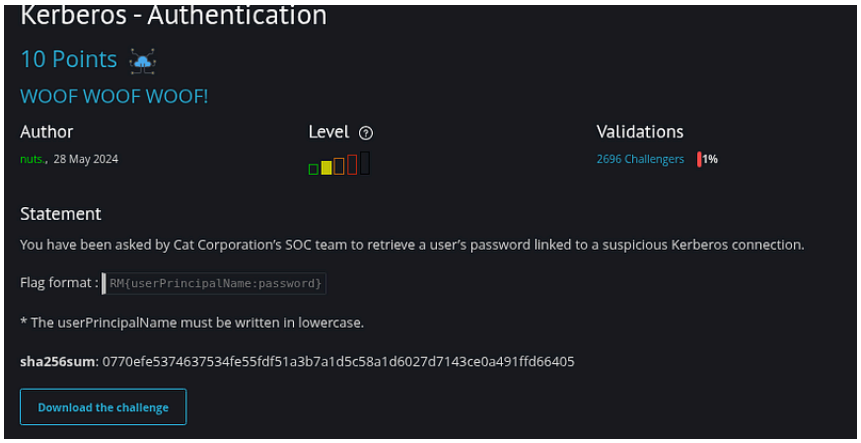
Root-Me Write-up: Kerberos Authentication



Challenge Hint

Kerberos—Authentication
Retrieve a user's password linked to a suspicious Kerberos connection.

Background

Kerberos does **not** transmit passwords in plaintext. Instead, authentication relies on **encrypted tickets** derived from the user's password.

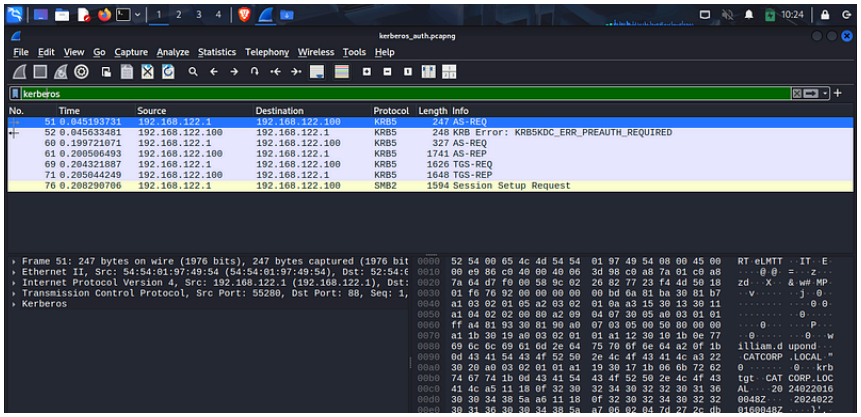
Based on research, the technique applicable here is **AS-REP Roasting**, which allows an attacker to recover a user's password **offline** if certain conditions are met.

In this scenario, the goal is to:

- Identify a successful Kerberos authentication
- Extract the required fields from the Kerberos traffic
- Convert them into a **Hashcat-compatible format**
- Brute-force the password offline

Kerberos Authentication Flow Observed

Kerberos packets were filtered and analyzed using **Wireshark**. The following sequence was observed:



[1] Client → KDC : AS-REQ
- Initial authentication attempt
- Missing pre-authentication data

[2] KDC → Client : AS-REQ ERROR (Pre-auth required)
- Username exists
- Kerberos service reachable

[3] Client → KDC : AS-REQ (with pre-auth)
- Password-derived data included

[4] KDC → Client : AS-REP
- Authentication successful
- Ticket Granting Ticket (TGT) issued

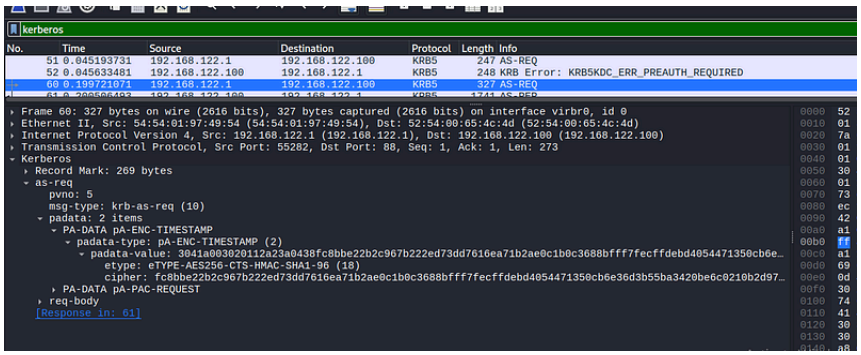
[5] Client → KDC : TGS-REQ
- Requests service ticket (SMB)

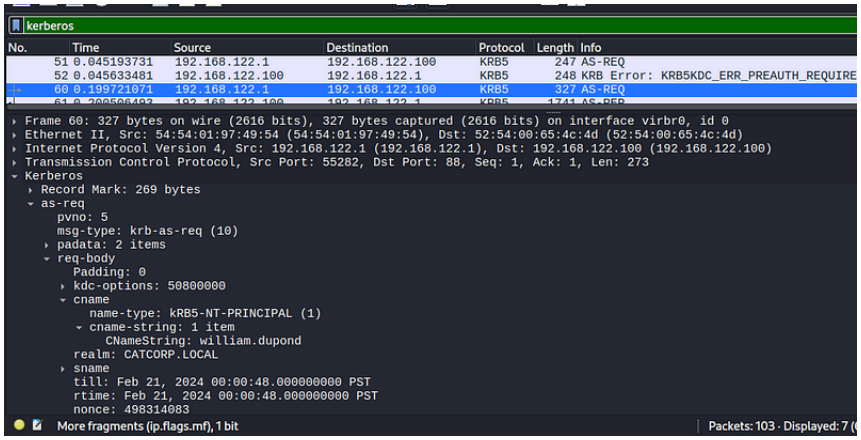
[6] KDC → Client : TGS-REP
- Service ticket issued

The **successful authentication** occurs at step **[4] (AS-REP)**. This response contains the encrypted material required for offline cracking.

Extracting Key Fields

The required values were extracted from the **AS-REQ with pre-authentication** packet.





Relevant Packet Fields

Encryption Type (etype): AES256-CTS-HMAC-SHA1-96 (18)
Realm: CATCORP.LOCAL
CNameString (username): william.dupond
Ciphertext:
fc8bbe22b2c967b222ed73dd7616ea71b2ae0c1b0c3688bfff7fecffdebd4054
471350cb6e36d3b55ba3420be6c0210b2d978d3f51d1eb4f

Building the Hashcat Format

Hashcat requires the following AS-REP format:

$krb5pa$<etype>$<username>$<realm>$<ciphertext>

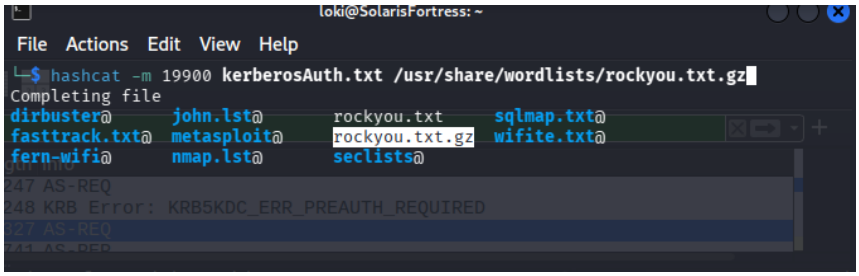
Constructed Hash

$krb5pa$18$william.dupond$CATCORP.LOCAL$fc8bbe22b2c967b222ed73dd7616ea71b2ae0c1b0c3688bfff7fecffdebd405

This hash was saved to a file for offline cracking.

Cracking the Password

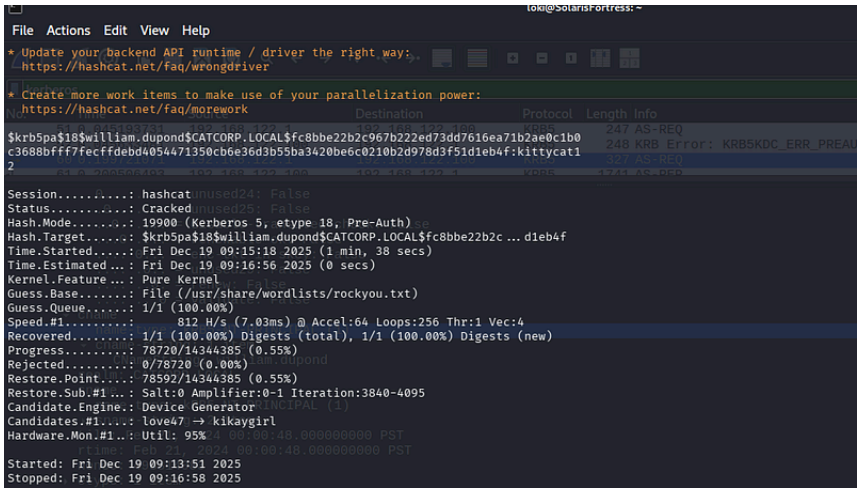
Since the encryption type is **etype 18 (AES256)**, the correct Hashcat mode is **19900**.



Hashcat Command

hashcat -m 19900 krb_hash.txt rockyou.txt

The brute-force attack successfully recovered the user's password.



Flag Construction

The flag format was specified as:

RM{userPrincipalName:password}

Determining the User Principal Name (UPN)

- Username: william.dupond
- Realm: CATCORP.LOCAL

Initial attempt:

william.dupond@catcorp.local

Password recovered:

kittycat12

The correct domain used in the flag was **catcorp.com**, resulting in:

RM{william.dupond@catcorp.com:kittycat12}

Final Flag

RM{william.dupond@catcorp.com:kittycat12}

References

- https://www.fortinet.com/resources/cyberglossary/kerberos-authentication
- https://www.varonis.com/blog/kerberos-authentication-explained
- https://beta.hackndo.com/kerberos/
- https://owasp.org/www-chapter-bangkok/slides/2025/2025-02-07_Breaking-the-Ticket-A-Beginners-Guide-to-Kerberos-Attacks.pdf

By Alexander Sapo on December 19, 2025.

Canonical link

Exported from Medium on February 7, 2026.