COMP 357: ADVANCED PENTESTING

MEGA HACKING - THE FINAL PROJECT: KERBEROASTING (GROUP 02)

COMPLETED BY: TWINKALJIT SINGH

PROFESSOR: ADAM ABERNETHY

(Biggs, 2025)

Mitigation: Defence + Validation

Prevent or reduce impact of Kerberoasting: After completing a successful Kerberoasting attack, I applied different defensive controls to reduce or prevent this type of attack in the future. Below each mitigation is ***what I did***, ***why it matters***, *and* ***how it reduces the risk***.

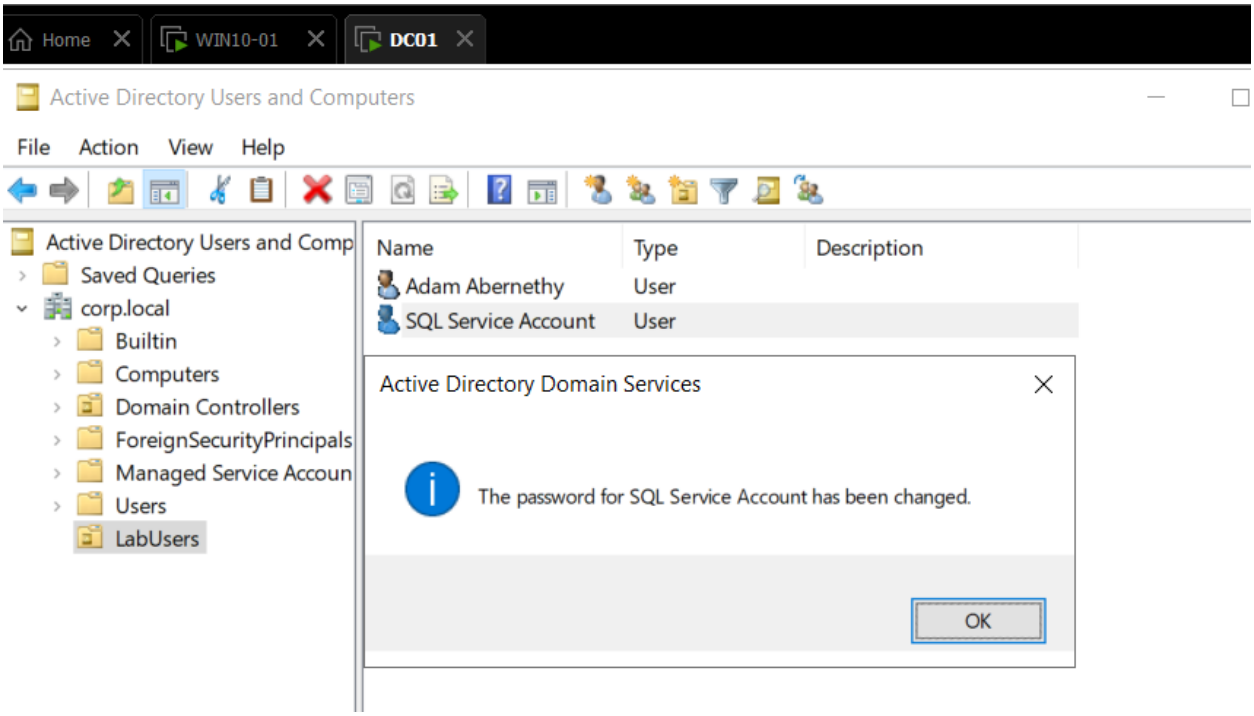
➢ Mitigation 1: Strengthening the Service Account Password

- On the Domain Controller (DC01), I opened Active Directory Users and Computers, located the service account svc-sql, and reset its password. Instead of a short weak password like *password123!*, I generated a strong, long password consisting of uppercase, lowercase, numbers, and symbols (minimum 25+ characters). I also configured the account so the password must be changed regularly.
- Kerberoasting succeeds mainly when:

1. The service account password is weak.
2. The attacker can crack the TGS ticket offline with a wordlist.
3. The password never rotates and becomes predictable over time.

When passwords are long and strong, cracking becomes extremely slow or practically impossible using dictionary or brute-force attacks.

- After resetting the password to a strong one, I attempted Kerberoasting again using Rubeus and Hashcat with the same wordlist. This time, **no valid password was cracked**, which proves that stronger passwords significantly reduce the risk. Even if a ticket is captured, cracking it is not feasible.



➢ Mitigation 2: Removing Domain Admin Privileges from the Service Account

- **Action Completed**

  I opened the service account properties and removed **Domain Admins** from svc-sql membership. Instead, the account was left as a normal user or restricted to only the required system privilege (least privilege model).

- **Reason for this Mitigation**

  Service accounts are often over-privileged in real environments. This creates a high-risk situation where cracking one service account = full domain compromise. By removing Domain Admin rights, even if an attacker cracks the password later, the account will **not give full domain control**, limiting the damage.

- **Impact on Attack**

  When re-testing, even if the ticket was captured again, the account no longer provided domain-wide access. The attacker may still authenticate, but cannot perform privileged actions like editing GPOs, accessing C$ shares, or managing users.
  The blast radius of breach reduces massively.

➢ Mitigation 3: Enabling Regular Password Rotation for Service Accounts

- **Action Completed**

  Password policy was updated to enforce:

  - Strong minimum length (> 14 characters recommended by Microsoft)
  - Complex structure requirement
  - Rotation frequency (e.g., every 60–90 days)

  In addition, service accounts should ideally use **Managed Service Accounts (MSA)** in production to eliminate human password handling and rotation issues.

- **Reason for this Mitigation**

  Kerberoasted hashes remain valid until a password changes. If the account keeps the same password for years, attackers can attempt cracking offline without time pressure. Frequent password rotation reduces exposure time and makes stale ticket captures useless.

- **Impact on Attack**

  If rotation is active, even if a hash was captured successfully but not cracked immediately, tickets become invalid after password change. Cracking efforts become useless and time-wasting for the attacker.

➢ Mitigation 4: Disable Weak Encryption Types like RC4

- **Action Completed**

  In a real enterprise configuration, organizations should disable RC4 encryption and force Kerberos to use AES. RC4 hashes are easier to crack, while AES significantly increases cracking difficulty.

- **Reason for this Mitigation**

  Kerberoasting mostly targets **RC4-HMAC encrypted service tickets (etype 23)** because they crack faster. If AES-only encryption is enabled, hash extraction becomes less useful, much slower to crack, and harder for attackers.

- **Impact on Attack**

  Even if Rubeus still extracts TGS tickets, cracking them becomes computationally expensive, reducing practicality of attack.

➢ Mitigation 5: Monitoring & Alerting for Abnormal Ticket Requests

- **Action Completed**

  In real environment, Security teams should monitor event logs like:

  - **Windows Event 4769 (Kerberos TGS Request)**
  - Sudden spike of TGS requests for a single service account
  - Unusual activity from a workstation user requesting privileged accounts

  SIEM tools (Splunk, Sentinel, Elastic) can alert when someone does mass SPN enumeration.

- **Reason for this Mitigation**

  Kerberoasting is quiet, but **not invisible**. If defenders monitor Kerberos traffic, they can detect the attack during ticket extraction instead of post-breach.

- **Impact on Attack**

  Attackers lose stealth advantage. Blue team can respond early by resetting account password, isolating machines, or blocking suspicious user accounts.

Mitigation Result Summary Table

Mitigation Applied	What Changed	Result on Attack
Strong password	Hash harder to crack	Hashcat failed to recover password
Removed Domain Admins	Reduced privileges	Even if cracked, no domain control
Enabled password rotation	Short exposure window	Captured hash becomes useless soon

Disable RC4	Uses AES encryption	Makes cracking more expensive
Monitoring Kerberos logs	Detects unusual TGS requests	Helps catch attacker early

Outcome

After applying the above defenses, I executed the attack again.
Although Rubeus still generated a ticket**,** Hashcat could not crack the password with the same dictionary, and even if the account was compromised later, it no longer provided Domain Admin access.

This proves that Kerberoasting risk can be greatly lowered using proper password hygiene, least privilege, and monitoring practices.