漏洞扫描

一、漏洞数据库

1.共享漏洞数据库或平台

2.searchsploit

kali自带，可与Nmap联动

先用Nmap扫描，结果放在一个.xml文件中：

`nmap 域名/ip -sV -oX nmap.xml`

然后输入命令：

`searchsploit --nmap nmap.xml`

找到对应漏洞利用程序执行

二、Nmap漏洞扫描

1.Nmap内置NSE脚本

安装在/usr/share/nmap下的scripts文件夹中，脚本有多种类型，其中vuln类别的脚本专门用于漏扫

输入命令：

`nmap --script vuln -T4 域名/ip`

Nmap就会运行所偶属于vuln类别的脚本

如果需要运行所有脚本，可以输入命令

`nmap -T4 -A -sV -vvv -d -oA target.output --script all --script-args vulns.show=all 域名/ip`

其中，-A表示启用操作系统和服务版本探测、脚本扫描和traceroute；-sV参数表示使用服务版本探测；-vvv表示增加输出的详细程度；-d表示增加调试输出的详细程度，-oA target.output表示将扫描结果输出到三种格式文件中(nmap、gnmap、xml)，并以target.output为文件名前缀；--script all表示适用所有的NSE脚本来扫描目标；--script-args表示提供参数给脚本，其中vulns.show=all表示显示所有的漏洞信息。

注：--script all选项非常危险，会运行所有NSE脚本，其中会包含一些可能造成损害或违法的脚本如dos、exploit、malware等，除非有明确目的，否则不建议使用该选项

###

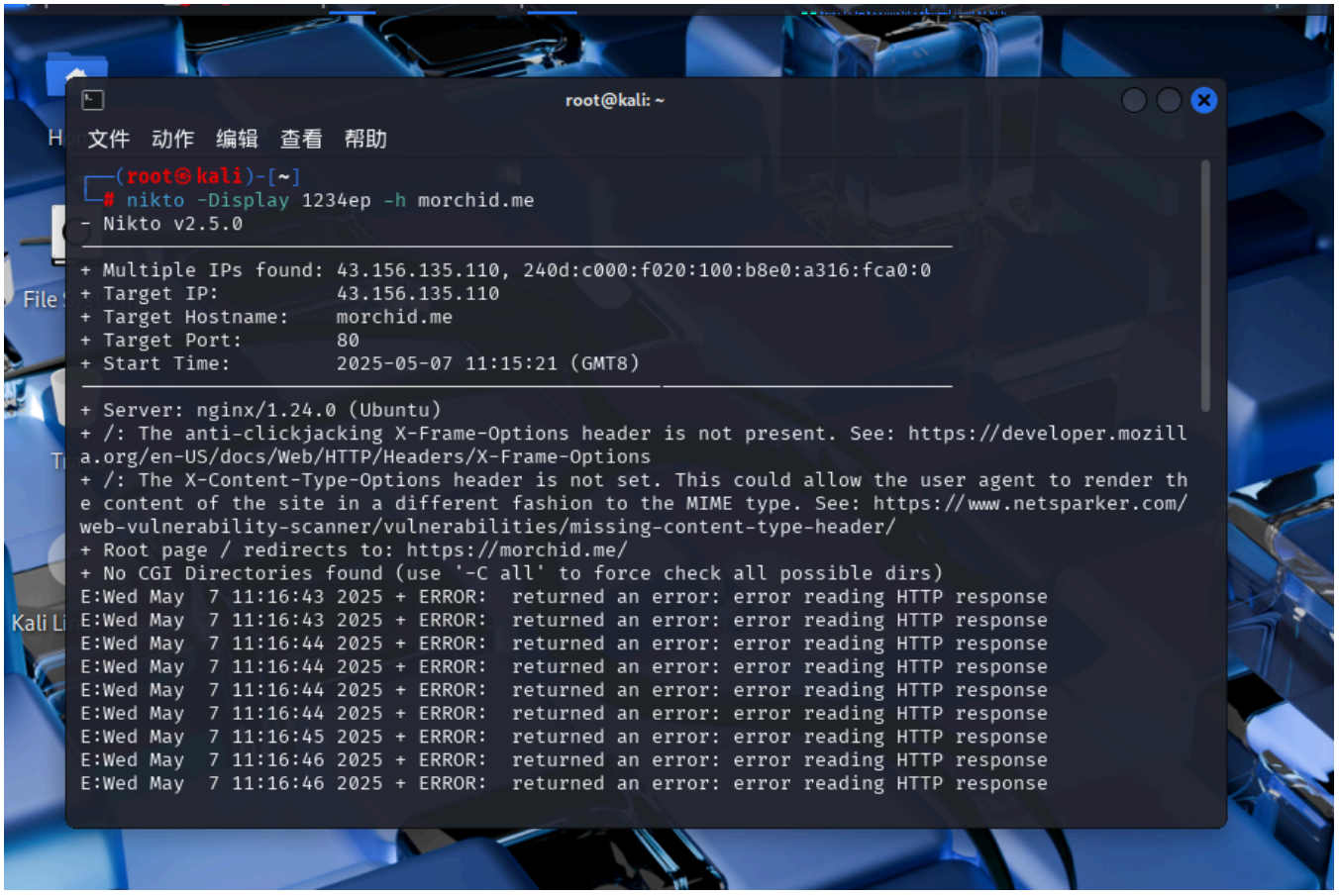
2.自定义NSE脚本

切换至Nmap的NSE脚本目录，以扫描网站是否存在phpinfo.php文件为例，创建一个phpinfo.nse文件，编写保存完毕后输入命令：`nmap --script-help phpinfo.nse`获取该脚本帮助，如果能显示帮助信息则证明该脚本可以正常运行。

然后输入：`nmap 域名/ip --script phpinfo`检测网站是否存在phpinfo.php文件。

phpinfo.nse

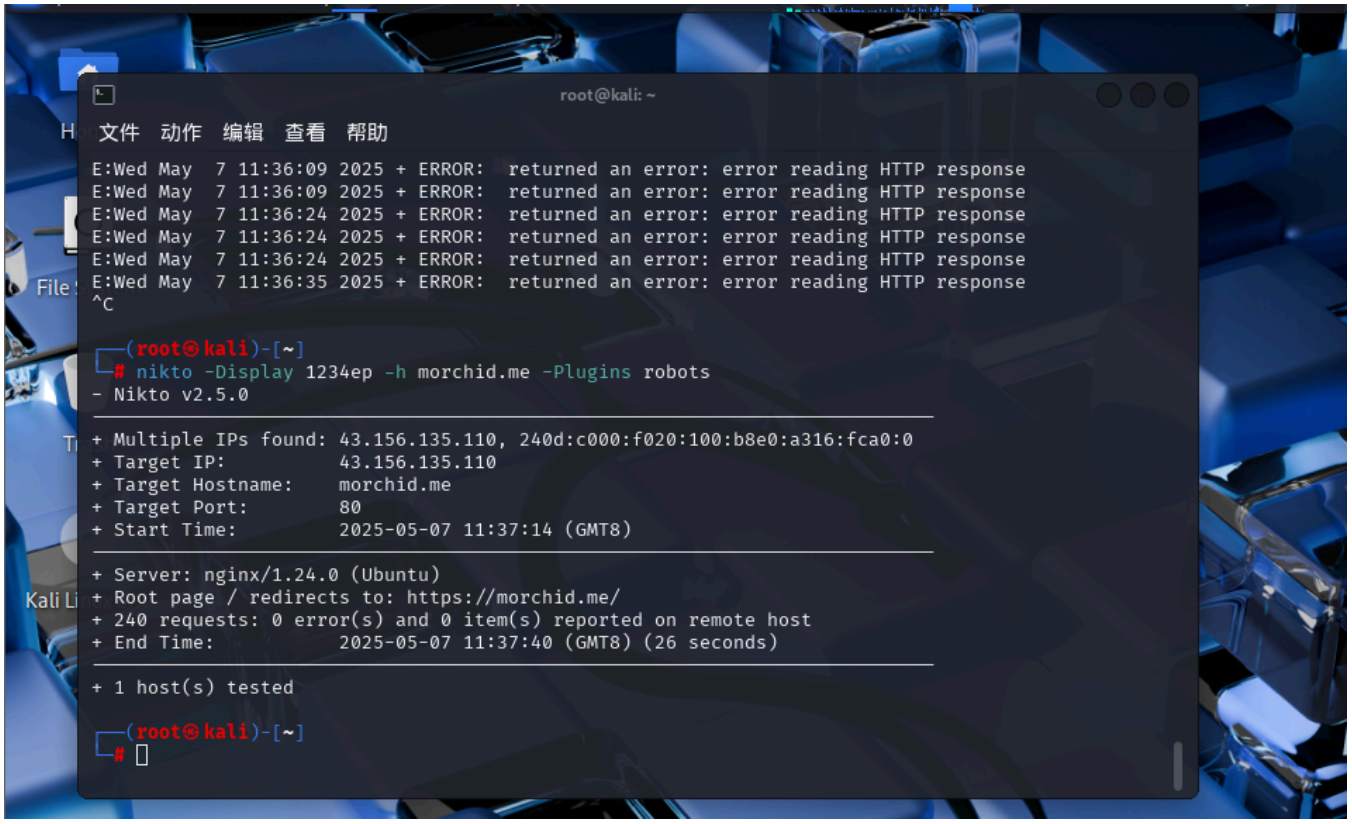
三、Nikto漏洞扫描



`nikto -Display 1234ep -h 域名/ip`

`nikto -Display 1234ep -h 域名/ip -o nikto.html` 可以将漏洞扫描结果保存到nikto.html文件中，通过浏览器打开可以更直观地查看扫描的漏洞。

Nikto拥有非常多的插件，可输入命令: `nikto -list-plugins` 来查看Nikto自带的插件，使用插件进行漏洞扫描可用命令：`nikto -Display 1234ep -h 域名/ip -Plugins 插件名称` 如果不使用-Plugins选项指定插件名称，Nikto会使用全部插件执行漏洞扫描



四、Wapiti漏洞扫描

wapiti，开源轻量级Web应用程序漏洞扫描工具，可对Web应用进行黑盒测试，寻找可以注入数据的脚本和表单，并利用各种攻击载荷检测漏洞。

进行简单漏扫的命令：`wapiti -u 域名/ip --color`，

扫描登录后的网站

可通过设置Cookie进行扫描。Wapiti读的是JSON格式的Cookie，需要用到Wapiti自带的获取Cookie的工具——wapiti-getcookie，输入命令：

```
wapiti-getcookie -u 域名/ip -c cookie.txt
```

然后填写表单登录即可。登录表单和Cookie会存储到cookie.txt文件中，其中-u用来指定URL，-c表示制定一个JSON格式文件。

生成完毕后可通过-c或-cookie来指定Cookie文件。如：

```
wapiti -u 域名/ip --cookie cookie.txt
```

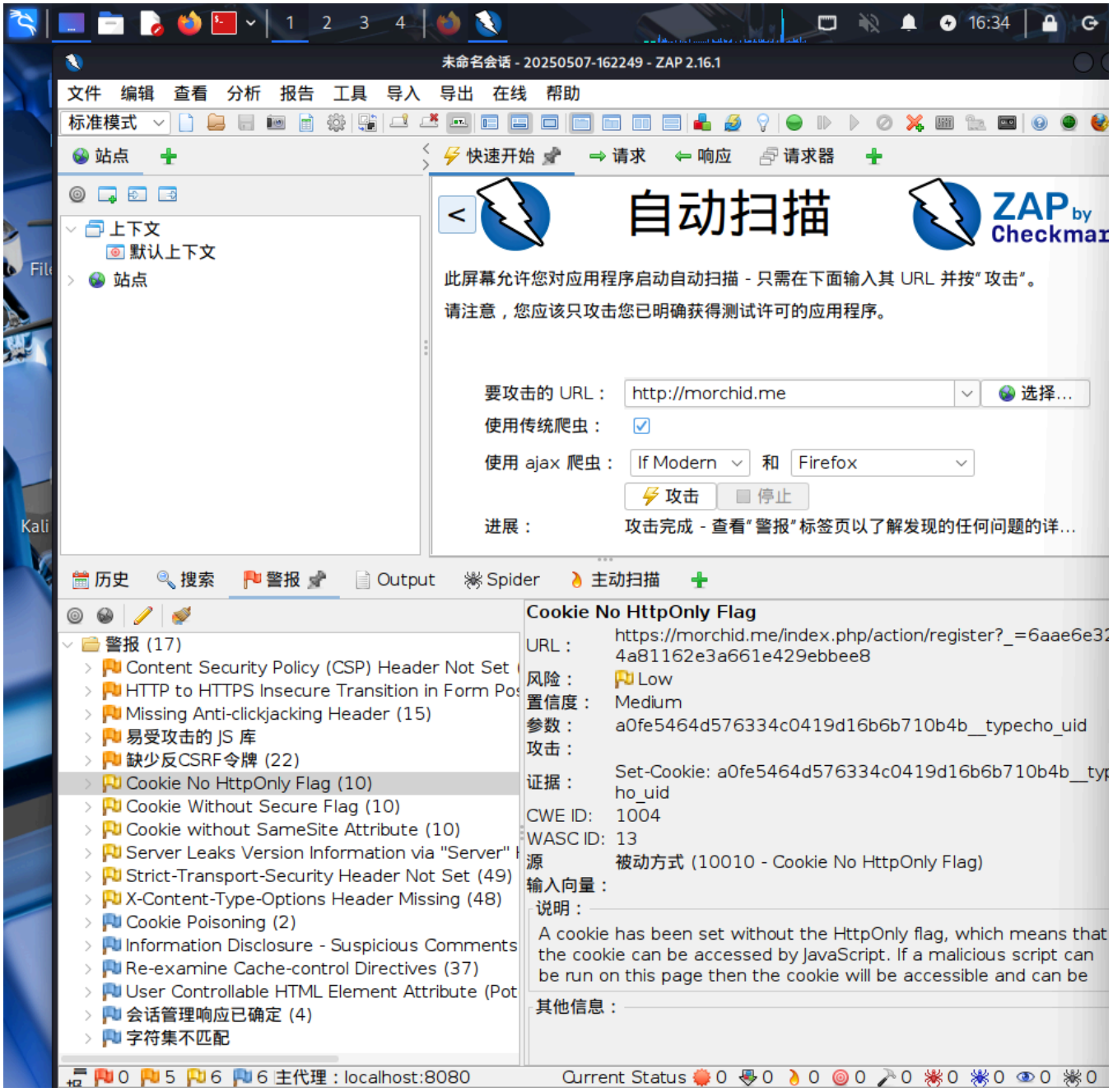
五、ZAP漏洞扫描

ZAP（Zed Attack Proxy）是由OWASP开发的免费开源的安全测试工具。较为全面和灵活。

1.使用ZAP主动扫描

1>设置代理，导入证书

2>执行主动扫描



扫描完成后可点击报告生成报告

2.使用ZAP手动探索

手动探索是另一种扫描方式，可以让我们通过代理浏览器来访问目标网站，并记录所有请求和响应。手动探索时ZAP不会自动发送攻击载荷，而是由渗透测试人员根据需要选择或编辑不同参数和值后自行发送。

1>设置代理

2>在ZAP中启动代理浏览器访问目标网站

六、CMS漏洞扫描

CMS类型漏洞

在计算机安全领域，CMS（内容管理系统）通常指的是用于管理网站内容的软件系统，例如WordPress、Joomla、Drupal等。CMS类型漏洞通常指的是在CMS软件中发现的漏洞，这些漏洞可以被恶意用户利用来攻击或破坏网站。以下是一些常见的CMS类型漏洞及其防范措施：

1. SQL注入

描述：攻击者通过在CMS的输入字段中插入恶意的SQL代码，尝试操纵数据库。

防范措施：

- 使用参数化查询。
- 对所有输入数据进行验证和清洗。
- 使用最新的数据库和CMS版本，因为新版本通常包含安全补丁。

2. 跨站脚本（XSS）

描述：攻击者将恶意脚本注入到CMS的网页中，当其他用户浏览这些页面时，脚本会在他们的浏览器中执行。

防范措施：

- 对所有输出数据进行编码，特别是来自用户的输入。
- 使用内容安全策略（CSP）来限制资源加载。

3. 文件上传漏洞

描述：攻击者上传恶意文件（如PHP脚本），这些文件可能被执行，从而控制服务器。

防范措施：

- 限制允许上传的文件类型。
- 对上传的文件进行内容检查。
- 将上传的文件存储在网站外部，且不可直接访问的目录中。

4. 权限管理漏洞

描述：攻击者可能通过不正确的权限设置获得非授权的访问权限。

防范措施：

- 实施最小权限原则，即为用户分配必要的最小权限。
- 定期审计和监控用户权限。

5. 插件和主题漏洞

描述：第三方插件或主题可能包含未被发现的安全漏洞。

防范措施：

- 定期更新CMS、插件和主题到最新版本。
- 从官方或信誉良好的源获取插件和主题。
- 使用插件和主题之前，检查其安全性和用户评价。

6. CSRF（跨站请求伪造）

描述：攻击者诱导用户执行非预期的操作，例如更改管理员密码或发布恶意内容。

防范措施：

- 使用CSRF令牌来验证请求的合法性。
- 对敏感操作实施双重验证。

1.WPScan漏洞扫描

WPScan是一款扫描WordPress网站漏洞的工具，它能够扫描WordPress本身的漏洞、插件漏洞和主题漏洞。

已预装在Kali Linux系统中，执行前需要输入命令：`wpscan --update`更新数据库

访问官网注册用户获取用于进行漏扫的API Token，同一个每天免费用25次

对网站进行基本扫描：`wpscan --url 域名/ip --api-token Token`

2.JoomScan漏洞扫描

JoomScan是一款开源的OWASP Joomla！漏洞扫描器，可以对使用Joomla！CMS开发的网站进行自动化的漏洞检测和报告，其轻量化和模块化的架构能够保证在扫描过程中不会留下过多痕迹。

使用前需要安装，命令：`apt install joomscan`

使用：`jomscan -u 域名/ip`

扫描结束后会在/usr/share/joomscan/reports/目录下生成以Joomla网站URL命名的文件夹，其中有HTML格式的报告

###