### 一、SSRF：服务端请求伪造

原因：

服务器相当于中间人，内网造成攻击。服务端提供了从其他服务器应用获取数据的功能，且没有对目标地址做过滤与限制。

危害：

1.内外网的端口和服务扫描

2.主机本地敏感数据的读取

3.内外网主机应用程序漏洞的利用

4.内外网Web站点漏洞的利用

......

### 二、SQL注入

原因：

未过滤输入，后台查询语句处理不当

危害：

信息泄露，身份验证绕过等

### 三、XSS：跨站脚本攻击

原因：

特殊字符没有被转义编码，对站内信任用户造成攻击

分类：

本地利用、反射型、存储型、DOM(特殊的反射型)

危害：

网络钓鱼，盗取cookie等信息窃取，脚本执行

### 四、CSRF：跨站请求伪造

原因：

依靠受信任用户，通过在授权用户访问的页面中包含链接或者脚本的方式工作，cookie具有持久性会话授权，没有多层级授权。

示例：

一个网站用户Bob可能正在浏览聊天论坛，而同时另一个用户Alice也在此论坛中，并且后者刚刚发布了一个具有Bob银行链接的图片消息。设想一下，Alice编写了一个在Bob的银行站点上进行取款的form提交的链接，并将此链接作为图片src。如果Bob的银行在cookie中保存他的授权信息，并且此cookie没有过期，那么当Bob的浏览器尝试装载图片时将提交这个取款form和他的cookie，这样在没经Bob同意的情况下便授权了这次事务。

危害：

劫持会话执行操作

### 五、暴力破解

原因：

验证机制缺陷

危害：

这个用脚想就知道

### 六、XXE：XML External Entity（XML外部实体注入攻击）

原因：

未过滤XML恶意内容，导致注入外部实体参数被恶意使用

危害：

任意文件读取

### 七、命令执行漏洞

原因：

当应用需要调用一些外部程序去处理内容的情况下，就会用到一些执行系统命令的函数。如PHP中的system，exec，shell\_exec等，当用户可以控制命令执行函数中的参数时，将可注入恶意系统命令到正常命令中，造成命令执行攻击。

危害：

自然不用多说了吧！

### 八、文件上传漏洞

原因：

文件上传路径变量过滤不严，以及相关目录的执行权限不当

危害：

webshell攻击

### 九、文件包含漏洞

原因：

服务器执行PHP文件时，可以通过文件包含函数加载另一个文件中的PHP代码，并且当PHP来执行，而文件包含函数加载的参数没有经过过滤或者严格的定义，可以被用户控制，包含其他恶意文件，导致了执行了非预期的代码。

类型：

本地无限制、session、有限制绕过（%00截断，路径长度阶段，点号截断）

远程无限制、有限制、问号、#

PHP伪协议（file://、http://、ftp://、php://、zlib://、data://、glob://、phar://、ssh2://、rar://、ogg://、expect://）

危害：

目录内容遍历

### 十、越权

原因：

身份验证逻辑出现问题，权限控制弱

类型：

水平越权

垂直越权，通常是提权操作

### 十一、解析漏洞

原因：

服务器解析机制问题，导致文件非法解析运行

危害：

上传文件非法获得shell

### 十二、编辑器漏洞

原因：

这个当然是各家编辑器的锅啦

危害：

各种各样很严重的危害，只有想不到没有做不到！

### 十三、逻辑漏洞

原因：

当然是程序设计逻辑出现问题，不严谨，或者太复杂，乱成一团

危害：

任意密码修改（没有旧密码验证）、越权访问、密码找回漏洞、交易支付金额篡改等

### 十四、URL重定向漏洞、跳转漏洞

原因：

服务器在告知浏览器跳转时，未对客户端传入的重定向地址进行合法性校验

危害：

网络钓鱼，信息窃取、会话劫持

### 十五、目录遍历漏洞

原因：

Web应用程序一般会有对服务器的文件读取查看的功能，用到提交的参数来指明文件名，以及其权限控制不当导致

危害：

信息泄露

### 十六、信息泄露

原因：

SQL注入、XSS、目录遍历、弱口令等均可导致信息泄露

危害：

网络裸奔