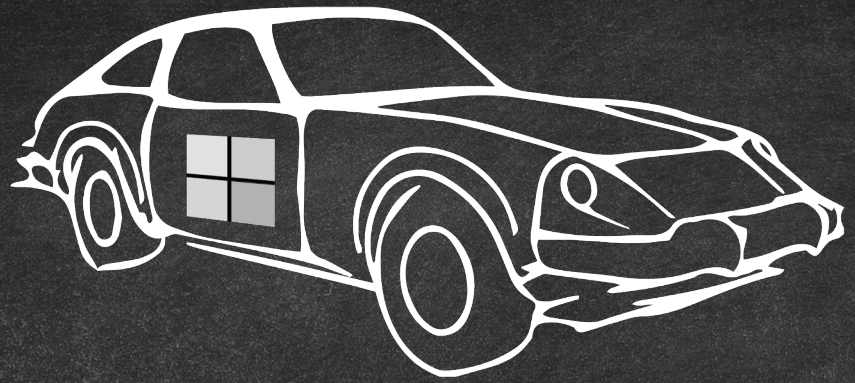


Certified Pre-Owned

Abusing Active Directory Certificate Services

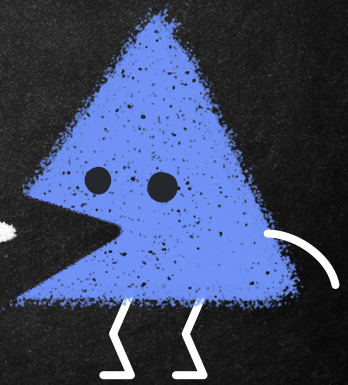
@harmjOy

@tifkin_

SPECTER OPS

TL;DR

- Background
- Account Persistence
- Domain Escalation
- Persistence with “Golden” Certificates



Active Directory Certificate Services

- AD CS is a server role that functions as Microsoft's public key infrastructure (PKI) implementation
  - *Used by organization for smart cards, SSL certificates, code signing, etc.*
- Clients send certificate signing requests (CSRs) to an (enterprise) CA, which signs issued certificates using the private key for the CA certificate

NTAuthCertificates

```
[*] NTAuthCertificates - Certificates that enable authentication:

    Cert SubjectName              : CN=theshire-CA-CA, DC=theshire, DC=local
    Cert Thumbprint               : C55C386A11CC7D0FE7B2B6644947C374835B5899
    Cert Serial                   : 55000000D357096D17908848C50000000000D3
    Cert Start Date               : 3/23/2021 4:18:03 PM
    Cert End Date                 : 3/23/2023 4:28:03 PM
    Cert Chain                    : CN=theshire-DC-CA,DC=theshire,DC=local ->

    Cert SubjectName              : CN=theshire-DC-CA, DC=theshire, DC=local
    Cert Thumbprint               : 187D81530E1ADBB6B8B9B961EAADC1F597E6D6A2
    Cert Serial                   : 14BFC25F2B6EEDA94404D5A5B0F33E21
    Cert Start Date               : 1/4/2021 10:48:02 AM
    Cert End Date                 : 1/4/2026 10:58:02 AM
    Cert Chain                    : CN=theshire-DC-CA,DC=theshire,DC=local
```

This is the root of domain-based certificate auth!

Certificate Enrollment

1. Client generates public/private key pair

2. Client sends a certificate request (CSR) to an Enterprise CA server

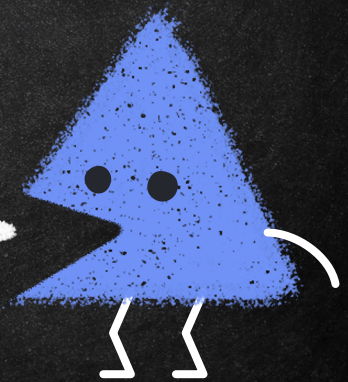
Enterprise CA

3. Does the certificate template exist? Are the settings in the CSR allowed by cert template?
Is the user allowed to enroll for a certificate?

4. CA generates a certificate and signs it using the CA private key

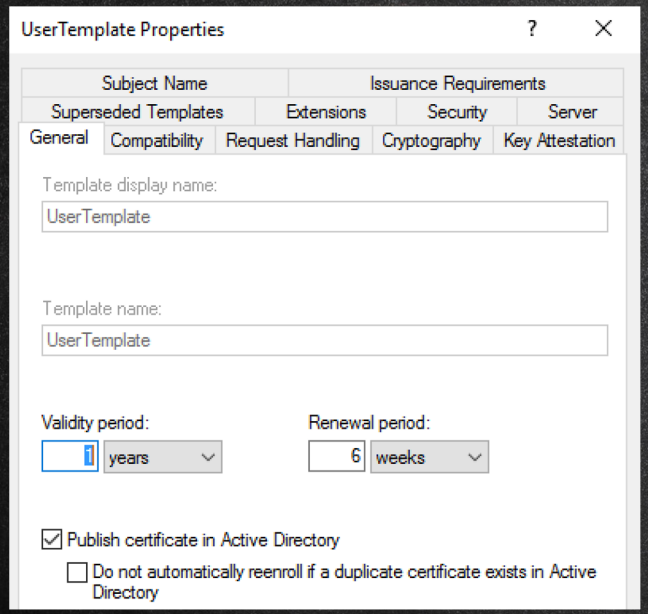
5. Enrollment CA returns certificate to client

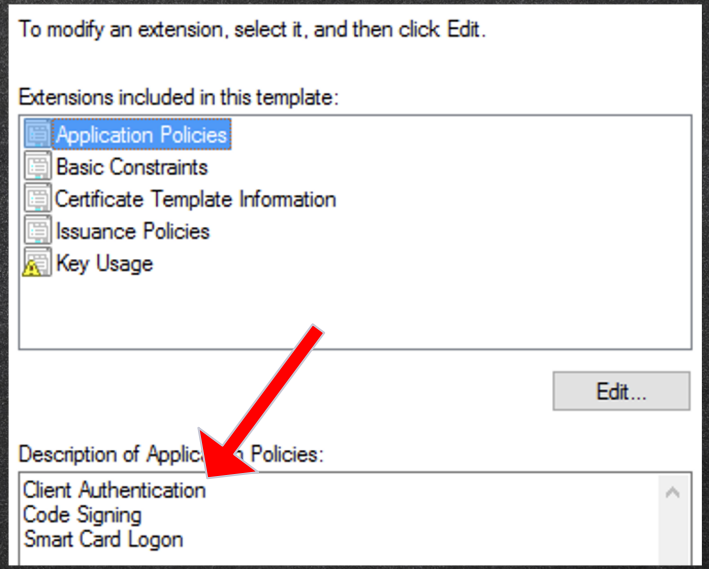
6. Client stores certificate in Windows Certificate store and uses to perform actions allowed by the certificate (authentication, code signing, etc.)

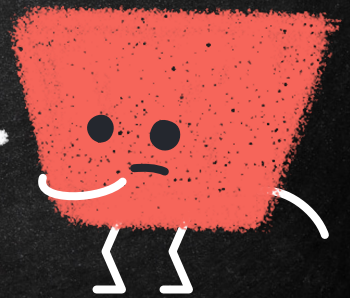


Certificate Templates

- CAs issue certificates with “blueprint” settings defined by certificate templates (stored as AD objects)

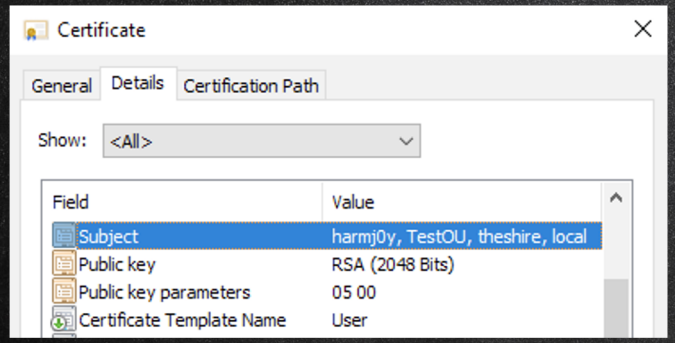


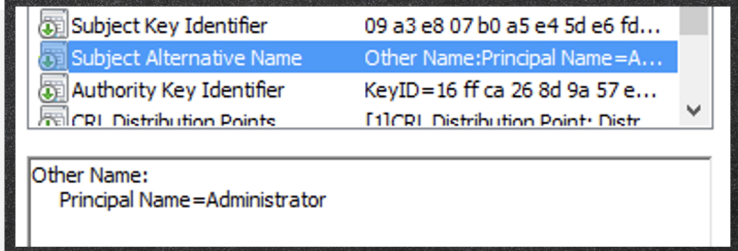




Subject Alternative Names (SANs)

- Allows additional identities to be bound to a certificate beyond the Subject
- Can be dangerous when combined with certificates that allow domain authentication!
  - *AD maps certificates to user accounts using the SAN*

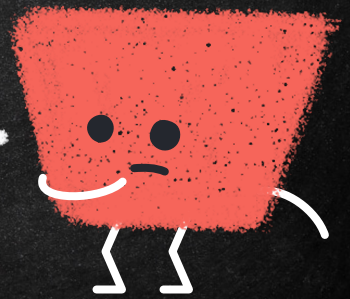


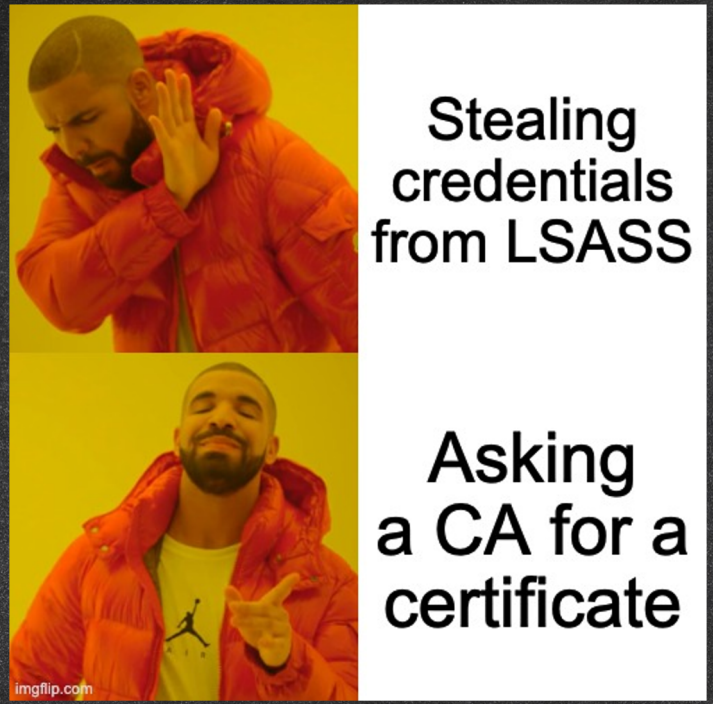


Aren't Smartcards Required?

- No! Rubeus and Kekeo support **Kerberos authentication** using certificates via PKINIT
  - **Schannel authentication** also supports certificates (e.g., LDAPS)

```
C:\Temp>Rubeus.exe asktgt /user:harmj0y /certificate:C:\Temp\harmj0y.pfx /password:Password123!

  _____          _
 (_____ \       | |
  _____) )_   _| | __  _____ _   _  _ __
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

 v1.6.1

[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subject: CN=harmj0y, OU=TestOU, DC=theshire, DC=local
[*] Building AS-REQ (w/ PKINIT preauth) for: 'theshire.local\harmj0y'
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFtDCCBbCgAwIBBaEDAgEWooIExDCCBMBhggS8MIIEuKADAgEFoRAbDlRIRVNISVJFLkxPQ0FMoiMw
      IaADAgECoRowGBsGa3JidGd0Gw50aGVzaGlyZS5sb2NhbKOCBHgwggR0oAMCARKhAwIBAqKCBGYEggRi
      k/yUw9I6uiPHZruYdWf4OovsYzaArBtEg1pgCjaIzCc9ikFhVJX2xAssFaol9XtGR2a3Y0TzzjM21Km9
```





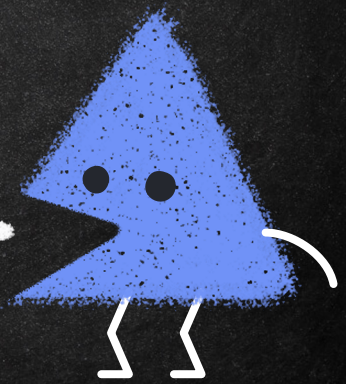
Account Persistence

a.k.a. Long Term LSASS-less Credential Theft

"Passive" Certificate Theft

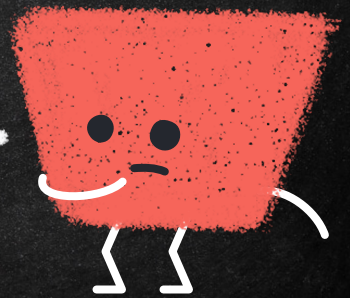
- If hardware protection is not used, existing user/machine certificates are stored using DPAPI
  - *Mimikatz and SharpDPAPI can steal such certs/private keys*

```
    Thumbprint       : 7AB2BA3046ACA6F5C16E03ABF619018583CC069D
    Issuer           : CN=theshire-DC-CA, DC=theshire, DC=local
    Subject          : CN=attacker, CN=Users, DC=theshire, DC=local
    Valid Date       : 5/21/2021 2:07:43 PM
    Expiry Date      : 5/21/2023 2:07:43 PM
    Enhanced Key Usages:
        Certificate Request Agent (1.3.6.1.4.1.311.20.2.1)
        Any Purpose (2.5.29.37.0)
         [!] Certificate can be used for client auth!

    [*] Private key file 0c65eb3c0cab72d6af2e594facceadb7_6c712ef3-1467-4f96-bb5c-6737ba66cfb0 was recovered:

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvQwqdu+Hrkjlf+ULjCmld3wa9hCsG/Md4xLCwihvWn39MrYO
/pW435cajGOtXyQyMdTwvK9Y1YOY/sozTrHVt4ChBkxzgw0qqPPeJsEmV87R8xpS
WYXCiujWP6eoLFL+A9Zyj1JMUFm5xU6m83GB9ZQAJUFe0lF5wpIeX+dWsd6uVK32
```



"Active" Certificate Theft

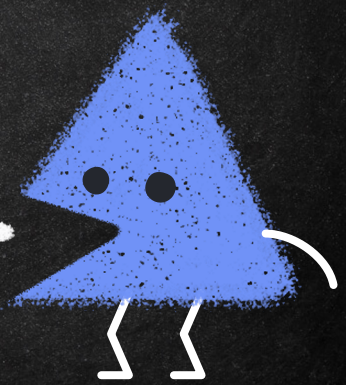
- Users/machines can enroll in any template they have **Enroll** permissions for
  - *By default the **User** and **Machine** templates are available*
- We want a template that allows for AD authentication
  - *Lets us get a user's TGT (and NTLM!)*
  - *Lets us compromise a computer through RBCD/S4U2Self*
- We can enroll through DCOM (Certify), RPC, and AD CS web endpoints

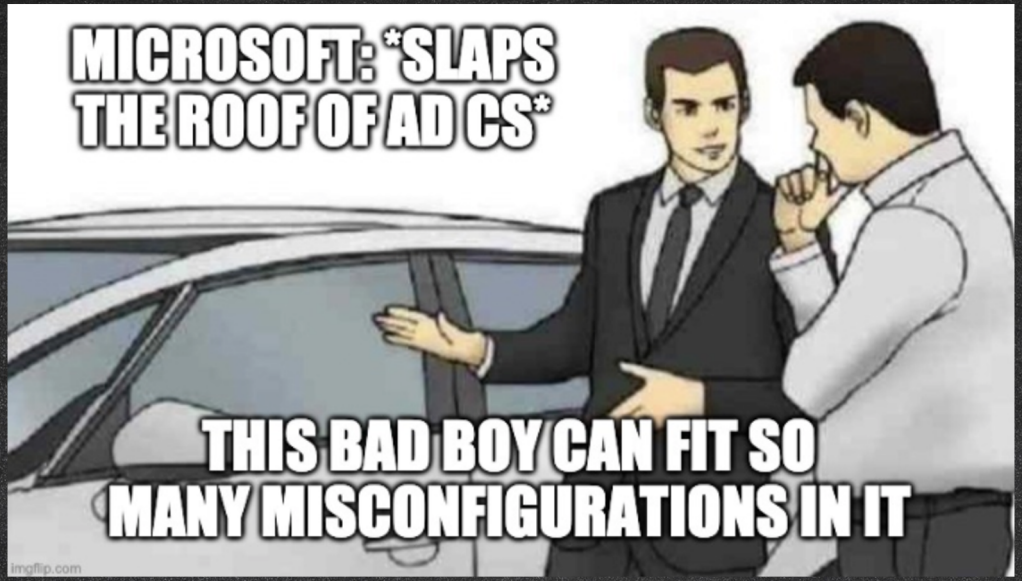


```
C:\Tools>Certify.exe request /ca:dc.theshire.local\theshire-DC-CA /template:User

  _____          _   _  __
 / ____|        | | (_)/ _|
| |     __ _ __| |_ _| |_ _   _
| |    / _\ '__| __| |  _| | | |
| |___| __/ |  | |_| | | | |_| |
 \_____\___|_|   \__|_|_|  \__, |
                              __/ |
                             |___./
 v0.5.2

[*] Action: Request a Certificates

[*] Current user context    : THESHIRE\harmj0y
[*] No subject name specified, using current context as subject.

[*] Template                : User
[*] Subject                 : CN=harmj0y, OU=TestOU, DC=theshire, DC=local

[*] Certificate Authority   : dc.theshire.local\theshire-DC-CA

[*] CA Response             : The certificate had been issued.
[*] Request ID              : 4614

[*] cert.pem         :

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAtLiaRTnJPiAARucYbJOwGeA7GCLndz+F2o39WhK1M8QTclmO
```

Offensive Advantages

- Doesn't touch lsass.exe's memory!
- Doesn't need elevation (for user contexts)!
- Few existing detection methods! (*currently* lesser known technique :)
- Separate credential material from passwords
  - *Works even if an account changes its password!*
  - *Long lifetime. By default, **User/Machine** templates issue certificates valid for **1 year**.*





Domain Escalation

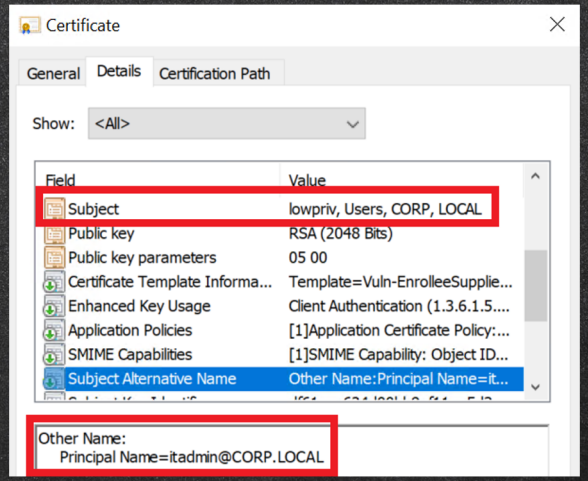
Domain User → Enterprise Admin

Template Misconfigurations: General Requirements

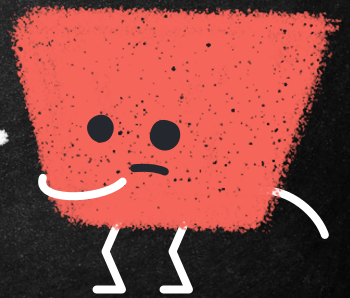
- The Enterprise CA grants low-privileged users enrollment rights
- Low privileged users can enroll in the template
  - Specified by the certificate template AD object's security descriptor
- Manager approval is disabled
- No "authorized signatures" are required

Key Misconfiguration: Templates that Process User-Supplied SANs

1. An attacker can specify an arbitrary SAN when requesting a certificate
2. The certificate enables domain authentication
3. The CA creates and signs a certificate using the attacker-supplied SAN

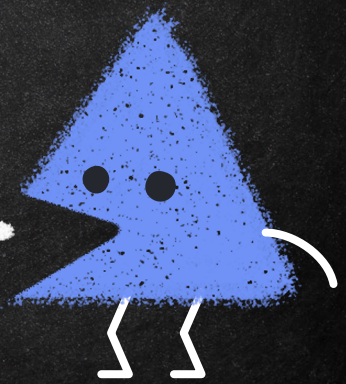


Then the attacker can become any account in the domain!



Escalation Scenarios

- **ESC1**
  - General Requirements
  - **[PKINIT] Client Authentication**, **Smart Card Logon**, **Any Purpose**, or **No EKU** (i.e., EKU allows auth)
  - The **ENROLLEE_SUPPLIES_SUBJECT** flag
- **ESC2**
  - General requirements
  - The **Any Purpose** EKU or **No EKU**
- **ESC3**
  - General requirements + no “enrollment agent restrictions”
  - The **Certificate Request Agent** EKU
  - Enrollment rights to template with a few other requirements



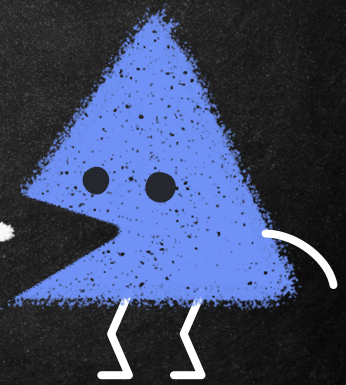
cmd.exe (running as THESHIRE\lowpriv)

```
C:\Tools>Certify.exe find /vulnerable /ca:dc.theshire.local\theshire-DC-CA
```

Finding vulnerable certificate templates

Escalation Scenarios (cont.)

- **ESC4**
  - Vulnerable certificate template access control
- **ESC5**
  - Vulnerable PKI object access control
- **ESC6**
  - **EDITF_ATTRIBUTESUBJECTALTNAME2** flag set on a CA
  - *(Allows CSRs for ANY template to specify a SAN!)*
- **ESC7**
  - Vulnerable CA access control
  - The **ManageCA** permission can be used to fixate ESC6



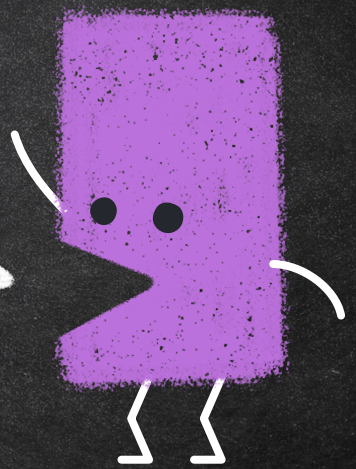
ESC8 – NTLM Relay to HTTP Enrollment Endpoints

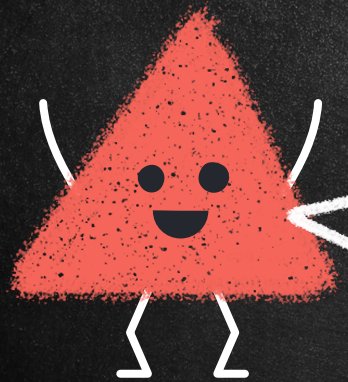
- AD CS web enrollment endpoints are optional roles (but commonly installed)
  - *All of these endpoints are vulnerable to NTLM relay!*
- If there is a user-enrollable auth template:
  - *Extends the window for user NTLM relay scenarios*
- If there is a machine-enrollable auth template:
  - *Combine with printer bug for coerced auth*
  - *I.e., take over **ANY** system in the domain running the spooler service!*

“

“We determined your finding is valid but does not meet our bar for a security update release.”

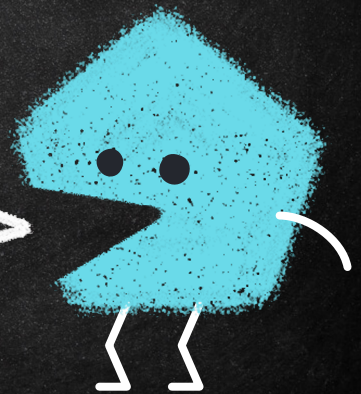
-MSRC





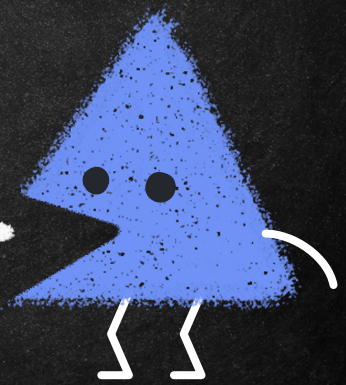
Domain Persistence

"One Certificate To Rule Them All"



Stealing CA Private Keys

- If the private key for a CA's certificate is not protected by a TPM/HSM, it's protected by DPAPI
  - *CAs sign issued certificates with this key*
- If we can steal private key of any CA cert in **NTAuthCertificates**, we can forge our own certificates as anyone in the domain!
- These forged certs can't be revoked!
  - *The certs are never actually "issued"!*
  - *Forged certs work as long as the CA cert is still valid :)*

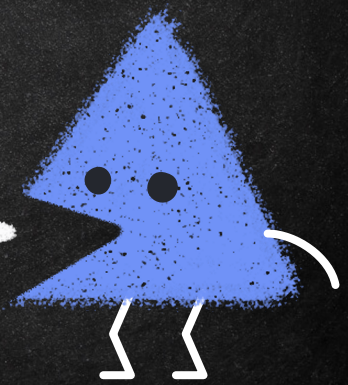


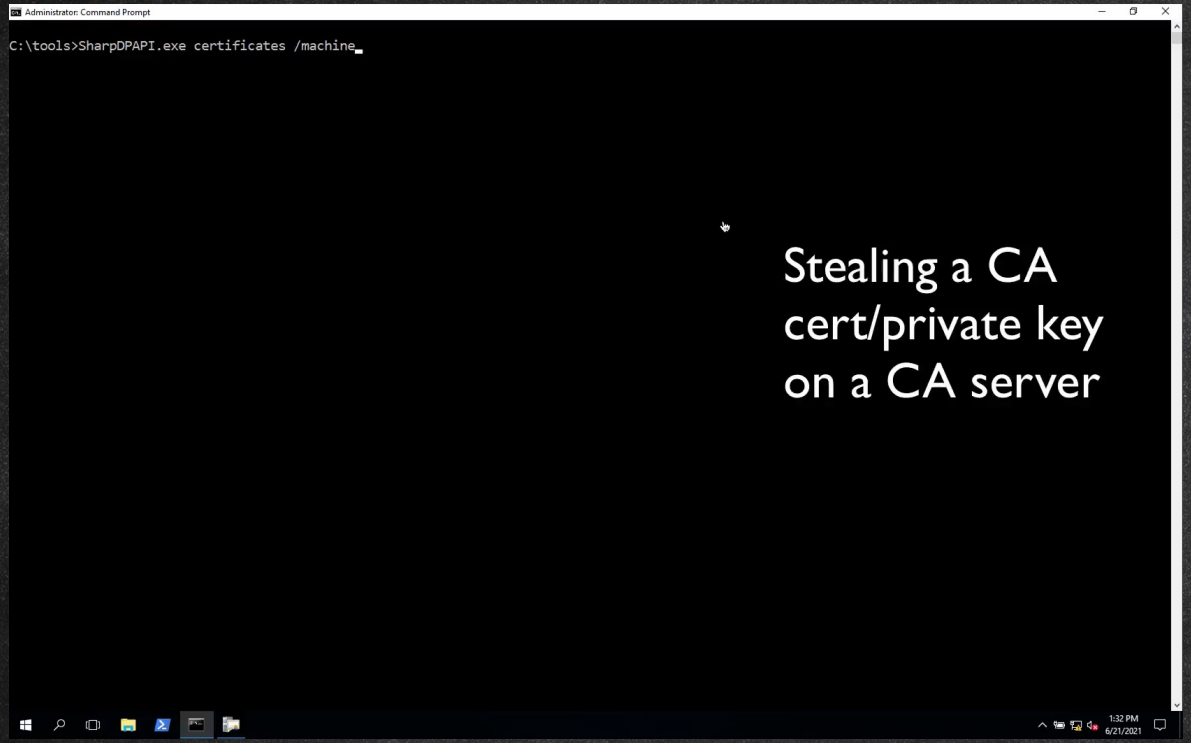

THEY SAID I
COULD BECOME ANYTHING
SO I BECAME A
CERTIFICATE AUTHORITY
imgflip.com

Bonus: PKINIT -> NTLM

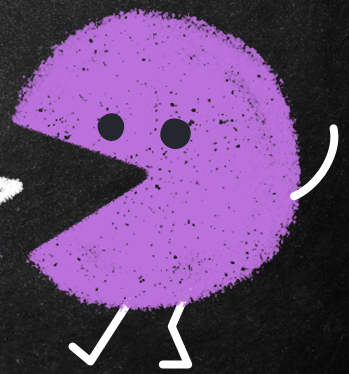
- As part of [MS-PKCA], for backwards compatibility a legitimate certificate can be used to retrieve the associated user/computer's NTLM hash
  - *First publicly/offensively detailed by @gentilkiwi*
  - *Recently integrated into Rubeus by @_ethicalchaos_ and @exploitph*

- If we combine this with "golden" certificates, we have an alternative way to obtain NTLM credentials for any active user/computer (i.e., an alternative to DCSync)



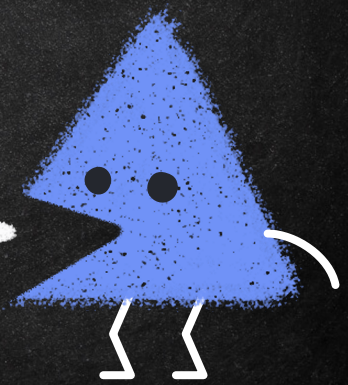


Stealing a CA cert/private key on a CA server



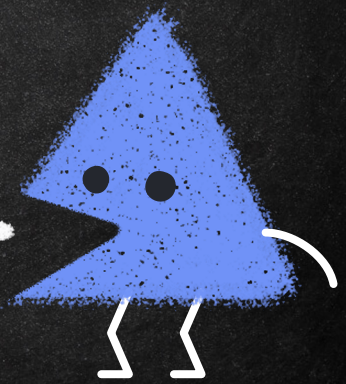
Summary

- AD CS has a lot of abuse potential!
  - User credential theft/machine persistence
  - Domain escalation and persistence
- Our 140 page whitepaper has complete details
  - Includes extensive defensive information and additional architectural considerations
  - https://bit.ly/3xLziQ9
- Certify and ForgeCert are now live in the GhostPack Github organization, along with PSPKIAudit for auditing your environment



Acknowledgements

- Previous work (see the paper for complete details):
  - Benjamin Delpy, PKISolutions, Christoph Falta, CQURE, Keyfactor, @Elkement, Carl Sörqvist, Brad Hill
- Ceri Coburn for PKINIT related Rubeus additions
- Special thanks to Mark Gamache for collaborating with us on parts of this work



Questions?