Web安全工程师面试（SQL、XSS、CSRF、SSRF）

原创

Hardworking666 已于 2022-03-29 22:07:22 修改 1701 收藏 4

分类专栏： 护网（HW） 文章标签： Web漏洞 SQL注入 xss csrf 安全

于 2021-12-21 20:03:53 首次发布

文章目录

一、Web 访问过程分析

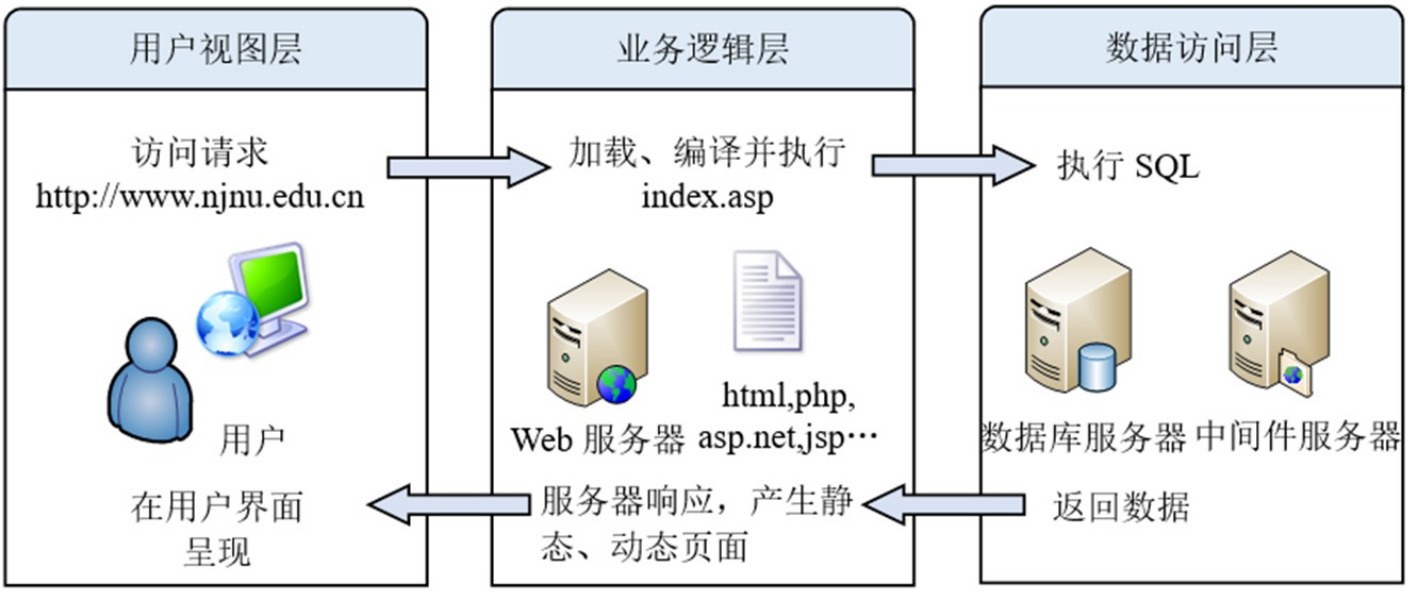
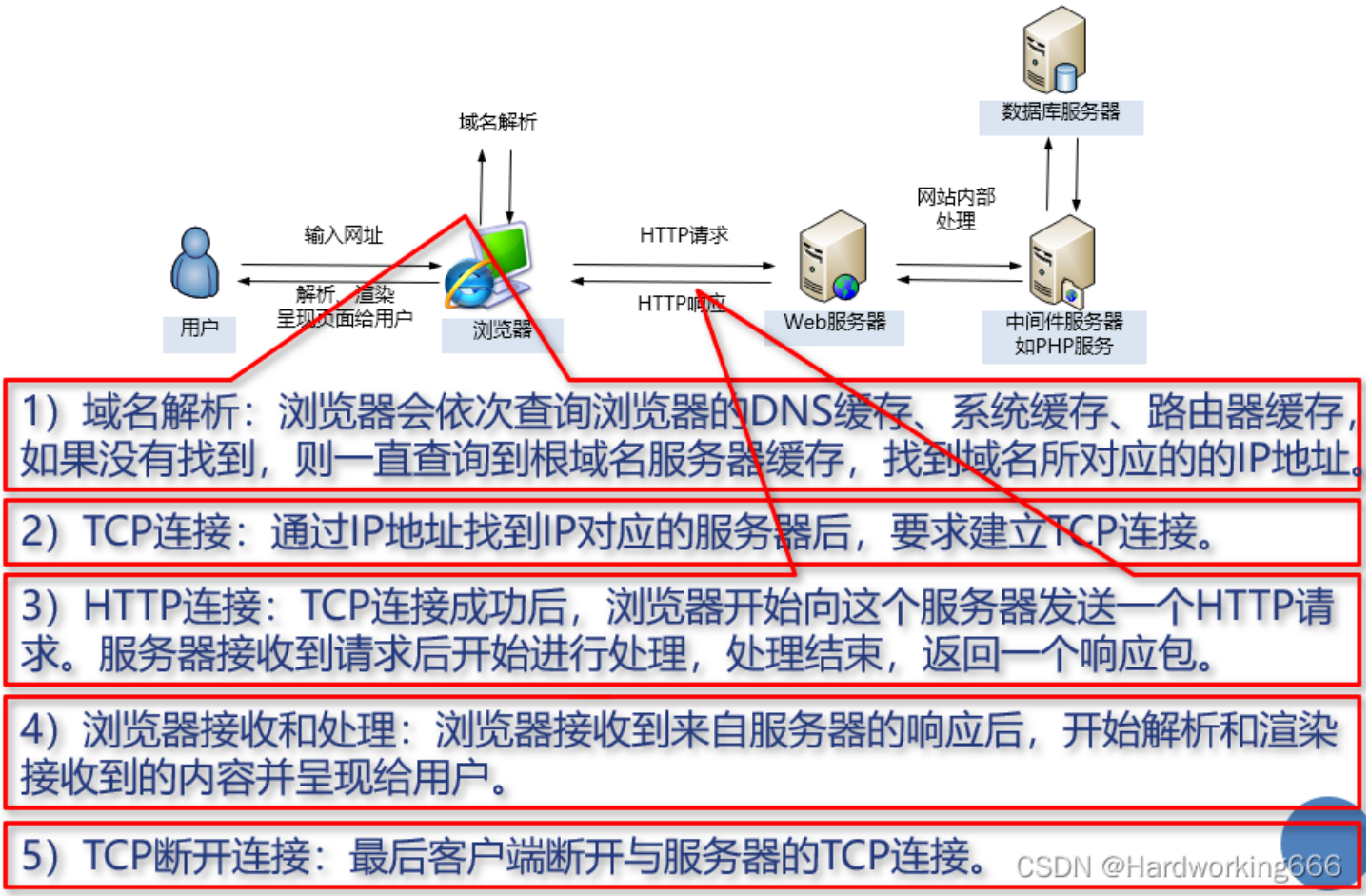


图 4-1　Web 三层通用架构

一次Web访问过程分析：**DNS域名解析、TCP连接、HTTP请求、处理请求返回HTTP响应、页面渲染和关闭连接。**



二、SQL注入

SQL注入漏洞是指，

攻击者能够利用现有Web应用程序，将**恶意的数据**插入**SQL查询**中，提交到**后台数据库引擎**执行**非授权操作**。

SQL注入攻击利用的工具是**SQL语法**。

1、SQL注入的危害

1、非法查询、修改或删除**数据库资源**。
2、执行**系统命令**。
3、获取承载主机操作系统和网络的**访问权限**。

2、SQL注入思路

1、注入点选择
2、数字型和字符型注入
3、通过Web端对数据库注入或者直接访问数据库注入

3、SQL注入的类型

1、报错注□
2、bool 型注□
3、延时注□
4、宽字节注□

4、SQL注入防护

1、使□安全的 API
2、对输□的特殊字符进□ Escape 转义处理
3、使□□名单来规范化输□验证□法
4、对客户端输□进□控制，不允许输□ SQL 注□相关的特殊字符
5、服务器端在提交数据库进□ SQL 查询之前，对特殊字符进□过滤、转义、替换、删除。
6、规范编码, 字符集

三、XSS跨站脚本

跨站脚本(Cross Site Script)，简称XSS。

XSS漏洞是指：应用程序没有对接收到的**不可信数据**经过适当的**验证或转义**就直接发给客户端浏览器。
原理：web浏览器可以执行HTML页面中嵌入的脚本命令，攻击者利用XSS漏洞将恶意脚本代码注入到网页中，当用户浏览该网页时，便会触发执行恶意脚本。

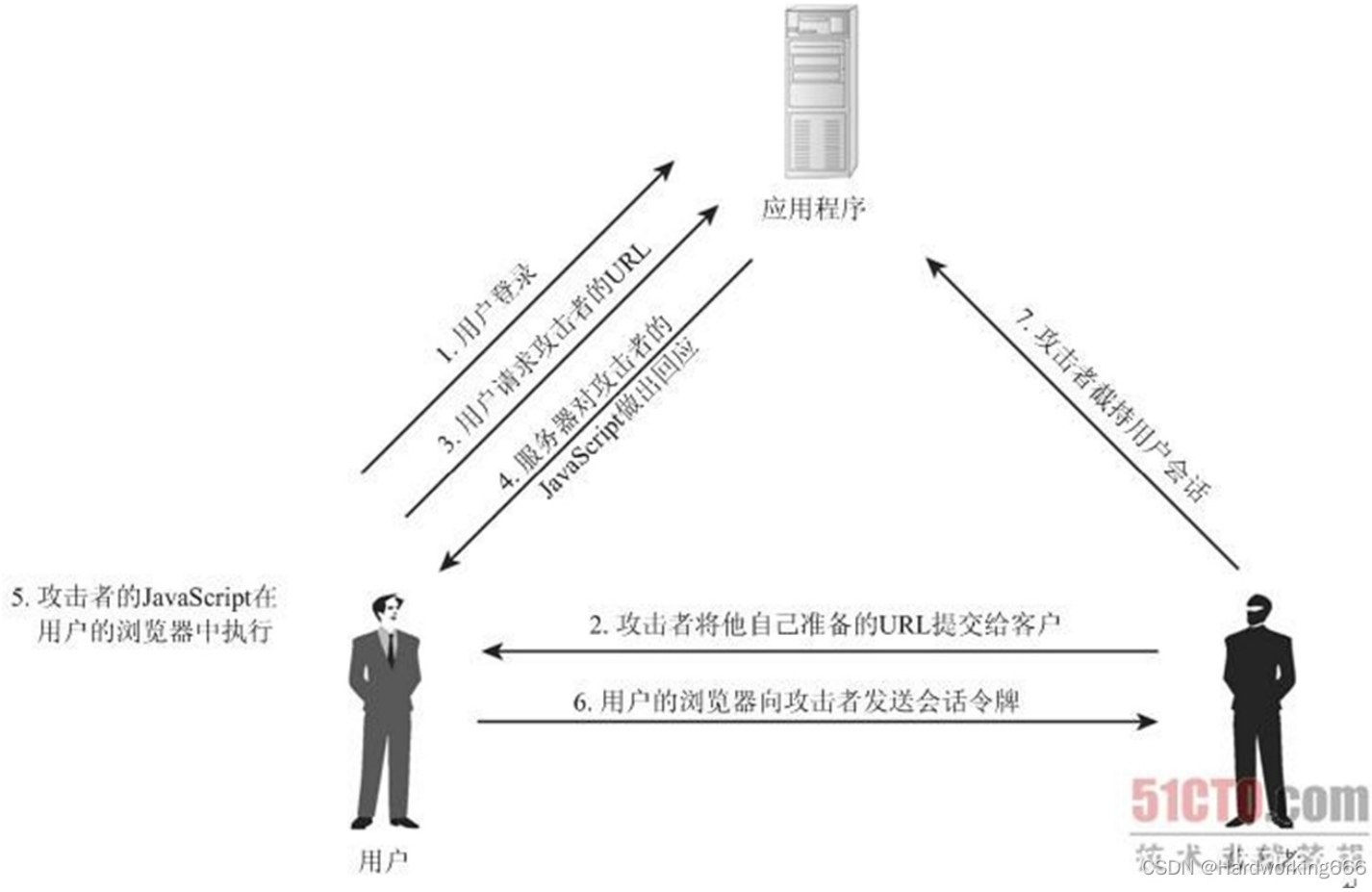
XSS漏洞主要**危害**
1 非法访问、篡改**敏感数据**
2 **会话劫持**
3 控制受害机器**向其他站点发起攻击**

1、反射型XSS漏洞原理

1）最普遍的一种类型。
2）服务器直接使用客户端提供的数据而没有对数据进行无害化处理，就会出现此漏洞。
3）特点：**用户单击时触发，而且只执行一次**，因此反射型XSS也称为非持久型XSS。

反射型XSS通常是由
攻击者诱使用户向有漏洞的Web应用程序**提供危险内容**，然后危险内容会**反射给用户**并由浏览器执行。

XSS漏洞潜在影响的一种攻击：可导致攻击者截获一名通过验证的用户的会话。
劫持用户的会话后，攻击者就可以**访问该用户经授权访问的所有数据和功能**。
实施这种攻击的步骤如图：



2、存储型XSS漏洞原理

存储型XSS也称为持久型XSS，它的危害更大。此类XSS **不需要用户单击特定的URL** 就能执行跨站脚本。攻击者事先将恶意脚本代码上传或者存储到存在漏洞的**服务器端数据库**中，只要用户浏览包含此恶意脚本的网页便会触发，遭受攻击。

存储型XSS漏洞通常在**留言板、个人资料、博客日志**等位置出现，并常被用于编写危害性更大的**XSS蠕虫**。

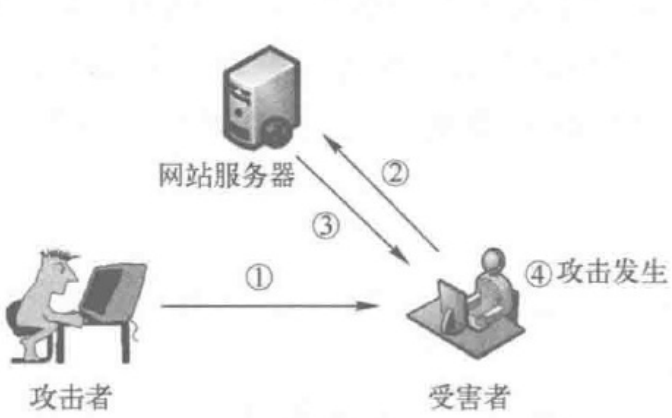


图 4-20　反射型 XSS 攻击过程

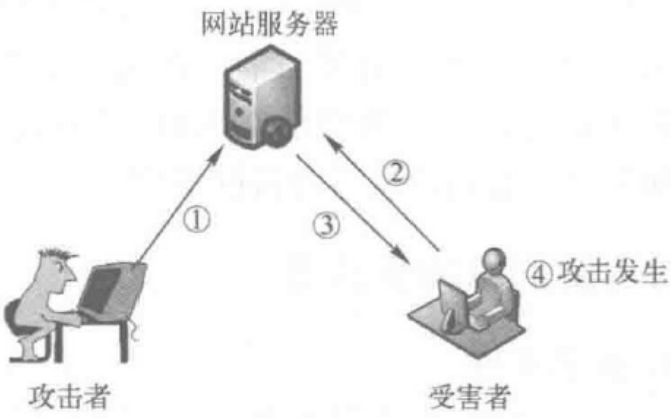


图 4-21　存储型 XSS 攻击过程

3、基于DOM的XSS

基于DOM的XSS又称为**本地XSS**，DOM 型 XSS 漏洞是基于**文档对象模型** （Document Objeet Model，DOM）的□种漏洞。
由于客户端浏览器JavaScript可以访问浏览器的DOM动态地检查和修改页面的内容，当HTML页面采用不安全的方式从document.location、document.URL、document.referrer或其他攻击者可以修改的对象获取数据时，如果数据包含恶意JavaScript脚本，就会触发基于DOM的XSS攻击。
基于DOM的XSS攻击与反射型XSS和存储型XSS不同，基于DOM的XSS攻击来源于客户端处理的脚本中，**无需服务器端的参与**。

文档对象模型 (DOM) **将 web 页面与到脚本或编程语言连接**起来。通常是指 JavaScript，但将 HTML、SVG 或 XML 文档建模为对象并不是 JavaScript 语言的一部分。DOM模型用一个逻辑树来表示一个文档，树的每个分支的终点都是一个节点(node)，每个节点都包含着对象(objects)。DOM的方法(methods)让你可以用特定方式操作这个树，用这些方法你可以改变文档的结构、样式或者内容。节点可以关联上事件处理器，一旦某一事件被触发了，那些事件处理器就会被执行。

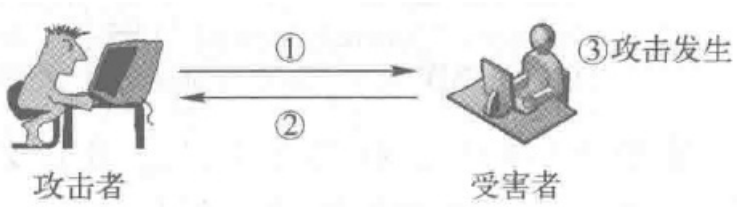


图 4-22　基于 DOM 的 XSS 攻击过程

表 4-1　XSS 漏洞类型比较

对　比　项	反　射　型	存　储　型	基于 DOM
是否需要用户单击 URL	是	否	是
是否与服务器交互	是	是	否
是否持久	否	是	否
危害程度	中等	高	高
防御难度	一般	难	CSDN @Hardworking666

4、XSS未给出具体位置的解决

如果安全应急响应中心（SRC，Security Response Center） 上报了□个 XSS 漏洞，payload 已经写□页面，但未给出具体位置，如何快速介□？

看是什么类型的 XSS，XSS 反射型看提交的地址，指的参数是哪个位置，通过这个□□进□ fuzzing 测试。如果是存储型□□查找关键字。
修复方式：对字符实体进□转义、使□ HTTP Only 来禁□ JavaScript 读取 Cookie 值、输□时校验、浏览器与 Web 应□端采□相同的字符编码。

5、DOM 型和 XSS 自动化测试或人工测试

人工测试思路：找到类似 document.write、innerHTML 赋值、outterHTML 赋值、window.location 操作、写 javascript: 后内容、eval、setTimeout 、setInterval 等直接执□之类的函数点。找到其变量，回溯变量来源观察是否可控，是否经过安全函数。

自动化测试参看道哥的博客，思路是从输入入手，观察变量传递的过程，最终检查是否有在危险函数输出，中途是否有经过安全函数。但是这样就需要有□个 javascript 解析器，否则会漏掉□些通过 js 执□带□的部分内容容。

四、CSRF跨站请求伪造

1、CSRF与XSS区别

跨站请求伪造(Cross-site request forgery)简称CSRF，尽管与跨站脚本漏洞名称相近，但它与跨站脚本漏洞不同。

XSS利用**站点内的信任用户**，而CSRF则通过**伪装来自受信任用户的请求**来利用受信任的网站。

CSRF和反射型XSS的主要区别是：反射型XSS的目的是在**客户端**执行脚本，CSRF的目的是在**Web应用**中执行操作。

CSRF跨站请求伪造攻击迫使登录用户的浏览器将伪造的HTTP请求，包括该用户的会话Cookie和其他认证信息，发送到一个存在漏洞的Web应用程序，而这些请求会被应用程序认为是用户的合法请求。

2、CSRF的危害

篡改□标□站上的□户数据、盗取□户隐私数据、传播 CSRF 蠕虫。

3、CSRF的防御

CSRF 防御原理：不让黑客那么容易伪造请求
1、cookie 中加□随机数，要求请求中带上，□攻击者获取不到 cookie中的随机数。
2、验证HTTP Referer 字段, 在请求地址中添加 takon 验证。

4、CSRF护网面试

token 和 referer 做横向对比，谁安全等级高？

token 安全等级更□，因为并不是任何服务器都可以取得 referer，如果从 HTTPS 跳到 HTTP，也不会发送 referer。并且 FLASH □些版本中可以□定义 referer。但是 token 的话，要保证其□够随机且不可泄露。(不可预测性原则)

对 referer 的验证，从什么角度去做？

对 header 中的 referer 的验证，□个是空 referer，□个是 referer 过滤或者检测不完善。为了杜绝这种问题，在验证的白名单中，正则规则应当写完善。

针对 token，会对 token的哪方面进□测试？

针对token的攻击，□是对它本身的攻击，重放测试□次性、分析加密规则、校验□式是否正确等，□是结合信息泄露漏洞对它的获取，结合着发起组合攻击信息泄露有可能是缓存、□志、get，也有可能是利□跨站很多跳转登录的都依赖token，有□个跳转漏洞加反射型跨站就可以组合成登录劫持了。另外也可以结合着其它业务来描述token的安全性及设计不好怎么被绕过□如抢红包业务之类的。

五、SSRF服务器端请求伪造

1、成因

SSRF(Server-Side Request Forgery，服务器端请求伪造)。是一种由攻击者构造形成**由服务端发起请求**的一个安全漏洞。
一般情况下，SSRF攻击的目标是**从外网无法访问的内部系统**。

很多Web应用都提供了**从其他服务器上获取数据**的功能。使用用户指定的URL，Web应用可以获取图片，下载文件，读取文件内容等。这个功能如果被恶意使用，可以利用存在缺陷的web应用**作为代理**攻击远程和本地服务器。

2、危害

可以对外网服务器所在的内网、本地进行端口扫描，获取一些服务的banner信息 。
攻击运行在内网或者本地的应用程序。
对内网web应用进行指纹识别，通过访问默认文件实现 。
攻击内外网的web应用。sql注入、struct2、redis等。
利用file协议读取本地文件等。

3、利用

1、可以对外网、内网、本地进行端口扫描，某些情况下端口的Banner会回显出来（比如3306的）；
2、攻击运行在内网或本地的有漏洞程序（比如溢出）；
3、可以对内网Web应用进行指纹识别，原理是通过请求默认的文件得到特定的指纹
4、攻击内网或外网有漏洞的Web应用
5、使用file:///协议读取本地文件

4、漏洞挖掘

一. WEB功能上查找
1、分享：通过URL地址分享网页内容
通过URL地址分享网页内容早期应用中，为了更好的用户体验，Web应用在分享功能中，通常会获取目标URL地址网页内容中标签或者<meta name="description"content=""/>标签中content的文本内容提供更好的用户体验。
2、转码服务：通过URL地址把原地址的网页内容调优使其适合手机屏幕浏览
3、在线翻译：通过 URL地址翻译对应文本的内容
4、图片加载与下载：通过URL地址加载或下载图片
图片加载远程图片地址此功能用到的地方很多，但大多都是比较隐秘，如有些公司中的加载自家图片服务器上的图片用于展示。（开发者为了有更好的用户体验通常对图片做些微小调整例如加水印、压缩等，就必须要把图片下载到服务器的本地，所以就可能造成SSRF问题）。
5、图片、文章收藏功能
6、未公开的api实现以及其他调用URL的功能
7、从URL关键字中寻找

二. 从URL关键字中寻找
Share、wap、url、link、src、source、target、u、3g、display、sourceURL、imageURL、domain

三. 通用的SSRF实例
Weblogic配置不当，天生ssrf漏洞
Discuz x2.5/x3.0/x3.1/x3.2 ssrf漏洞

5、防御方法

1、过滤返回信息，验证远程服务器对请求的响应是比较容易的方法。如果web应用是去获取某一种类型的文件。那么在把返回结果展示给用户之前先验证返回的信息是否符合标准。
2、统一错误信息，避免用户可以根据错误信息来判断远端服务器的端口状态。
3、限制请求的端口为http常用的端口，比如，80,443,8080,8090。
4、黑名单内网ip。避免应用被用来获取获取内网数据，攻击内网。
5、禁用不需要的协议。仅仅允许http和https请求。可以防止类似于file:///,gopher://,ftp:// 等引起的问题。

6、绕过方法

1、@
http://abc@127.0.0.1
2、添加端口号
http://127.0.0.1:8080
3、短地址
http://dwz.cn/11SMa
4、可以指向任意ip的域名：xip.io
5、ip地址转换成进制来访问
115.239.210.26 ＝ 16373751032

六、学习链接

GitHub上的安全知识框架：
项目文件一览
Security

安全工具 - 各类安全工具的使用介绍

安全资源

靶机
HTB
VulnHub
DC Serial - DC 系列靶场,难度简单至中等,可以学习各种提权和CMS漏洞利用,推荐初学者挑战
It's_October
Kioptrix Serial - Kioptrix 系列靶场,难度简单至中等,推荐初学者挑战
Mission-Pumpkin - 难度适中,偏向于加解密比较多,漏洞利用内容较少
symfonos Serial - 挺有难度的靶场,内容丰富,难度中等,漏洞利用内容很多,推荐有一定经验者挑战
Wargames
Bandit
BlueTeam

分析 - 分析工具与分析案例
加固 - 系统、应用加固的方法和工具资源
监察 - 有关查杀、监控、蜜罐的资源
取证 - 内容涉及操作系统的取证、web 的取证、文件的取证
应急 - 应急资源、溯源案例
笔记 - 涉及磁盘取证、内存取证、USB取证等内容
实验 - 涉及流量分析实战、安防设施搭建等内容
Crypto

Crypto - 介绍各种编码和加密算法及相关的工具
CTF

CTF - 收集 CTF 相关的工具和 writeup 资源
writeup - 自己参与的一些比赛记录
ICS

工控协议 - 总结各类工控协议的知识点
上位机安全 - 总结上位机安全相关的知识点
PLC攻击 - 总结 PLC 攻击的相关知识点
S7comm相关 - 记录 S7comm 相关错误类型、功能码和相关参数
实验 - 仿真环境搭建和 PLC 攻击实验
IOT

固件安全
固件安全 - 记录 IOT 固件分析的知识点,包括固件提取、固件分析、固件解密等
实验 - 分析固件实验
无线电安全
实验 - 无线电安全实验
硬件安全
Device-Exploits - 嵌入式设备相关漏洞利用,不太熟悉这一块,内容不多
HID - 和组员制作的 HID 实物记录
MobileSec

Android安全 - 记录一些安卓安全相关的内容,这块掌握较少
RedTeam

安防设备
Bypass技巧 - 记录 waf 绕过手段
SecDevice-Exploits - 常见的安全设备的漏洞利用方法
后渗透
后渗透 - 后渗透知识点的大纲
权限提升 - 操作系统和数据库的提权方法
权限维持 - 权限维持的各种方法和资源
实验
软件服务安全
CS-Exploits - 收集软件、业务应用服务漏洞的渗透手段和 cve 漏洞
DesktopApps-Exploits - 收集桌面软件的渗透手段和 cve 漏洞
协议安全
Protocol-Exploits - 按照协议归类各种漏洞、攻击手段
笔记
信息收集
端口安全 - 记录端口渗透时的方法和思路
空间测绘 - 收集搜索引擎语法资源
信息收集 - 记录信息收集方面各类技术，如漏扫、IP 扫描、端口扫描、DNS 枚举、目录枚举、指纹等
语言安全
语言安全
云安全
云安全 - 云主机利用工具,渗透案例,相关知识点
OS安全
Linux安全 - 包含 Linux 口令破解，漏洞利用、获取Shell
OS-Exploits - 收集操作系统的 cve 漏洞
Windows安全 - 包含 windows pth、ptt，漏洞利用、提权、远程执行命令
实验
Web 安全
前端攻防 - 前端解密,绕过访问
BS-Exploits - 全面收集 web 漏洞 POC | Payload | exp
IDOR - 整个部分结构大部分基于乌云的几篇密码找回、逻辑漏洞类文章,在其基础上记录和归纳
靶场
Web_Generic
Web_Tricks
Reverse

Reverse
实验
FILE
Develop

版本控制

Git学习笔记 - 记录 git 的用法和平时使用 github 遇到的问题
标记语言

HTML
JSON
XML
可视化

gnuplot
正则

regex - 常用正则表达式和相关资源
Web

Speed-Web
HTTP
笔记
Integrated

数据库

Power-SQL
Speed-SQL
笔记
实验
虚拟化

Docker
Linux

God-Linux - 记录 Linux 下的骚操作,收集的较少,后面会慢慢添加
Power-Linux - 配置指南,记录各种服务搭建与配置过程
Secure-Linux - Linux 加固+维护+应急响应参考
Speed-Linux - 命令速查手册,记录各种基本命令操作
笔记
实验 - 各种 linux 服务的搭建过程和案例
Network

不同厂商 - 记录不同厂商配置服务命令的区别
方向实验 - 按方向分类记录配置
速查 - 速查各类帧、报文格式、掩码等
SDN笔记 - 记录以前比赛时 SDN 的题目和命令
TCP-IP - 记录 TCP/IP 协议栈的协议
VPN-Security - 记录 VPN 领域的协议
Windows

Secure-Win - Windows 加固+维护+应急响应参考
Speed-Win - 记录 windows 下 CMD 常用命令
笔记
实验 -涉及域环境搭建、基础服务搭建
Powershell
Plan

Misc-Plan - 各种小技巧
Team-Plan - 团队协作解决方案
Thinking-Plan - 问题解决方式的记录和学习
VM-Plan - VMWare 常见问题记录