HTTP学习

- **Web及网络基础**
  - **HTTP的诞生**
    - 1.0版本诞生于1996年5月
    - 1.1版本于1997年1月公布，仍然是目前最主流的HTTP 协议版本
    - 2.0版本于2015年5月正式发表
  - **网络基础 TCP/IP**
    - TCP/IP 协议簇
      > 分层管理
    - 负责传输的IP协议
    - 确保可靠性的TCP协议
    - 负责域名解析的DNS服务
  - **URI和URL**
- **简单的 HTTP 协议**
  - 特点
    - 用于客户端和服务端之间的通信
    - 不保存状态，即无状态协议——后续引入Cookie 技术
    - 请求URI定位资源
  - 方法
    - Get
    - Post
    - Put
    - Head
    - Delete
    - Options
    - Trace
    - Connect
  - 持久连接节省通信量
    - 持久连接
    - 管线化
- **HTTP 报文**
  - **请求及响应报文的结构**
    - 请求行
    - 状态行
    - 首部字段

    - 其他（比如Cookie 等）
- **编码提升传输效率**
    - 报文主体和实体主体的差异
    - 压缩传输的内容编码
    - 分割发送的分块传输编码
- **功能**
    - 发送多种数据的多部分对象集合
    - 范围请求——网络中断可恢复
    - 内容协商返回最合适的内容

- **HTTP 状态码**
    - 结构
        - 以三位数字和原因短语组成
          > 例如 200 OK，数字第一位指定了响应类别
        - 分为5种响应
    - 1XX，信息性状态码，接收的请求正在处理
    - 2XX，成功状态码，请求正常处理完毕
        - 200 OK：客户端发送的请求被服务器正常处理
        - 204 No Content：服务器正常处理接收的请求，但返回响应报文没有任何实体的主体
        - 206 Partial Content：客户端采用范围请求的状态码
    - 3XX，重定向状态码，需要进行附加操作以完成请求
        - 301 Moved Permanently：永久性重定向。表示请求的资源已被分配了新的 URI
        - 302 Found：临时性重定向。
        - 303 See Other：表示请求的资源存在另一个 URI，应使用 GET 方法定向获取请求的资源。
        - 304 Not Modified：发生了为满足条件的情况
        - 307 Temporary Redirect：临时重定向。与 302 Found 有相同的含义，但 307 不会从 POST 变成 GET
    - 4XX，客户端错误状态码，服务器无法处理请求
        - 400 Bad Request：请求的报文存在语法错误
        - 401 Unauthorized：请求需要通过 HTTP 认证（BASIC 认证、DIGEST 认证）的认证信息
        - 403 Forbidden：对请求资源的访问被服务器拒绝了。
        - 404 Not Found：服务器上找不到请求的资源
    - 5XX，服务器错误状态码，服务器处理请求出错
        - 500 Internal Server Error：表示服务器端在执行请求时发生错误。也可能是 Web 应用存在的 bug 或者某些临时的故障
        - 503 Service Unavailable：表示服务器暂时处于超负载或正在进行停机维护，无法处理请求

- **Web 服务器**

- **虚拟机实现多个域名，也就可以搭建多个网站**
- **代理：有转发功能的应用程序**
  - 缓存代理：预先将资源的副本(缓存)保存在代理服务器上
    - 利用缓存可以减少对服务器的访问，节省通信流量和通信时间
    - 缓存的有效期限
    - 客户端的缓存主要是在浏览器内
  - 透明代理：对报文不做任何加工操作
- **网关：转发其他服务器通信数据的服务器**
  - 可以使通信线路上的服务器提供非HTTP协议服务
  - 可以在客户端和网关之间通信线路加密提高通信的安全性
- **隧道：在相隔很远的客户端和服务器之间进行中转，并保持双方通信连接的应用程序**
  - 目的是确保客户端和服务器进行安全的通信
  - 不会解析 HTTP 请求

HTTP 首部

报文首部

- 请求报文首部由方法、URI、HTTP 版本、HTTP 首部字段等部分构成
- 响应报文首部由HTTP 版本、状态码、HTTP 首部字段 3 部分组成

HTTP 首部字段

通用首部字段

- Cache-Control：操作缓存的工作机制
- Connection
  - 控制不再转发给代理的首部字段
  - 管理持久连接
- Date：表明 HTTP 报文的创建日期和时间
- Pragma：仅作为与 HTTP/1.0 的向后兼容而定义
- Trailer：事先说明在报文主体后记录了哪些首部字段
- Transfer-Encoding：规定了传输报文主体时采用的编码方式
- Upgrade：用于检测 HTTP 协议及其他协议是否可使用更高的版本进行通信
- Via：追踪客户端和服务器之间的请求和响应报文的传输路径
- Warning：告知用户一些与缓存相关的问题的警告

请求首部字段

- Accept：告知服务器，用户代理能够处理的媒体类型和对应的相对优先级
- Accept-Charset：通知服务器用户代理支持的字符集和字符集的相对优先顺序
- Accept-Encoding：用户代理支持的内容编码及其优先级
- Accept-Language：用户代理能够处理的自然语言集（指中文或英文等）以及优先级
- Authorization：告知服务器用户代理的认证信息（证书值）
- Expect：期望出现的某种特定行为。若服务器无法理解客户端的期望作出回应而发生错误时，会返回状态码 417 Expectation Failed
- From：告知服务器使用用户代理的用户的电子邮件

- Host：告知服务器，请求的资源所处的互联网主机名和端口号
- If-Match：属于附带条件之一，告知服务匹配资源所使用的实体标记（ETag）值
- If-Modified-Since：属于附带条件之一，若该字段值早于资源的更新时间，希望处理该请求
- If-None-Match：属于附带条件之一，和 If-Match 作用相反
- If-Range：属于附带条件之一，告知服务器若指定的该字段值和请求资源的 ETag 值或时间一致时，则作为范围请求处理，反之，返回全体资源
- If-Unmodified-Since：和 If-Modified-Since 作用相反。
- Max-Forwards：通过 TRACE 方法或 OPTIONS 方法，发送本字段值，指定可经过的服务器最大数量。
- Proxy-Authorization：接收到代理服务器发来的认证质询时，发送包含该字段的请求，以告知服务器认证所需要的信息。
- Range：告知服务器资源的指定范围
- Referer：告知服务器请求的原始资源的 URI
- TE：告知服务器能够处理响应的传输编码方式及相对优先级，还可以指定伴随 trailer 字段的分块传输编码方式
- User-Agent：创建请求的浏览器和用户代理名称等信息

- **响应首部字段**
  - Accept-Ranges：告知客户端服务器是否能够处理范围请求，以指定获取服务器端某个部分的资源。可处理范围请求时字段值为 bytes，否则是 none
  - Age：告知客户端，源服务器在多久前创建了响应，字段值单位是秒。
  - ETag：告知客户端，实体标识。服务器会为每份资源分配对应的 ETag 值，资源更新时，该值也需要更新
    - 强 ETag 值：不论实体发生多细微的变化都会改变其值
    - 弱 ETag 值：只用于提示资源是否相同，只有资源发生了根本改变，产生差异时才会改变该值，这时会在字段值最开始处附加 W/
  - Location：可以将响应接收方引导至某个与请求 URI 位置不同的资源。基本上，该字段会配合 3xx：Redirection的响应，提供重定向的 URI。
  - Proxy-Authenticate：将代理服务器要求的认证信息发送给客户端
  - Retry-After：告知客户端，多久之后再次发送请求。主要配合状态码 503 Service Unavailale 或 3xx Redirect 响应一起使用
  - Server：告知客户端，当前服务器上安装的 HTTP 服务器应用程序的信息。包括程序名称、版本号和安装时启用的可选项
  - Vary：可对缓存进行控制
  - WWW-Authenticate：用于 HTTP 访问认证。告知客户端，适用于访问请求 URI 所指定资源的认证方案（Basic 或 Digest）和带参数提示的质询
- **实体首部字段**

  补充了资源内容更新时间等与实体有关的信息
  - Allow：告知客户端，能够支持 Request-URI 指定资源的所有 HTTP 方法
  - Content-Encoding：告知客户端，服务器对实体的主体部分选用的内容编码方式。内容编码是在不丢失实体信息前提下进行的。主要采用4种
    - gzip
    - compress
    - deflate

        - identity
    - Content-Language：告知客户端，实体主体使用的自然语言
    - Content-Length：表明了实体主体部分的大小，单位是字节。
    - Content-Location：表示报文主体返回资源对应的 URI
    - Content-MD5：是一串由 MD5 算法生成的值。用于检查报文主体在传输过程是否保持完整和确认传输到达
    - Content-Range：针对范围请求，告知客户端，作为响应返回的实体的哪个部分符合范围请求，单位是字节
    - Content-Type：说明了实体主体内对象的媒体类型，和首部字段 Accept 一样，字段值用 type/subtype 形式赋值。
    - Expires：将资源失效的日期告知客户端
    - Last-Modified：指明资源最终修改的时间。
- **还根据是否缓存代理分为两种类型**
    - 端到端首部（End-to-end Header）
    - 逐跳首部（Hop-by-hop Header）--8个字段
        - Connection
        - Keep-Alive
        - Proxy-Authenticate
        - Proxy-Authorization
        - Trailer
        - TE
        - Transfer-Encoding
        - Upgrade
- **为 Cookie 服务的首部字段**
    - Set-Cookie：服务器开始管理客户端状态时，会事先告知各种信息
    - Cookie：告知服务器，客户端想获得 HTTP 状态管理支持
- **其他首部字段**

    HTTP 首部字段是可以拓展的，所以在 Web 服务器和浏览器应用上，会出现各种非标准的首部字段
    - X-Frame-Options：属于 HTTP 响应首部，用于控制在其他 Web 网站的 Frame 标签内的显示问题，防止点击劫持（clickjacking）攻击
    - X-XSS-Protection：属于 HTTP 响应首部，针对跨站脚本攻击（XSS)的一种对策，用于控制浏览器 XSS 防护机制的开关
    - DNT（Do Not Track）：属于 HTTP 请求首部，用于拒绝被精准广告追踪的一种方法
    - P3P：属于 HTTP 响应首部，通过利用 P3P(The Platform for Privacy Preferences，在线隐私偏好平台)技术，可以让 Web 网站上的个人隐私变成一种仅供程序可理解的形式，以达到保护用户隐私的目的。

- **HTTP的缺点**
    - 通信使用明文（不加密），内容可能会被窃听
    - 不验证通信方的身份，因此可能遭遇伪装
        - 有可能遭遇伪装的 Web 服务器
        - 有可能遭遇已伪装的客户端

- 无法确定正在通信的对方是否具备访问权限
- 无法判定请求是来自何方、出自谁手
- 即使是无意义的请求也会照单全收。无法阻止海量请求下的 DoS 攻击(Denial of Service，拒绝服务攻击)

- 无法证明报文的完整性，所以有可能已遭篡改
  - 请求或响应在传输途中，遭攻击者拦截并篡改内容的攻击被称为中间人攻击（Man-in-the-Middle attack, MITM）
- 解决方案
  - 加密处理防止被窃听
    - 通信加密--SSL(Secure Socket Layer，安全套接层)和 TLS(Transport Layer Security，安全传输层协议)，组合 SSL 的 HTTP 被称为 HTTPS(HTTP Secure，超文本传输安全协议)或 HTTP over SSL
    - 内容的加密--前提是要求客户端和服务器同时具备加密和解密机制
  - 查明对手的证书
  - 常用 MD5 和 SHA-1 等散列值校验的方法，以及用来确认文件的数字签名方法，来确认报文的完整性。

HTTPS

HTTP + 加密 + 认证 + 完整性保护 = HTTPS

- **HTTPS 不是应用层的一种新协议，它是 HTTP 通信接口部分采用 SSL 和 TLS 协议代替，是身披 SSL 协议外壳的 HTTP。**
- **SSL 是独立于 HTTP 的协议，是当今世界上应用最为广泛的网络安全技术。**
- ### 公开密钥加密技术--SSL 采用的加密处理方式

  近代加密算法是公开的，但密钥却是保密的

  - **共享密钥加密（Common Keycrypto systerm），也称为对称密钥加密**，是指加密和解密共用一个密钥的方式，在发送密钥过程可能遭遇窃听的风险
  - **公开密钥加密**--采用一对非对称的密钥，分别是私有密钥和公有密钥，发送方采用对方的公开密钥加密，而对方采用私有密钥进行解密，公开密钥可以随意发布，但私有密钥不能让任何人知道。
    - 处理速度相比共享密钥加密方法要慢
    - 存在无法证明公开密钥就是货真价实的公开密钥的问题
    - 可以使用由数字证书认证机构（CA，Certificate Authority）和其他相关机关颁发的公开密钥证书
  - **HTTPS 采用混合加密机制**--混合上述两种方法
  - **EV SSL 证书（Extended Validation SSL Certificate)**
    - 特点
      - 证明作为通信一方的服务器是否规范
      - 确认对方服务器背后运营的企业是否真实存在
    - 意图是防止用户被钓鱼攻击（Phishing），但很多用户并不了解 EV SSL 证书相关的知识，不会太留意它
  - **客户端证书**
    - 证书的获取需要用户自行安装，但客户端证书需要付费购买
    - 安全性极高的认证机构可颁发客户端证书，但仅用于特殊用途的业务，比如网上银行

        - 证书无法证明用户本人的真实有效性
    - 采用 OpenSSL 开源程序可以构建自己的认证机构，这种认证机构叫做自认证机构，颁发的证书叫做自签名证书。
    - 中级认证机构的证书可能会变成自认证证书
- **SSL 会导致 HTTPS 的处理速度变慢**
    - 通信慢，采用 SSL，会比采用 HTTP，网络负载可能变慢 2 到 100 倍，因为需要进行 SSL 通信，整体上处理通信量不可避免会增加。
    - SSL 必须进行加密处理，这会大量消耗 CPU和内存等资源，导致处理速度变慢。

认证

常用认证方法

- 密码：只有本人才会知道的字符串信息
- 动态令牌：仅限本人持有的设备内显示的一次性密码
- 数字证书：仅限本人（终端）持有的信息
- 生物认证：指纹和虹膜的等本人的生理信息
- IC 卡等：仅限本人持有的信息

HTTP 采用的认证方式

BASIC 认证（基本认证）

- 采用 Base64 编码方式，但并不是加密处理，会有被窃听的风险
- 想再进行一次 BASIC 认证时，一般的浏览器却无法实现认证注销操作
- 使用不够便捷灵活，且达不到多数 Web 网站期望的安全性登记，因此并不常用

DIGEST 认证（摘要认证）

- 采用质询/响应的方式，提供了防止密码被窃听的保护机制，但不存在防止用户伪装的保护机制
- 使用同样不够便捷灵活，安全性也不够高，适用范围也有所受限

SSL 客户端认证

- 借由 HTTPS 的客户端证书完成认证的方式
- 不仅仅依靠证书完成认证，还会和基于表单认证形成一种**双因素认证**

FormBase 认证（基于表单认证）

- 该认证方式不是在 HTTP 协议中定义的
- 目前的认证大多数是用 Web 应用程序各自实现的基于表单的认证方式
- 一般采用 Cookie 来管理 Session（会话）

基于 HTTP 的功能追加协议

HTTP 的瓶颈

- 一条连接上只可发送一个请求
- 请求只能从客户端开始。客户端不可以接收除响应以外的指令。
- 请求 / 响应首部未经压缩就发送。首部信息越多延迟越大
- 发送冗长的首部。每次互相发送相同的首部造成的浪费较多。
- 可任意选择数据压缩格式。非强制压缩发送。

- **Ajax 的解决方法**

  Ajax（Asynchronous JavaScript and XML，异步 JavaScript 与 XML 技术）

  - 一种有效利用 JavaScript 和 DOM 的操作，以达到局部 Web 页面替换加载的异步通信手段。
  - 核心技术是名为 XMLHttpRequest 的 API，通过 JavaScript 脚本语言的调用就能和服务器进行 HTTP 通信。
  - 利用 Ajax 实时地从服务器获取内容，可能会导致大量请求产生。
  - Ajax 仍未解决 HTTP 协议本身存在的问题
- **Comet 的解决方法**
  - Comet 是在服务器有更新内容后直接给请求返回响应。是一种通过延迟应答，模拟实现服务器端向客户端推送的功能
  - 内容上虽然可以做大实时更新，但为了保留响应，一次连接的持续时间也变长了。
  - 为了维持连接，会消耗更多的资源，并且也未解决 HTTP 协议本身的问题。
- **SPDY**
  - 没有完全改写 HTTP 协议，而是在 TCP/IP 的应用层和传输层之间通过新加会话层的形式运作，并且规定通信中使用 SSL
  - SPDY 给 HTTP 协议加入了额外的功能
    - **多路复用流。**通过单一的 TCP 连接，可以无限处理多个 HTTP 请求，并且所有请求在一条 TCP 连接上完成，提高了 TCP 的处理效率
    - **赋予处理优先级。**
    - **压缩响应首部。**
    - **推送功能。**支持服务器主动向客户端推送数据的功能
    - **服务器提示功能。**服务器可以主动提示客户端请求所需要的资源，避免不必要的请求。
  - SPDY 基本上只是将单个域名（IP 地址）的通信多路复用，但当一个 Web 网站上使用多个域名下的资源，改善效果会受到限制。
  - 是一种可有效消除 HTTP 瓶颈的技术，但很多 Web 存在的问题并非仅仅是由 HTTP 瓶颈所导致。
- **WebSocket--使用浏览器进行全双工通信**
  - 建立在 HTTP 协议上，连接的发起方仍是客户端，但一旦建立 WebSocket 通信连接，任意一方都可以直接向对方发送报文。
  - 主要特点
    - 推送功能
    - 减少通信量
  - 为了实现 WebSocket 通信，在 HTTP 连接建立后，需要完成一次“握手”(Handshaking)的步骤，其中响应会返回状态码 101 Switching Protocols
- **WebDAV--Web 服务器上管理文件的分布式文件系统**
  - 可以对 Web 服务器上的内容直接进行文件复制、编辑，创建文件和删除文件，以及文件创建者管理、文件编辑过程中禁止其他用户内容覆盖的加锁功能、对文件内容修改的版本控制功能

构建 Web 内容的技术

- **HTML（HyperText Markup Language，超文本标记语言）**

为了发送 Web 上的超文本而开发的标记语言

- HTML 的版本--HTML 4.0 和 HTML5
- 设计应用 CSS（Cascading Style Sheets，层叠样式表）
- **动态 HTML**
  - **DOM（Document Object Model，文档对象模型）**可以将 HTML 内的元素当做对象操作，比如取出元素内的字符串、改变其 CSS 的属性，使页面的设计发生改变
  - **JavaScript--客户端脚本语言**，可以实现对 HTML 的 Web 页面的动态改造

- **Web 应用**
  - CGI（Common Gateway Interface，通用网关接口）指 Web 服务器在接收到客户端发送过来的请求后转发给程序的一组机制。
  - Servlet 是一种能在服务器上创建动态内容的程序。它是用 Java 语言实现的一个接口，属于面向企业级 Java（JavaEE，Java Enterprise Edition）的一部分。其运行的环境叫做 Web 容器或 Servlet 容器
- **数据发布的格式及语言**
  - **XML（eXtensible Markup Language，可拓展标记语言）**是一种可按应用目标进行拓展的通用标记语言。
    - 采用标签构成树形结构，并且可自定义拓展标签
    - 读取 XML 文档的数据比起 HTML 更加简单
    - 更容易复用数据
  - **发布更新信息的 RSS / Atom**，它们都是发布新闻或博客日志等更新信息文档的格式的总称，都用到了 XML。
  - **JSON（JavaScript Object Notion）**是一种以 JavaScript 的对象表示法为基础的轻量级数据标记语言。
    - 可以处理 false / true / null / 数组/对象/数字/字符串 7种数据类型
    - 让数据更轻更纯粹
    - 多种编程语言都提供实现 JSON 的类库

- **Web 攻击技术**
  - 针对 Web 的攻击技术
    - HTTP 不具备必要的安全功能
    - 在客户端即可篡改请求
    - **攻击模式**
      - 主动攻击：指攻击者通过直接访问 web 应用，把攻击代码传入的攻击模式。
        - 代表性的攻击是 SQL 注入攻击和 OS 命令注入攻击
      - 被动攻击：指利用圈套策略执行攻击代码的攻击模式。在被动攻击过程中，攻击者不直接对目标 Web 应用访问发起攻击
        - 利用用户身份攻击企业内部网络
  - 安全漏洞
    - **因输出值转义不完全引发的漏洞**

      输出值转义：从数据库或文件系统、HTML、邮件等输出 Web 应用处理的数据之际，对输出做转义处理是一项至关重要的安全策略。

- **跨站脚本攻击（Cross-Site Scripting XSS）**: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 HTML 标签或 JavaScript 进行的一种攻击。属于**被动攻击模式。**
  - 利用虚假输入表单获取用户个人信息
  - 利用脚本窃取用户的 Cookie 值，被害者在不知情的情况下，帮助攻击者发送恶意请求
  - 显示伪造的文章或图片
- **SQL 注入攻击（SQL Injection）**： 指针对 Web 应用使用的数据库，通过运行非法的 SQL 而产生的攻击。

  SQL 注入是攻击者将 SQL 语句改变成开发者意想不到的形式以达到破坏结构的攻击
  - 非法查看或篡改数据库内的数据
  - 规避认证
  - 执行和数据库服务器业务关联的程序等
- **OS 命令注入攻击（OS Command Injection）**： 指通过 Web 应用，执行非法的操作系统命令达到攻击的目的。只要在能调用 Shell 函数的地方就有存在被攻击的风险。

  OS 注入攻击可执行 OS 上安装着的各种程序。
- **HTTP 首部注入攻击（HTTP Header Injection）**： 指攻击者通过在响应首部字段内插入换行，添加任意响应首部或主体的一种攻击。属于**被动攻击模式。**
  - 设置任何 Cookie 信息
  - 重定向至任意 URL
  - 显示任意的主体（**HTTP 响应截断攻击--向首部主体内添加内容的攻击**）
- **邮件首部注入攻击（Mail Header Injection）**： 指 Web 应用中的邮件发送功能，攻击者通过邮件首部 To 或 Subject 内任意添加非法内容发起的攻击。
- **目录遍历（Directory Traversal）攻击**：指对本无意公开的文件目录，通过非法截断其目录路径后，达成访问目的的一种攻击。有时也被成为**路径遍历（Path Traversal）攻击。**
  - 对这种攻击，固然存在输出值转义的问题，但更应该关闭指定对任意文件名的访问权限
- **远程文件包含漏洞（Remote File Inclusion）**是指当前部分脚本内容需要从其他文件读入时，攻击者利用指定外部服务器的 URL 充当依赖文件，让脚本读取之后，就可运行任意脚本的一种攻击。
  - 主要是 PHP 存在的安全漏洞，对 PHP 的 include 或 require 来说，这是一种可通过设定，指定外部服务器的 URL 作为文件名的功能。但是该功能太危险，PHP 5.2.0 之后默认此功能无效
  - 固然存在输出值转义问题，但更应该控制对任意文件名的指定。

因设置或设计上的缺陷引发的安全漏洞

- **强制浏览**：指从安置在 Web 服务器的公开目录下的文件中，浏览那些原本非自愿公开的文件。
  - 泄露顾客的个人信息等重要情报
  - 泄露原本需要具有访问权限的用户才可以查阅的信息内容
  - 泄露未外连到外界的文件

- **不正确的错误信息处理**：指 Web 应用的错误信息包含对攻击者有用的信息。
  - Web 应用抛出的错误信息
  - 数据库等系统抛出的错误信息
    - PHP 或 ASP 等脚本错误
    - 数据库或中间件的错误
    - Web 服务器的错误
- **开放重定向（Open Redirect）**：对指定的任意 URL 作重定向跳转的功能。与此相关的安全漏洞是指，假如重定向 URL 到某个具有恶意的 Web 网站，那么用户就会被诱导至那个 Web 网站。

因会话管理疏忽引发的安全漏洞

会话管理是用来管理用户状态的必备功能

- **会话劫持（Session Hijack）**指通过某种手段拿到用户的会话 ID，并非法使用此会话 ID 伪装成用户，达到攻击的目的。
  - 通过非正规的生成方法推测会话 ID
  - 通过窃听或 XSS 攻击盗取会话 ID
  - 通过会话固定攻击（Session Fixation）强行获取会话 ID
- **会话固定攻击（Session Fixation）**会强制用户使用攻击者指定的会话 ID，属于被动攻击。
- **跨站点请求伪造（Cross-Site Request Forgeries，CSRF）**指攻击者通过设置好的陷阱，强制对已完成认证的用户进行非预期的个人信息或设定信息等某些状态更新，属于被动攻击
  - 利用已通过认证的用户权限更新设定信息等
  - 利用已通过认证的用户权限购买商品
  - 利用已通过认证的用户权限在留言板上发表言论

其他安全漏洞

- **密码破解攻击（Password Cracking）即算出密码，突破认证，攻击不局限于 Web 应用，还包括其他的系统（如 FTP 或 SSH 等）**
  - 通过网络的密码试错
    - **穷举法（Brute-force Attack，又称为暴力破解法）**指对所有密钥集合构成的密钥空间（Keyspace）进行穷举
    - **字典攻击**指利用事先收集好的候选密码（经过各种组合方式后存入字典），枚举字典中的密码，尝试通过认证的一种攻击手法。
  - 对已加密密码的破解（指攻击者入侵系统，已获得加密或散列处理的密码数据的情况）
    - **通过穷举法或者字典攻击进行类推**
    - **彩虹表（Rainbow Table）**是由明文密码及与之对应的散列值构成的一种数据库表，是一种通过事先制作庞大的彩虹表
    - **拿到密钥**
    - **加密算法的漏洞**
- **点击劫持（Clickjacking）**指利用透明的按钮或链接做成陷阱，覆盖在 Web 页面只上，然后诱使用户在不知情的情况下，点击那个链接访问内容的一种攻击手段，又被称为**界面伪装（UI Redressing）**

- **DoS 攻击（Denial of Service attack）**是一种让运行中的服务呈停止状态的攻击，有时也叫做服务停止攻击或拒绝服务攻击，攻击对象不仅限于 Web 网站，还包括网络设备和服务器等。
  - 攻击方式
    - 集中利用访问请求造成资源过载，资源用尽的同时，实际上服务也就呈停止状态。
    - 通过攻击安全漏洞使服务停止
  - **DDos 攻击（Distributed Denial of Service attack）**是指多台计算机发起的 DoS 攻击。它通常利用那些感染病毒的计算机作为攻击跳板。
- **后门程序（Backdoor）是指开发设置的隐藏入口，可不按正常步骤使用受限功能。**
  - 开发阶段作为 Debug 调用的后门程序
  - 开发者为了自身利益植入的后门程序
  - 攻击者通过某种方法设置的后门程序