- Filename: eccouncil-ceh31250-v12-6-2-1-windows-authentication.md
- Show Name: CEHv12 (312-50)
- Topic Name: System Hacking Phases and Attack Techniques - System Hacking
- Episode Name: Windows Authentication
  ==

Windows Authentication

Objectives:

- Explain the process used for authentication by the Security Account Manager, NTLM, and Kerberos

---

- <u>**Windows Security Accounts Manager(SAM) Database**</u>
    - Located in the Registry
        - `%SystemRoot%/system32/config/SAM`
    - Stores hashed user passwords
        - LM/NTLM hashes
    - Special lock on the SAM to keep safe
        - SAM can't be copied or moved while system is running
        - It can be accessed directly from memory
          <P>
- <u>**NT LAN Manager(NTLM) Authentication**</u>
    - Used to be THE auth mechanism for Windows
        - Now just there as a back-up to Kerberos

1.
2. A user accesses a client computer and provides a

```
+ domain name
+ user name
+ password.
  - The client computes an RC4 UTF-16-LE hash of the password
    + discards the actual password
      - The client sends the user name to the server (in plaintext).
```

2. The server generates a 16-byte random number

```
+ Called a 'CHALLENGE'
  - Sends it back to the client
```

3. Client encrypts this CHALLENGE with the hash of the user's password

```
+ Returns the result to the server
  - This is called the 'RESPONSE'.
```

4. The server sends the following three items to the domain controller:

```
+ User Name
+ CHALLENGE sent to the client
+ RESPONSSE received from the client
```

5. The domain controller uses the user name to retrieve the hash of the user's password and uses it to encrypt the CHALLENGE

```
+ It compares its CHALLENGE with the CHALLENGE in the RESPONSE by the client
  - If they match, authentication is successful
```

<P>

- <u>**Kerberos**</u>
    1. User's client generates an authenticator and is encrypted with the User's

password

- Authenticator = info about the user + timestamp

2. Client sends the encrypted authenticator to the KDC
3. KDC looks up the username and password (*also checks the timestamp*)
4. KDC tries to decrypt the authenticator with the password
5. KDC sends back a TGT to client

- TGT also timestamped and encrypted with the same key as the authenticator

6. Client decrypts the TGT with user's password key
7. Client uses TGT to access other resources

- Client requests access to Sever_A
    - TGT + Server_A Access Request
- KDC accepts request because of TGT
- KDC generates a updated ticket for Server_A access
- Client receives new ticket and sends copy to Server_A
- Server_A decrypts ticket with its own password