# Web应用安全

参考链接

<https://segmentfault.com/a/1190000004585199>

<https://blog.csdn.net/tiankai30/article/details/54892108>

<https://github.com/CHYbeta/Web-Security-Learning>

<https://github.com/FallibleInc/security-guide-for-developers/blob/master/security-checklist-zh.md>

## 入门知识

### 安全分类

#### 认证与授权

受保护资源页面不能未认证授权下进行访问

认证：访问网站时需要登录，用户名和密码的提交校验就是认证的过程。

授权：登录网站后，每个人所能访问的页面不尽相同，就是授权的作用。

如何测试：得到需要登录才能访问的页面（可通过页面属性、抓包等工具），直接拷贝在浏览器地址栏中访问，如果加了安全机制，直接访问会提示错误；否则，不安全。

#### Session、Cookie

session和cookie提高了用户体验，但不应该存放一些敏感数据，如身份证、手机号等。   
另外，不同的web站点的作用域应区分开，以避免出现不必要的麻烦。

如何测试：浏览器的开发者工具、抓包工具等都可以看到相应的信息。

#### DDOS 拒绝服务攻击

DdoS的攻击方式有很多种，最基本的DoS攻击就是利用合理的服务请求来占用过多的服务资源，从而使合法用户无法得到服务的响应。

Ddos攻击利用的就是合理的服务请求，所以但凡网站都存在这一风险。既然不可避免，就加强防御吧。

#### 文件上传漏洞

文件上传漏洞指攻击者利用程序缺陷绕过系统对文件的验证与处理策略将恶意程序上传到服务器并获得执行服务器端命令的能力。

常见方式有：

上传web脚本程序，使web容器解释执行上传的文件；

病毒、恶意程序，并诱导用户下载执行；

包含脚本的图片，某些浏览器低版本会执行。

#### XSS攻击

xss跨站攻击即跨站脚本攻击，百度百科中介绍如下：用户在浏览网站、使用即时通讯软件、甚至在阅读电子邮件时，通常会点击其中的链接。攻击者通过在链接中插入恶意代码，就能够盗取用户信息。攻击者通常会用十六进制（或其他编码方式）将链接编码，以免用户怀疑它的合法性。网站在接收到包含恶意代码的请求之后会产成一个包含恶意代码的页面，而这个页面看起来就像是那个网站应当生成的合法页面一样。

许多流行的留言本和论坛程序允许用户发表包含HTML和javascript的帖子。假设用户甲发表了一篇包含恶意脚本的帖子，那么用户乙在浏览这篇帖子时，恶意脚本就会执行，盗取用户乙的session信息。

#### 跨站请求伪造

跨站请求伪造（Cross-site request forgery）也被称为one-click attack或者session riding，通常缩写为CSRF或者XSRF， 是一种挟制用户在当前已登录的Web应用程序上执行非本意的操作的攻击方法。   
跟跨网站脚本（XSS）相比，XSS利用的是用户对指定网站的信任，CSRF 利用的是网站对用户网页浏览器的信任。

#### SQL 注入

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。

### 防止措施

#### 参数校验

对用户输入的数据进行全面安全检查或过滤，尤其注意检查是否包含SQL 或XSS特殊字符。验证用户输入的数据，包括值、类型、范围等等，用验证控件进行验证 RequiredFieldValidator RangeValidator RegularExpressionValidator，这些检查或过滤必须在服务器端客户端都进行。客户端服务端都要进行验证，客户端为了提高用户体验，服务端才可以有效防止威胁

#### 禁止Root账号

不要使用管理员权限的数据库连接

#### 敏感信息

机密信息加密，不能使用明文

session中的敏感信息需要加密，避免在永久cookies中存储敏感信息，重要的cookie标记为http only

#### 认证授权

访问页面时，对用户合法性进行判断

#### 漏洞扫描

项目发布前使用使用漏洞扫描软件，如IBM appScan，UnisWebScanner，对安全性进行评估

#### 避免SQL注入

不要使用动态拼接sql

## 开源实现

Js：<https://github.com/cure53/DOMPurify>