

一.网安教育

1.网络安全法

《中华人民共和国网络安全法》第27条明确规定，任何个人和组织不得从事非法侵入他人网络、干扰他人网络正常功能、窃取网络数据等危害网络安全的活动；不得提供专门用于从事侵入网络、干扰网络正常功能及防护措施、窃取网络数据等危害网络安全活动的程序、工具；明知他人从事危害网络安全的活动的，不得为其提供技术支持、广告推广、支付结算等帮助。

- 非法授权扫描漏洞属于犯法。
- 找到漏洞时不能获取内部隐私数据。
- 找到漏洞-报告-分析漏洞的危害程度
- 甲方漏洞要写明自己的责任时间，例如：漏洞维护至...
- 实例：网警发现企业系统存在漏洞，若在规定时间内不整改会面临巨额惩罚。
- 不扫公网IP、政府网址漏洞
- 2021“数据安全法”、“个人隐私保护法”

2.渗透测试攻击需要的权限

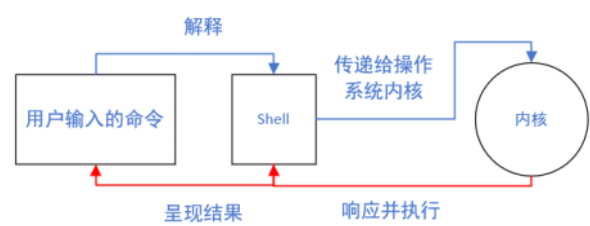
渗透测试提权

webshell 网站控制权  
shell 操作系统控制权

(1) shell:

①介绍:

英文单词含义为外壳，剥壳。在计算机中被用作一种与内核操作系统交互的界面。



②常见两种形式:

- GUI:Graphical (图形) User (用户) Interface (界面)
- CLI:Command-Line (命令行) interface (界面)

③常见的类型:

- (Linux系统) **Bash**: Bourn Again (=Born Again重生) Shell (外壳)
- (Linux系统+Mac系统+Windows系统) **Zsh**: Z Shell
- (Windows系统) **Cmd**: Command(命令) Prompt (提示符)
- (Windows系统) **Powershell**: Windows的高级命令行界面，集成了.NET 框架，功能更强大。

(2) Webshell

①介绍:

Webshell 是一种通过 Web 服务器上的网页接口提供远程访问和控制的工具。

它通常是一个恶意脚本文件，攻击者可以上传到目标服务器，通过 Web 浏览器访问它来执行命令和进行各种恶意操作。

②特点和功能:

- 文件上传/下载: 攻击者可以通过 Webshell 上传和下载文件。
- 命令执行: 可以在服务器上执行操作系统命令。
- 数据库操作: 能够连接和操作服务器上的数据库。
- 权限提升: 尝试提升在服务器上的权限，以获取更高的控制权。
- 持久化后门: 创建持久化访问方式，以便在未来继续访问。

③常见的Webshell语言:

- PHP (最常见)
- ASP: 用于Windows服务器上的Webshell。
- JSP: 用于Java服务器上的Webshell。

二.课程体系



1.系统安全加固

(1) 操作系统安全加固

例: 基线检查

- 弱口令
- 敏感文件和敏感目录及权限
- 敏感账号设置是否正确
- 端口开放情况 (测定白名单) 等。

(2) 中间件 (服务支撑软件) 安全加固



常见的中间件:

- Web服务器和应用服务器: Apache, Nginx, Microsoft Internet Information Services (IIS), Tomcat
- 数据库中间件: MySQL、PostgreSQL、Oracle Database、Microsoft SQL Server、MongoDB

- 消息队列和消息中间件: RabbitMQ、Apache Kafka、ActiveMQ、Redis
- 服务中间件: Spring Boot、Django、Express.js、Flask
- API网关和服务网格: Kong、Apigee、Istio、Envoy
- 缓存中间件: Memcached、Redis

(3) 数据库安全加固

数据库 mysql Oracle MS-SQL redis

• redis往往是内网薄弱点

(4) 云主机安全加固

云主机: 阿里云 百度云 华为云 腾讯云 天翼云 laas Pass SaaS

2.渗透测试基础

3.渗透测试

- 最关键的是“信息搜集”
- 打开渗透思路,主站不好渗透,从薄弱资产入手信息搜集。

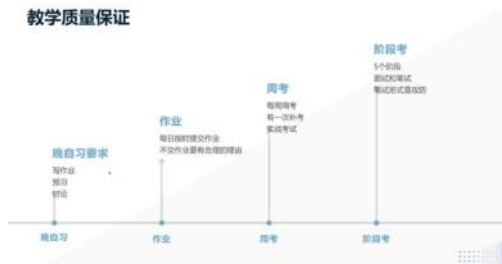
4.内网渗透

第五部分实训

三.教学质量保证

每日晚自习(当天有作业)+每周周考+阶段考(5次,面试加笔试)

教学质量保证



四.技术内容

- Web渗透主要是B/S架构模式
  - APP渗透主要是C/S架构模式
- APP渗透客户端面临的问题: 反编译、个人隐私信息保护、数据泄露以及传输接口是否安全?

• TCP/IP协议书籍阅读

Web渗透:

浏览器--服务器模式 瘦客户端 B/S

客户端--服务器模式 胖客户端 C/S

五.fiddler挖掘业务逻辑漏洞

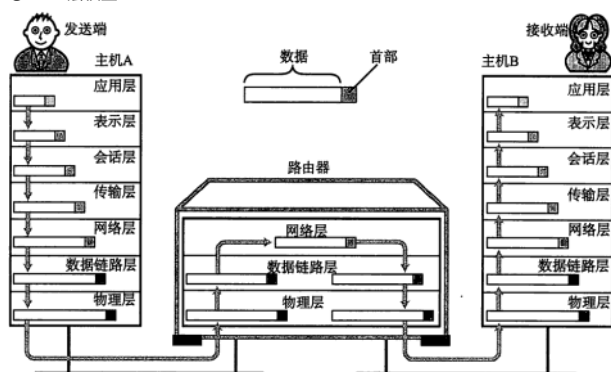
1.介绍

①fiddler抓取的是应用层的流量包, BP也是, Wireshark全流量抓包。

2.必备知识

(1) 网络体系结构

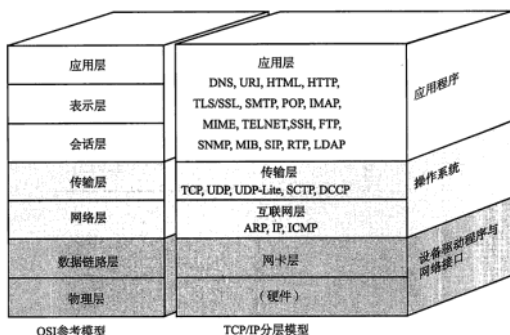
①OSI七层模型



②TCP/IP五层模型

从上至下: 应用层-传输层-网络层-数据链路层-物理层

- 应用层协议 http https ftp smtp pop3 dns 面向用户
- 传输层协议 tcp udp 传输分组 差错检测、流量控制
- 网络层 IP ICMP: ping IP路由 IP包头
- 数据链路层 ARP RARP 局域网寻址 增加MAC地址包头
- 物理层 比特流



(2) fiddle操作

- 抓包的目的是看流量，观察流量中是否有攻击流量，数据包是否异常。
- 动手做：对数据包进行修改。

- ctrl x清除当前列表
- ctrl f搜索

- Raw格式：行格式（数据包基本模式）

### ①特点

- 请求-响应模型

- 无连接导致无状态

- 长连接机制（几分钟，保证带宽不

②数据包格式:

POST <http://www.showbu>

①请求行: 请求方法 URL 协议版本 --有空格

- GET功能：请求从服务器获取指定资源。

- 面试题:GET与POST的区别?

- GET请求:

• POST请求: 通常用于提交数据给服务器

- POST请求：通常用于提交数据给服务器
- 提交表单数据、上传文件等。

2. 参数传递方式不同:

- GET请求：参数通过URL传递，URL中包含查询字符串。
    - 示例如<http://example.com/page?param1=value1&param2=value2>。
  - POST请求：参数包含在请求体中，浏览器不会在地址栏中显示提交的数据。
2. 安全性不同。

- GET请求：因为参数在URL中明文传输，所以敏感数据容易暴露在URL中，不适合传递敏感信息。

- POST请求：参数在请求体中传输，相对而言更为安全，可以传递敏感数据。
- 但并不是绝对安全，仍需加密（如HTTPS）。

- GET请求: GET请求的数据长度较短。

- 由于URL长度限制（通常是2048个字符，但不同浏览器和服务器的限制可能不同），
- POST请求：理论上没有数据长度限制，可以传递大量数据。
- 实际限制取决于服务器配置和内存大小。

- GET请求：GET请求是幂等的，即对同一资源的多个相同的GET请求应返回相同的结果。

- GET请求通常会被浏览器缓存。
- POST请求: POST请求不是幂等的, 每次请求可能会导致服务器状态的改变(如新增记录)。
  - 浏览器一般不缓存POST请求。

1. 基础格式: `scheme://userinfo@host:port/path?query#fragment`

①scheme (协议)

②userinfo (用户信息, 可选):

- ✧ 包含用户名和密码，通常用于需要认证的服务器访问。
- ✧ 格式为: `username:password`
- ✧ 例如: `user:pass@`

- ✧ 定义服务器的主机名或IP地址。

- ✧ 例如: [www.example.com](http://www.example.com)、192.168.1.1。

- ☆ 定义服务器使用的端口号。如果省略，默认使用协议的标准端口（如HTTP默认80，HTTPS默认443）。

- ◇ 例如: :8080。

- ☆ 定义资源在服务器上的路径 可以是多级路径

- ☆ 例如: /path/to/resource.

⑥query (查询子串, 可选):  
 包含要发送给服务器的查询参数。通常用于GET请求。

- ✧ 包含要发送给服务器的查询参数，
  - ✧ 格式为键值对，多个参数用&分隔。
  - ✧ 例如：?key1=value1&key2=value2。
- ⑦ fragment (片段标识符，可选)：

- ✧ 用于在资源内定位特定部分，通常用于HTML文档的锚点。
- ✧ 例如：[#id=1](#)

- ✧ 例如: #section1。

- URL中的参数用?引用，多个参数用&符号连接
- URL: UniqueResourceLocation统一资源定位符

下面是一个完整的URL示例及其组成部分：

- ◇ scheme: https
- ◇ userinfo: user:pass
- ◇ host: www.example.com
- ◇ port: 8080
- ◇ path: /path/to/resource
- ◇ query: key1=value1&key2=value2
- ◇ fragment: #section1

②请求头：一系列键值对来说明请求的属性。

- Cookie值如果前后没有发生改变，存在漏洞。
- Referer和Cookie 容易被注入
- UA:User-Agent说明浏览器内核情况

③空行

④请求体

(II) 响应包格式

①响应行：协议 状态码 描述 状态码描述 一有空格

- 状态码：
  - 2xx： 正常响应
  - 3xx： 跳转 进一步响应
  - 4xx： 客户端错误响应
    - 404 客户端请求URL有误
    - 403 客户端请求权限不足
    - 415 客户端请求类型不对
  - 5xx： 服务端响应错误

②响应头：键值对说明响应属性

- Server： 旗标信息，说明服务器中间件的类型和版本。
  - 安全角度考虑：名字最好自定义且要隐藏版本-属于服务器版本（敏感信息）泄露
  - 中间件编译安装可以任意取名，其中的BWS/1.1就是编译安装的自定义名字。

```
Isprivate: 1
Server: BWS/1.1
```

- 示例：京东网站的Server信息，nginx为中间件

```
HTTP/1.1 200 OK
Server: nginx
```



③空行

④响应体：

- 常见的格式如下：
  - ◇ HTML：HyperText Markup Language（超文本标记语言）。用于网页内容的展示，通常用于Web浏览器的响应。

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
```
  - ◇ XML：eXtensible Markup Language（可拓展标记语言）。具有严格的语法规则，用于结构化数据。适用于需要复杂数据结构和验证的场景。

```
<person>
  <name>John Doe</name>
  <age>30</age>
  <city>New York</city>
</person>
```
  - ◇ 纯文本（Plain Text）：简单的文本数据，没有格式。

```
Hello, World!
```
  - ◇ JSON：JavaScript Object Notation（JavaScript对象表示法）。轻量级的数据交换格式，可读性强，易于解析和生成。

```
{
  "name": "John Doe",
  "age": 30,
  "city": "New York"
}
```
  - ◇ YAML：YAML Ain't Markup Language（YAML不是一种标记语言）。另一种数据序列化格式，常用于配置文件。

```
name: John Doe
age: 30
city: New York
```
  - ◇ CSV：Comma-Separated Values（逗号分隔值）。用于存储表格数据（数字和文本）的纯文本格式。

```
name,age,city
John Doe,30,New York
```
  - ◇ 二进制数据（Binary Data）：用于传输文件或其他非文本内容。例如，图像、音频、视频文件。

### 3.Fiddle抓包

(1) 抓https数据包

①s是指SSL协议：

- ◇ 含义：Secure Sockets Layer（安全套接层）。
- ◇ 功能：它在传输层对网络连接进行加密，防止数据在传输过程中被窃听、篡改或伪造。
- ◇ 特点：SSL协议通过使用加密算法和身份验证技术来确保通信的机密性和完整性。
- ◇ 版本：SSL的最新版本是TLS（Transport Layer Security，传输层安全）协议，TLS是SSL的后续版本，并且更为安全和强大。

②实现方式：使用SSL证书。

证书内容：拥有者的公钥，拥有者的相关信息等。

(2) 抓手机模拟器的包

①先决条件

在同一局域网内  
支持抓取https的包

②配置

xposed框架解决抓包对抗问题 一夜神模拟器商店内下载  
原因：安卓7.0以上会有抓包对抗，防止程序进行流量抓取。



### (3) fiddler的自动响应器

请求的重定向AutoResponder

**小作业：重定向能重定向到什么？**

- ✧ HTTP/HTTPS请求：可以将某个特定的URL请求重定向到另一个URL。这在测试不同服务器的响应或模拟特定环境下非常有用。
- ✧ 本地文件：可以将请求重定向到本地文件。这允许开发人员在不需要访问远程服务器的情况下测试本地资源，例如将对某个JavaScript文件的请求重定向到本地磁盘上的另一个JavaScript文件。
- ✧ 自定义响应：可以创建并返回自定义的HTTP响应。这可以用于模拟服务器的各种响应情况，例如返回特定的状态码、头信息和响应体内容。
- ✧ 替换资源：可以替换某些请求资源的内容，例如用本地修改后的CSS文件替换服务器上的CSS文件，以便快速进行样式调试和测试。

示例：修改百度界面的前端展示



### (4) fiddler的断点（正在进行时）

①请求前断点

②响应后断点

③全局断点：给所有请求打上断点

fiddler中下方的空格设置

↑为请求前断点



↓为响应后断点



改请求、看响应

改请求、改响应

④局部断点

Bpu：局部请求前断点。http连接前需要Connect请求，全局断点会拦截，而局部则不会

带参数（有？的URL情况）会有URL编码问题，导致局部拦截失败

例如：<http://www.shouhuola.com/c-20>（不带参数的URL）可以修改c-20为c-21来切换到不同的界面。

如何取消bpu，直接在下方输入bpu（后面不带任何）



Bpafter：局部响应后断点

功能与全局响应后断点相同，不过可以针对具体的URL资源。

取消也是在fiddler最下方黑色区域只输入bpafter即可



### (5) fiddler的构造器（完成时）

①模块：composer

②功能：请求，重发

③攻击作用：用于重发攻击、应用层的暴力破解、修包改包、ajax请求、异步处理等。

小技巧：如何快速找包：

- ①比对URL的主机host
- ②比对URL
- ③看返回值body数据包大小
- ④看细节

### (6) BurpSuite靶场注册

！注意：不要翻译成中文，否则无法进入邮箱的注册界面

PortSwigger

Create your account

Please enter your email address to register.

Email address:

Captcha:

Register

Already registered? Click [here](#) to log in.

Login

Please enter your email address and password to log in.

Email address:

Password:

[Forgot your password?](#)

☐ Remember me on this computer

Login Create account

初始密码是一堆字符，需要备份，第一次登陆之后可以修改密码，但是新密码也是一堆字符，自动生成的。

进入靶场网页：

<https://portswigger.net/web-security/all-labs#business-logic-vulnerabilities>

此次的练习BurpSuite靶场网址

<https://portswigger.net/web-security/authentication/other-mechanisms/lab-password-reset-broken-logic>

### Lab: Password reset broken logic

**Authentication** LAB Not solved

This lab's password reset functionality is vulnerable. To solve the lab, reset Carlos's password then log in and access his "My account" page.

- Your credentials: `wiener:petee`
- Victim's username: `carlos`

#### (7) 业务逻辑漏洞

##### ①漏洞分类：

- ✧ 通用型：常规漏洞如owasp top10, sql注入, xss, csrf, 有工具能扫。
- ✧ 业务逻辑漏洞：全屏个人经验，没有工具能扫出来。一般从业务逻辑处：登录注册，修改密码，购物车等。

##### ②业务逻辑漏洞类型

横向越权：用户A能看到B的数据。

纵向越权：用户A能获得管理员Administer的权限。

#### (3) 举例

- 刷赞，控制赞的数量

#### (4) BurpSuite靶场实战

### Login

Username

Password

[Forgot password?](#) 找回密码

Login

利用找回密码来水平越权。（找回密码往往有token验证）

有token时，观察每次发同样的包是否一致，一致则容易存在注入问题。

测试点：

- 删除token观察是否正常
- 若token修改后仍然能正常使用，尝试修改密码和账号进行水平越权

S1进入找回密码页面，输入账号wiener

Please enter your username or email

wiener

Submit

WebSecurity Academy

Password reset broken logic

[Back to lab home](#) [Email client](#) [Back to li](#)

Please check your email for a reset password link.

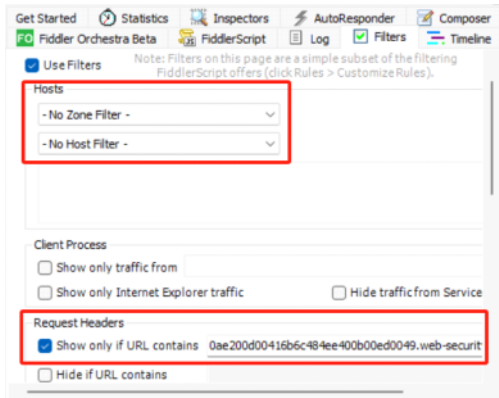
request: 1/1000

S2此时打开Fiddler进行抓包

进入该链接



#### S3 设置Fiddler的过滤litter模块



#### S4 修改密码

这里修改为了123456



#### S5 在提交时准备抓包

观察请求方法，提交是POST请求，所以锁定该包



#### S6 观察包的内容，找可修改的部分

①URL中存在token



②请求体中存在token

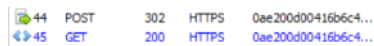


#### S7 两者为同一token，尝试删除token.观察是否正常运行

URL中删除后

temp-forget-password-token= HTTP/1.1

请求体中删除后



发现能正常访问，说明该包容易被修改

#### S8 修改账户为目标账户carlos

temp-forget-password-token=username=carlos&new-password-1=123456&new-password-2=123456

再次登录，用密码123456




# Login

Username

carlos

Password

123456



[Forgot password?](#)

Log in

Your username is: carlos

Your email is: carlos@carlos-montoya.net

登录成功 ✓