

Web 安全

实验复现

Brute Force 暴力破解

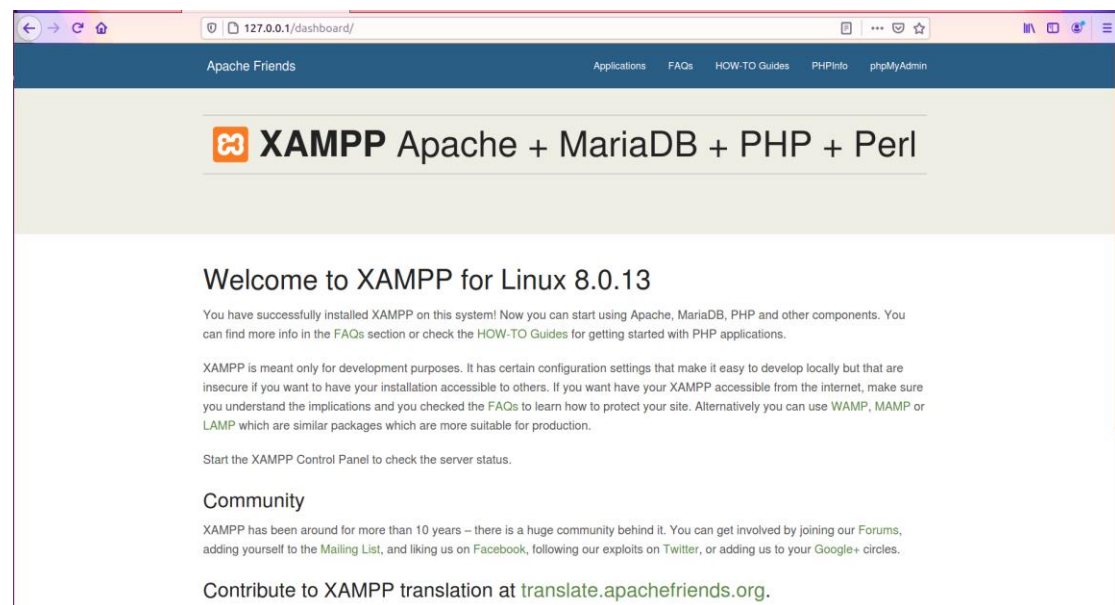
首先卸载 Apache2

然后下载 xampp, 运行安装程序 `sudo ./xampp-linux-x64-8.0.13-0-installer.run`
`cd /opt/lamp`

`sudo ./manager-linux-x64.run` 运行 xampp

启动三个服务 MySQL Database, ProFTPD, Apache Web Server

打开 `127.0.0.1/dashboard`, 服务启动成功

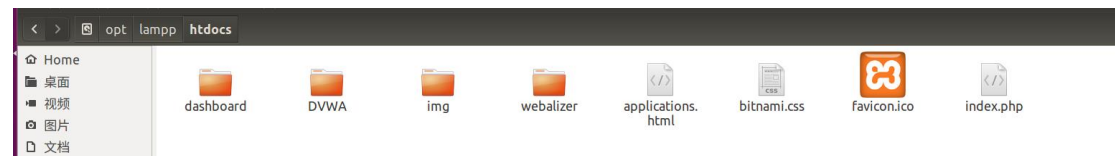


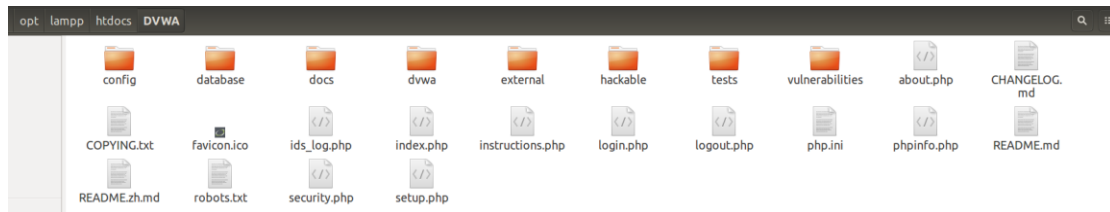
开始搭建靶场

进入到路径 `/opt/lampp/htdocs/`

`sudo apt-get install git`

`git clone http://github.com/digininja/DVMA.git` 安装 github 上的一个 DVMA 仓库, 执行这个命令需要 root 权限



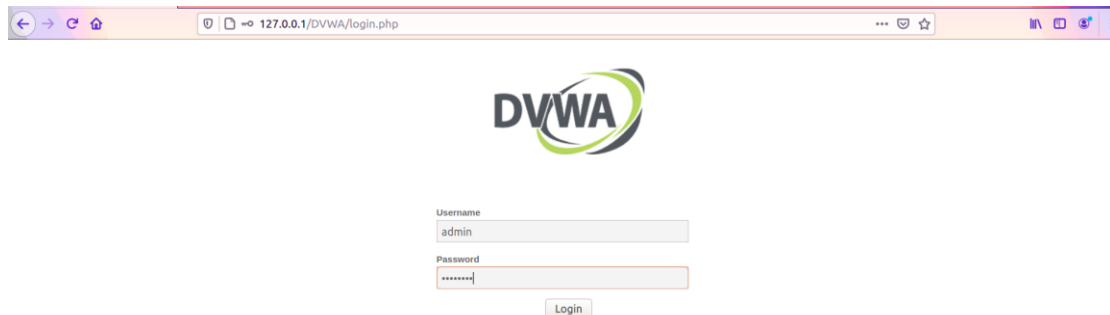


更改 config 里的配置文件, 拷贝一份 config.inc.php.dist, 重命名为 config.inc.php
然后, 修改 config.inc.php 的部分内容
将 password 字段删空, 将 user 字段修改为 root

```
$_DVWA = array();  
$_DVWA[ 'db_server' ] = '127.0.0.1';  
$_DVWA[ 'db_database' ] = 'dvwa';  
$_DVWA[ 'db_user' ] = 'root';  
$_DVWA[ 'db_password' ] = '';  
$_DVWA[ 'db_port' ] = '3306';
```

登录靶场, 创建数据库

默认用户名 admin, 默认密码 password



点击最下方创建数据库

Setup Check

Web Server SERVER_NAME: **127.0.0.1**

Operating system: ***nix**

PHP version: **8.0.13**

PHP function display_errors: **Disabled**

PHP function safe_mode: **Disabled**

PHP function allow_url_include: **Disabled**

PHP function allow_url_fopen: **Enabled**

PHP function magic_quotes_gpc: **Disabled**

PHP module gd: **Installed**

PHP module mysql: **Installed**

PHP module pdo_mysql: **Installed**

Backend database: **MySQL/MariaDB**

Database username: **root**

Database password: ***blank***

Database database: **dvwa**

Database host: **127.0.0.1**

Database port: **3306**

reCAPTCHA key: **Missing**

[User: root] Writable folder /opt/lampp/htdocs/DVWA/hackable/uploads/: **No**

[User: root] Writable file /opt/lampp/htdocs/DVWA/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: **No**

[User: root] Writable folder /opt/lampp/htdocs/DVWA/config: **No**

Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your `php.ini` file and restart Apache.

`allow_url_fopen = On`

`allow_url_include = On`

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

将安全等级调到最低 **low**，否者后面实验会翻车

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Low



Submit

然后安装 **Brute Force** 准备暴力破解

安装版本与实验指导保持一致

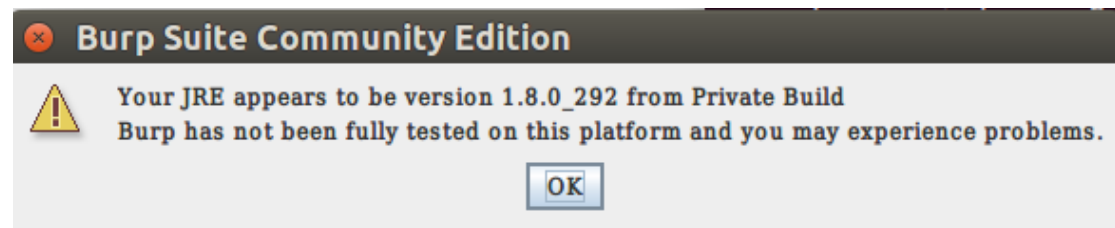
```
sudo apt-get install openjdk-8-jdk
```

这个 JDK 和 JRE 的版本和老师给的有点不一样，但是后面的实验是能正常做完的

```
justin@ubuntu-server:~$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~16.04.1-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

```
sudo java -jar burpsuite_community_v1.7.36.jar
```

burpsuite 有专业版和社区版，专业版有一些更加专业的功能，但我们此次实验社区版就足够了



修改浏览器代理

连接设置

配置访问互联网的代理服务器

☐ 不使用代理服务器(Y)

☐ 自动检测此网络的代理设置(W)

☐ 使用系统代理设置(U)

☒ 手动配置代理(M)

HTTP 代理(X)

127.0.0.1

端口(P)

8080

☒ 也将此代理用于 FTP 和 HTTPS

HTTPS Proxy

127.0.0.1

端口(O)

8080

FTP 代理

127.0.0.1

端口(R)

8080

SOCKS 主机

端口(T)

0

☐ SOCKS v4

☒ SOCKS v5

☐ 自动代理配置的 URL (PAC)

重新载入(E)

不使用代理(N)

例如: .mozilla.org, .net.nz, 192.168.1.0/24

☐ 如果密码已保存, 不提示身份验证(I)

帮助(H)

取消

确定

修改浏览器高级设置，包括两项

proxy.allow

☐ 仅显示修改过的首选项

network.proxy.allow_hijacking_localhost

true

这一步是为了保证能抓到 127.0.0.1 的包

network.captive-portal-service.backoffFactor	5.0
network.captive-portal-service.enabled	false
network.captive-portal-service.maxInterval	1500000
network.captive-portal-service.minInterval	60000

这一步是过滤 `detectportal.firefox.com` 的包

给浏览器安装 burp 的证书

<http://burp> 下载 burp 的证书, 或者在 burp 软件里面生成证书 export CA certificate

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Add

Edit

Remove

Running	Interface	Invisible	Redirect	Certificate
<input checked="" type="checkbox"/>	127.0.0.1:8080			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for use in other tools or another installation of Burp.

Import / export CA certificate

Regenerate CA certificate

然后将证书导入浏览器

证书管理器

您的证书

认证决策

个人

服务器

证书颁发机构

您有用来识别这些证书颁发机构的证书文件

证书名称	安全设备
▼ AC Camerfirma S.A.	
Chambers of Commerce Root - 2008 Builtin Object Token	
Global Chambersign Root - 2008	Builtin Object Token
▼ AC Camerfirma SA CIF A82743287	
Camerfirma Chambers of Comme...	Builtin Object Token
Camerfirma Global Chambersign ...	Builtin Object Token

查看(V)...

编辑信任(E)...

导入(M)...

导出(X)...

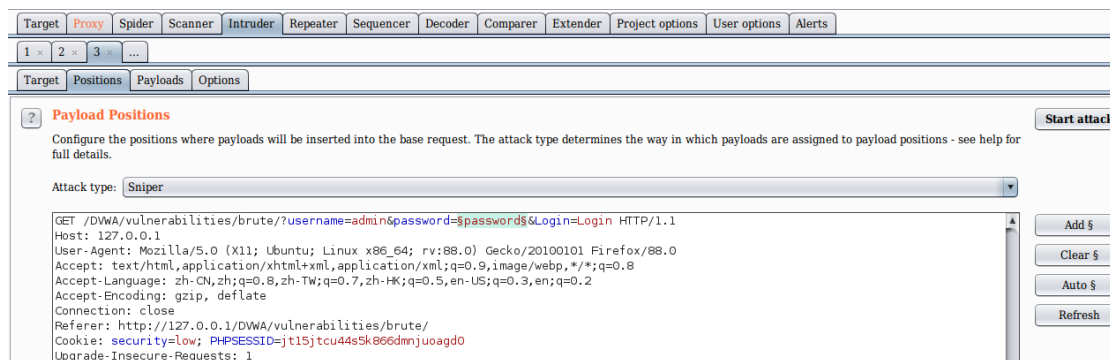
删除或不信任(D)...

确定

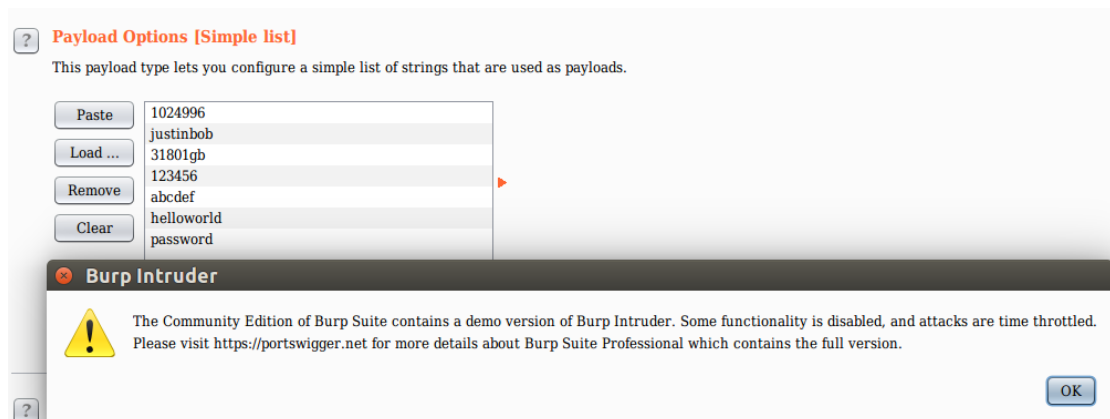
开始操作



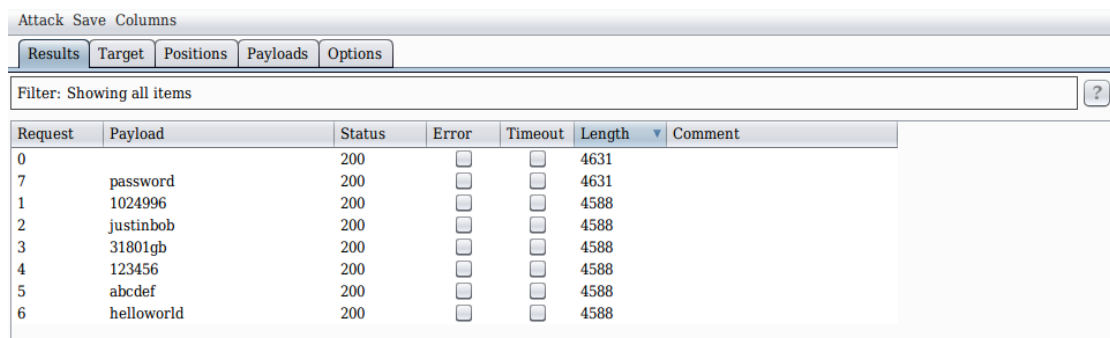
拦截到登录时的包，账号密码都是明文传输



将上一步拦截到的包发送给 Intruder 模块，指定爆破的属性字段，准备暴力破解



这里我就没生成字典，自己随便编了几个密码试试，当然也可以导入字典，不过那样应该破解起来挺慢的



这是爆破的结果，密码就是 password，爆破成功

SQL 注入

Login

Username:

admin' or '1'='1

Password:

Login

TargetProxySpiderScannerIntruderRepeaterSequencerDecoderComparerExtenderProject optionsUser optionsAlerts

InterceptHTTP historyWebSockets historyOptions

Request to http://127.0.0.1:80

ForwardDropIntercept is onAction

RawParamsHeadersHex

GET /DVWA/vulnerabilities/brute/?username=admin%27+or+%271%27%3D%271&password=&Login=Login HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://127.0.0.1/DVWA/vulnerabilities/brute/?username=admin&password=password&Login=Login
Cookie: security=low; PHPSESSID=jt15jtcu44s5k866dmnjuoagd0
Upgrade-Insecure-Requests: 1

Burp 拦截到的包


Login

Username:

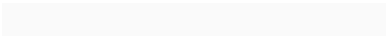
Password:

Login

Welcome to the password protected area admin' or '1'='1



成功登录



ARP Spoofing

在 Attacher 上安装 ettercap 和 driftnet

```
sudo apt-get install ettercap-common
```

```
sudo apt-get install driftnet
```

sudo ettercap -G 打开 ettercap, 进行 ARP 欺骗操作

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
5 hosts added to the hosts list...
Host 192.168.1.1 added to TARGET1
Host 192.168.1.111 added to TARGET2
```

ARP poisoning victims:

GROUP 1 : 192.168.1.1 54:75:95:DD:EE:18

GROUP 2 : 192.168.1.111 00:0C:29:A6:5F:24

扫描局域网内的主机, ARP 攻击的 Target1 就是我们的网关 192.168.1.1, Target2 就是我们的 Victim 192.168.1.111

Host List ✖		
IP Address	MAC Address	Description
192.168.1.1	54:75:95:DD:EE:18	
fe80::816d:8bd3:b0cd:3eda	90:78:41:E8:17:A5	
192.168.1.100	B6:C5:BE:EF:C7:A0	
192.168.1.107	90:78:41:E8:17:A5	
192.168.1.111	00:0C:29:A6:5F:24	

这是 Attacker 的网络信息

```
gaoben@ubuntu-client:~$ ifconfig
ens33  Link encap:以太网 硬件地址 00:0c:29:b0:48:68
        inet 地址:192.168.1.102 广播:192.168.1.255 掩码:255.255.255.0
        inet6 地址: fe80::3d12:54e1:a60e:add2/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1
        接收数据包:35123 错误:0 丢弃:0 过载:0 帧数:0
        发送数据包:14580 错误:0 丢弃:0 过载:0 载波:0
        碰撞:0 发送队列长度:1000
        接收字节:28482825 (28.4 MB) 发送字节:7360206 (7.3 MB)

lo      Link encap:本地环回
        inet 地址:127.0.0.1 掩码:255.0.0.0
        inet6 地址: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 跃点数:1
        接收数据包:392 错误:0 丢弃:0 过载:0 帧数:0
        发送数据包:392 错误:0 丢弃:0 过载:0 载波:0
        碰撞:0 发送队列长度:1000
        接收字节:32356 (32.3 KB) 发送字节:32356 (32.3 KB)
```

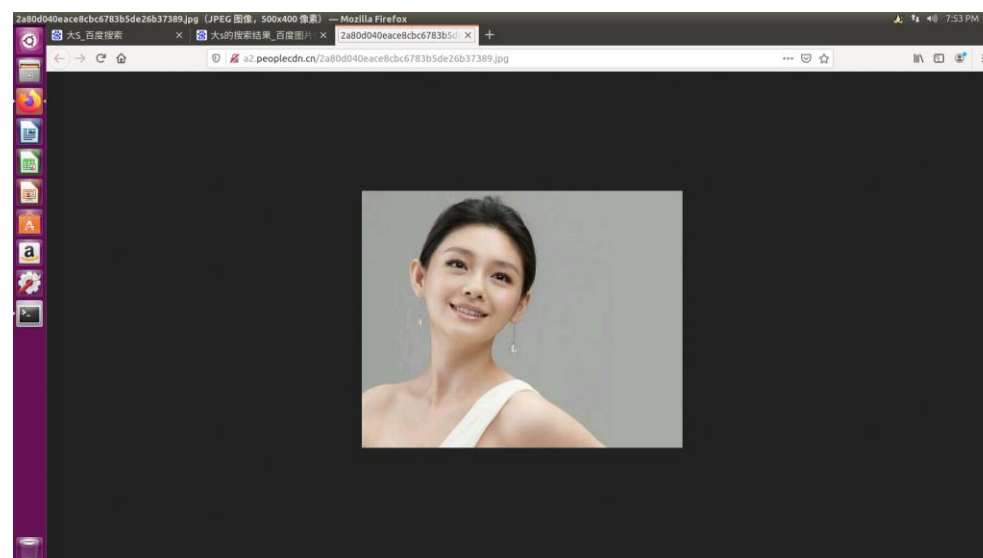
操作完成后查看 Victim 的 arp 表, 攻击成功


```
justin@ubuntu:~$ ifconfig
ens33  Link encap:以太网 硬件地址 00:0c:29:a6:5f:24
       inet 地址:192.168.1.111 广播:192.168.1.255 掩码:255.255.255.0
       inet6 地址: fe80::6677:ddf1:1457:2323/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1
       接收数据包:6045 错误:0 丢弃:0 过载:0 帧数:0
       发送数据包:2448 错误:0 丢弃:0 过载:0 载波:0
       碰撞:0 发送队列长度:1000
       接收字节:7681340 (7.6 MB) 发送字节:214422 (214.4 KB)

lo     Link encap:本地环回
       inet 地址:127.0.0.1 掩码:255.0.0.0
       inet6 地址: ::1/128 Scope:Host
       UP LOOPBACK RUNNING MTU:65536 跃点数:1
       接收数据包:309 错误:0 丢弃:0 过载:0 帧数:0
       发送数据包:309 错误:0 丢弃:0 过载:0 载波:0
       碰撞:0 发送队列长度:1000
       接收字节:26351 (26.3 KB) 发送字节:26351 (26.3 KB)

justin@ubuntu:~$ arp -a
? (192.168.1.1) 位于 00:0c:29:b0:48:68 [ether] 在 ens33
? (192.168.1.102) 位于 00:0c:29:b0:48:68 [ether] 在 ens33
justin@ubuntu:~$
```

然后在 Victim 主机浏览器上通过 HTTP 打开一张大 S 的图片

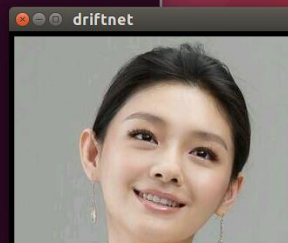


Attacker 捕获成功

```
gaoben@ubuntu-client:~$ ifconfig
ens33  Link encap:以太网 硬件地址 00:0c:29:b0:48:68
       inet 地址:192.168.1.102 广播:192.168.1.255 掩码:255.255.255.0
       inet6 地址: fe80::3d12:54e1:a60e:add2/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1
       接收数据包:7468 错误:0 丢弃:0 过载:0 帧数:0
       发送数据包:3290 错误:0 丢弃:0 过载:0 载波:0
       碰撞:0 发送队列长度:1000
       接收字节:9167862 (9.1 MB) 发送字节:294067 (294.0 KB)

lo     Link encap:本地环回
       inet 地址:127.0.0.1 掩码:255.0.0.0
       inet6 地址: ::1/128 Scope:Host
       UP LOOPBACK RUNNING MTU:65536 跃点数:1
       接收数据包:392 错误:0 丢弃:0 过载:0 帧数:0
       发送数据包:392 错误:0 丢弃:0 过载:0 载波:0
       碰撞:0 发送队列长度:1000
       接收字节:32356 (32.3 KB) 发送字节:32356 (32.3 KB)

gaoben@ubuntu-client:~$ sudo driftnet -i ens33
Corrupt JPEG data: 64 extraneous bytes before marker 0xef
Unsupported marker type 0x69
日 12月 05 19:51:11 2021 [driftnet] warning: driftnet-61aca7af6b8b4567.jpeg: bog
us image (err = 4)
```



原理、预防措施

Brute Force 暴力破解

原理

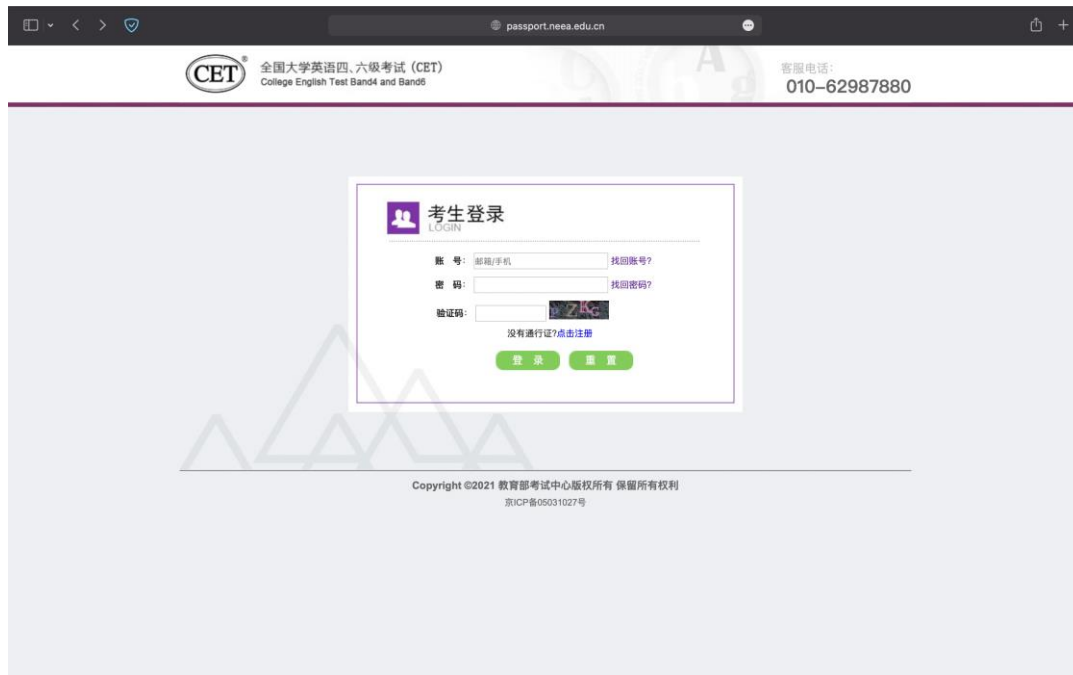
暴力破解其实就是穷举，猜测字符有多少位，猜测每个位置是哪个字符。我见过最多的就是那种带密码的压缩包，通过导入字典进行破解，通过暴力破解来破解登陆密码几乎是不可能。防止暴力破解有以下几个办法，主要就是通过增大密码复杂度和惩罚攻击者来实现

解决办法

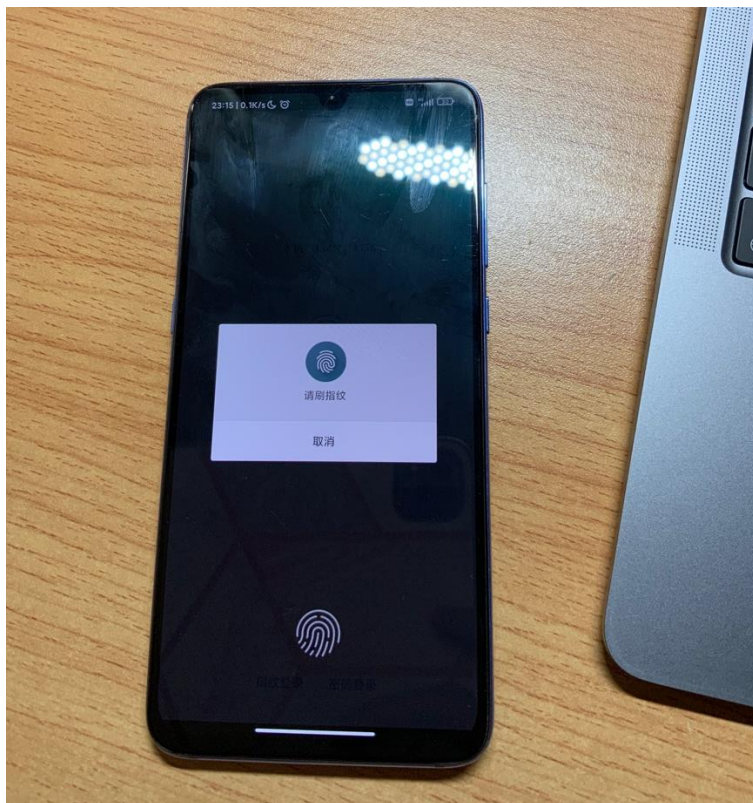
1. 对账号登陆时密码错误次数限制，超过次数后直接锁定账号，或者在指定时间内账号锁定，时间指数增加，保护信息财产安全



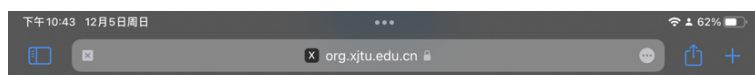
2. 对每个 IP 试错次数限制，次数达到后封禁这个 IP，让其无法攻击，攻击者就必须更换设备或者更换设备 IP，类似爬虫封 IP
3. 登录时候，加入验证码，有几种形式的验证码，有的是识别图像中的物品路灯、汽车、椰子树什么的，有的是识别图片中的字符，有的是需要点击一下，有的是将图片中的小缺块拖到指定位置，有的是按要求顺序点击图中的字符，进行人机验证，每次都得人工输入验证码，无法暴力破解



4. 登录的时候，需要短信验证码或者邮箱验证码，人脸识别，指纹识别等等
下面这个是中国银行 app 登录的一种方式，哈哈，不让截图



5. 用户创建密码的时候，对密码进行强制限制，比如说长度区间，各种特殊字符要有，不能单是数字，保证创建的是强密码，比如 Apple 的 Safari 创建账号的时候，会帮忙创建强密码，在通过 iCloud 同步到其他设备
四六级官网的密码对特殊字符的要求好像很奇怪，每次登录的时候我都得重置密码



统一身份认证登录

正在登录至(Logging in to) **【bb平台】** **【教务处**

】-管理员-刘老师-联系电话：82668307-

账号 2194214342

密码 poghyp-0bAhka-fugtep 强密码

验证码登录 | 新用户认证

忘记密码

Verification code... | New user

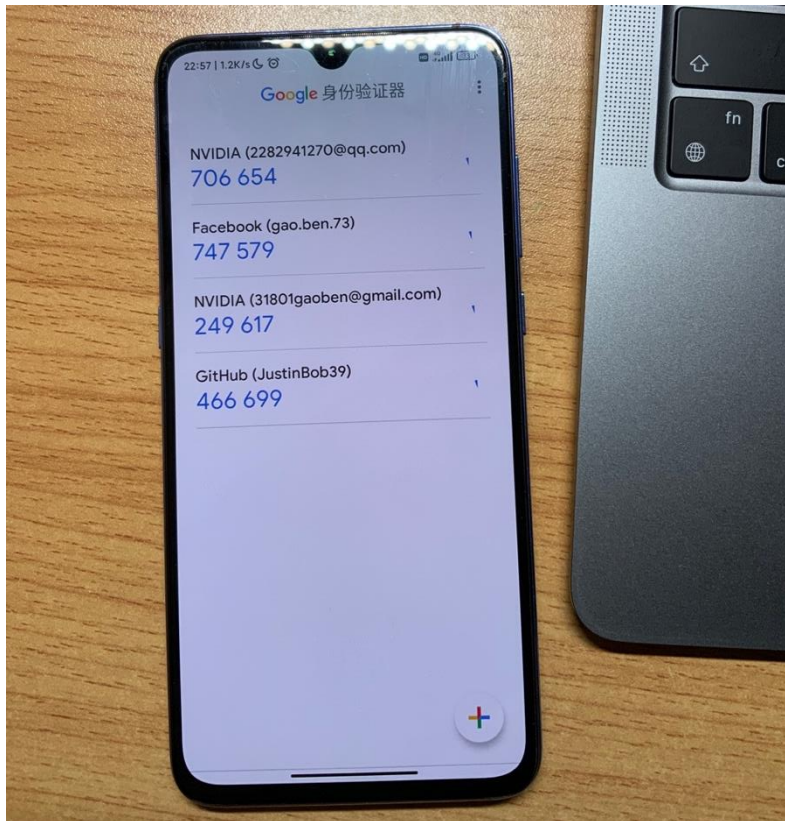
Forget password

开发维护：西安交通大学网络信息中心

账号密码问题：029-82667777

登录

6. 现在 Google 有一种新技术，叫做动态验证码，就是在云端不断生成验证码，一段时间内有效，用户凭借这个验证码就可以登陆



SQL Injection

原理

SQL 注入的原理是将 SQL 代码伪装到输入参数中，传递到服务器解析并执行的一种攻击手法。也就是说，在一些对 server 端发起的请求参数中植入一些 SQL 代码，server 端在执行 SQL 操作时，会拼接对应参数，同时也将一些 SQL 注入攻击的“SQL”拼接起来，导致会执行一些预期之外的操作。

比如说我们在登录框输入 `admin' or '1'='1`，密码为空，然后点击提交
服务器端拿到 POST 请求后，首先连接数据库，然后后台对请求参数中携带的用户名、密码进行参数校验，即 SQL 的查询过程，相当于调用了

```
SELECT * FROM user WHERE username = 'admin' or '1'='1'
```

因为 `'1'='1'` 一直为 True，所以我们就能登录成功

如果使用 `' or 1=1 #` 作为用户名参数

```
select * from users where username='' or 1=1#'
```

#后面都被理解为注释

```
select * from users where username='' or 1=1
```

```
select * from users
```

查询出所有的登录用户

还有多种办法，叫做万能登录密码

admin' or 1=1#	'or'="a'='a
'or 1=1	'or' '1'='1'
"or "a"="a	'or' '='
"or 1=1--	'or' '='or' '='
"or"="	'or'='1'
"or"="a'='a	'or'='or'
"or1=1--	'or.'a.'='a
"or=or"	'or1=1--
' 'or'='or'	1'or'1'='1
') or ('a'='a	a'or' 1=1--
'.).or>('.a.'='.a	a'or'1=1--
'or 1=1	or 'a'='a'
'or 1=1--	or 1=1--
'or 1=1/*	or1=1--

但然，这也是为什么我们的靶场 Security Level 要调到 Low 的原因，因为安全等级一高这种低级的 SQL 注入就无法成功

解决办法

1. SQL 是一种弱类型语言，我们可以通过限制输入，输入时进行正则匹配，不合法报错提示用户，保证提交的是合法的类型、长度、格式，能在一定程度上减少输入中加入 SQL 参数，比如说禁止一些 '#'，';'，'-' 等 SQL 敏感的字符，给定下拉框、单选框限定用户输入。现在主流账号比如说 QQ 账号都是一串数字，可以被解释为类似 int 类型强类型
2. 参数化查询，是一种 SQL 预编译方法，也是目前最有效的预防 SQL 注入的方法，指在设计与数据库链接并访问数据时，在需要填入数值或数据的地方，使用参数来给值。在使用参数化查询的情况下，SQL 不会将参数的内容视为 SQL 指令的一部分来处理，而是在数据库完成 SQL 指令的编译后，才套用参数运行，因此就算参数中含有具破坏性的指令，也不会被数据库所运行。
MySQLi 扩展可以提供参数化查询功能，PHP 5.1 在处理数据库时提出了一个更好的方法 PHP 数据对象（PDO）。
3. 存储过程，是一种在数据库中存储复杂程序，以便外部程序调用的一种数据库对象，就是 SQL 的代码封装与重用。为了完成特定功能的 SQL 语句集，经编译创建并保存在数据库中，用户可通过指定存储过程的名字并给定参数来调用执行，相当于指定了参数执行的过程，防止 SQL 注入执行了不在预期内的语句
4. SQL escape 转义，用户输入如果没有任何限制的话，则必须对特殊字符进行变换。如果对单引号不进行变换，轻者不能正常执行功能，重则会发生数据库错误，甚至可

能导致系统崩溃。对单引号、通配符、全角字符、转义符、字符类型、模糊查询 `like`、特殊字符都要进行相应的转义处理

PHP 中提供了 `mysql_real_escape_string()` 语句

5. 避免管理员权限，最小化用户权限，比如用户只是需要查询，那就只给 `select` 权限就可以了，不要给用户赋予 `update`、`insert` 或者 `delete` 权限。防止用户干坏事破坏数据库，确保每个应用程序都有自己的数据库凭证，并且这些凭证具有应用程序需要的最低权限。

创建用户的时候限制用户的登录主机，一般是限制成指定 IP 或者内网 IP 段

初始化数据库的时候删除没有密码的用户。安装完数据库的时候会自动创建一些用户，这些用户默认没有密码

定期清理不需要的用户，回收权限或者删除用户

6. Web 应用防火墙，可以阻止以下的攻击方式

- ✧ SQL injection
- ✧ Cross-site scripting (XSS)
- ✧ Session hijacking
- ✧ Distributed denial of service (DDoS) attacks
- ✧ Cookie poisoning
- ✧ Parameter tampering

在服务器前运行运行 Web Application Firewall，监控进出的流量，通过默认策略、自定义的网络安全规则策略，快速响应及时识别出威胁，阻止恶意行为。

ARP Spoofing

原理

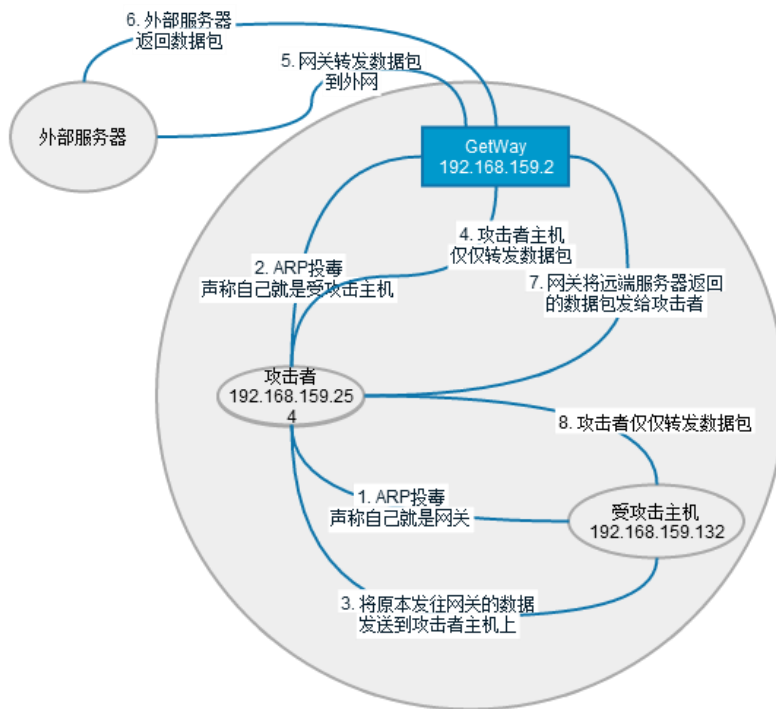
ARP 的作用

Address Resolution Protocol，地址解析协议，工作在网络层，负责将 IP 地址转化为对应的 MAC 地址，能非常好的解释为什么有的网络互联设备加载两层协议，有的加载三层协议。

源主机和目的主机之间通过广域网进行通信，首先我们的源主机不管是已知目的主机 IP 地址还是通过 DNS 解析获得目的主机 IP 地址，将目的 IP 地址和自己的 IP 地址封装进网络层，然后将自己的 MAC 地址和 ARP 表中网关的 MAC 地址封装进数据链路层，数据包发送给网关，网关连接着广域网，然后通过 IP 地址进行路由，每一跳的过程中源 IP 和目的 IP 是不变的，但是 MAC 地址一直在变化，最终到达目的主机的网关，该网关查找 ARP 表将数据包发送给目的主机。目的主机回送一个响应，数据包经过前面的类似过程发送，最终到达局域网网关，网关通过广播获得每个 IP 地址对应的 MAC 地址，前提是每台主机都是诚实的，因此它获得这个数据包后，就知道该转发给局域网内的正确的主机。

PPT 上的这张图太经典了，描述的是 **Man-in-the-Middle (MitM) Attack**，利用 ARP 协议的漏洞，攻击者通过冒充自己是网关，受害者把数据包都发给攻击者，攻击者再把数据包发送给实际网关，实际网关数据也只会交付给攻击者，攻击者再发送给受害者，原理有点类似代理服务器。也就是说，可以在受害者完全不知情的情况下，窥探他的一举一动，非常恐怖，不过也不必过于惊恐，因为现在主流的杀毒软件都会检测 ARP 欺骗。

当然，ARP 欺骗其实也有一些合理的应用。



ARP 攻击有三种形式

- ✧ Man-in-the-Middle (MitM) Attack
- ✧ Denial of Service (DoS) Attack
- ✧ Session Hijacking

ARP 抓包分析

下面我们来分析一下 ARP 攻击的过程，

53 26.754485860	Vmware_b0:48:68	Tp-LinkT_dd:ee:18	ARP	42 192.168.1.111 is at 00:0c:29:b0:48:68
54 26.754612758	Vmware_b0:48:68	Vmware_a6:5f:24	ARP	42 192.168.1.1 is at 00:0c:29:b0:48:68 (duplicate use of 192.168.1.111 detected!)

Attacker 不断地向网关发送 ARP 报文，欺骗网关 Victim 的 IP 地址 192.168.1.111 对应的 MAC 地址为自己的 MAC 地址

Attacker 同时还不断地向 Victim 发送 ARP 报文，欺骗 Victim 网关的 MAC 地址就是自己的 MAC 地址

最终就达到了 PPT 中图片描述的那种情况，Victim 的一举一动 Attacker 都十分清楚

我们的 Wireshark 也很给力，检测出了异常，发现了 duplicate use of 192.168.1.1 和 192.168.1.111

▶ Frame 53: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: Vmware_b0:48:68 (00:0c:29:b0:48:68), Dst: Tp-LinkT_dd:ee:18 (54:75:95:dd:ee:18)
▶ Destination: Tp-LinkT_dd:ee:18 (54:75:95:dd:ee:18)
▶ Source: Vmware_b0:48:68 (00:0c:29:b0:48:68)
Type: ARP (0x0806)
▼ Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: Vmware_b0:48:68 (00:0c:29:b0:48:68)
Sender IP address: 192.168.1.111
Target MAC address: Tp-LinkT_dd:ee:18 (54:75:95:dd:ee:18)
Target IP address: 192.168.1.1

ARP 层:

- ✚ 硬件类型, 2 字节, 标识链路层协议, 值为 1, 表明是链路层协议是以太网 Ethernet
- ✚ 协议类型, 2 字节, 标识网络层协议, 值为 0x0800, 表明是网络层协议为 IPv4, 相应的还有 IPv6
- ✚ 硬件地址大小, 1 字节, 标识 MAC 地址长度, 这里是 6 个字节, 48bit
- ✚ 协议地址大小, 1 字节, 标识 IP 地址长度, 这里是 4 个字节, 32bit
- ✚ 操作码, 2 字节, 标识 ARP 包的类型, 1 标识请求, 2 标识响应
- ✚ 发送方 MAC 地址, Attacker 的 MAC 地址 00:0c:29:b0:48:68
- ✚ 发送方 IP 地址, Attacker 伪造 Victim 的 IP 地址, 192.168.1.111
- ✚ 目的 MAC 地址, 网关的 MAC 地址 54:75:95:dd:ee:18
- ✚ 目的 IP 地址, 网关的 IP 地址 192.168.1.1

数据链路层:

- ◆ 目的 MAC 地址, 网关的 MAC 地址 54:75:95:dd:ee:18
- ◆ 源的 MAC 地址, Attacker 的 MAC 地址 00:0c:29:b0:48:68
- ◆ 上层协议类型, 0x0806, 表明是 ARP 而不是 IP

```
▶ Frame 54: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: Vmware_b0:48:68 (00:0c:29:b0:48:68), Dst: Vmware_a6:5f:24 (00:0c:29:a6:5f:24)
  ▶ Destination: Vmware_a6:5f:24 (00:0c:29:a6:5f:24)
  ▶ Source: Vmware_b0:48:68 (00:0c:29:b0:48:68)
  Type: ARP (0x0806)
  ▶ [Duplicate IP address detected for 192.168.1.1 (00:0c:29:b0:48:68) - also in use by 54:75:95:dd:ee:18 (frame 53)]
  ▶ [Duplicate IP address detected for 192.168.1.111 (00:0c:29:a6:5f:24) - also in use by 00:0c:29:b0:48:68 (frame 53)]
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Vmware_b0:48:68 (00:0c:29:b0:48:68)
  Sender IP address: 192.168.1.1
  Target MAC address: Vmware_a6:5f:24 (00:0c:29:a6:5f:24)
  Target IP address: 192.168.1.111
```

为了减少冗余, 和上面分析过一样的字段我后面就精简掉了, 主要分析不同的字段

ARP 层:

- 发送方 MAC 地址, Attacker 的 MAC 地址 00:0c:29:b0:48:68
- 发送方 IP 地址, Attacker 伪造网关的 IP 地址, 192.168.1.1
- 目的 MAC 地址, Victim 的 MAC 地址 00:0c:29:a6:5f:24

```
▶ Frame 10: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlo1, id 0
▼ Ethernet II, Src: IntelCor_e8:17:a5 (90:78:41:e8:17:a5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: IntelCor_e8:17:a5 (90:78:41:e8:17:a5)
  Sender IP address: 192.168.1.106
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.1.1
```

这是我之前网络实验中的一个 ARP 请求, 请求网关的 MAC 地址

比较特殊的是, 数据链路层目的 MAC 地址为 ff:ff:ff:ff:ff:ff, 表明这是一个广播, 接收到的节点检查自己的 IP 地址与目标 IP 地址 192.168.1.1 是否匹配, 匹配的话就回应自己的 MAC 地址

解决办法

1. 静态 ARP 表, 通过分配固定 IP 而不是 DHCP, 将 MAC 地址静态地映射到 IP 地址, 设备

有变动的話，整个局域网内的主机 ARP 表都得更新，相比自学习，需要人工维护。适合一些小型家庭局域网、保密单位

2. 交换机 ARP 检测，企业级交换机会在每个端口上启用 DAI，除了连接到其他交换机的端口，DAI 就是 Dynamic ARP Inspection，DAI 会拦截所有的 ARP 请求和响应，根据可信数据库中 MAC 地址与 IP 地址的绑定关系来确认 ARP 是否合法，这个数据库是通过 DHCP Snooping 建立的，因为得确保如果 IP 地址变动能及时与 MAC 地址对应，DAI 会丢弃那些可疑或者恶意的 ARP 响应，并且限制每个主机发送给交换机 ARP 包的速率
3. 交换机端口检测，只允许每个端口只能有一个 MAC 地址，防止攻击者有多个 MAC 地址多重身份
4. 防止恶意的人接入局域网，ARP 攻击必须是在同一个局域网内发生的，我们可以使用 802.1x, WI-FI Protected Access 等进行身份认证，确保只有我们信任的主机设备才可以接入局域网。还有一种情况，局域网内主机中恶意病毒了，被远程操控进行 ARP 攻击，这时候就防不胜防，所以装杀毒软件还是很有必要的
5. 网络隔离，合理组网，分配网络号和子网掩码，将主机分为几个不同的子网，将重要的资源、信息集中到一个专用子网，ARP 攻击只能发生在同一子网内
6. 信息加密，虽然无法防止 ARP 攻击，但是能帮助减少损失。ARP 攻击一个重要的目的就是获取他人的登录密码等信息，如果密码以明文传输的话就十分危险了，所以登录的时候密码得进行对称加密或者非对称加密，让攻击者无法破解出正确密码
这也是为什么实验要求我们访问的事 HTTP 网站，因为如果访问 HTTPS 网站的话，数据都是通过 SSL/TLS 机制加密的，Attacker 就解析不出来正确的数据
7. 使用虚拟专用网络 VPN，设备和 Internet 之间的连接通过一条加密隧道，所有的数据都会被加密，当然这些加密的数据攻击者拿到是 Worthless 的
8. 及时发现 ARP 攻击，现在电脑手机上的杀毒软件都有 ARP 监测这一项，如果我们真的被攻击了，不急，Wireshark 抓个包，查看是谁在冒充我们，因为他发送 ARP 应答报文的时候需要附上自己的 IP，拿到他的 IP，我们是不是可以进行一些报复呢