

网络与信息安全课内实验 4： WEB 安全实验报告

班级：计算机 2101 班

姓名：田濡豪

学号：2203113234

1 实验平台及环境

本次实验使用个人计算机完成。

使用 WSL（Windows Subsystem for Linux）完成，所安装的 Ubuntu 系统版本为：

Ubuntu 22.04.4 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

对于需要 Ubuntu 图形化界面或需要多台虚拟机同时运行的实验步骤，选择使用 Docker 进行容器化部署。Docker 镜像可以方便的进行修改、销毁、容器并发，性能高于不同虚拟机且不影响宿主机，为实验提供了高灵活性和容错空间。所选用的镜像版本为：

kasmweb/desktop:1.16.1-rolling-weekly

对于每个实验步骤的具体环境配置，将在对应章节说明。

2 实验步骤

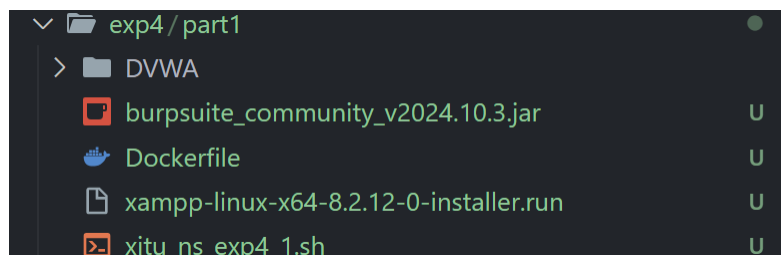
1. 爆破登录
2. SQL 注入
3. 中间人攻击

3 实验过程

3.1 爆破登录

3.1.1 环境配置

下载并准备好实验中需要用到的 XAMPP 安装包，DVWA 文件夹和 Burp Suite JAR 文件，复用在实验 2 中配置的 Dockerfile。



修改 Dockerfile，将文件复制到 root 用户目录。观察 XAMPP 的命令行安装过程，将所需安装输入一并添加到 Dockerfile 的层中。

```
# copy xampp installer, DVWA folder, and burpsuite jar to home directory
COPY ./xampp-linux-x64-8.2.12-0-installer.run $HOME
COPY ./DVWA $HOME/DVWA
COPY ./burpsuite_community_v2021.8.1.jar $HOME
```

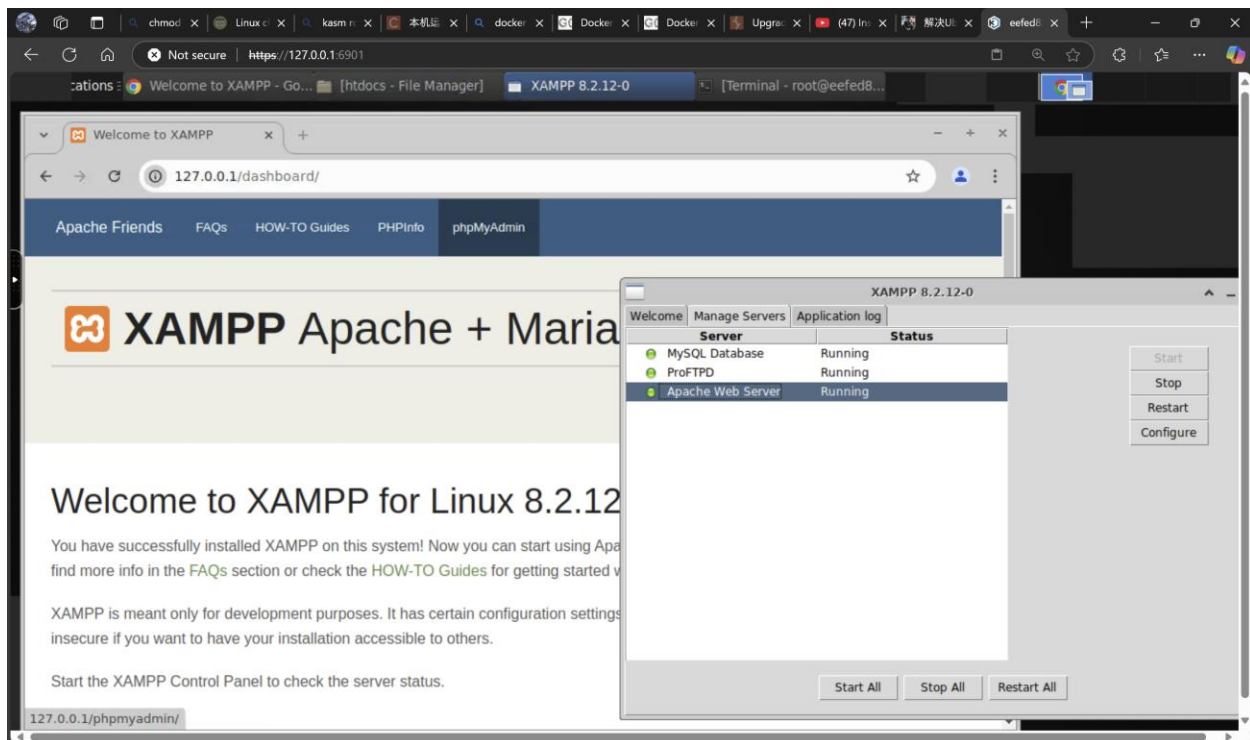
```
...# ca certificates
...&& sudo apt-get install -y ca-certificates \
...# xampp
...&& chmod +x $HOME/xampp-linux-x64-8.2.12-0-installer.run \
...&& echo -e "Y""Y""\n""Y" | $HOME/xampp-linux-x64-8.2.12-0-installer.run
```

构建并运行镜像，发现无法打开 XAMPP 的图形界面。查阅资料后发现这是因为 XAMPP 需要 su 权限，但是 xhost 政策不允许非登录用户映射图形窗口到登录用户的显示器。使用 xhost + 命令修改访问控制权限，即可正常运行 XAMPP。

```
Terminal - root@eefed8307507: /opt/lampp
File Edit View Terminal Tabs Help

:1.0
default:/opt/lampp$ sudo -s
root@eefed8307507:/opt/lampp# ./manager-linux-x64.run
No protocol specified
No protocol specified
Unknown Error couldn't connect to display ":1.0"
root@eefed8307507:/opt/lampp# export DISPLAY=:0.0
root@eefed8307507:/opt/lampp# ./manager-linux-x64.run
Unknown Error couldn't connect to display ":0.0"
root@eefed8307507:/opt/lampp# export DISPLAY=:1.0
root@eefed8307507:/opt/lampp# ./manager-linux-x64.run
No protocol specified
No protocol specified
Unknown Error couldn't connect to display ":1.0"
root@eefed8307507:/opt/lampp# xhost +
No protocol specified
xhost: unable to open display ":1.0"
root@eefed8307507:/opt/lampp# exit
exit
default:/opt/lampp$ xhost +
access control disabled, clients can connect from any host
default:/opt/lampp$ sudo -s
root@eefed8307507:/opt/lampp# ./manager-linux-x64.run
```

运行 Server 服务，可见 XAMPP 已经成功安装。



完整的 Docker 镜像配置文件如下：

```

#Dockerfile
#sudo as root role.

FROM kasmweb/desktop:1.16.1-rolling-weekly
USER root

ENV HOME=/home/kasm-default-profile
ENV STARTUPDIR=/dockerstartup
ENV INST_SCRIPTS=$STARTUPDIR/install
WORKDIR $HOME

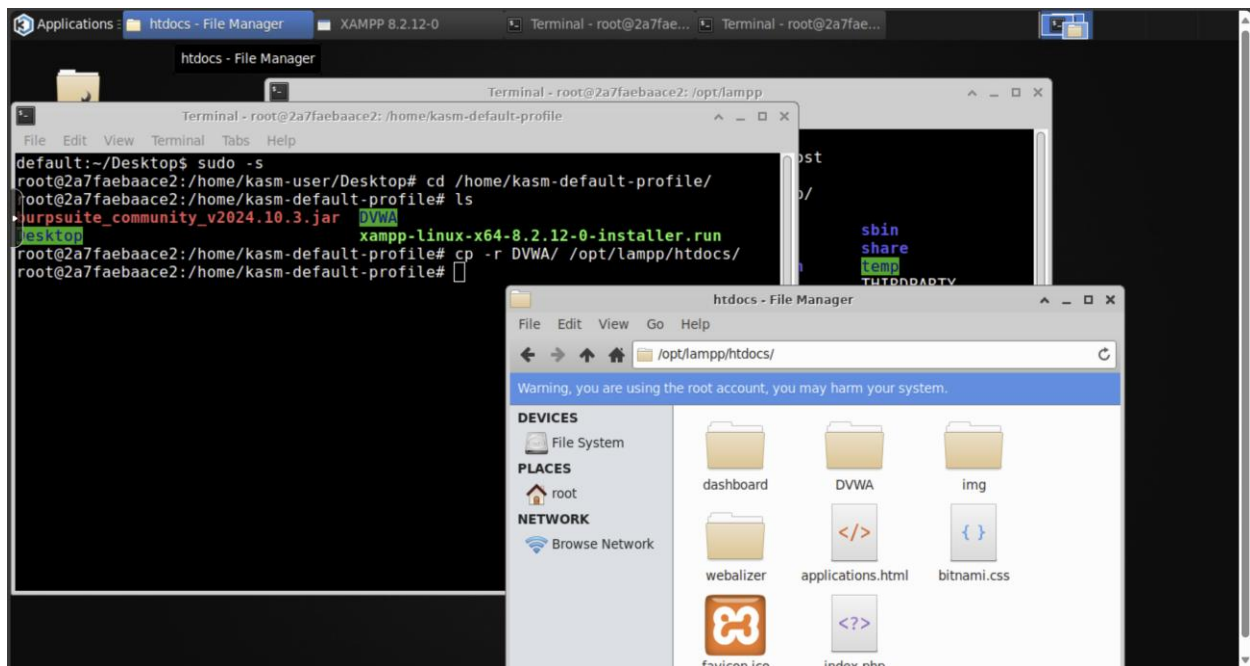
##### Customize Container Here #####
ENV TZ=Asia/Shanghai
# copy xampp installer, DVWA folder, and burpsuite jar to home directory
COPY ./xampp-linux-x64-8.2.12-0-installer.run $HOME
COPY ./DVWA $HOME/DVWA
COPY ./burpsuite_community_v2024.10.3.jar $HOME
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone \
    && apt-get update \
    && apt-get install -y sudo \
    && echo 'kasm-user ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers \
    && rm -rf /var/lib/apt/list/* \
    && chown 1000:0 $HOME \
    && $STARTUPDIR/set_user_permission.sh $HOME \
    # basic tools
    && sudo apt-get install -y gedit \
    && sudo apt install -y iputils-ping \
    # wireshark
    && apt install -y wireshark \
    # ca certificates
    && sudo apt-get install -y ca-certificates \
    # xmapp
    && chmod +x $HOME/xampp-linux-x64-8.2.12-0-installer.run \
    && echo -e "Y""Y""\n""Y" | $HOME/xampp-linux-x64-8.2.12-0-installer.run
##### End Customizations #####
ENV HOME=/home/kasm-user
WORKDIR $HOME
RUN mkdir -p $HOME && chown -R 1000:0 $HOME
USER 1000

```

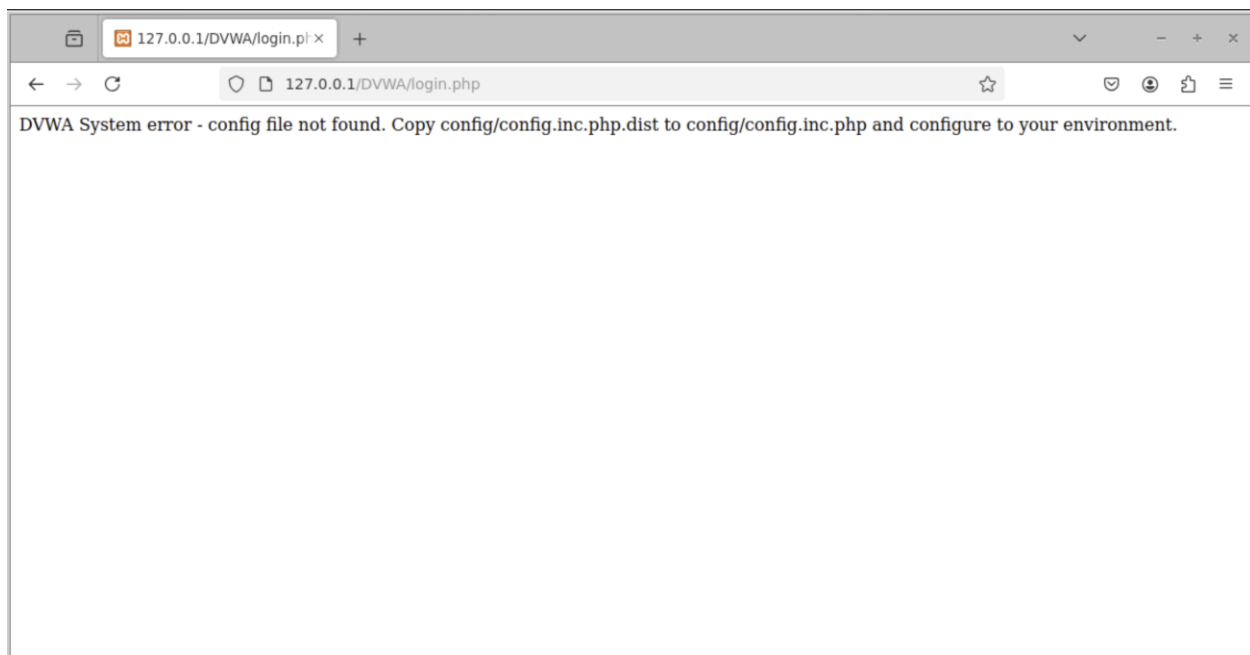
其中 xhost +命令需要登陆后手动执行。

3.1.2 DVWA 靶场搭建

复制靶场文件。



出现报错，按照教程提示更改并应用配置文件。




```
Open + config.inc.php /opt/lampp/htdocs/DVWA/config Save - + x
2
3 # If you are having problems connecting to the MySQL database and all of the variables below
  are correct
4 # try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to
  sockets.
5 # Thanks to @digininja for the fix.
6
7 # Database management system to use
8 $DBMS = getenv('DBMS') ?: 'MySQL';
9 $DBMS = 'PGSQL'; // Currently disabled
10
11 # Database variables
12 # WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
13 # Please use a database dedicated to DVWA.
14 #
15 # If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
16 # See README.md for more information on this.
17 $DVWA = array();
18 $DVWA['db_server'] = getenv('DB_SERVER') ?: '127.0.0.1';
19 $DVWA['db_database'] = getenv('DB_DATABASE') ?: 'dvwa';
20 $DVWA['db_user'] = getenv('DB_USER') ?: 'root';
21 $DVWA['db_password'] = getenv('DB_PASSWORD') ?: '';
22 $DVWA['db_port'] = getenv('DB_PORT') ?: '3306';
23
24 # ReCAPTCHA settings
25 # Used for the 'Insecure CAPTCHA' module
26 # You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
27 $DVWA['recaptcha_public_key'] = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
28 $DVWA['recaptcha_private_key'] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';
29
30 # Default security level
31 # Default value for the security level with each session.
32 # The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or
  impossible'.
```

更改后访问出现白屏，仍不能正常显示。打开 XAMPP 中 SQL 数据库的 log 文件，发现该数据库使用 MariaDB。根据 config 文件中的提示，这种情况下无法直接使用 root 用户，必须创建一个专门的 DVWA user。

```
Open + 33c40142d660.err /opt/lampp/var/mysql Save - + x
config.inc.php 33c40142d660.err
35 2024-12-17 17:22:07 0 [Note] Added new Master info '' to nash table
36 2024-12-17 17:22:07 0 [Note] /opt/lampp/sbin/mysqld: ready for connections.
37 Version: '10.4.28-MariaDB' socket: '/opt/lampp/var/mysql/mysql.sock' port: 3306 Source distribution
38 2024-12-17 17:22:49 9 [Warning] Access denied for user 'dvwa'@'localhost' (using password: YES)
39 2024-12-17 17:33:22 0 [Note] /opt/lampp/sbin/mysqld (initiated by: unknown): Normal shutdown
40 2024-12-17 17:33:22 0 [Note] InnoDB: FTS optimize thread exiting.
41 2024-12-17 17:33:22 0 [Note] InnoDB: Starting shutdown...
42 2024-12-17 17:33:22 0 [Note] InnoDB: Dumping buffer pool(s) to /opt/lampp/var/mysql/ib_buffer_pool
43 2024-12-17 17:33:22 0 [Note] InnoDB: Instance 0, restricted to 250 pages due to innodb_buf_pool_dump_pct=25
44 2024-12-17 17:33:22 0 [Note] InnoDB: Buffer pool(s) dump completed at 241217 17:33:22
45 2024-12-17 17:33:23 0 [Note] InnoDB: Removed temporary tablespace data file: "ibtmp1"
46 2024-12-17 17:33:23 0 [Note] InnoDB: Shutdown completed; log sequence number 1764275; transaction id 2369
47 2024-12-17 17:33:23 0 [Note] /opt/lampp/sbin/mysqld: Shutdown complete
48
49 2024-12-17 17:33:23 1817 mysqld_safe mysqld from pid file /opt/lampp/var/mysql/33c40142d660.pid ended
50 2024-12-17 17:33:27 8432 mysqld_safe Starting mysqld daemon with databases from /opt/lampp/var/mysql
51 2024-12-17 17:33:27 0 [Note] Using unique option prefix 'key_buffer' is error-prone and can break in the future. Please use the full name
  'key_buffer_size' instead.
52 2024-12-17 17:33:27 0 [Note] Starting MariaDB 10.4.28-MariaDB source revision c8f2e9a5c0ac5905f28b050b7df5a9ffd914b7e7 as process 8583
53 2024-12-17 17:33:27 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
54 2024-12-17 17:33:27 0 [Note] InnoDB: Uses event mutexes
55 2024-12-17 17:33:27 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
56 2024-12-17 17:33:27 0 [Note] InnoDB: Number of pools: 1
57 2024-12-17 17:33:27 0 [Note] InnoDB: Using SSE2 crc32 instructions
58 2024-12-17 17:33:27 0 [Note] mysqld: 0 TMPFILE is not supported on /tmp (disabling future attempts)
59 2024-12-17 17:33:27 0 [Note] InnoDB: Initializing buffer pool, total size = 16M, instances = 1, chunk size = 16M
60 2024-12-17 17:33:27 0 [Note] InnoDB: Completed initialization of buffer pool
61 2024-12-17 17:33:27 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man
  page of setpriority().
62 2024-12-17 17:33:27 0 [Note] InnoDB: 128 out of 128 rollback segments are active.
63 2024-12-17 17:33:27 0 [Note] InnoDB: Creating shared tablespace for temporary tables
64 2024-12-17 17:33:27 0 [Note] InnoDB: Setting file /opt/lampp/var/mysql/ibtmp1 size to 12 MB. Physically writing the file full; please...
```

根据 DVWA README 中的提示，登录 MariaDB 并创建一个新用户。

```
Terminal - root@33c40142d660: /opt/lampp/bin
File Edit View Terminal Tabs Help
-> Ctrl-C -- exit!
Aborted
root@33c40142d660:/opt/lampp/bin# ./mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.4.28-MariaDB Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.001 sec)

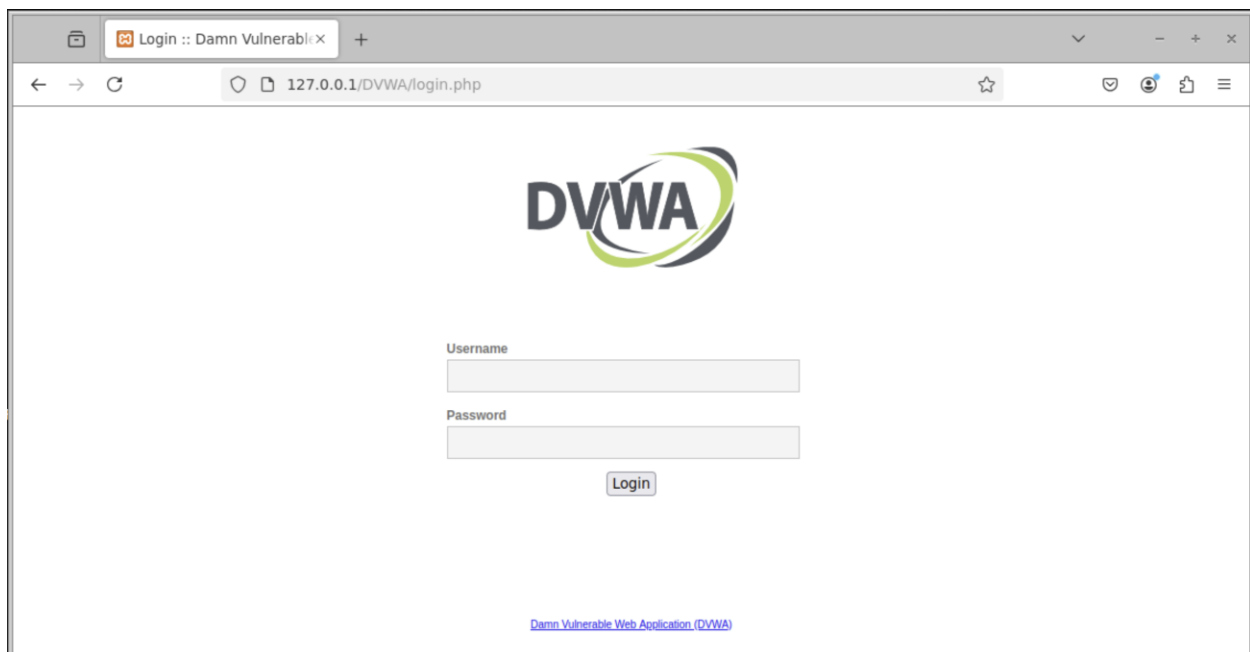
MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;
Query OK, 0 rows affected (0.001 sec)

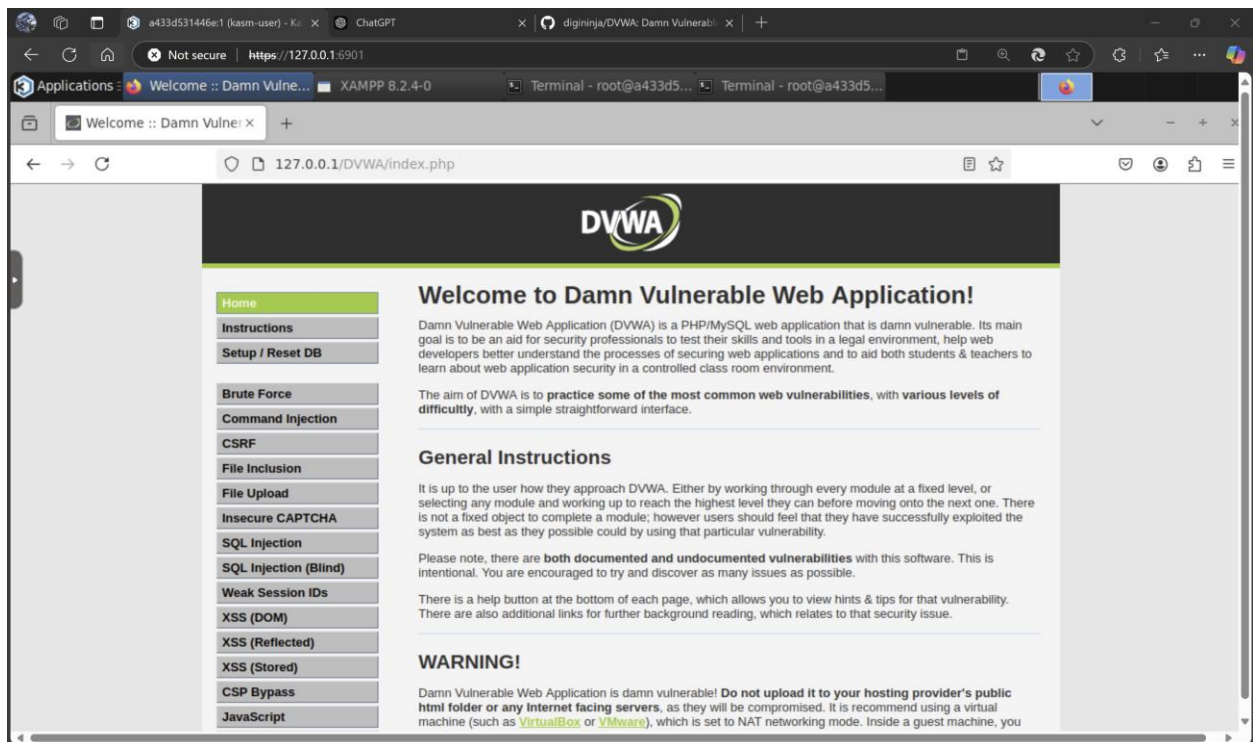
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> 
```

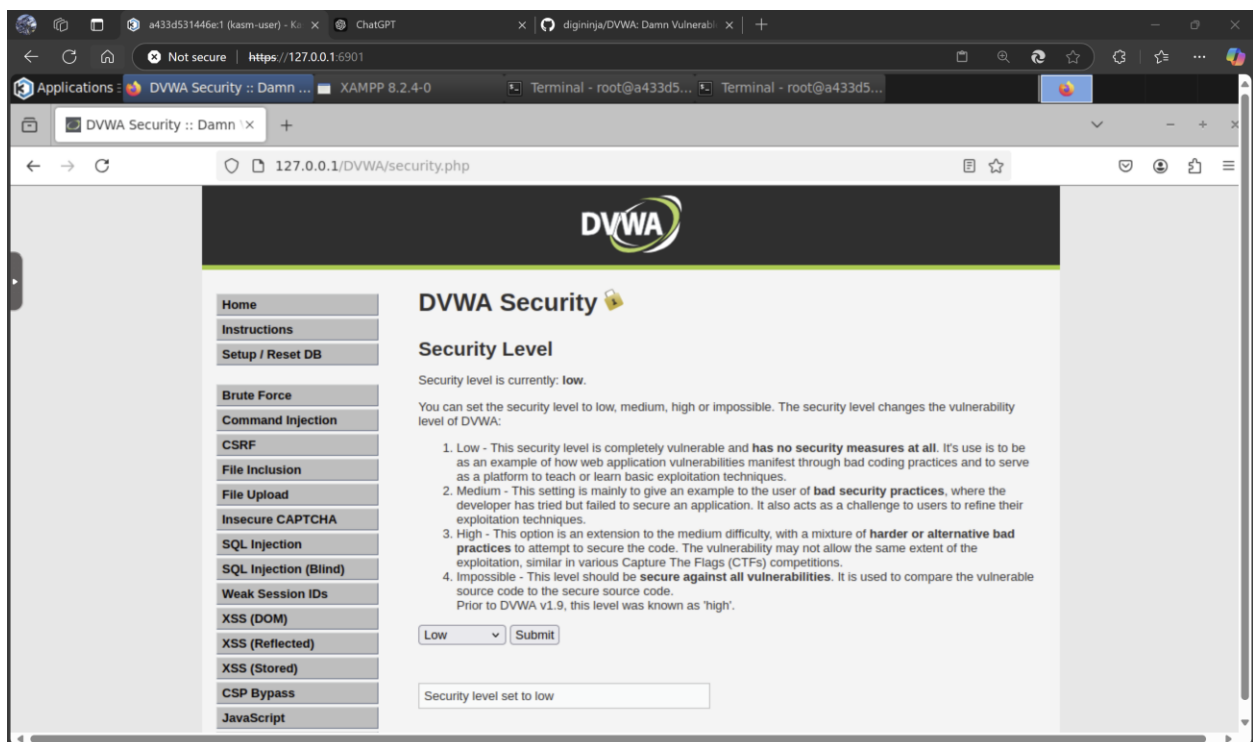
同时，还原 config 中的用户，使其密码、名称与在 MariaDB 中创建的用户一致。再次尝试方位 DVWA 界面，即可成功登录。



登陆后创建数据库，并使用默认用户名 admin 和默认密码 password 登录数据库。



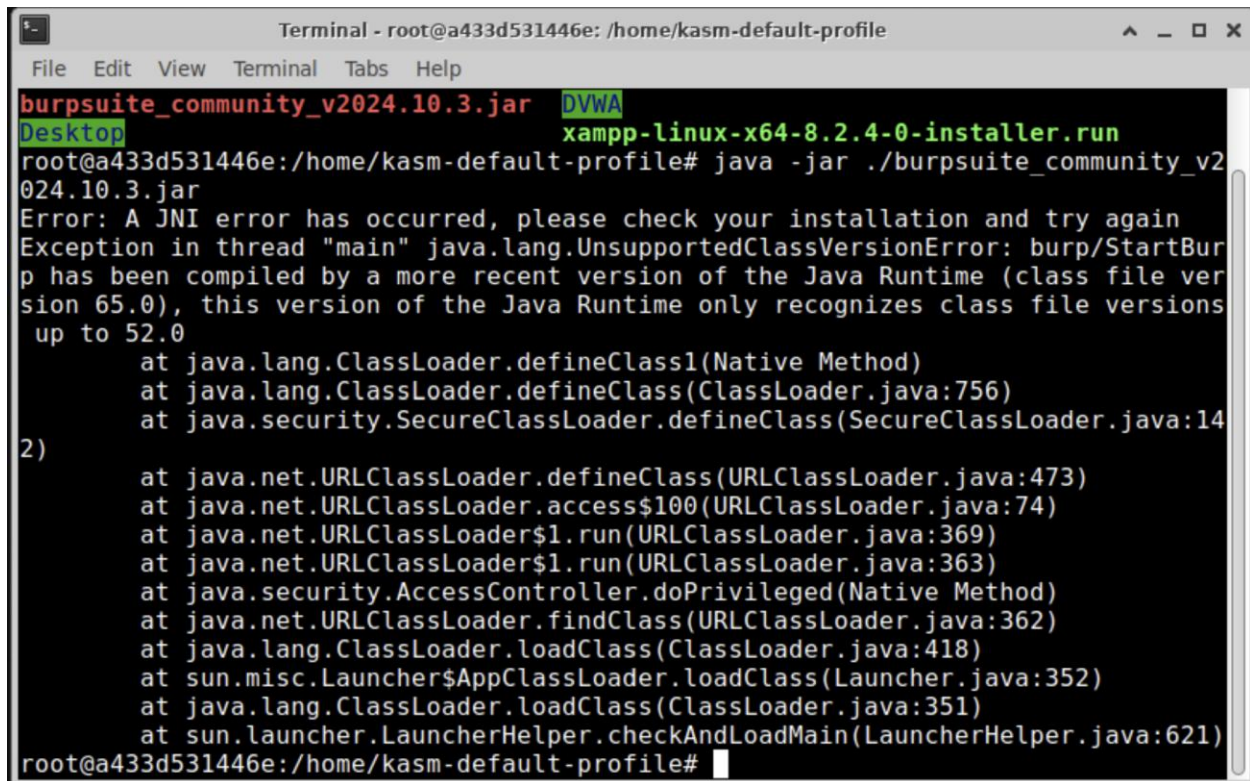
将安全等级设置为 low。



至此，靶场搭建完成。

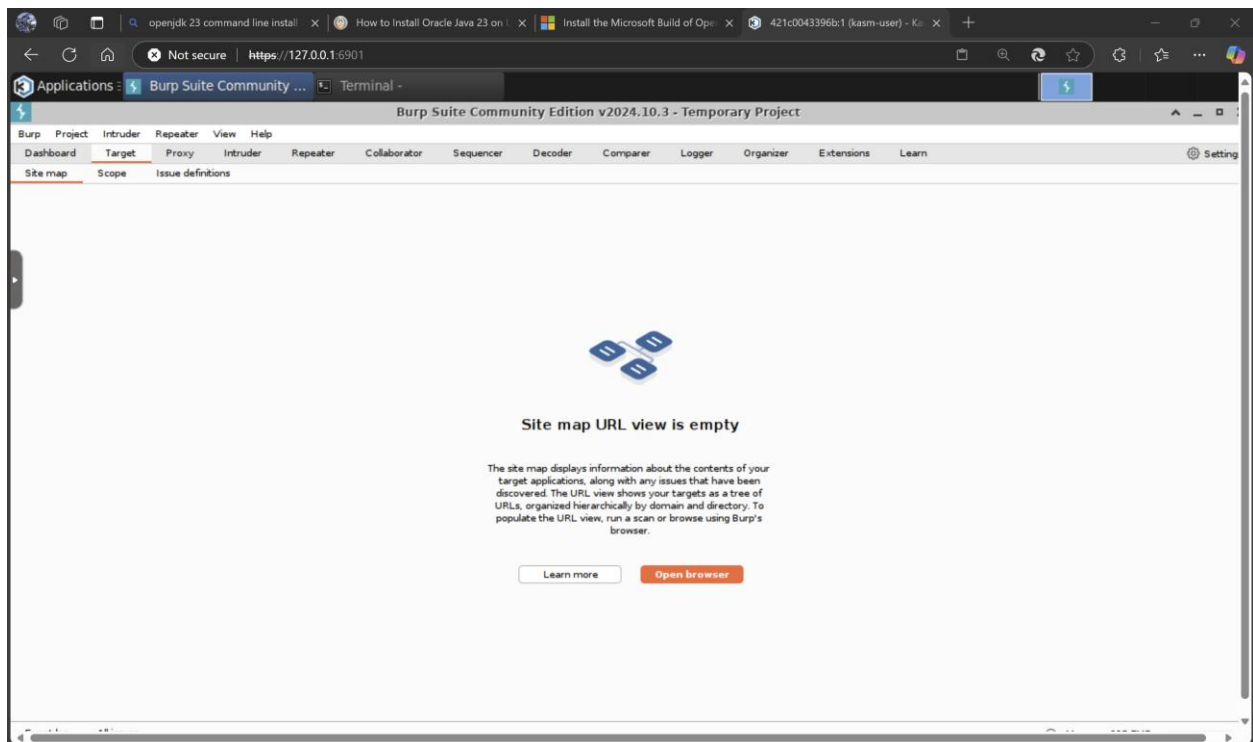
3.1.3 Burpsuite Brute Force 暴力破解

首次运行 Burpsuite 未能成功。

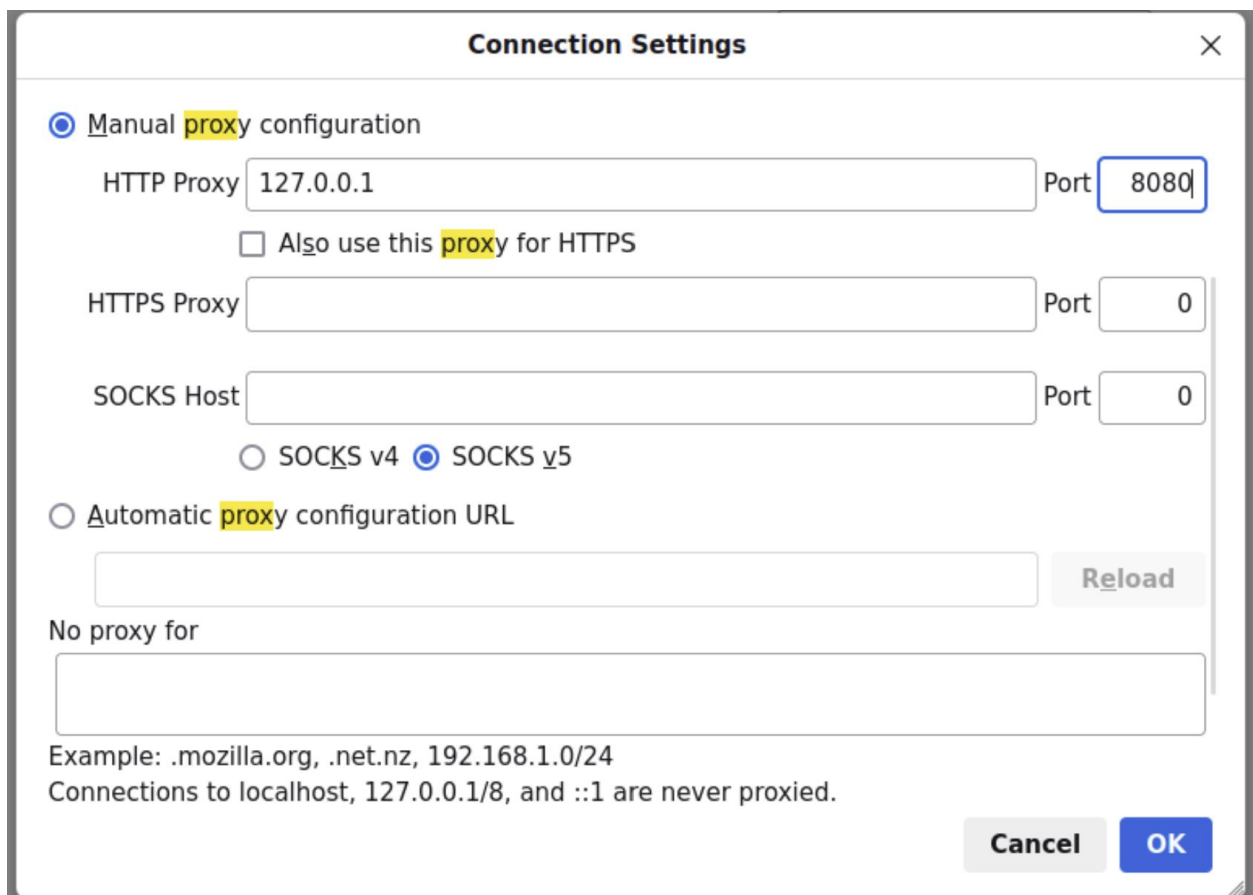
A terminal window titled "Terminal - root@a433d531446e: /home/kasm-default-profile" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the command `java -jar ./burpsuite_community_v2024.10.3.jar` being executed. The output is an error message: "Error: A JNI error has occurred, please check your installation and try again" followed by a stack trace. The stack trace indicates a `java.lang.UnsupportedClassVersionError: burp/StartBurp` because the class file version (65.0) is newer than what the Java Runtime (version 52.0) supports. The stack trace includes frames from `java.lang.ClassLoader`, `java.security.SecureClassLoader`, `java.net.URLClassLoader`, and `sun.launcher.LauncherHelper`.

```
Terminal - root@a433d531446e: /home/kasm-default-profile
File Edit View Terminal Tabs Help
burpsuite_community_v2024.10.3.jar DVWA
Desktop xampp-linux-x64-8.2.4-0-installer.run
root@a433d531446e:/home/kasm-default-profile# java -jar ./burpsuite_community_v2
024.10.3.jar
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.UnsupportedClassVersionError: burp/StartBur
p has been compiled by a more recent version of the Java Runtime (class file ver
sion 65.0), this version of the Java Runtime only recognizes class file versions
up to 52.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:756)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:14
2)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:473)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:74)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:369)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:363)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:362)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
    at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:621)
root@a433d531446e:/home/kasm-default-profile#
```

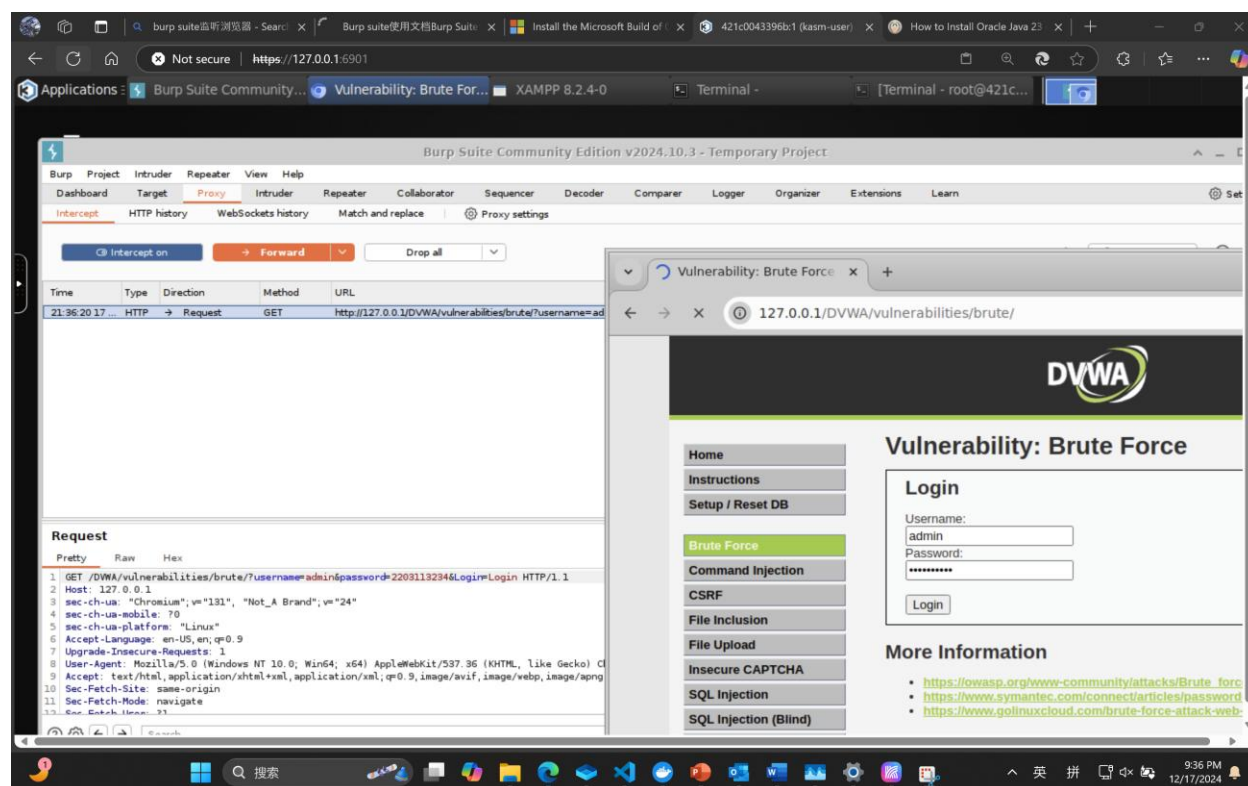
经过检查，发现是安装的 openjdk 版本过于老旧，无法支持新版 burpsuite 的运行，使用命令行安装 openjdk-21-jdk 后可以正常运行。



手动设置浏览器代理地址。



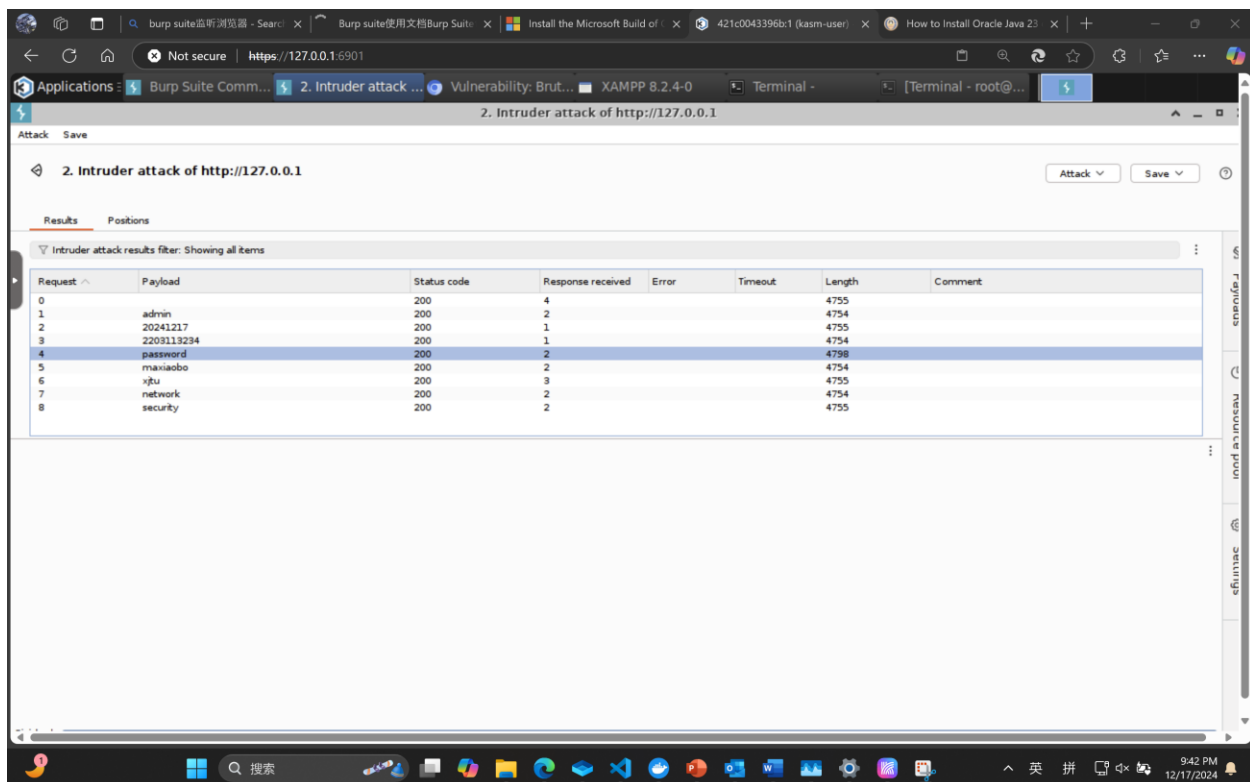
在页面中输入用户名 admin，密码是我的学号 2203113234，可见成功拦截到了包。



将其发送到 intruder 并准备爆破。随机输入一系列字符，其中包括正确密码。



开始爆破。



注意选中的蓝色条目。其他的条目长度均是 4755 或 4754，只有蓝色条目长度是 4798。蓝色条目对应的密码就是设置的初始密码 password。可见通过爆破的方式成功暴力破解了密码。

3.1.4 暴力破解的可能防范措施

经过网上的资料查找，我发现防范暴力破解一般有以下几种方式：

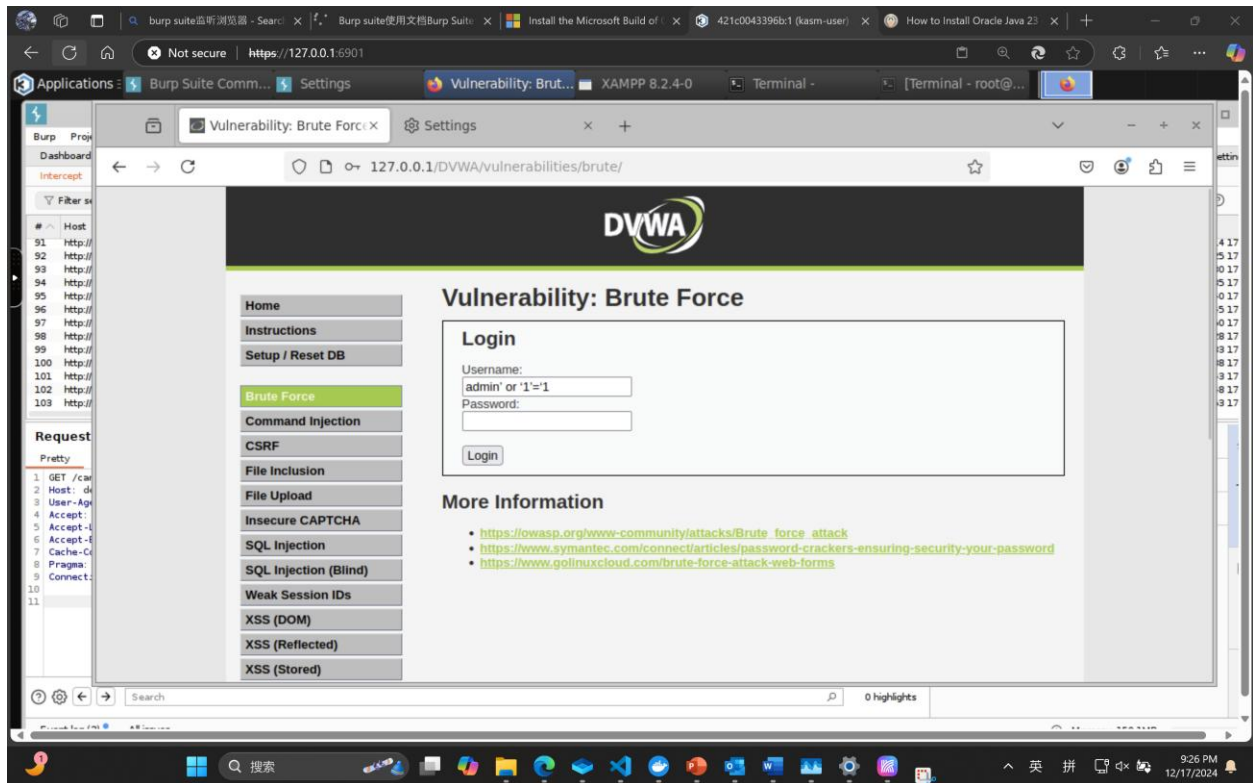
- 增加密码强度：增加密码强度使暴力破解的成本不可接受。
- 使用多重因素验证：如常见的手机验证码、邮箱验证码等多因素辅助验证，使仅仅暴力破解密码无法完成登录。
- 密码安全策略：例如限制一定时间内可以重复尝试密码的次数，以此大幅度延长暴力破解所需时间。

除此以外，在本次实验中明文传输是使用 Burpsuite 构建爆破包的前提。猜想可以使用 HTTPS 等加密手段阻止攻击方获取登陆时的报文格式，根本上杜绝爆破包的创建。

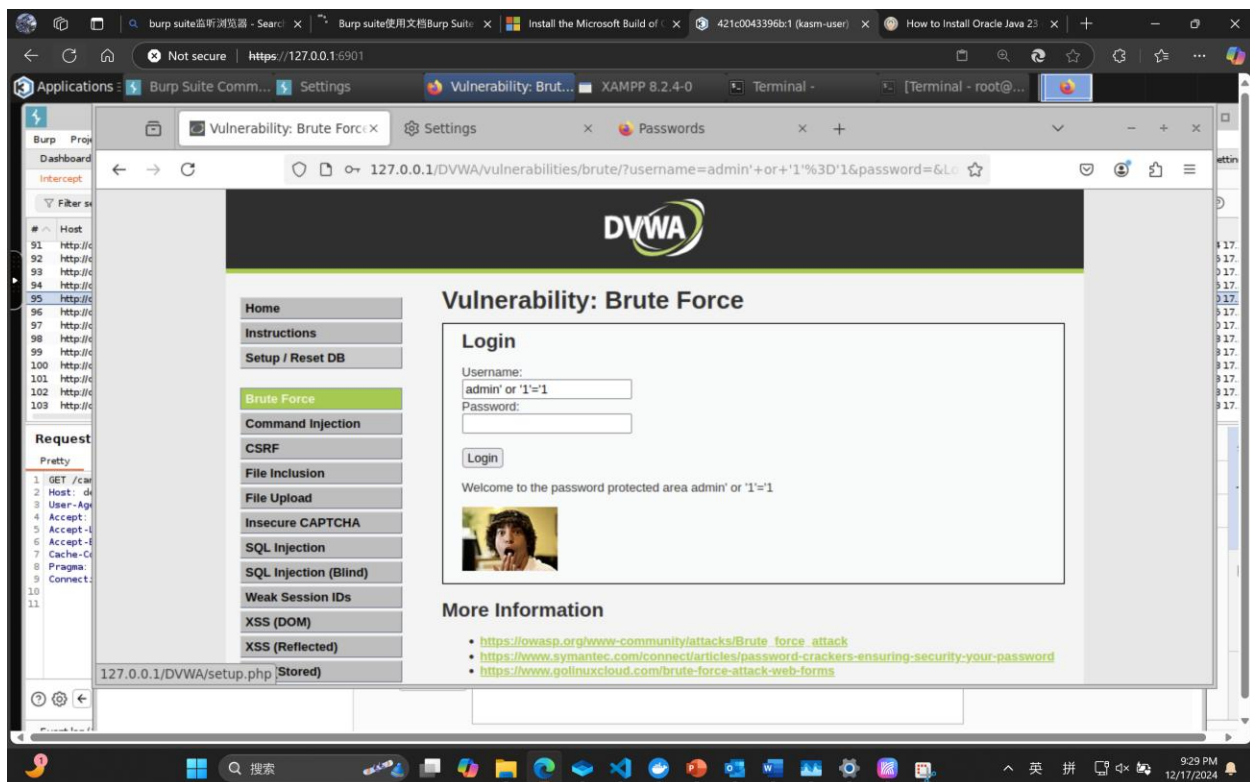
3.2 SQL 注入

3.2.1 SQL 注入实验

复用上一节中的实验容器，在 Brute Force 页面的用户名栏填入注入语句，不填写密码并提交。



仍可以正常登录，验证了 sql 注入漏洞。



3.2.2 SQL 注入的防范方法

结合本次实验和资料查找，可以总结出几种防御 SQL 攻击的简单方法：

- 限制输入内容：避免输入可以构成 SQL 注入语句的符号，如 ‘=’ 等。
- 固定查询参数到前端：不提供查询语句，把每个查询语句都做成 UI，用户选择语句并使用输入框输入参数。避免用户直接提交查询语句。
- 使用 ORM 框架：同样避免手动拼接 SQL 语句。
- 引入带有先验知识的语言模型：大预言模型和人类一样拥有 SQL 语言的先验知识，可以分辨出带有 SQL 注入特征的语句。

3.3 中间人攻击

3.3.1 环境配置

更改先前实验的 Docker 镜像配置文件，创建适合本实验的镜像。

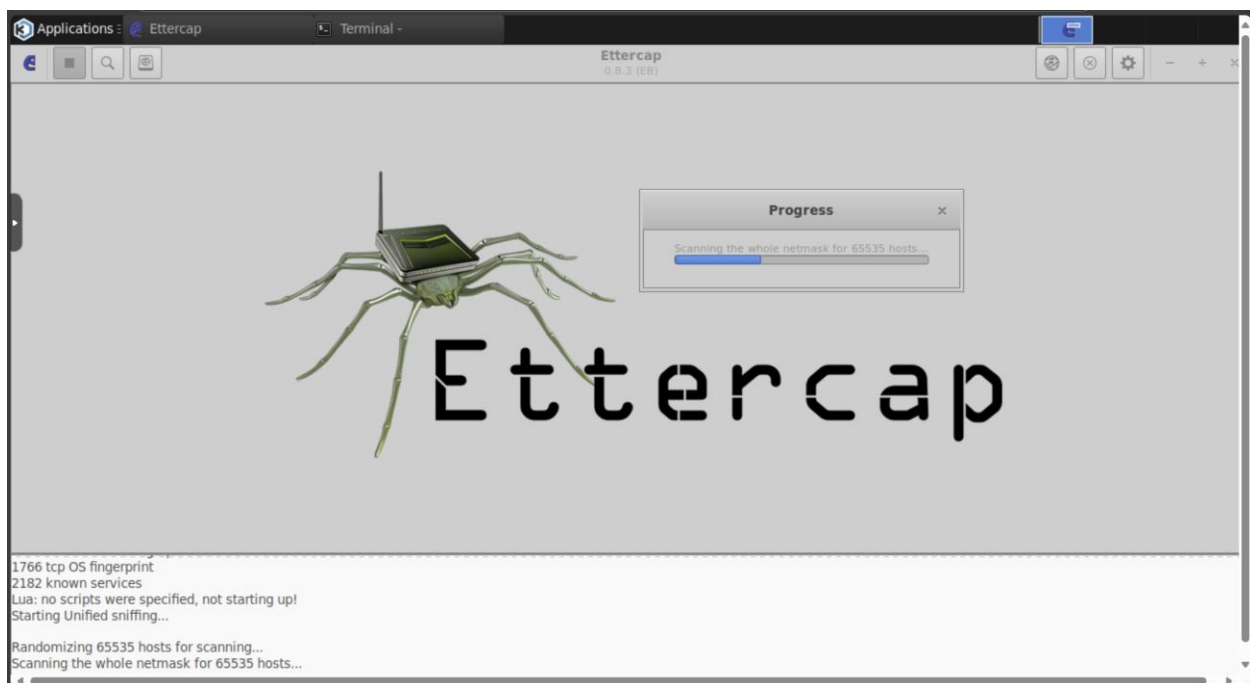
```
#Dockerfile
#sudo as root role.

FROM kasmweb/desktop:1.16.1-rolling-weekly
USER root

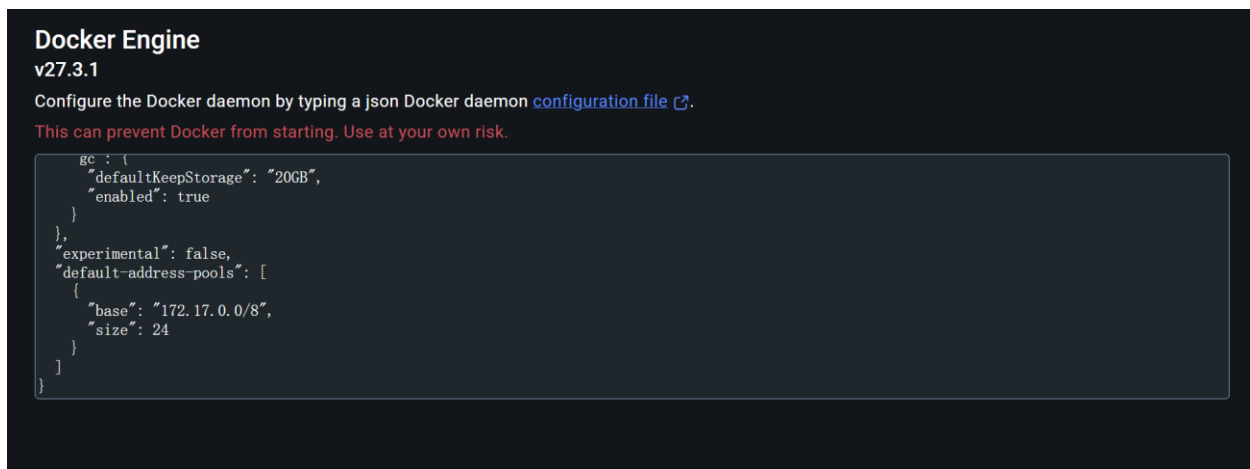
ENV HOME=/home/kasm-default-profile
ENV STARTUPDIR=/dockerstartup
ENV INST_SCRIPTS=$STARTUPDIR/install
WORKDIR $HOME
```

```
##### Customize Container Here #####
ENV TZ=Asia/Shanghai
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone \
    && apt-get update \
    && apt-get install -y sudo \
    && echo 'kasm-user ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers \
    && rm -rf /var/lib/apt/list/* \
    && chown 1000:0 $HOME \
    && $STARTUPDIR/set_user_permission.sh $HOME \
    # basic tools
    && sudo apt-get install -y gedit \
    && sudo apt install -y iputils-ping \
    # wireshark
    && apt install -y wireshark \
    # ettercap
    && sudo apt-get install ettercap-common -y \
    # drift net
    && sudo apt-get install driftnet -y
##### End Customizations #####
ENV HOME=/home/kasm-user
WORKDIR $HOME
RUN mkdir -p $HOME && chown -R 1000:0 $HOME
USER 1000
```

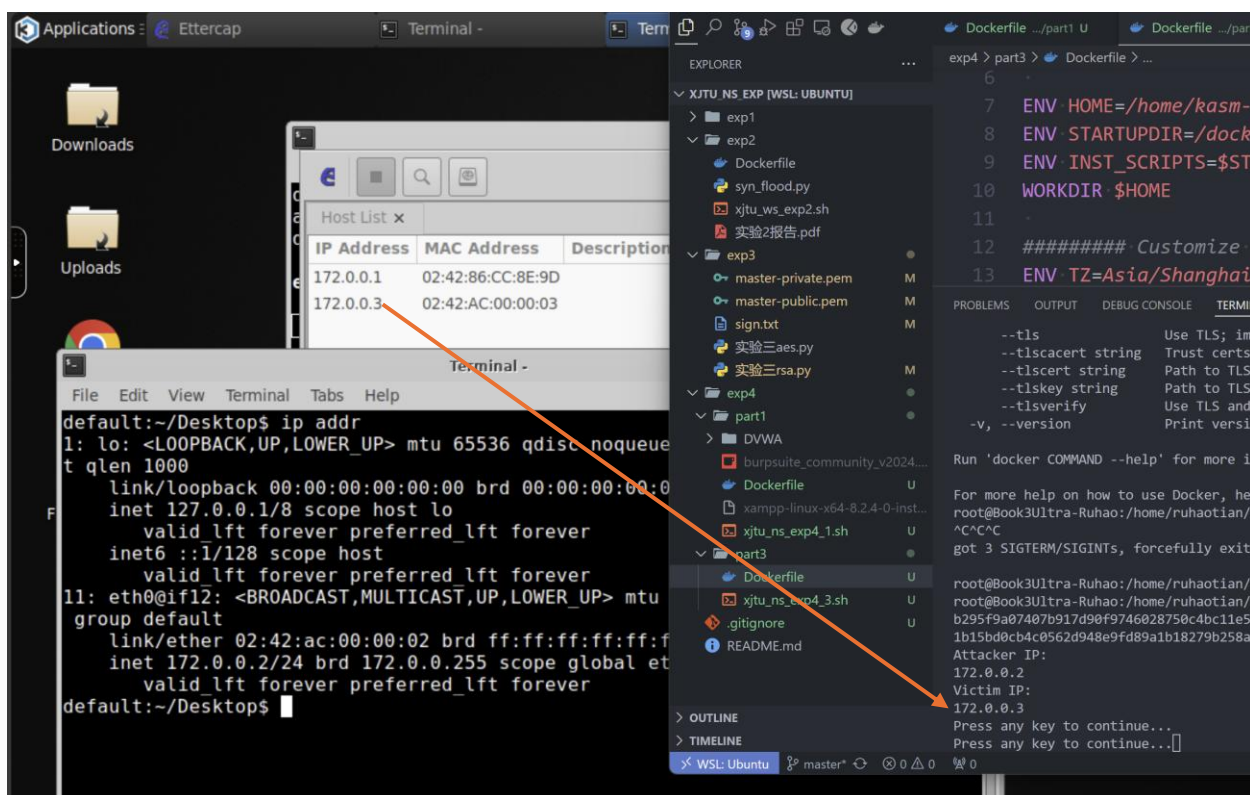
登录 Attacker，同样使用 xhost + 允许 root 用户映射到图形界面。Ettercap 启动后搜索所有 host。



可见由于 Docker 默认网桥的子网掩码为 16 位，host 搜索非常慢。因此修改 Docker 的默认配置文件，将子网掩码增加到 24 位。

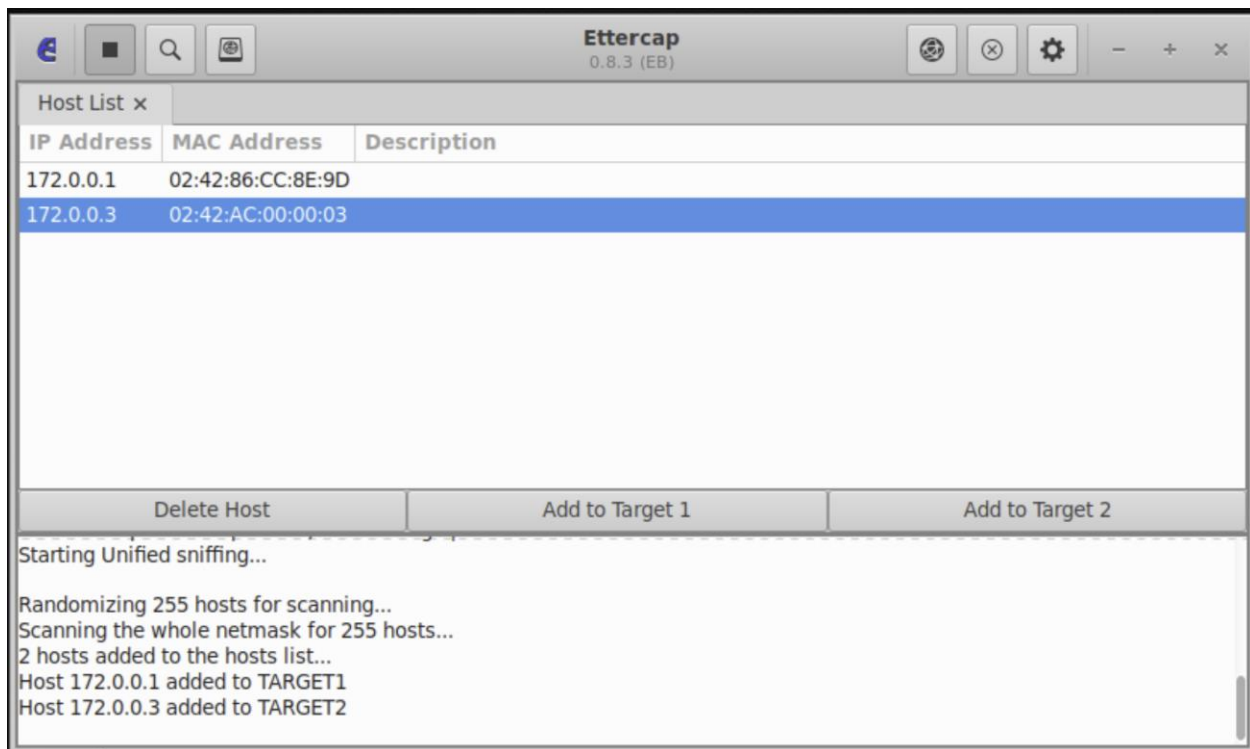


重新启动 Docker 后搜索很快完成。将搜索到的 host 与 docker 的局域网容器地址对比，可见已经成功添加了 victim 容器的 ip 地址。

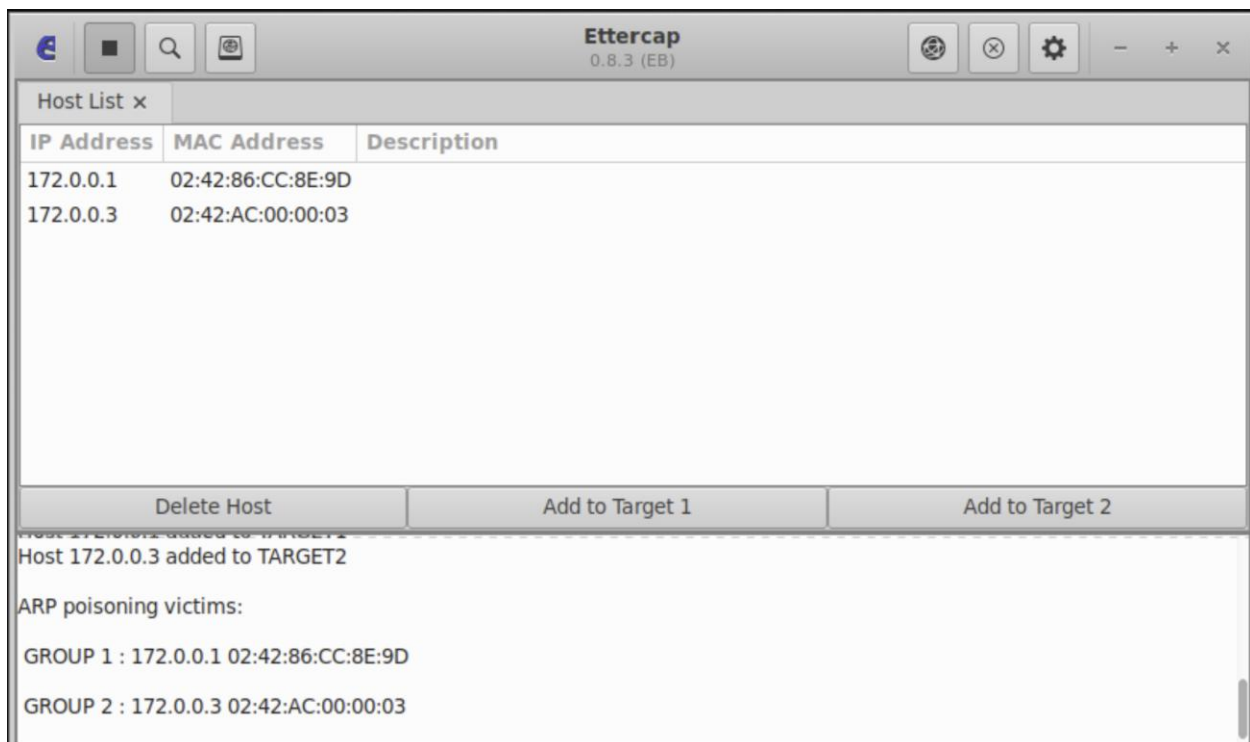


3.3.2 中间人攻击

将网关和 Victim 容器分别设置为 Target1 和 Target2。



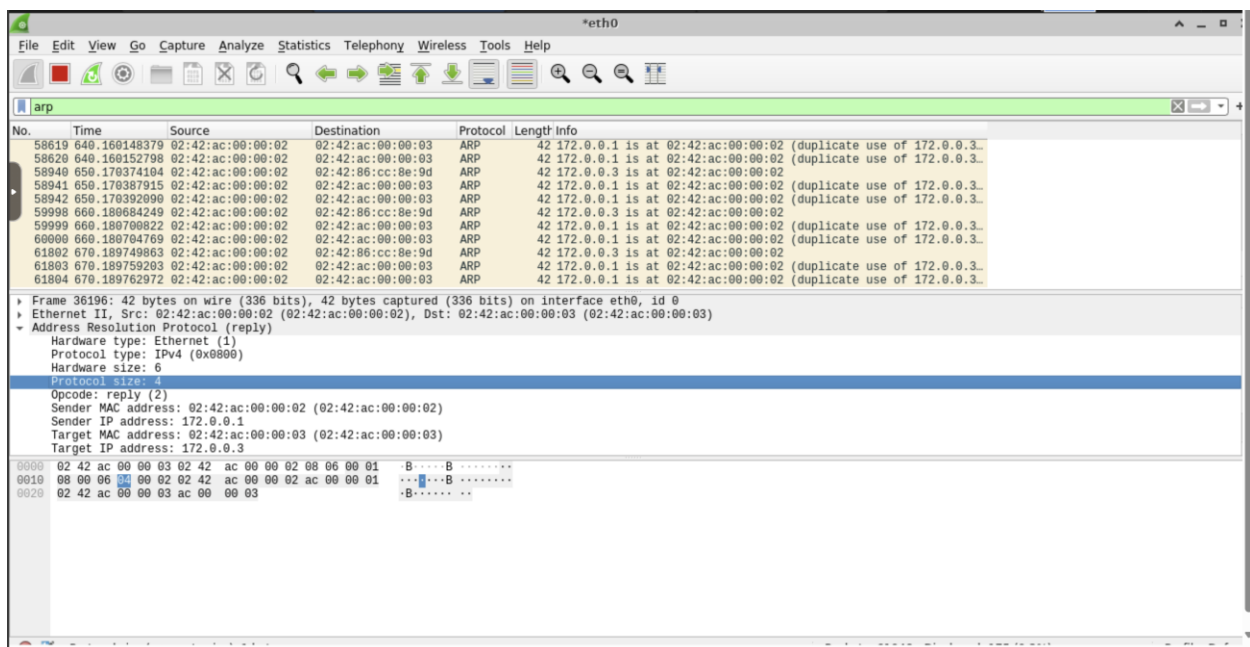
开始 ARP 中间人攻击。



随后切换到 Victim 容器，发现图形界面卡死，这是因为容器的图形界面传输也会经过网关 172.0.0.1。换言之，ARP 攻击同样阻碍了容器的图形界面传输。由于无法再使用图形界面，使用命令行登录容器并查看 arp 缓存列表，可见攻击者已经出现在了 arp 缓存中，证明攻击成功。

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE COMMENTS
○ (xjtu_ns) (base) ruhaotian@Book3Ultra-Ruhao:~/xjtu_ns_exp$ docker exec -it victim /bin/bash
default:~$ arp -a
? (172.0.0.1) at 02:42:ac:00:00:02 [ether] on eth0
? (172.0.0.2) at 02:42:ac:00:00:02 [ether] on eth0
default:~$
```

由于阻碍了图形界面传输，后续使用 Victim 浏览器访问网页的操作难以完成，但仍然可以在 Attacker 容器中对 ARP 数据包进行捕获和分析：



可以看到，该 ARP 数据包的 Target IP address 是 172.0.0.3，这是 Victim 的地址。说明 Attacker（172.0.0.2）成功捕获到了网关（172.0.0.1）和 Victim 之间的通信，验证了 ARP 攻击。

3.3.3 中间人攻击的防范方法

根据资料查找，防范 ARP 攻击的方法包括：

- 静态绑定网关 IP 与 MAC 地址：由于局域网一般适用于组织内，规模小且易管控，可以由管理员手动设置好网关并静态绑定，不接受其他所有 ARP 表项。但是这种方法随着网络规模的扩大成本会愈发高昂。
- 网络隔离：使用多个 VLAN 虚拟局域网，最大程度限制 ARP 攻击的攻击面。
- 使用带有安全功能的专用硬件设备：例如，部分交换机支持 ARP 流量检测，可以识别并拦截局域网中发送方 IP 和 MAC 地址不匹配的情况。

4 心得体会

本次实验从很多方面增强了我的计算机知识（并不限于网络安全），我从 linux 系统的显示器控制策略、Docker 配置、MySQL 服务器配置以及网络安全等多个方面均有收获。诚然，本次实验中的内容都是比较经典的攻击方法。但我认为相比这些知识本省，其最重要的意义是让我感受到了网络安全的重要性以及当前网络安全面临的严峻挑战。现代的电子产品都力求用户对安全无感以创造良好体验，可这在无形中助长了我对网络安全的轻视。我很感谢这门课能让我看到现代网络波涛汹涌的另一面。