  - BloodhoundAD[2]
  - PowerView[3]
  - Certipy[4]
  - Snaffler[5]
  - Kerbrute[6]
  - Get-GPPPassword[7]

[2] https://github.com/SpecterOps/BloodHound
[3] https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1
[4] https://github.com/ly4k/Certipy
[5] https://github.com/SnaffCon/Snaffler
[6] https://github.com/ropnop/kerbrute
[7] https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Get-GPPPassword.ps1

- Does our user have local admin rights?
- Can we connect via RDP to another machine?
- Passwords where they shouldn't be?
  - Group Policies
  - User / Computer descriptions
  - File Shares
  - AD-Attributes
- Users with weak passwords?
- Any old systems with known vulnerabilities?

Misconfigurations in User / Group Relations

- BloodhoundAD is perfect for this!
- You can find out quickly if your user has indirect control of another object
- Example:
  - You just pwned a helpdesk user
  - Helpdesk users have the ability to reset passwords of other users
  - Now you can reset the password of an IT-Administrator
  - Use their account to connect to a server and run mimikatz to gather even more passwords
  - If a Domain-Admin had a session on this server, you just pwned everything

- There are a lot of places for (almost) cleartext passwords to be stored in AD
- Group Policies can store AutoLogon passwords which can be decrypted
  - Use Get-GPPPassword.ps1[8]
- Some administrators are not aware that descriptions can be read by everyone

```
nxc ldap <hostname> -u <user> -p <pass> -M get-desc-users
```

- Other attributes can store passwords as well:

```
nxc ldap <hostname> -u <user> -p <pass> -M get-unixUserPassword -M
    getUserPassword
```

[8] https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Get-GPPPassword.ps1

- Usually, there is a lockout policy for number of password attempts
- Instead of trying many passwords for one user...
- we are going to use one password for many users!
- Users (and Admins!) tend do use guessable passwords
  - username = password
  - Summer2025!
  - 'Company'1234!
  - Init01!
  - etc.

Spraying Passwords

- Start by acquiring a list of domain users:

```
nxc ldap <hostname> -u <user> -p <pass> --active-users > active.txt
tail active.txt -n+5 | awk -F ' ' '{ print $5 }' > domain_users.txt
```

- Highly recommended: read password policy:

```
nxc smb <hostname> -u <user> -p <pass> --pass-pol
```

- Then, use kerbrute to spray your passwords:

```
kerbrute passwordspray -d <domain> domain_users.txt Winter2022
kerbrute passwordspray -d <domain> domain_users.txt --user-as-pass
```

- Most companies use some sort of knowledgebase
- Searching through those is recommended for every engagement
- Snaffler[9] automates this process for shares
- But sometimes you may want to do it more manually
- From PowerView.ps1[10], you can use Check-ShareAccess:

```
Find-DomainShare -CheckShareAccess
```

- And then search through them manually
- Also search local filesystems of servers / workstations (e.g. C:\tmp)

[9] https://github.com/SnaffCon/Snaffler
[10] https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1

Spoofing / Coercion

- Goal: Get other systems to authenticate to us
- Tooling
  - Responder[11]
  - Inveigh[12]
  - mitm6[13]
  - Powermad[14]
  - Coercer[15]

[11] https://github.com/lgandx/Responder
[12] https://github.com/Kevin-Robertson/Inveigh
[13] https://github.com/dirkjanm/mitm6
[14] https://github.com/Kevin-Robertson/Powermad
[15] https://github.com/p0dalirius/Coercer

- LLMNR / NBNS Spoofing
- Adding a DNS Wildcard
- Create a fake DHCPv6 Server that provides a fake DNS Server
- Use Print Spooler / other RPC calls to force remote authentication
- Crack the captured hashes or **relay** them

- Lots of legacy protocols still in use
- LLMNR (Link Local Multicast Name Resolution) / NBNS (NetBIOS Name Service) are multicast without any authentication
- Windows queries various protocols for name resolution:
  - Local hosts file
  - DNS-Server
  - LLMNR / NBNS
- Anyone can answer!

- Sometimes, the AD-DNS Server allows creation of DNS Records
- This is useful for machines to add their own name
- Sometimes, all users or even ”Anonymous” can add records
- You can add wildcard entries
- Further reading: https://www.netspi.com/blog/technical-blog/network-pentesting/exploiting-adidns/

- By default, Windows (since Vista) prefers IPv6 to IPv4
- If a network does not provide a DHCPv6 server...
- ...become one yourself!
- By becoming the preferred DHCP, you can set a preferred DNS server too
- Further reading: https://redfoxsec.com/blog/ipv6-dns-takeover/

- There are several methods you can use to get a Computer (for example the Domain Controller) to connect to another system
  - PetitPotam[16]
  - PrinterBug[17]
  - DFSCoerce[18]
  - There are many more!
- You'll usually want the target to connect to you though...

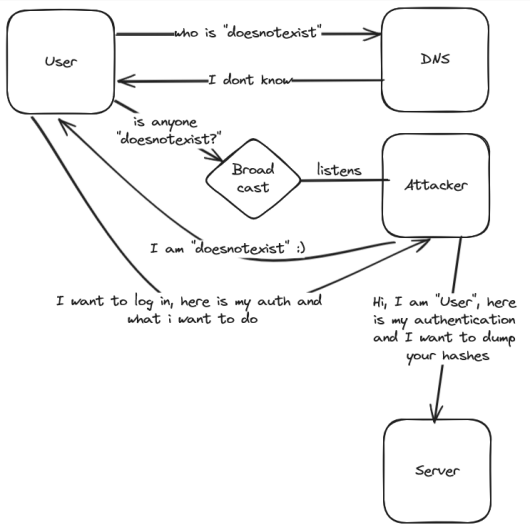
[16] https://www.prosec-networks.com/blog/petit-potam-ntlm-relay-angriff/
[17] https://www.thehacker.recipes/ad/movement/mitm-and-coerced-authentications/ms-rprn
[18] https://www.bleepingcomputer.com/news/microsoft/new-dfscoerce-ntlm-relay-attack-allows-windows-domain-takeover/

NTLM Relaying

- Goal: Don't want to crack NTLM Authentication Responses? Relay them!
- Tooling
  - ntlmrelayx[19]
  - LdapRelayScan[20]
  - Inveigh[21]

[19] https://github.com/fortra/impacket/blob/master/examples/ntlmrelayx.py
[20] https://github.com/zyn3rgy/LdapRelayScan
[21] https://github.com/Kevin-Robertson/Inveigh

- If you manage to get an authentication request from another system / user, you can use the ”authentication” part with a different ”payload”
- Relay authentication from higher-privileged accounts
- You can relay to many services:
  - SMB: Allows code execution if account is an administrator
  - LDAP: Allows reading / writing LDAP Attributes
  - HTTP: Attack Certificate Services (or Exchange etc.)
- Further Reading:
  https://trustedsec.com/blog/a-comprehensive-guide-on-relaying-anno-2022


User
who is "doesnotexist"
DNS
I dont know
is anyone "doesnotexist?"
Broad cast
listens
Attacker
I am "doesnotexist" :)
I want to log in, here is my auth and what i want to do
Hi, I am "User", here is my authentication and I want to dump your hashes
Server

Attacking Active Directory Certificate Services

- Goal: Exploit various misconfigurations in ADCS to gain elevated privileges
- Tooling
  - certipy[22]
  - Certify[23]
- There are various documented misconfigurations in Certificate Templates that allow attacks
- They are dubbed ESC1 - ESC15, some are easy to exploit, some pretty hard

[22] https://github.com/ly4k/Certipy
[23] https://github.com/GhostPack/Certify

Attack Scenarios

- ESC1
  - A user can enroll a certificate and specify a custom UPN
  - This allows them to create a certificate that is valid for anyone they choose (like ”Administrator@domain.local”)
  - You can authenticate with a valid certificate, giving you instant Domain-Admin rights
- ESC4
  - A user can edit a Certificate Template, allowing them to enable ESC1
- ESC8
  - NTLM Relay to the Certificate Service HTTP endpoint
  - Remember Coercion? You can coerce a Domain Controller to connect to you, then relay their connection to the HTTP-Endpoint, and get a Certificate for a Domain Controller!

Attack Scenarios cont.

- The other ESCs are a little more advanced and would take quite some time to explain
- ESC1 - ESC8: https://posts.specterops.io/certified-pre-owned-d95910965cd2
- ESC9 and ESC10: https://research.ifcr.dk/certipy-4-0-esc9-esc10-bloodhound-gui-new-authentication-and-request-methods-and-more-7237d88061f7
- ESC11: https://blog.compass-security.com/2022/11/relaying-to-ad-certificate-services-over-rpc/
- ESC12: https://pkiblog.knobloch.info/esc12-shell-access-to-adcs-ca-with-yubihsm
- ESC13: https://posts.specterops.io/adcs-esc13-abuse-technique-fda4272fbd53
- ESC14: https://posts.specterops.io/adcs-esc14-abuse-technique-333a004dc2b9
- You can find a good overview here: https://www.thehacker.recipes/ad/movement/ad-cs

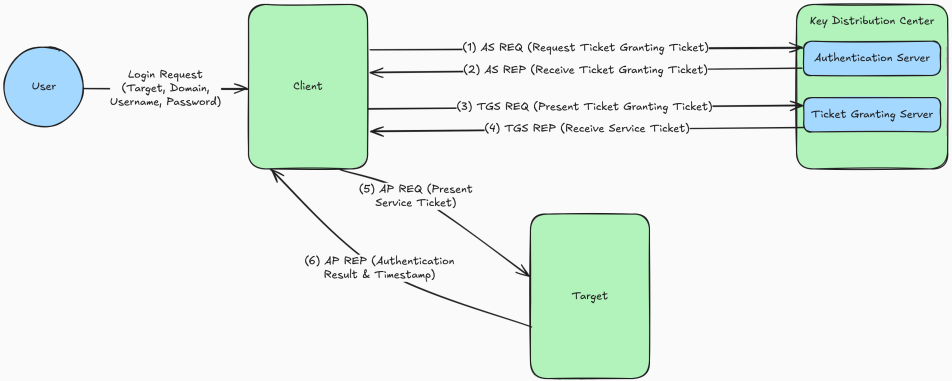
Attacks on Kerberos

- Goal: Exploit features in NTLM's successor, Kerberos
- Tooling
  - Rubeus[24]
  - Mimikatz[25]
  - hashcat[26]

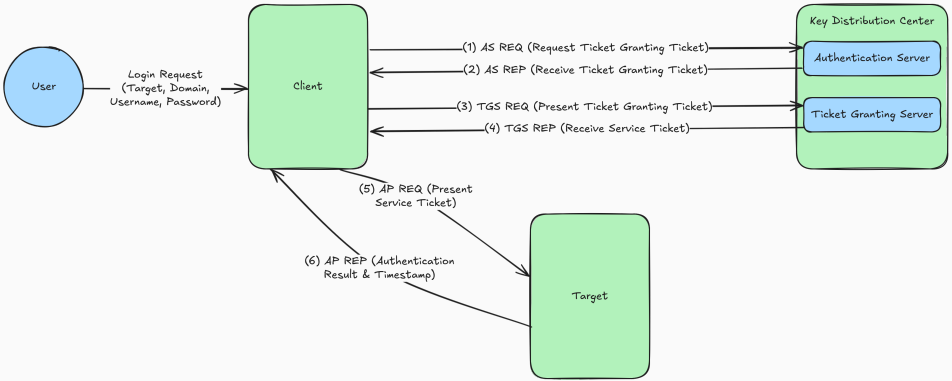
[24] https://github.com/GhostPack/Rubeus
[25] https://github.com/gentilkiwi/mimikatz
[26] https://hashcat.net/hashcat/

- Kerberoasting / AS-REP Roasting
- Constrained / Unconstrained Delegation

- You can send a TGS-REQ (Service Ticket Request) for any Service-Account (SPN) in the domain
- You need a TGT first, so you need to do the AS-REQ stuff as a Domain-User first
- The TGS-REP you get back is encrypted with the SPN's hash
- This hash can be cracked offline, although it is pretty hard


User
Login Request
(Target, Domain,
Username, Password)
Client
Key Distribution Center
Authentication Server
Ticket Granting Server
(1) AS REQ (Request Ticket Granting Ticket)
(2) AS REP (Receive Ticket Granting Ticket)
(3) TGS REQ (Present Ticket Granting Ticket)
(4) TGS REP (Receive Service Ticket)
(5) AP REQ (Present
Service Ticket)
(6) AP REP (Authentication
Result & Timestamp)
Target

- This works very similar to Kerberoasting
- The AS-REQ contains a username, the desired service to access, and a timestamp encrypted with the user's password
- The Authentication Service then checks if it can decrypt the timestamp using the password hash the Domain-Controller has stored
- For users that have a special flag (do not require pre-authentication) set, however, you can skip the whole timestamp stuff
- Which means, you do NOT need a valid domain user password for this attack!
- However, you do need to know the username of the account you want to request a TGT for
- If you manage to do that, you can try to crack the encrypted password stored in the TGT you received


User
Login Request
(Target, Domain,
Username, Password)
Client
Key Distribution Center
Authentication Server
Ticket Granting Server
(1) AS REQ (Request Ticket Granting Ticket)
(2) AS REP (Receive Ticket Granting Ticket)
(3) TGS REQ (Present Ticket Granting Ticket)
(4) TGS REP (Receive Service Ticket)
(5) AP REQ (Present
Service Ticket)
(6) AP REP (Authentication
Result & Timestamp)
Target

- Thankfully, you don't have to actually understand it:

```
nxc ldap <hostname> -u <user> -p <pass> --kerberoasting output.txt
hashcat -m 13100 -a 0 -O output.txt <wordlist.txt>

nxc ldap <hostname> -u <user> -p '' --asreproast output.txt
hashcat -m 18200 -a 0 -O output.txt <wordlist.txt>
```

- A system with ”Unconstrained Delegation” enabled will store tickets in memory
- Which means, if you gain administrative rights on such a system, you can dump and use saved tickets
- You can now Coerce another System to authenticate to this Unconstrained Delegation System, and use its ticket
- (This is very similar to NTLM relaying attacks)
- Further reading: https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/domain-compromise-via-unrestricted-kerberos-delegation

Case Study

Showing a PingCastle report and checking vulnerabilities.

PLAB{h0w_t0_h4ck_m1cr0s0ft??}