

实验四： WEB 安全

一、实验目的

1. 搭建安全靶场，熟悉常见的 WEB 安全漏洞
2. 了解并掌握相关工具寻找漏洞及注入点
3. 掌握漏洞的保护方式

二、实验平台

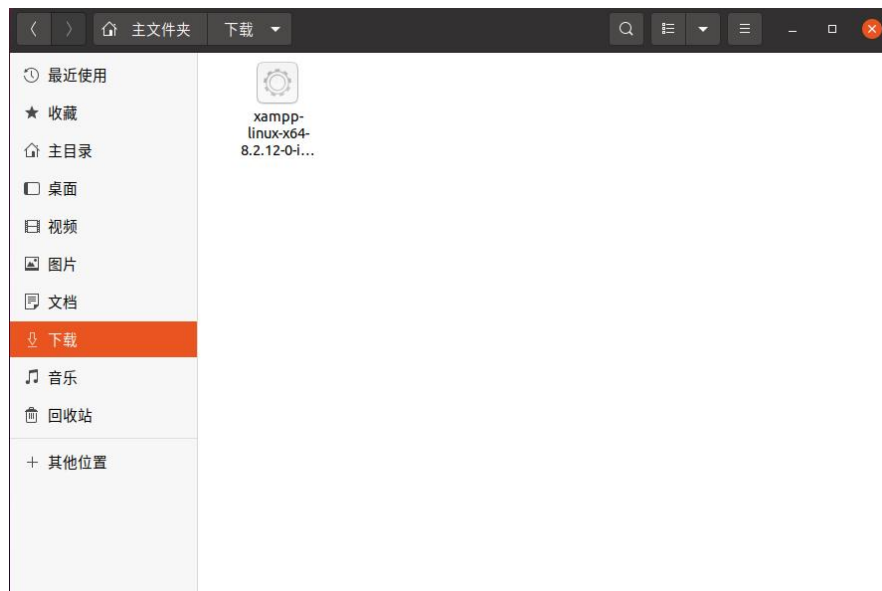
Server: ubuntu 虚拟机

Attacker: ubuntu 虚拟机，与 server 处于同一网段（局域网）

三、实验过程及结果分析

步骤一：DVWA 靶场搭建

本步骤中，首先卸载原本安装的 apache2 服务。再 XAMPP 网站下载对应版本的 XAMPP，后使用 `sudo ./xampp-linux-x64-8.2.12-0-installer.run` 命令进行安装。安装过程如下：

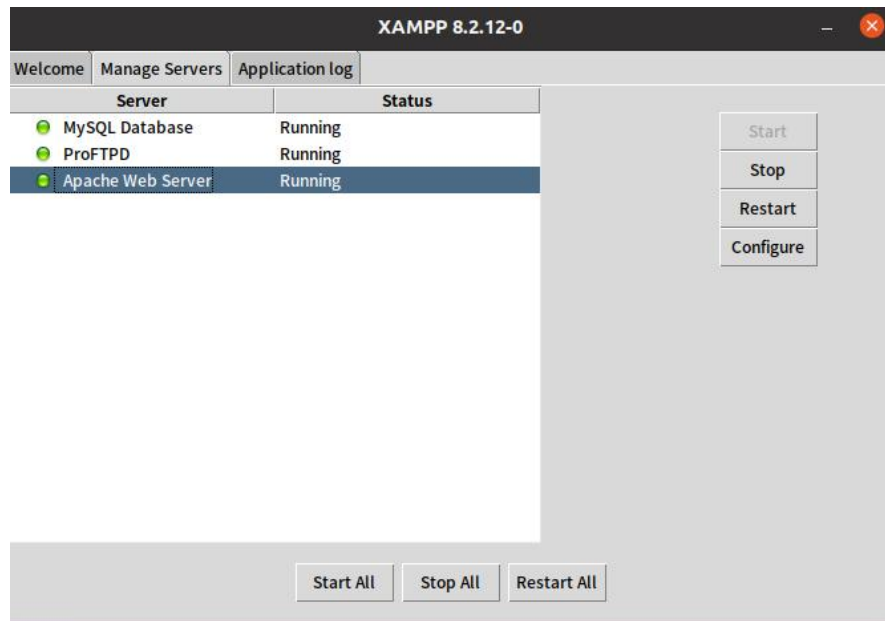


在命令行中输入 `chmod +x xampp-linux-x64-8.2.12-0-installer.run` 为安装包增加执行权限后使用 `sudo ./xampp-linux-x64-8.2.12-0-installer.run` 运行安装。

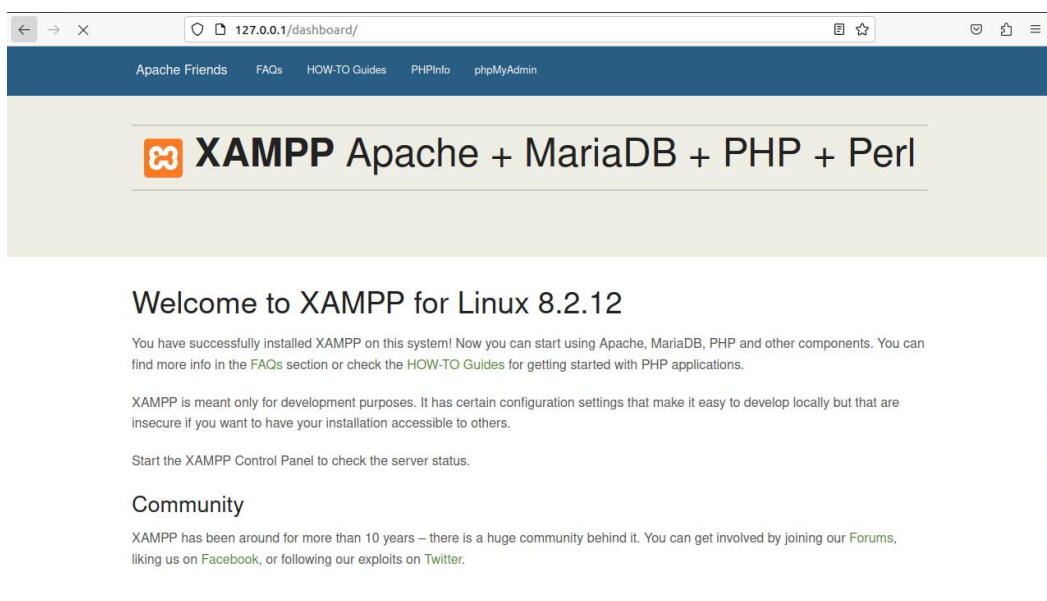
```
nakano9331@nakano9331-VirtualBox:~/下载$ chmod +x xampp-linux-x64-8.2.12-0-installer.run
nakano9331@nakano9331-VirtualBox:~/下载$ sudo ./xampp-linux-x64-8.2.12-0-installer.run
```

进入安装界面后，按实验指导要求进行操作，执行 `cd` 命令进入 `/opt/lampp` 文件夹，使

用 `sudo ./manager-linux-x64.run` 命令将下面各项服务开启。



开启服务后进入浏览器出现如下界面安装成功。

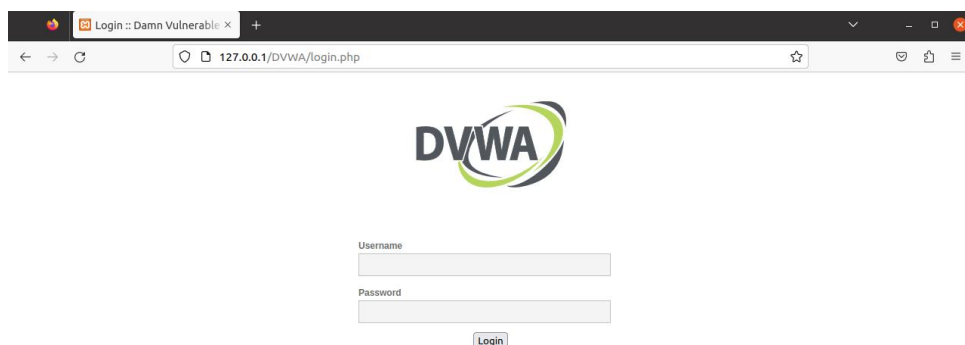


点击 open application folder, 进入 htdocs 页面，发现文件夹中并未存在 DVWA 靶场文件，因此执行 `sudo git clone https://github.com/digininja/DVWA.git` 将靶场文件复制到环境中。

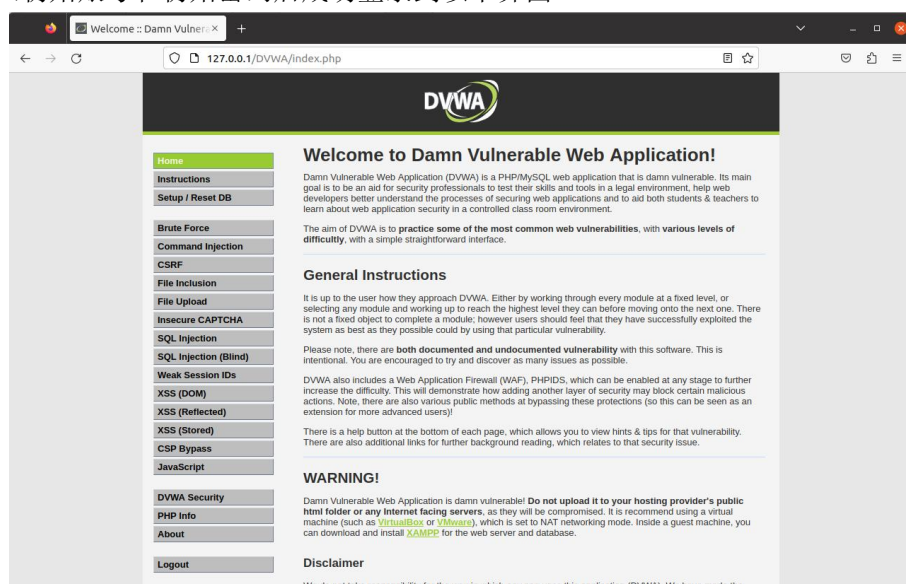


在浏览器中再次打开靶场, 登陆 `127.0.0.1/DVWA/login.php` 发现出现报错, 由实验指导书中步骤执行 `sudo cp /opt/lampp/htdocs/DVWA/config/config.inc.php.dist /opt/lampp/htdocs/DVWA/config/config.inc.php` 命令后将 user 参数和 password 参数分别修改为 “root” 和 “” 之后再次访问登录页面。

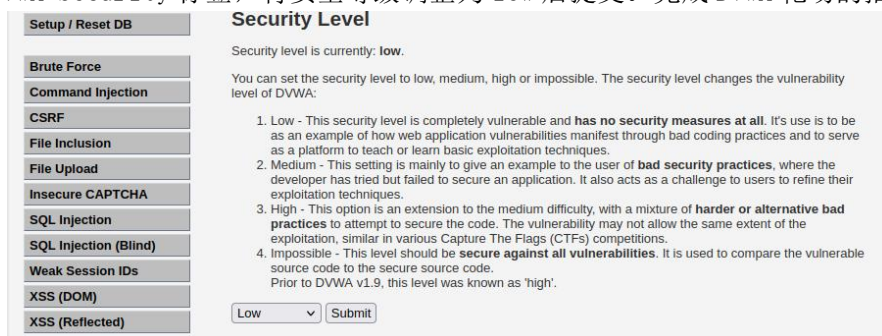
此时发现登录页面报错消失但是显示空白，通过查找相关 DVWA 使用指导，访问 <http://127.0.0.1/DVWA-master/setup.php> 后点击创建数据库，成功进入登录页面。



输入初始账号和初始密码后成功登录到以下界面。



选择 DVWA Security 标签，将安全等级调整为 low 后提交。完成 DVWA 靶场的搭建。



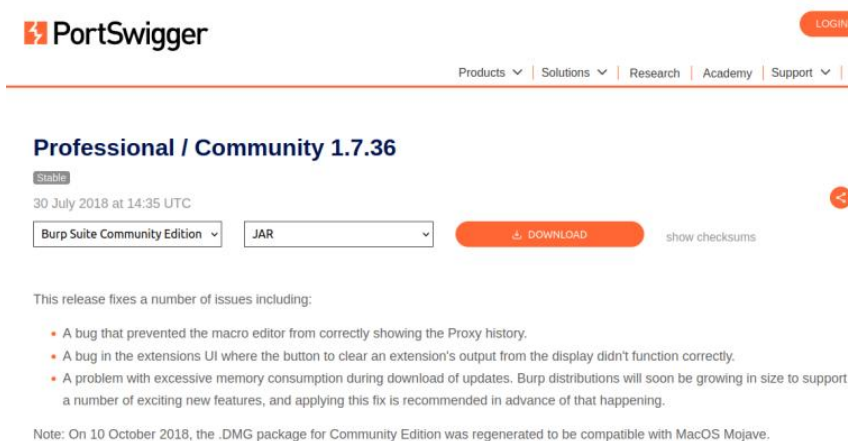
步骤二：Brute Force 暴力破解

本步骤需要用到的工具为 Burp Suite，Burp Suite 是一个集成化的渗透测试工具，它集合了多种渗透测试组件，使我们自动化地或手工地能更好的完成对 web 应用的渗透测试和攻击。

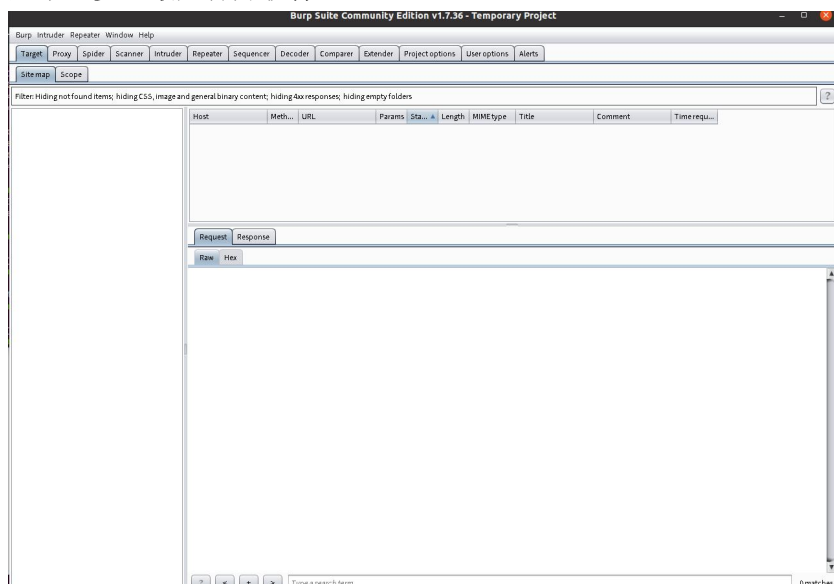
为了完成相关实验操作，首先使用 `sudo apt-get install openjdk-8-jdk` 命令进行安装 java 环境。安装后查看当前版本。

```
nakano9331@nakano9331-VirtualBox:~$ java -version
openjdk version "1.8.0_432"
OpenJDK Runtime Environment (build 1.8.0_432-8u432-ga~us1-0ubuntu2~20.04-ga)
OpenJDK 64-Bit Server VM (build 25.432-bga, mixed mode)
nakano9331@nakano9331-VirtualBox:~$
```

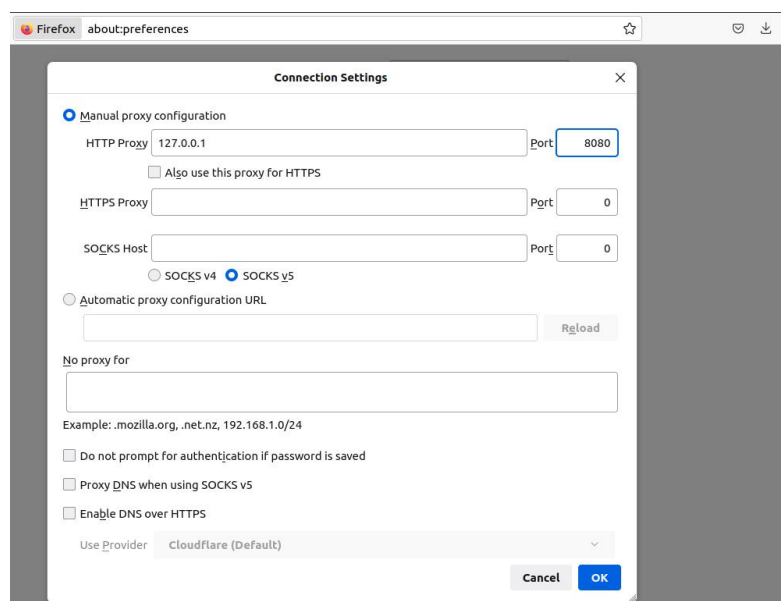
为运行 burpsuite，首先在访问官网下载相关安装包。



后在下载文件夹执行 `sudo java -jar burpsuite_community_v1.7.36.jar` 命令安装并使用默认配置下一步，最后打开软件。



首先，对浏览器的代理 ip 进行设置，ip 设置为 127.0.0.1，端口为 8080。



接下来填写账号密码提交登录，可以发现软件拦截到了包，并且发现账号和密码是明文传输的。

Request to http://127.0.0.1:80

Forward Drop Intercept on Action

Raw Params Headers Hex

GET /DWA/vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1

Host: 127.0.0.1

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/111.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: close

Referer: http://127.0.0.1/DWA/vulnerabilities/brute/

Cookie: security=low; PHPSESSID=b37cbgnj4fhsjinn8afp3vvepk

Upgrade-Insecure-Requests: 1

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: same-origin

Sec-Fetch-User: ?1

GET request to /DWA/vulnerabilities/brute/

Type	Name	Value
URL	username	admin
URL	password	password
URL	Login	Login
Cookie	security	low
Cookie	PHPSESSID	b37cbgnj4fhsjinn8afp3vvepk

点击 action 选择 send to Intruder，将包发送到 Intruder 模块，准备构造爆破包。该模块默认会将所有参数自动标记，但由于 low 模式下只需要爆破密码，因此点击 clear 先清除自动标记的参数，然后手动选中密码，按 add 标记参数。

Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions- see help for full details.

Attack type: Sniper

GET /DWA/vulnerabilities/brute/?username=admin&password=\$password&Login=Login HTTP/1.1

Host: 127.0.0.1

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/111.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: close

Referer: http://127.0.0.1/DWA/vulnerabilities/brute/

Cookie: security=low; PHPSESSID=b37cbgnj4fhsjinn8afp3vvepk

Upgrade-Insecure-Requests: 1

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: same-origin

Sec-Fetch-User: ?1

Add \$

Clear \$

Auto \$

Refresh

点击 payload 标签，在 simple list 中添加字典，添加完成后点击 start attack。后查看结果，寻找与其他包返回的长度不一样的爆破正确密码包。

Requ...	Payload	Status	Error	Time...	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4584	
1	pass1rd	200	<input type="checkbox"/>	<input type="checkbox"/>	4584	
2	plssword	200	<input type="checkbox"/>	<input type="checkbox"/>	4584	
3	p2ssw0rd	200	<input type="checkbox"/>	<input type="checkbox"/>	4584	
4	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4627	
5	pk1sw0rd	200	<input type="checkbox"/>	<input type="checkbox"/>	4584	
6	p123w09d	200	<input type="checkbox"/>	<input type="checkbox"/>	4584	

然后返回页面，输入密码登录，发现登陆成功。

Vulnerability: Brute Force


Login

Username:

Password:

Login

Welcome to the password protected area admin



保持原界面，在用户名一栏输入 admin' or '1' = '1，密码留空，发现还是可以成功登陆，说明该页面存在 SQL 注入漏洞，成功完成相关实验。



步骤三：基于 ARP 协议漏洞的中间人攻击

首先在在两台虚拟机上分别在命令行中输入 ifconfig，查看 IP 地址。得到攻击端的 IP 地址为 192.168.1.100，被攻击端的 IP 地址为 192.168.1.120，后面分别执行 ping 命令。

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe5d:e3db prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:5d:e3:db txqueuelen 1000 (Ethernet)
    RX packets 11493 bytes 1066989 (1.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5482 bytes 418165 (418.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4744 bytes 502805 (502.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4744 bytes 502805 (502.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

nakno@nakno-VirtualBox:~$ ping 192.168.1.120
PING 192.168.1.120 (192.168.1.120) 56(84) bytes of data.
64 bytes from 192.168.1.120: icmp_seq=1 ttl=64 time=0.246 ms
64 bytes from 192.168.1.120: icmp_seq=2 ttl=64 time=0.336 ms
64 bytes from 192.168.1.120: icmp_seq=3 ttl=64 time=0.259 ms
```

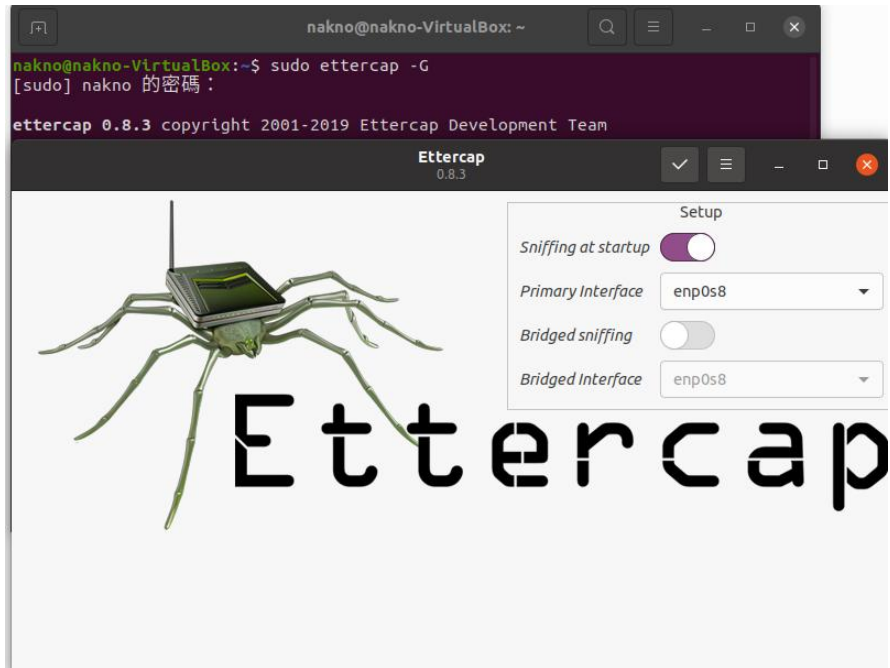
```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.120 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe26:6bbc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:26:6b:bc txqueuelen 1000 (以太网)
    RX packets 7884 bytes 777753 (777.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8938 bytes 575691 (575.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 5757 bytes 597806 (597.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5757 bytes 597806 (597.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

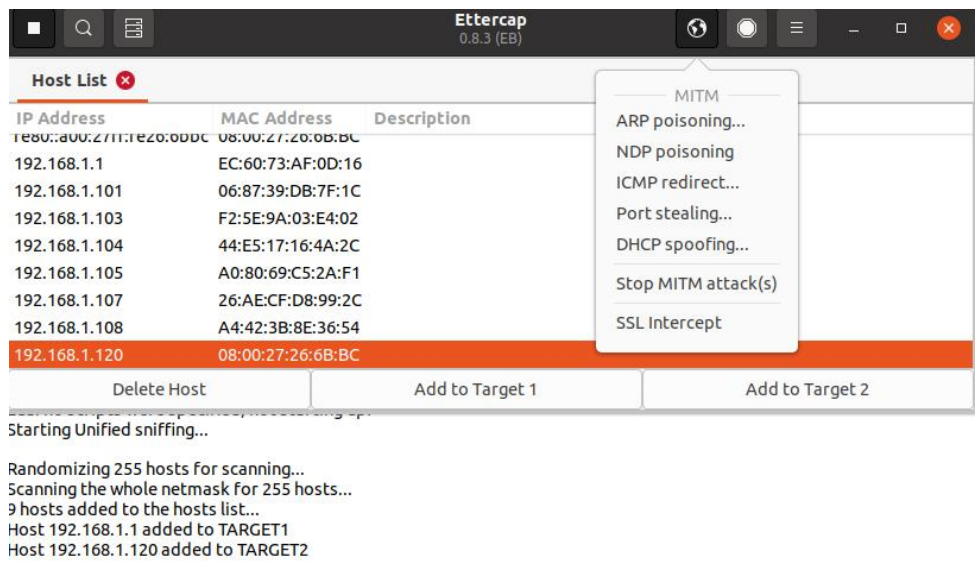
nakano9331@nakano9331-VirtualBox:~$ ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 字节, 来自 192.168.1.100: icmp_seq=1 ttl=64 时间=0.221 毫秒
64 字节, 来自 192.168.1.100: icmp_seq=2 ttl=64 时间=0.228 毫秒
64 字节, 来自 192.168.1.100: icmp_seq=3 ttl=64 时间=0.180 毫秒
```

观察到其二者可以双向 ping 通，后在 ubuntu 终端窗口输入分别执行下面的两条指令：
sudo apt-get install ettercap-common 和 sudo apt-get install driftnet，执行后安装完成后输入 sudo ettercap -G 打开软件，选择 sniff 标签下的 unified sniffing，具体

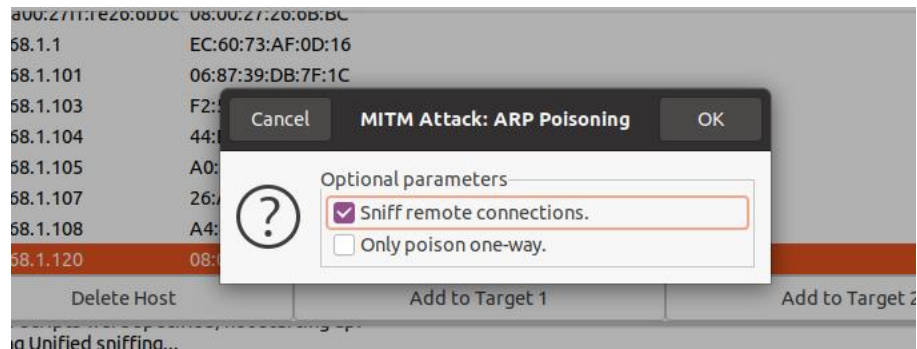
操作如下所示。



选择 host 下的 scan for hosts 扫描主机，完成后选择 hosts list 打开，点击网关，选择 add target1，再点击受害者的主机，选择 add target2。



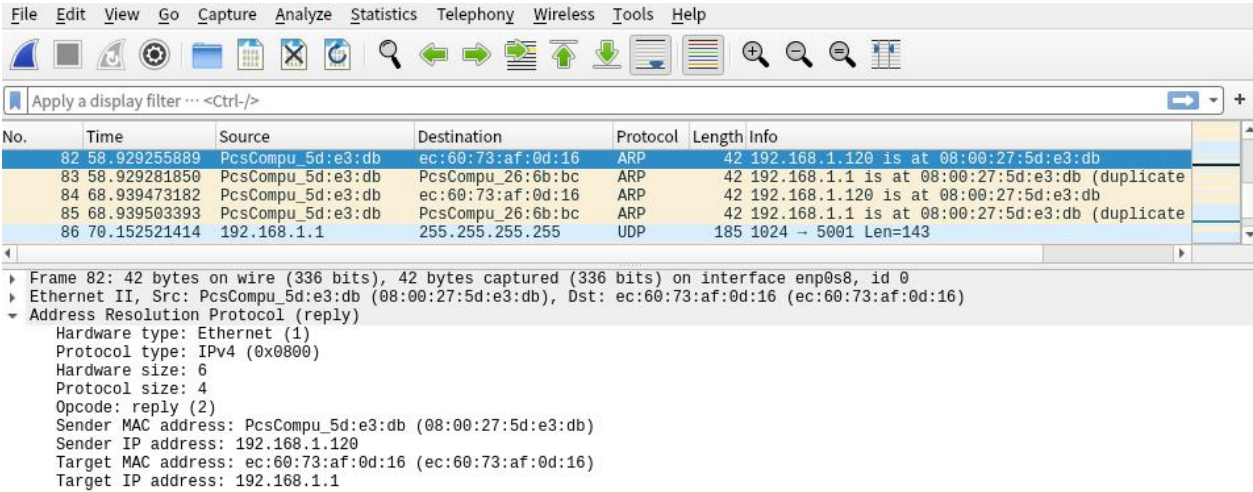
完成后选择标签 Mitm 中 ARP poisoning, 勾选第一个选项然后确定，这个时候就已经成功的发动了中间人攻击，受害者主机与网关之间的通信都会通过当前的攻击机



通过在受害主机上查看 arp 缓存表执行命令 `arp -a`，可以看到接口的地址是攻击者主机说明攻击成功。

```
nakano9331@nakano9331-VirtualBox:~$ arp -a
? (192.168.1.1) 位于 08:00:27:5d:e3:db [ether] 在 enp0s8
? (192.168.1.100) 位于 08:00:27:5d:e3:db [ether] 在 enp0s8
? gateway (10.0.2.2) 位于 52:54:00:12:35:02 [ether] 在 enp0s3
```

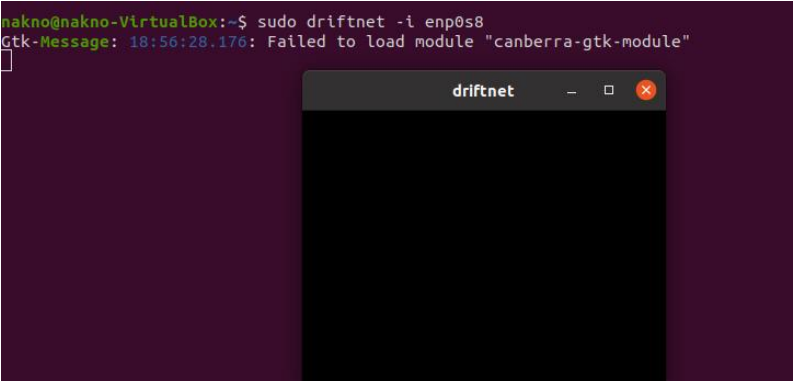
在攻击者主机上打开 wireshark 进行抓包，查看 arp 协议数据包的内容并解释。



ARP 数据包主要有两种类型分别是 ARP 请求和 ARP 响应。ARP 请求是由源主机发出的，请求网络中的所有主机告诉源主机某个 IP 地址对应的 MAC 地址。从图中看出 ARP 响应数据包中 Src 显示源主机的硬件地址为 00:00:27:5d:e3:db。Dst 显示发起请求的主机的硬件地址为 ec:00:73:af:0d:16。

其中 Hardware Type 表示硬件类型为 1，表示以太网。Protocol Type 是协议类型为 0x0800 表示 IP 协议。Hardware Size 是硬件地址长度，以字节为单位，对于以太网是 6。Protocol Size 协议地址长度以字节为单位，对于 IPv4 是 4。Operation 表示操作类型其中 2 表示 ARP 响应。Sender MAC Address 表示请求或响应包中发起方的 MAC 地址为 00:00:27:5d:e3:db。Sender IP Address 显示请求或响应包中发起方的 IP 地址为 192.168.1.120。Target MAC Address 表示请求包中的目标 MAC 地址为 ec:00:73:af:0d:16。Target IP Address 显示请求或响应包中的目标 IP 地址为 192.168.1.1。

在攻击者主机上执行 `sudo driftnet -i enp0s8` 后，在受害者主机上打开一个未加密的 http 网页，然后观察 driftnet 的嗅探情况。



这里选择旧版 XJTU 教务处网页 <http://cmsm.xjtu.edu.cn/jdjwc/> 进入观察，可以发现 driftnet 已嗅探到了图片，说明中间人攻击生效。



步骤四：漏洞防御方法思考

对于暴力破解（Brute Force）攻击，可以采取强化密码策略，确保用户使用复杂密码，包括字母、数字和特殊字符的组合。定期要求用户更换密码，以减少攻击者利用长期有效密码的机会。或是限制登录尝试次数，即对每个用户在一定时间内的登录尝试次数进行限制，超过预定的失败次数后，暂时锁定账户，阻止进一步尝试。此外，还可以设置账户锁定时间，减缓攻击者的破解速度。在此基础上，引入双因素认证，在用户名和密码的基础上增加第二层身份验证，例如通过短信、移动应用或硬件令牌生成验证码，从而进一步提高账户安全性。

对于防御 SQL 注入攻击，可以采用参数化查询或预编译语句的方法，避免用户输入直接嵌入 SQL 语句，从而有效防止注入攻击。同时严格验证和过滤用户输入以限制用户输入中只能包含合法字符和格式，拒绝任何可能的恶意载荷。并实施最小权限原则，为数据库用户分配最小权限，避免使用高权限账户，限制潜在攻击者的访问范围。为了防止系统僵化，定期更新系统、定期安全审计。及时应用数据库系统和应用程序的安全更新，修复已知漏洞。记录数据库访问和查询日志，及时发现并响应潜在威胁。

在这些措施的基础是，可以使用如 Hibernate 或 Entity Framework ORM 的框架，提供更安全的数据库访问接口。并利用数据库防火墙监控网络中的 SQL 注入攻击，检测异常活动并采取相应措施。

对于基于 ARP 协议漏洞的中间人攻击，可以启用静态 ARP 绑定，将 MAC 地址与 IP 地址绑定，减少动态 ARP 攻击的风险。同时，使用 ARP 防护工具监控网络中的 ARP 请求和响应，检测异常活动并采取措施。并采用网络隔离将敏感信息存储在专用子网中，降低攻击风险。还可以实施流量加密，使用 TLS/SSL 加密网络通信，防止攻击者在中间截取或篡改数据。在此基础上，定期检查网络设备的安全设置，确保没有漏洞可以被利用进行 ARP 攻击。

四、实验原理

1. 搭建安全靶场

在本实验中搭建了一个安全靶场，主要目的是为测试常见的 WEB 安全漏洞提供环境。使用了 XAMPP 和 DVWA，即 Damn Vulnerable Web Application 作为测试平台。XAMPP 提供了一个集成的服务器环境来支持应用程序的运行，而 DVWA 则是一个专门设计用来测试和演示多种 WEB 安全漏洞的应用。

2. 暴力破解密码

暴力破解密码是一种通过系统地尝试所有可能的密码组合来破解密码的攻击方式。暴力破解的攻击者会尝试所有可能的字符组合，直到找到正确的密码。这个过程可以通过编写程序来自动化完成，通常通过“穷举法”暴力列举密码空间中的所有可能的密码，逐一尝试每个密码。

密码空间是所有可能密码的集合，取决于密码的长度和字符集的大小。攻击者利用自动

化工具进行密码爆破，工具会按照一定的规则，如字符集、密码长度等逐个生成密码组合，然后将这些密码与目标系统的加密密码进行比对，直到找到正确的密码。

3. SQL 注入漏洞

SQL 注入是一种常见的网络攻击方式，攻击者通过将恶意的 SQL 命令插入到 Web 表单或页面请求的查询字符串中，欺骗服务器执行这些恶意的 SQL 命令，从而实现未授权的数据库操作。攻击者利用这种漏洞可以进行数据库的读取、修改、删除等操作，甚至获取敏感信息。

实验原理

SQL 注入漏洞的根本原因是 Web 服务未能对用户输入的数据进行适当的过滤或验证，特别是没有过滤掉 SQL 命令中的关键字或特殊字符。通常，Web 应用程序会将用户的输入直接传递给 SQL 查询，而如果这些输入包含特殊的 SQL 字符(如单引号 '、双引号 "、分号 ; 等)，就可能导致 SQL 命令被篡改，产生恶意的执行效果。

SQL 注入的发生主要是由于 Web 应用没有对用户输入的数据进行正确的过滤，导致恶意用户输入的 SQL 命令直接被执行。常见的 SQL 注入攻击方式包括：①错误注入：通过在输入框中输入恶意 SQL 代码，引发错误，暴露数据库结构信息。②盲注：在页面没有错误反馈的情况下，通过逐步推测数据库信息来进行攻击。③联合查询注入：利用 SQL 的 UNION 操作符，将额外的查询结果合并到原查询中，从而泄露敏感数据。

4. 中间人攻击

在本实验中，使用了 Ettercap 和 Driftnet 等工具进行 ARP 欺骗攻击。在攻击者主机上伪装成网关与受害者通信，从而可以拦截和篡改受害者与网关之间的通信。ARP 欺骗攻击通过伪造网络包，将攻击者的 MAC 地址欺骗性地插入到网络中的 ARP 表里，使得网络流量经过攻击者主机。攻击者可以监控、篡改或阻止流量，甚至发起其他更复杂的攻击。

五、实验小结

本次网络安全实验涵盖了 WEB 安全漏洞 的探索与防御,包括对暴力破解(Brute Force)、SQL 注入攻击、以及基于 ARP 协议的中间人攻击的防御策略的实践操作。

暴力破解攻击通过配置强密码策略、限制登录尝试次数、引入双因素认证等措施防御，有效降低了暴力破解的成功率。通过限制用户登录的次数和强制使用复杂密码，可以有效减少攻击者通过暴力破解的机会。

SQL 注入通过使用参数化查询、输入验证、最小权限原则等方式防御，能够防止 SQL 注入漏洞的发生。尤其是使用 ORM 框架与数据库防火墙，有助于更安全地访问数据库，并及时监控和防范异常活动。

ARP 协议中间人攻击可以通过启用静态 ARP 绑定、使用 ARP 防护工具和加密通信等方法防御，能够有效防止基于 ARP 协议的中间人攻击。在实际应用中，加强网络隔离和定期审查网络配置也是降低攻击风险的重要步骤。

实验中使用了诸如 Burp Suite、Wireshark、Ettercap 和 Driftnet 等工具，这些工具可以更好地分析网络攻击的原理以及了解如何是进行的攻击。通过模拟攻击和拦截数据包，可以清晰地看到暴力破解和中间人攻击的实际效果。

六、遇到的问题解决方法

本次实验遇到的问题主要在第一步骤对于靶场的搭建,由于自身设备网络原因一直下载失败,后来通过反复更换下载节点成功完成下载配置。

其次是第三步骤，在 Ettercap 和 Driftnet 的使用时，由于操作不熟悉，一直无法再 host 列表中扫描到网关地址，加上对于相关设置不理解，多次尝试后均无法成功。后来通过搜索网络资料，最终在 <https://ask.csdn.net/questions/7992315> 处初步了解大致原因，后进行尝试重新配置桥接网络后完成实验。

七、实验心得

本次实验不仅让我在技术层面上提升了对网络安全攻击与防护的理解，更加深了我对安全的整体意识。在实验过程中，我认识到，网络安全不仅仅是技术问题，还涉及到管理和流程层面的防护。只有持续关注和更新安全策略，确保系统在不同阶段都能得到有效的保护，才能最大程度地降低攻击的风险。

此次实验是一次非常有价值的学习经历，不仅让我掌握了网络攻击的基本原理，也加深了我对防护措施的理解。通过实践，我认识到，面对网络安全威胁，技术防护只是其中的一部分，良好的安全意识、及时的更新和适当的工具使用，都是确保系统安全不可忽视的环节。在未来的学习和工作中，我将继续加强网络安全方面的知识，提升自己的安全防护能力。