Post-Compromise Attacks

Tuesday, July 9, 2024 4:35 AM

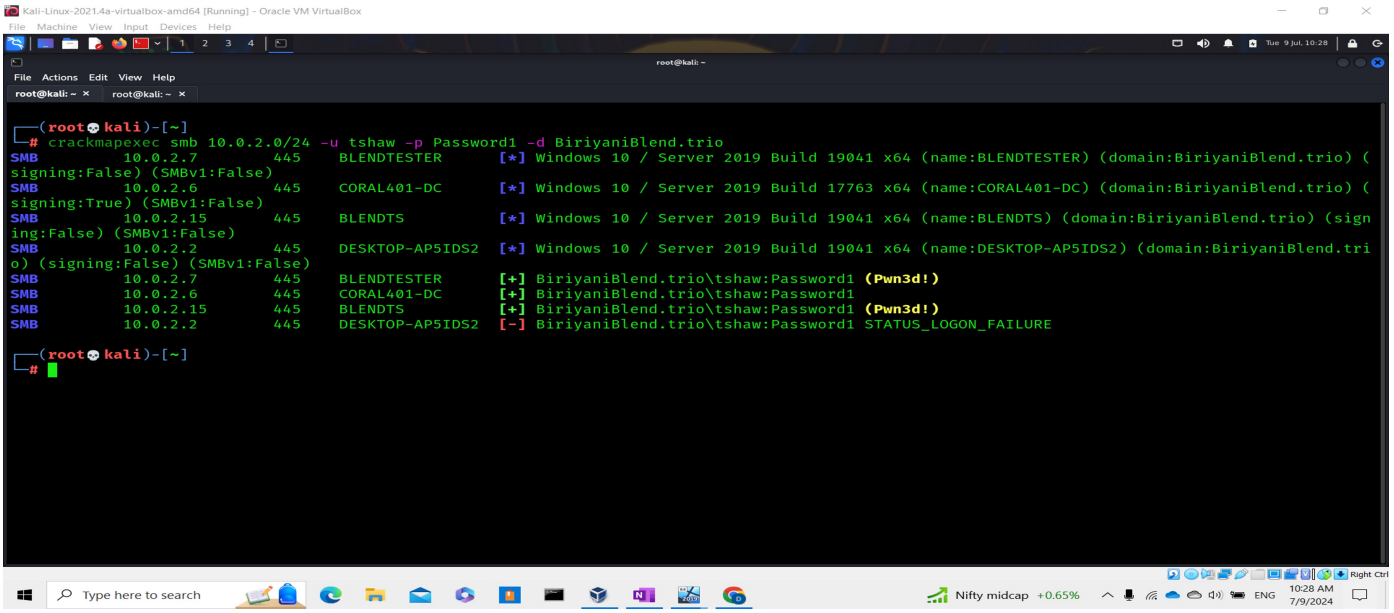
1) Pass The Hash/ Pass The Password:

We have a valid username and password that we captured in our earlier attacks now in this stage of post compromise attack we will use the username and the password of the domain user and pass it around in the entire subnet using a tool called 'crackmapexec' you see how many devices in the network use same user and password and try to dump SAM hashes and further we will pass those hashes and try to compromise as many devices as we can in the entire domain network.

The idea here is very simple, what we are trying to achieve is once we have a access on SAM hashes of local account users we can do two things with that, the first is we can take those hashes offline and try to crack those hashes using tools like hashcat or john and get cleartext password of those users which can be used further to access different machine within the Active Directory domain through 'Pass The Password' attack and we can also make a comment on the password policy of the domain based on how many hashes we are able to crack.

Second we can directly use those hashes and pass them around within the entire subnet and see if we can pawn them and this process will facilitate a lateral movement the Active Directory domain network which could further be escalated into getting a domain admin access through attacks like token impersonation etc. also if we get the hashes offer local administrator we can use to pass it around to see if we can get admin access on different devices.

Step 1: Execute crackmapexec to sweep the entire domain network.

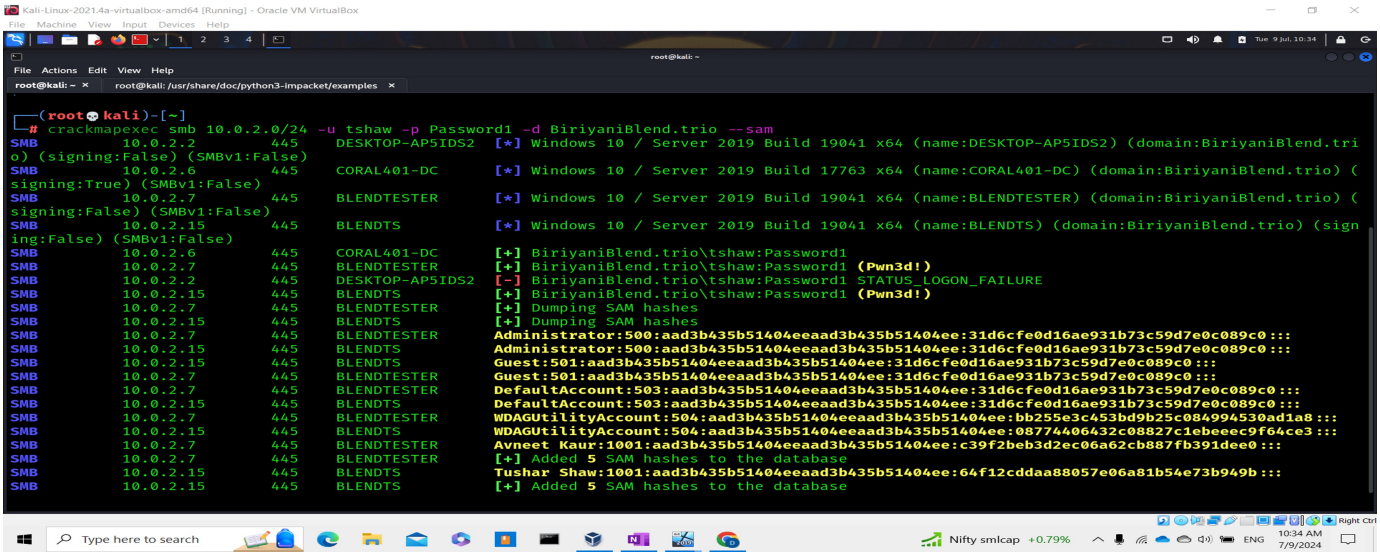


smb: service used for connection and authentication
10.0.2.0/24: CIDR network address
-u: username
-p: password
-d: fully qualified domain name

As we can see we have successfully pwned BLENDTESTER and BLENDTS which means domain user 'tshaw' is a local admin on both the machines. Hence we are all sent to dump the hashes from them.

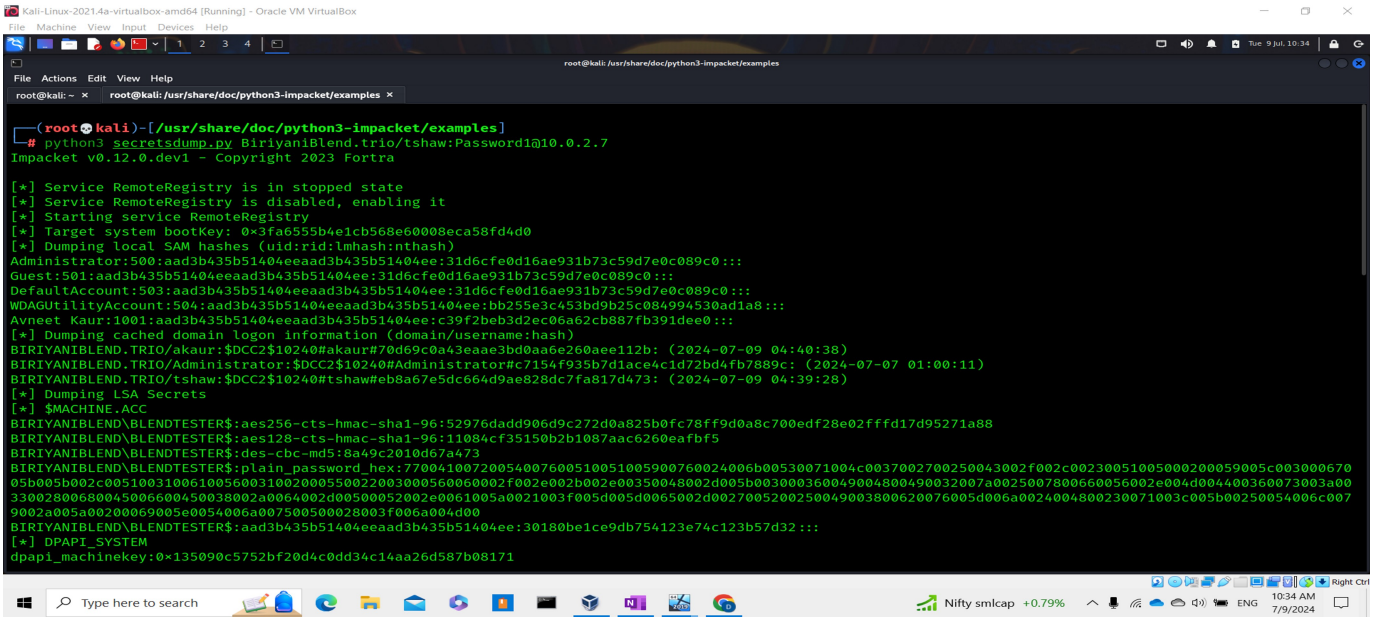
We can move ahead to dump hashes with 'crackmapexec.py' or we can use another impacket tool which is 'secretsdump.py'.

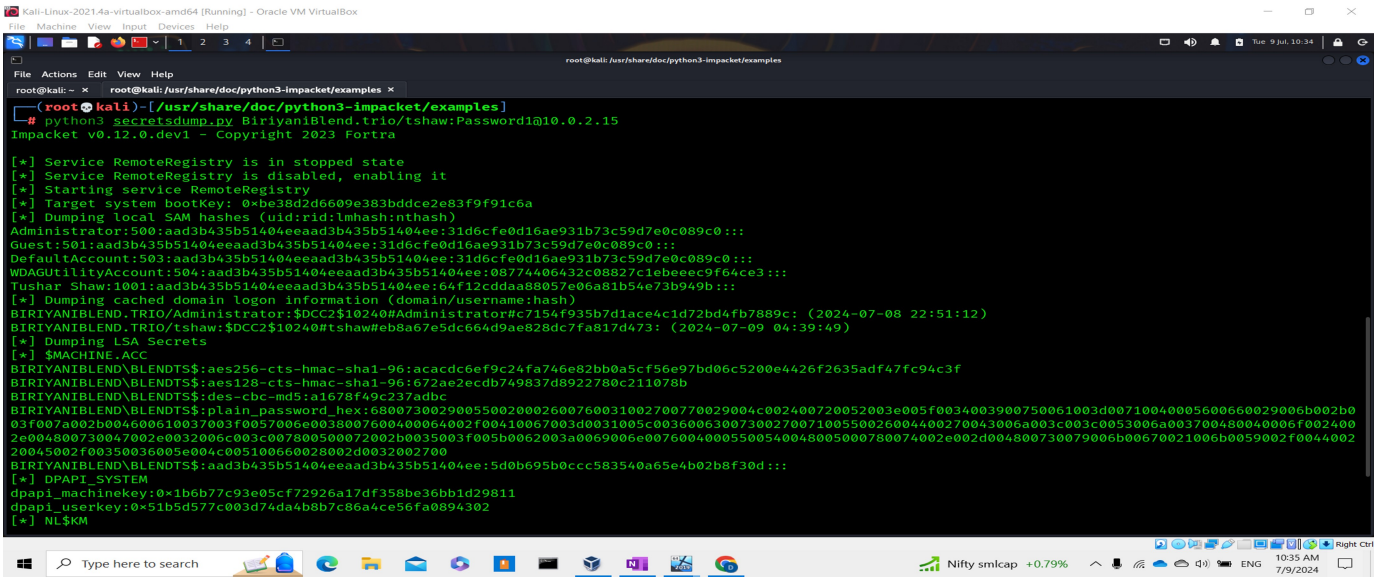
Below is the screenshot of dumping ntlm hashes of local users from the sam file on these domain devices



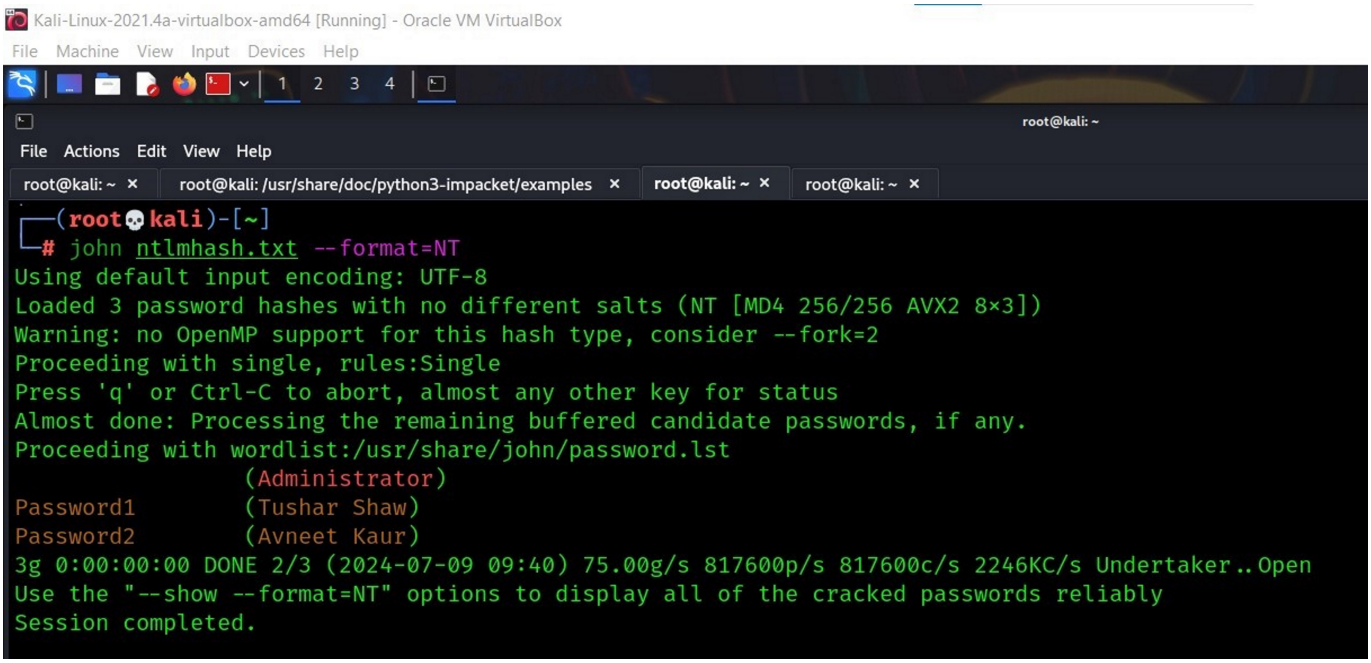
--sam: flag used to dump hashes from crackmapexec.py.

And below are the screenshots of the tool secretsdump.py to dump hashes from BLENDTESTER and BLENDTS respectively.



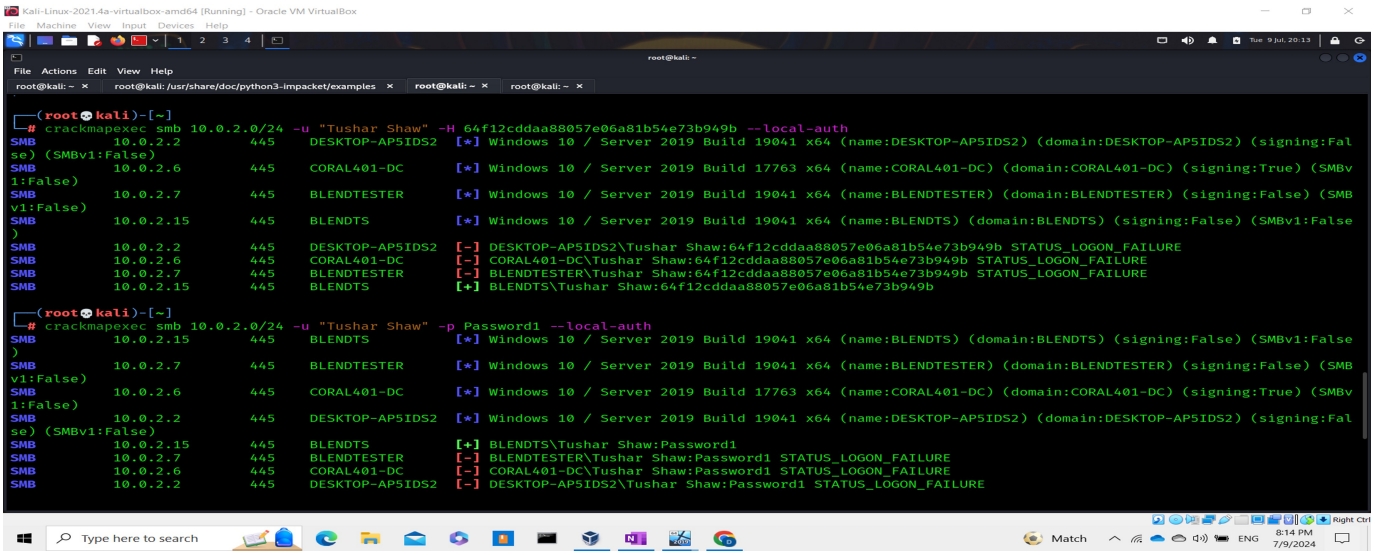


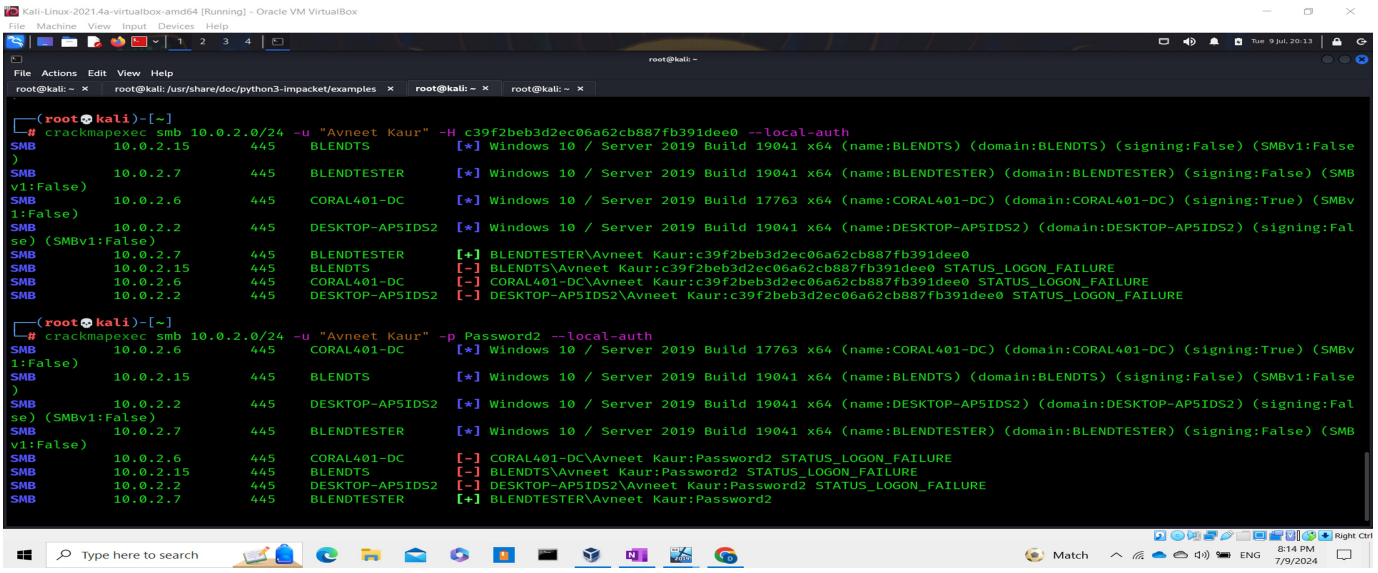
Now we are interested in user and administrator accounts, which are Tushar Shaw, Avneet Kaur and Administrator. We will put there hashes in a separate file called and ntlmhash.txt and try cracking it using 'hashcat'/'john'. In the mean time we will pass these hashes using crackmapexec.py to compromise more machine in the domain network.



Above screenshot shows 'john' has prompted us back with the password of both users and we can say that the password policy in this domain network is week.

Below is the screenshot of using crackmapexec.py to pass the hash locally in the domain.





-H: flag is for the NT hash of users password
--local-auth: flag to authenticate locally
In this case we can see apart from respective devices we are not able to get successful authentication anywhere. and that's completely fine, In an actual pentest environment we can also go ahead with the administrator's hash and se if we can get any successful login and admin access.

And another tool that we can use here is 'psexec.py', to login into the devices in the domain network using <LM:NT> hash

```
┌──(root💀kali)-[/usr/share/doc/python3-impacket/examples]
└─# python3 psexec.py "Tushar Shaw":@10.0.2.7 -hashes aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[-] SMB SessionError: code: 0×c000006d - STATUS_LOGON_FAILURE - The attempted logon is invalid. This is either due to a bad username or authentication inform
ation.

┌──(root💀kali)-[/usr/share/doc/python3-impacket/examples]
└─# python3 psexec.py "Tushar Shaw":@10.0.2.15 -hashes aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b                    1 x
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on 10.0.2.15.....
[-] share 'ADMIN$' is not writable.
[-] share 'C$' is not writable.
[*] Found writable share Share
[*] Uploading file joBfqgwD.exe
[*] Opening SVCManager on 10.0.2.15.....
[-] Error opening SVCManager on 10.0.2.15.....
[-] Error performing the installation, cleaning up: Unable to open SVCManager
```

So the whole objective of this attack it's too find security access management dumps and move laterally in the domain expecting that will find a user local computer having sam dump of a user having an account on the domain controller as well(part of admin group as well) and hence we will get access on the domain controller, this is one of the easiest way of accessing the domain controller and getting admin access over it.

Mitigation:

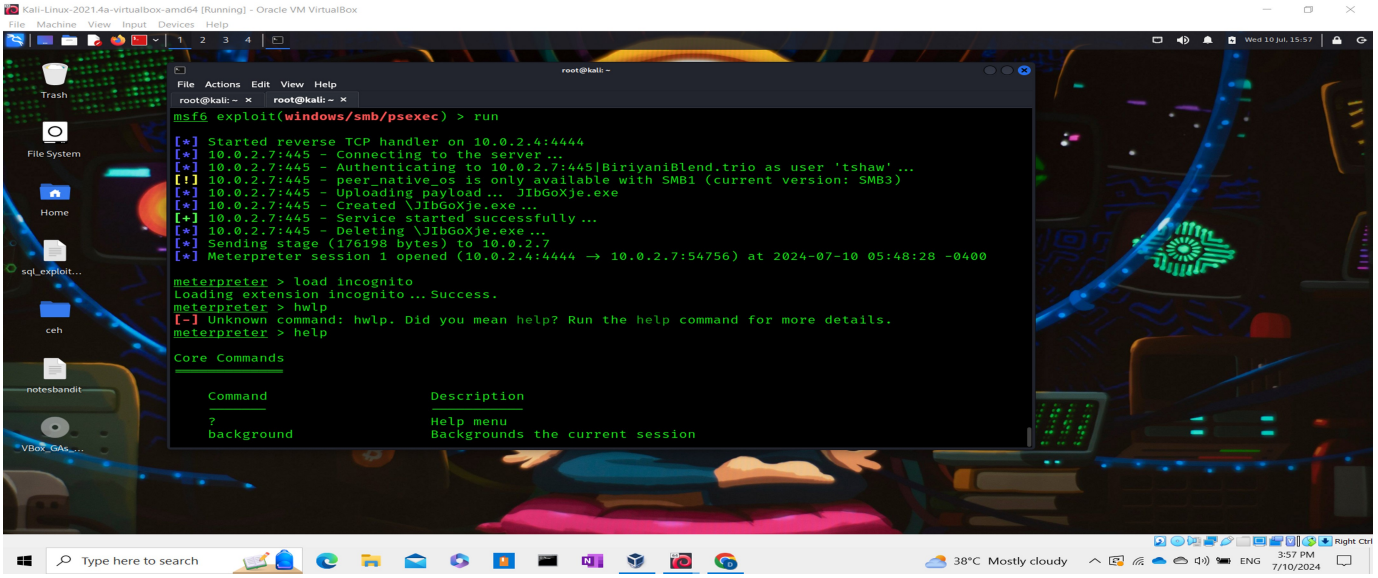
Hard to completely prevent but we can make it more difficult on an attacker:

- *Limit account re-use:*
  - *Avoid re-using local admin password*
  - *Disable Guest and Administrator accounts*
  - *Limit who is a local administrator (least privilege)*
- *Utilize strong passwords:*
  - *The longer the better (>14 characters)*
  - *Avoid using common words*
  - *I like long sentences*
- *Privilege Access Management (PAM)*
  - *Check out/in sensitive accounts when needed*
  - *Automatically rotate passwords on check out and check in*
  - *Limits pass attacks as hash/password is strong and constantly rotated*

2) Token Impersonation

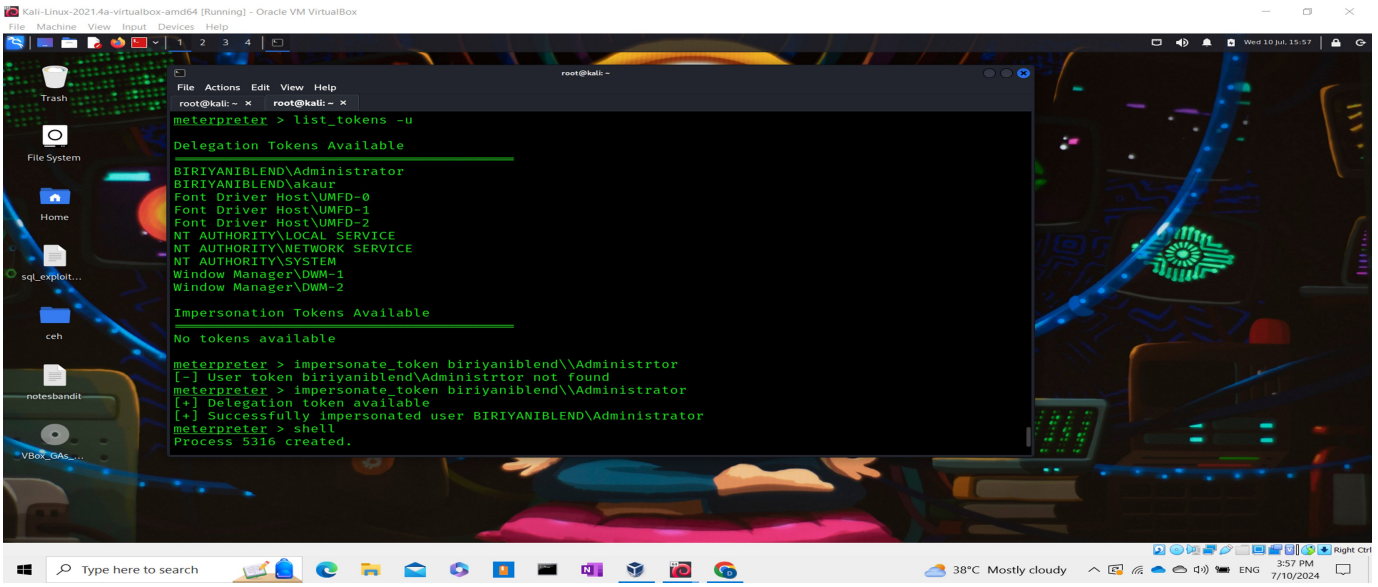
This attack often used in post exploitation scenario is used to escalate privileges. Now just for the sake of little bit of context, within windows environment there are something called as access tokens, granted to a user after authenticated. These access tokens are categorized based on various security levels and through these security levels one can determine the kind of privilege a user have. Our motive here is to find all the access tokens granted by the system security to users logon to the system and impersonate those tokens in order to run processes under the privilege of impersonated user and if there is a logon session of domain admin on the compromised machine, we can impersonate the access token of that admin, hence will be able to escalate our privilege or promote ourselves as domain admin user on that particular workstation within the Active Directory domain and hence can execute processes with the privilege of admin user.

To perform this attack first we will set up metasploit. We will use exploit/windows/smb/psexec that uses a default payload of windows/meterpreter/reverse_tcp.
Configure the tool by setting all the options like RHOSTS, SMBDomain, SMBUser, SMBPassword and set target to native upload.
Once done exploit and get a meterpreter session.

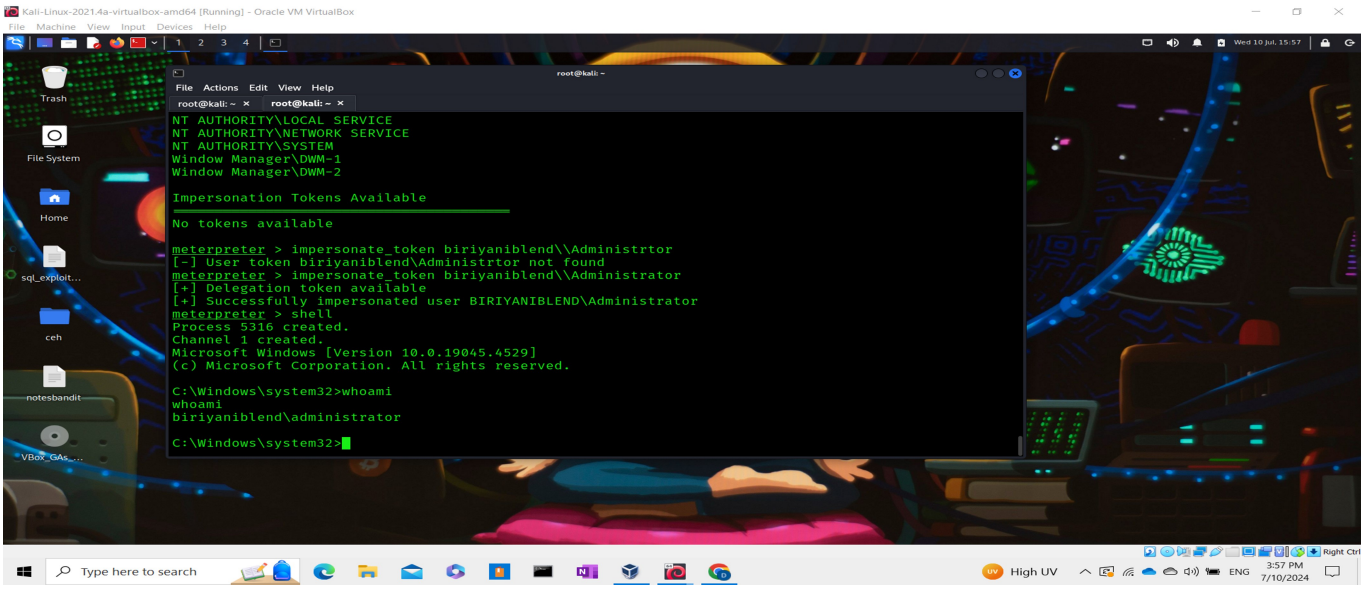


Above screenshot show that in the meterpreter session we have loaded the incognito module using the command 'load incognito' and the below screenshot shows list of token available on the machine using the command 'list_tokens -u'.

After listing all the tokens we impersonated the Administrator token using 'impersonate_token biriyaniblend\\Administrator'.



Hence we can access the shell as the impersonated user who is no one but Administrator.



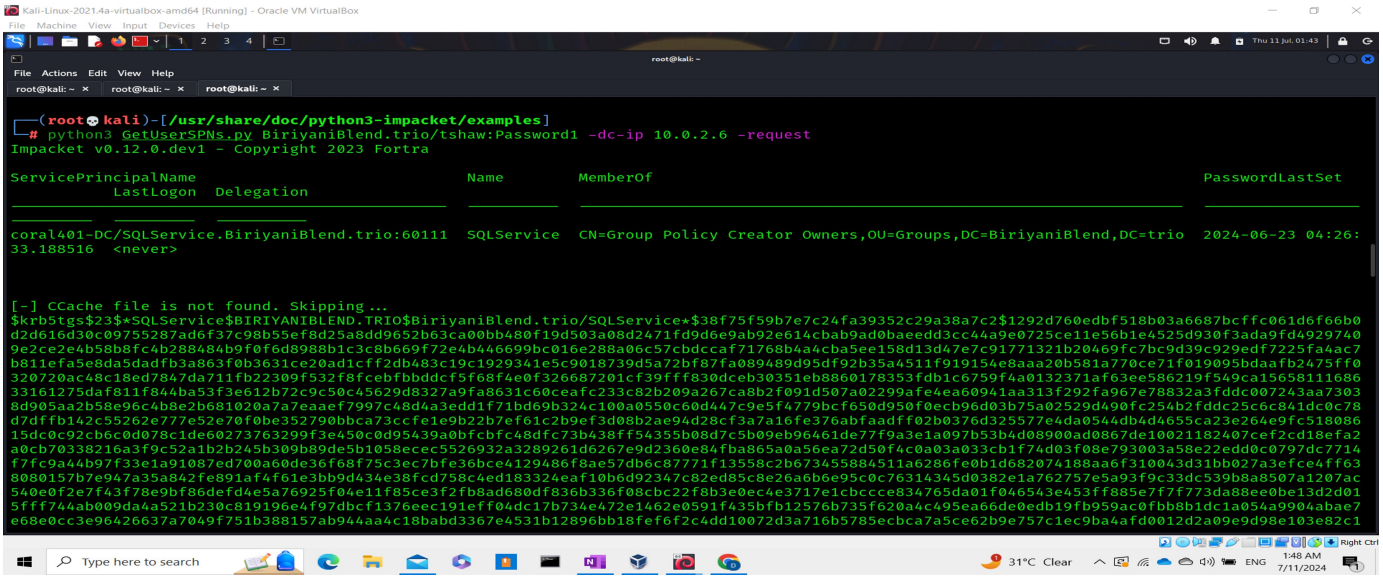
Mitigation Strategies:

- *Limit user/group token creation permissions*
- *Account tiering*
- *Local admin restriction*

3) Kerberosting

Kerberos is the default network authentication protocol within the windows domain. Kerberos authentication is a multi-step process involving the Key Distribution Center (KDC), which consists of two main components: the Authentication Server (AS) and the Ticket Granting Server (TGS).
Our objective here is to grab use the valid credentials of the compromised domain and complete the kerberos authentication in order to TGST(Ticket Granting Service Ticket) which consists of hash of the domain service account credential. We will then take that hash offline to crack it.

Firstly we will use a tool from impacket tool kit called GetUserSPNs.py in order to authenticate and grab TGT.



We can examine tat the service principle name(SPN) is SQLService. And the hash for the SQLService account is also present below.
Now taking the hash and trying to crack it using John.

```
┌──(root💀kali)-[~]
└─# john kerberhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
MYpassword123#   (?)
1g 0:00:00:14 DONE (2024-07-10 16:12) 0.07122g/s 772485p/s 772485c/s 772485C/s MZCARAMEL96..MYface123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now if the service account is also a member of domain admin account which it is in most of the enterprise environment then we can easily compromise the domain controller and then create new users, policies etc.

Mitigation:

- Strong Passwords
- Least Privileges

4) Golden Ticket Attack

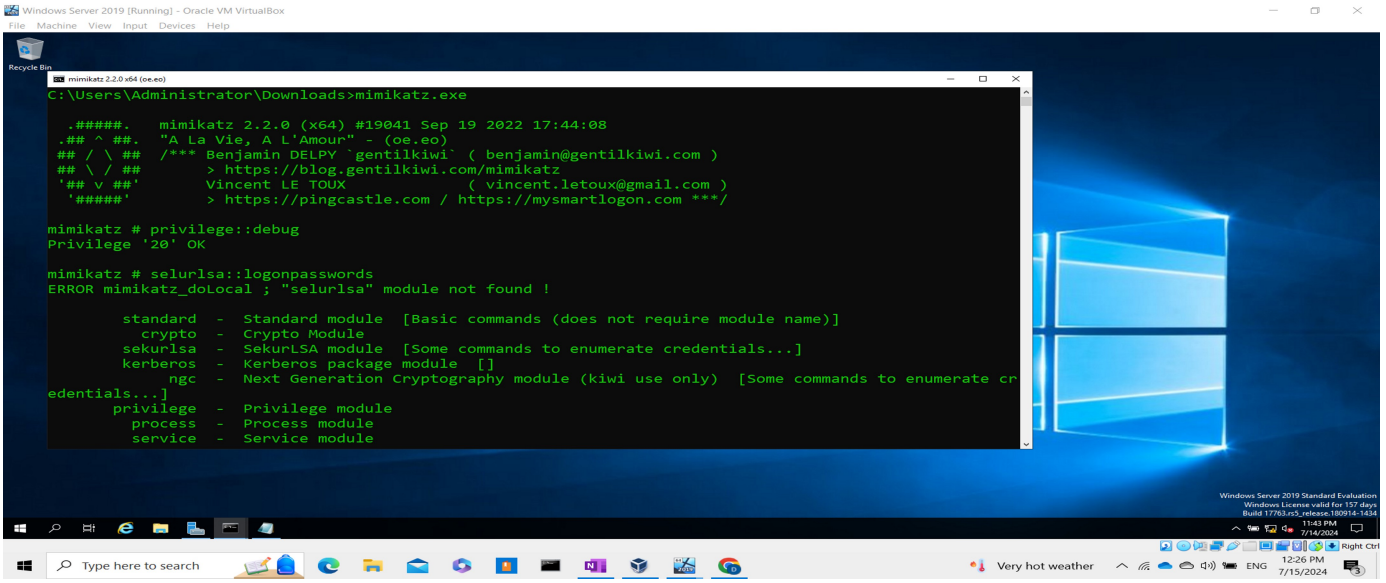
This attack can be executed through two tools. One of them is 'mimikatz' and other is impacket tool kit. Here we will utilize 'mimikatz'.

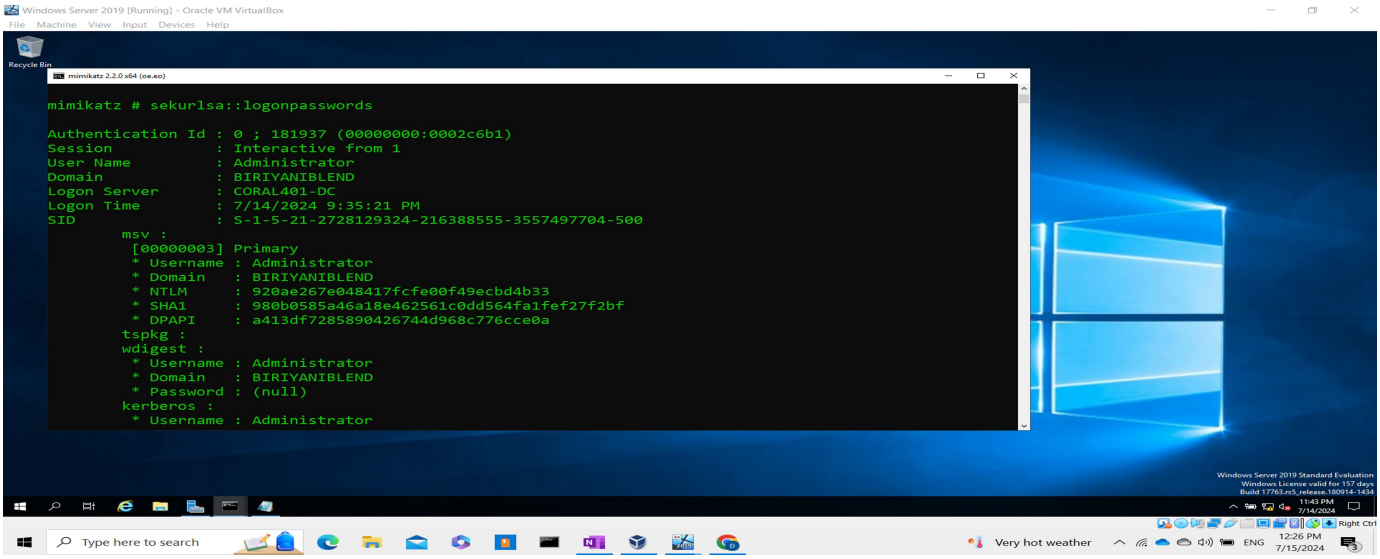
To install and execute mimikatz on the domain controller we need to access it first. It ca be an active session through RDP or through some reverse shell. In case of reverse shell we will utilize powershell.

Here we have access on the domain controller hence we will mimikatz executable on the machine and execute it.

The first command that we fire is 'privilege::debug' to give this process an access to debug other process stored in the memory and secure by system security.

Then following that command we want to fetch all the logon passwords using 'sekurlsa::logonpasswords'.

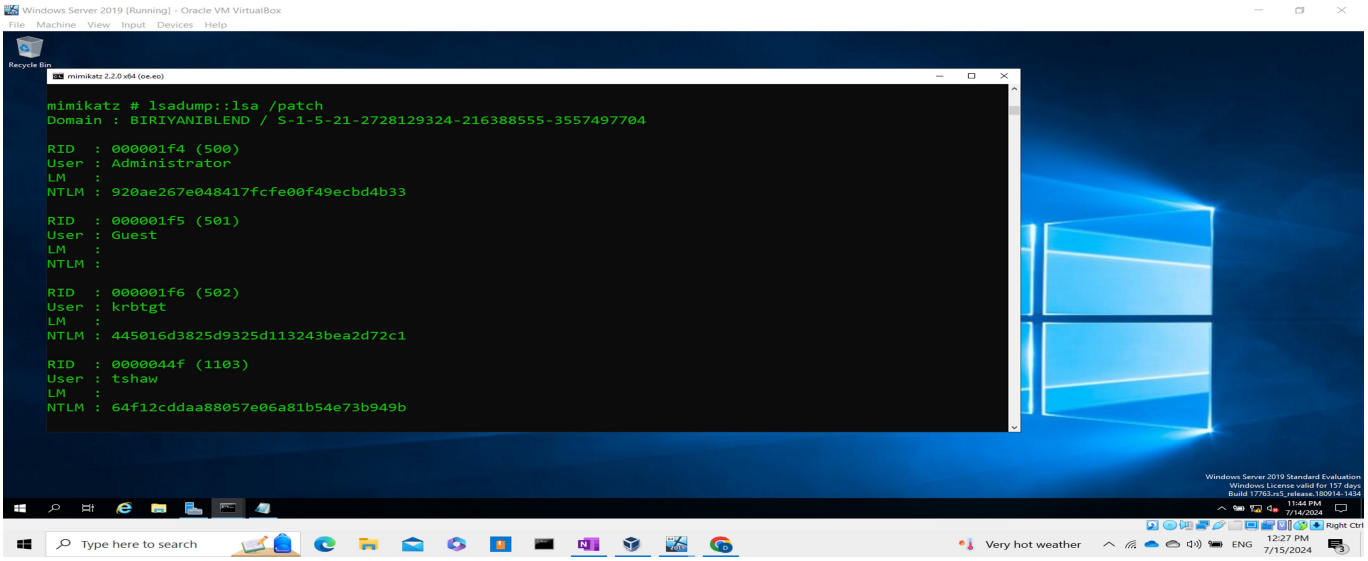




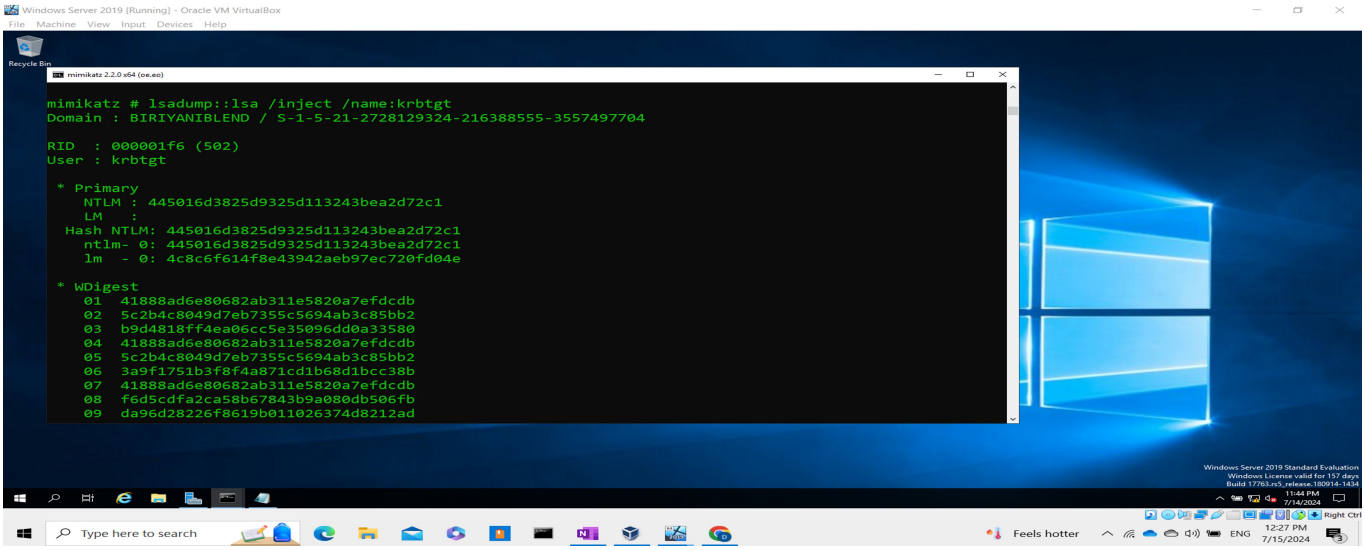
Further we can also dumps lsa using 'lsadump::lsa /patch'.

We can take all these usernames and password hashes offline and try to crack them. Based on how many passwords we are able to crack we can comment on the password policy of the organization objectively.

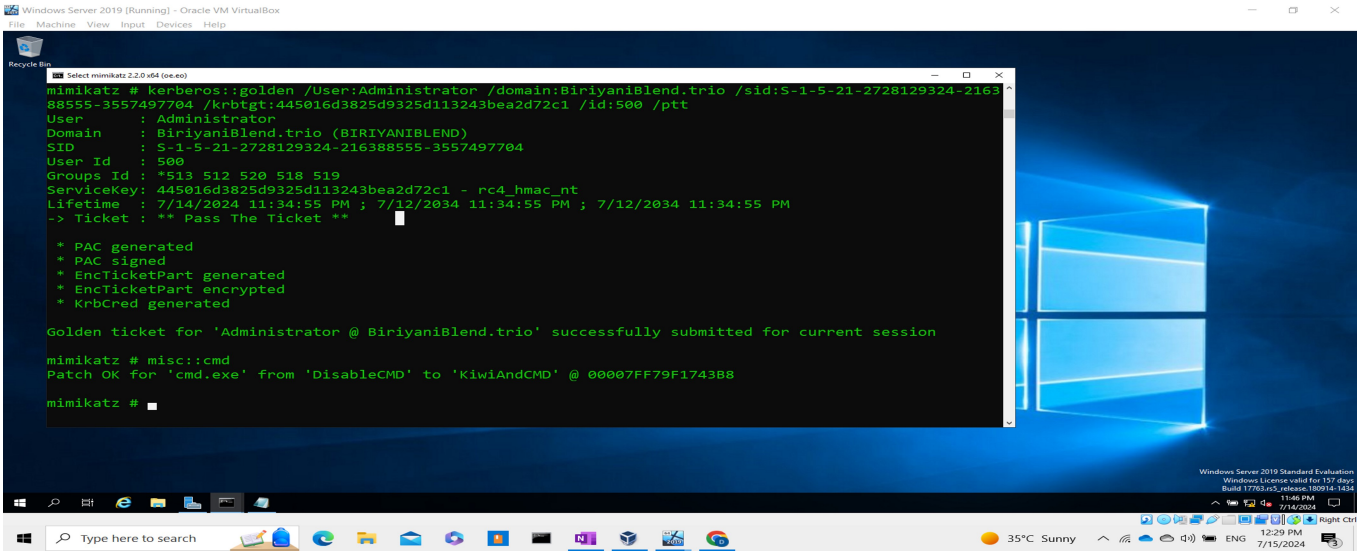
Here we are only interested in 'krbtgt' account password hash.



To particularly search for thee 'krbtgt' account hash we will use 'lsadump::lsa /inject /name:krbtgt'



Now to perform Golden ticket Attack we will use the following command:
'kerberos::golden /User:<any uname> /domain:BiriyaniBlend.trio /sid:<fetch from the above command execution> /krbtgt:<hash> /id:500 /ptt'
/ptt is used to pass the ticket into the current session.
Once the forged TGT which is our Golden Ticket is generated and submitted into our current session we will fire up a shell session using the command 'misc::cmd'.



And finally we can access any computer any share in the entire network.