1. **HTTP报文格式**

**应用层：HTTP协议**

客户端：Request

请求方式

相对URI

协议版本

请求首部字段  
通用首部字段

实体首部字段

请求报文主体

CR+LF

请求行

报文首部

空行

报文主体

服务端：Response

协议版本

状态码

[提示信息]

响应首部字段  
通用首部字段

实体首部字段

响应报文主体

CR+LF

状态行

报文首部

空行

报文主体

链路安全：HTTPs

用户身份认证

追加协议

常见web攻击

1. **HTTP协议版本**

* HTTP/0.9 ～ 1991
* HTTP/1.0 ～ 1996
* HTTP/1.1 ～ 1999
* HTTP/2.0 ～ 2015
* HTTP1.X：采用keep-alive持久连接；管线化
* HTTP2.0：并发请求
* 持久连接：只要任意一端没有明确提出断开连接，则保持TCP连接状态
* 管线化：可以同时并行发送多个请求，而不需要等待上一个请求接收到响应才能发送请求；本质是并行处理

1. **URI & URL**

**（1）含义**

* URI：统一资源标识符，用字符串标识某一互联网资源
* URL：统一资源定位符，表示资源在互联网上所处位置或地点

**（2）关系**

URL是URI的子集

**（3）分类**

* 绝对URL
* 相对URL
* 绝对URI
* 相对URI：从浏览器中基本URI处制定的URI（请求报文中用的就是相对URI）

**（4）请求中的相对URI**

* 普通URI：资源定位
* \*：不是访问特定资源，而是对服务器本身发起请求

1. **请求方式**
2. **GET：获取资源**

* 携带信息：在URL中明文携带、显示；大小受限(几K)；不安全
* 幂等请求

1. **POST：传输实体主体**

* 携带信息：在报文主体中携带；大小一般不受限；相对安全
* 非幂等请求

1. **PUT：传输文件**

* 功能：在请求报文主体中包含文件内容，然后保存到请求URI指定的位置

1. **HEAD：获得报文首部**

* 功能：与GET方法相同，但不返回报文主体部分
* 作用：用于确认URI的有效性以及资源更新的日期时间等

1. **DELETE：删除文件**

* 功能：与PUT方法相反，按请求URI删除指定位置处的资源

1. **OPTIONS：询问服务器支持的方法**

* 功能：向服务器查询针对请求URI指定的资源支持的方法(Allow:GET,POST)

1. **TRACE：追踪路径**

* 功能：与Max-Forwards连用，每经过一个服务器端就将该数减1，当Max-Forwards值为0时，停止继续传输，最后接收到请求的服务器返回状态码为200 OK的响应
* 缺陷：容易引起跨站追踪

1. **CONNECT：要求用隧道协议连接代理**

* 功能：要求在与代理服务器通信时创建隧道，实现用隧道协议进行TCP通信

|  |  |  |
| --- | --- | --- |
| 方法 | 说明 | 支持的HTTP协议版本（1.0，1.1，2.0） |
| GET | 获取资源 | 2.0、1.1、1.0、0.9 |
| POST | 传输实体主体 | 2.0、1.1、1.0 |
| PUT | 传输文件 | 2.0、1.1、 |
| HEAD | 获得报文首部 | 2.0、1.1、1.0 |
| DELETE | 删除文件 | 2.0、1.1、 |
| OPTIONS | 询问服务器支持的方法 | 2.0、1.1 |
| TRACE | 追踪路径 | 2.0、1.1 |
| CONNECT | 要求用隧道协议连接代理 | 2.0、1.1 |
| LINK | 建立和资源之间的联系 | 1.0 |
| UNLINK | 断开连接关系 | 1.0 |

1. **状态码**
2. **状态码类别**

|  |  |  |
| --- | --- | --- |
|  | 类别 | 原因短语 |
| 1XX | Informational（信息性状态码） | 接收的请求正在处理 |
| 2XX | Success（成功状态码） | 请求正常处理完毕 |
| 3XX | Redirection（重定向状态码） | 需进行附加操作完成请求 |
| 4XX | Client Error（客户端错误状态码） | 服务器无法处理请求 |
| 5XX | Server Error（服务器错误状态码） | 服务器处理请求出错 |

1. **2XX 成功**

* 200 OK：表示从客户端发来的请求在服务器端被正常处理了（包括HEAD）
* 204 No Content：表示服务器接收请求已成功处理，但返回的响应报文中不含也不允许含实体的主体部分
* 206 Partial Content：表示成功的范围请求

1. **3XX重定向**

* 301 Moved Permanently：永久性重定向。表示请求的资源已被分配了新的URI，以后应使用资源现在所指的URI
* 302 Found：临时性重定向。表示请求的资源已被分配了新的URI，希望用户本次能使用心得URI访问
* 303 See Other：类似302，但303状态码明确表示客户端应当采用GET方法获取资源
* 304 Not Modified：请求存在附带条件，服务器资源已找到，但未符合要求
* 307 Temporary Redirect：类似302，禁止POST变成GET请求

1. **4XX客户端错误**

* 400 Bad Request：表示请求报文中存在语法错误
* 401 Unauthorized：表示需要进行HTTP认证，或认证错误
* 403 Forbidden：表示请求资源的访问被服务器拒绝
* 404 Not Found：表示服务器上无法找到请求资源

1. **5XX服务器错误**

* 500 Internal Server Error：表明服务器端在执行请求时发生了错误
* 503 Service Unavailable：表明服务器暂时处于超负载或正在进行停机维护，现在无法处理请求

1. **HTTP首部**
2. **首部字段结构：**键值对
3. **首部字段重复情况：**按照浏览器内部处理逻辑处理
4. **首部字段分类**

* 端到端首部（end-to-end）：整个请求过程中一直存在
* 逐跳首部（hop-by-hop）：只对单次转发有效

Connection，Keep-Alive，Proxy-Authenticate，Proxy-Authorization，Tralier，TE，Transfer-Encoding，Upgrade

1. **首部字段一览表**
2. **通用首部字段**

|  |  |
| --- | --- |
| **首部字段名** | **说明** |
| Cache-Control | 控制缓存的行为 |
| Connection | 逐跳首部、连接管理 |
| Date | 创建报文的日期时间 |
| Pragma | 报文指令 |
| Trailer | 报文末端的首部一览 |
| Transfer-Encoding | 指定报文主体的传输编码方式 |
| Upgrade | 升级为其他协议 |
| Via | 代理服务器的相关信息 |
| Warning | 错误通知 |

1. **请求首部字段**

|  |  |
| --- | --- |
| **首部字段名** | **说明** |
| Accept | 用户代理可处理的媒体类型 |
| Accept-Charset | 优先的字符集 |
| Accept-Encoding | 优先的内容编码 |
| Accept-Language | 优先的语言（自然语言） |
| Authorization | Web认证信息 |
| Expect | 期待服务器的特定行为 |
| From | 用户的电子邮箱地址 |
| Host | 请求资源所在服务器 |
| If-Match | 比较实体标记（ETag） |
| If-Modified-Since | 比较资源的更新时间 |
| If-None-Match | 比较实体标记，与If-Match相反 |
| If-Range | 资源未更新时发送实体Byte的请求 |
| If-Unmodified-Since | 比较资源的更新时间，与If-Modified-Since相反 |
| Max-Forwards | 最大传输逐条数 |
| Proxy-Authorization | 代理服务器要求客户端的认证信息 |
| Range | 实体的字节范围请求 |
| Referer | 对请求中URI的原始获取 |
| TE | 传输编码的优先级 |
| User-Agent | HTTP客户端程序的信息 |

1. **响应首部字段**

|  |  |
| --- | --- |
| **首部字段名** | **说明** |
| Accept-Ranges | 是否接收字节范围请求 |
| Age | 推算资源创建经过时间 |
| ETag | 资源的匹配信息 |
| Location | 令客户端重定向只指定URI |
| Proxy-Authenticate | 代理服务器对客户端的认证信息 |
| Retry-After | 对再次发起请求的时机要求 |
| Server | HTTP服务器的安装信息 |
| Vary | 代理服务器缓存的管理信息 |
| WWW-Authenticate | 服务器对客户端的认证信息 |

1. **实体首部字段**

|  |  |
| --- | --- |
| **首部字段名** | **说明** |
| Allow | 资源可支持的HTTP方法 |
| Content-Encoding | 实体主体使用的编码方式 |
| Content-Language | 实体主体的自然语言 |
| Content-Length | 实体主体的大小（单位：字节） |
| Content-Location | 替代对应资源的URI |
| Content-MD5 | 实体主体的报文摘要 |
| Content-Range | 实体主体的位置范围 |
| Content-Type | 实体主体的媒体类型 |
| Expires | 实体主体过期的日期时间 |
| Last-Modified | 资源的最后修改日期 |

1. **通用首部字段**
2. **Cache-Control**

* **请求**：Cache-Control:缓存请求指令

no-cache：不接受缓存的响应

no-store：服务器不能缓存

max-age：缓存资源的缓存时间比指定时间数值更小，那么客户端会接受缓存

max-stale：缓存资源即使过期，但在max-stale指定时间内，客户端便会接受

min-fresh：接受指定时间内/后未过期的缓存

no-transform：缓存不能改变实体主体的媒体类型

only-if-cached：只接受缓存响应

cache-extension：扩展

* **响应**：Cache-Control:缓存响应指令

public：表明其他用户可用缓存

private：表明响应对特定用户使用缓存，其他用户则不会用缓存

no-cache：缓存服务器不能对资源进行缓存

no-store：客户端不能缓存

no-transform：缓存不能改变实体主体的媒体类型

must-revalidate：代理会向源服务器再次验证即将返回的响应缓存目前是否仍然有效

proxy-revalidate：缓存服务器在接收到客户端带有该指令的请求返回响应之前，必须再次验证缓存的有效性

max-age：允许缓存服务器缓存资源的最长时间

s-maxage：同max-age，只适用于提供多位用户使用的公共缓存服务器

cache-extension：扩展

**注：优先级**

HTTP1.1：s-maxage>max-age>Expires

HTTP1.0：s-maxage>Expires>max-age

1. **Connection**

* 控制不再转发给代理的首部字段。Connection: 不再转发的首部字段名
* 管理持久连接

Connection: Keep-Alive

Connection: Close

1. **Data**

* 表明创建HTTP报文的日期和时间

Data: Tue, 03 Jul 2012 04:40:59 GMT

1. **Pragma**

* 要求所有的中间服务器不返回缓存的资源

Pragam: no-cache

1. **Trailer**

* 事先说明在报文主体后记录了哪些首部字段

Trailer: Expires

1. **Transfer-Encoding**

* 规定传输报文主体时采用的编码方式

Transfer-Encoding: chunked

1. **Upgrade**

* 协商是否可以使用更高版本进行通信

Upgrade: TLS/1.0

Connection: Upgrade

1. **Via**

* 追踪客户端与服务器之间的请求和响应报文的传输路径，通常和Trace方法一起使用

Via: 1.0 gw.hacker.jp, 1.1 a1.example.com

1. **Warning**

* 告知用户一些与缓存相关的问题的警告

Warning: [警告码][警告的主机:端口号]“[警告内容]([日期时间])”

1. **请求首部字段**
2. **Accept**

* 通知服务器，用户代理能够处理的媒体类型以及媒体类型的优先级；q越大，优先级越高；不指定优先级的话默认为1;媒体类型采用type/subtype这种形式

Accept: text/html, application/json;q=0.9

1. **Accept-Charset**

* 通知服务器用户代理支持的字符集以及字符集的相对优先顺序

Accept-Charset:iso-8859-5, Unicode-1-1;q=0.9

1. **Accept-Encoding**

* 通知服务器用户代理支持的内容编码以及对内容编码的优先级顺序；也可以使用\*作为通配符，指定任意的编码格式

Accept-Encoding: gzip, deflate

1. **Accept-Language**

* 通知服务器用户代理能够处理的自然语言集，以及自然语言集的相对优先级

Accept-Language:zh-cn, en-us;q=0.3

1. **Authorization**

* 告知服务器，用户代理的认证信息

Authorization:Basic dweshdjslskaja==

1. **Expect**

* 告知服务器，期望出现某种特定行为

Exprct: 100-continue (期望状态码 100 Continue)

1. **From**

* 告知服务器，使用用户代理的用户的电子邮件地址

From: info@hacker.jp

1. **Host**

* 告知服务器，请求资源所处互联网主机名和端口号

当同一个服务器上部署了多个服务的时候，采用DNS服务解析域名后，得到的IP是相同的；因此，在发送HTTP请求时，必须在Host首部内完整指定主机名或域名的URI。

Host:www.waf.com

1. **If-Match**

* 服务器会比对If-Match的字段值和资源的ETag的值，仅当两者一致时，才会执行请求；反之，返回412 Precondition Failed

If-Match: “123456”

1. **If-Modified-Since**

* 告知服务器，如果If-Modified-Since字段值早于资源更新时间，则处理该请求；反之，返回304 Not Modified

If-Modified-Since: Thu, 15 Apr 2004 00:00:00 GMT

1. **If-None-Match**

* 与If-Match作用相反；服务器会比对If-Match的字段值和资源的ETag的值，仅当两者**不**一致时，才会执行请求

If-None-Match: “123456”

1. **If-Range**

* 告知服务器，如果If-Range字段值(ETag值或时间)和请求资源的ETag值或时间相一致时，则**作为范围请求处理**；反之，返回全部资源

If-Range: “123456”

Range: bytes=5001-10000

1. **If-Unmodified-Since**

* 与If-Modified-Since的作用相反；告知服务器，指定的请求只有在字段值指定的时间之后未发生更新，则处理请求；反之，返回412 Precondition Failed

If-Unmodified-Since: Thu, 15 Apr 2004 00:00:00 GMT

1. **Max-Forwards**

* 通过TRACE或OPTIONS方法，发送首部包含Max-Forwards字段的请求时，该字段以十进制形式指定可经过的服务器最大数目

Max-Forwards: 10

1. **Proxy-Authorization**

* 接收从代理服务器发来的认证质询时，客户端通过Proxy-Authorization

告知服务器认证所需信息

Proxy-Authorization:Basic dweshdjslskaja==

1. **Range**

* 范围请求；告知服务器资源的指定范围

Range: bytes=5001-10000

1. **Referer**

* 告知服务器请求的原始资源的URI

Referer: http://www.hackr.jp/index.htm

1. **TE**

* 告知服务器客户端能够处理响应的**传输编码方式**及相对优先级

TE: trailers

1. **User-Agent**

* 将创建请求的浏览器和用户代理名称等信息传送给服务器

User-Agent: Mozilla/5.0

1. **响应首部字段**
2. **Accept-Ranges**

* 服务器告知客户端是否能处理范围请求

Accept-Ranges: none （不可以）

Accept-Ranges: bytes （可以）

1. **Age**

* 告知客户端，源服务器在多久前创建了响应

Age: 600 （单位：s）

1. **ETag**

* 一种可将资源以字符串形式唯一标识的方式；服务器会为每份资源分配对应的ETag值；当资源更新时，ETag值也需要更新
* 强ETag值：不论实体发生多么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，才会改变ETag值

ETag: “123456”

1. **Location**

* 提供重定向的URI

Location: http://www.uasg.jp/index.html

1. **Proxy-Authenticate**

* 把由代理服务器所要求的认证信息发送给客户端

Proxy-Authenticate: Basic realm=”Usagidesign Auth”

1. **Retry-After**

* 告知客户端应该在多久之后再次发送请求

Retry-After: 120 （120s之后）

Retry-After: 具体时间 （具体时间之后）

1. **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

Server: Apache/2.2.17 (Unix)

1. **Vary**

* 当代理服务器接收到带有Vary首部字段指定获取资源的请求时，如果使用的Accept-Language字段的值相同，那么就直接从缓存返回响应。反之，则需要先从源服务器端获取资源后才能作为响应返回

Vary: Accept-Language

1. **WWW-Authenticate**

* 用于客户端到源服务器的HTTP访问认证

WWW-Authenticate: Basic realm=”Usagidesign Auth”

1. **实体首部字段**
2. **Allow**

* 用于通知客户端能够支持Request-URI指定资源的所有HTTP方法

Allow: GET, HEAD

1. **Content-Encoding**

* 用于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

Content-Encoding: gzip

1. **Content-Language**

* 用于告知客户端，实体主体使用的自然语言

Content-Language: zh-CN

1. **Content-Length**

* 表示实体主体部分的大小，单位字节；计算方式复杂

Content-Length: 15000

1. **Content-Location**

* 给出与报文主体部分相对应的URI

Content-Location: http://www.hackr.jp/index.html

1. **Content-MD5**

* 一串由MD5算法生成的值，其目的在于检查报文主体在传输过程中是否保持完整，以及确认传输到达；并不安全

Content-MD5：xxxxxxxxxxxxxxxxxxxxxxx

1. **Content-Range**

* 针对范围请求，返回响应时使用的首部字段，能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

Content-Range: bytes 5001-10000/10000

1. **Content-Type**

* 表明实体主体内对象的媒体类型

Content-Type: text/html; charset=UTF-8

1. **Expires**

* 将资源失效的日期告诉客户端

Expires: Wed, 04 Jul 2012 08:26:05 GMT

1. **Last-Modified**

* 指明资源最终的修改时间

Last-Modified: Wed, 04 Jul 2012 08:26:05 GMT

1. **其他**
2. **Cookie**

* 响应：Set-Cookie
* 请求：Cookie

Set-Cookie: status=enable; expires=xxx; path=/; domain=example.com; secure; HttpOnly

Cookie: status=enable

* expires:cookie的过期时间
* path:限制指定Cookie的发送范围的文件目录
* domain:domian值指定的域名与URI结尾一致就可以使用该Cookie
* secure: 仅在HTTPS安全通信时才会发送Cookie
* HttpOnly: 加以限制，使Cookie不能被javascript脚本访问（document.cookie便无法读取该内容）

1. **X-Frame-Options**

* 用于控制网站内容在其他web网站的Frame标签内的显示问题，主要是为了防止点击劫持

X-Frame-Options: DENY （拒绝）

X-Frame-Options: SAMEORIGN （仅允许同源域名下的页面）

1. **X-XSS-Protection**

* 用于控制浏览器XSS防护机制的开关，主要是为了防止XSS攻击

X-XSS-Protection: 0 （将XSS过滤设置成无效状态）

X-XSS-Protection: 1 （将XSS过滤设置成有效状态）

1. **DNT**

* Do Not Attack，意为拒绝个人信息被收集，表示拒绝被精准广告追踪的一种方法

DNT: 0 （同意被追踪）

DNT: 0 （拒绝被追踪）

1. **P3P**

* 在线隐私偏好平台，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

1. **报文主体**
2. **报文(message) VS 实体(entity)**

* 报文(message)：HTTP通信的基本单位，由8位组字节流组成，通过HTTP通信传输
* 实体(entity)：作为请求或响应的有效载荷数据被传输，其内容由实体首部和实体主体组成
* HTTP报文主体：传输请求或响应的实体主体

1. **提升传输速率的方法**
2. **实体编码**

* gzip（GNU zip）
* compress（UNIX系统的标准压缩）
* deflate（zlib）
* identity（不进行编码）

1. **分块传输编码(Chunked Transfer Coding)**

* 将实体主体分成多个部分（块）。每一块都会用十六进制来标记块的大小，而实体主体的最后一块会使用“0(CR+LF)”来标记

1. **报文主体的多部分对象集合：报文主体内可含有多种类型数据**

* **请求**：Content-Type：multipart/form-data; boundary=AaB03x
* **响应：**206 Partial Content

Content-Type：multipart/byteranges; boundary=THIS\_STRING\_SEPARATES

说明：使用boundary字符串来划分多部分对象集合知名的各类实体

1. **范围请求**

* **请求：Range: bytes = 5001-10000 （50001～10000字节）**

**Range: bytes = 5001- （从5001字节之后全部的内容）**

**Range: bytes = 0-3000，5000-7000 （多重范围）**

* **范围响应：**206 Partial Content

Content-Type：multipart/byteranges;

* **无法进行范围响应：200 OK**

**完整实体内容**

1. **内容协商**

**请求：**Accept，Accept-Charset，Accept-Encoding，Accept-Language

**响应：**Content-Language

1. **HTTPs**
2. **HTTP的缺点**

* 通信使用明文，内容可能会被窃听
* 不验证通信方身份，有可能遭遇伪装
* 无法证明报文的完整性（准确性），可能已遭篡改

1. **两种加密方式**
2. **对称密钥加密/共享密钥加密**

* 含义：加密和解密用同一个密钥的方式
* 优点：计算量小，速度快，适合加密大数据
* 缺点：密钥易泄漏；一个用户对应一个密钥，服务器管理麻烦
* 常用：AES，DES，3DES

1. **非对称加密/公开加密**

* 含义：加密和解密使用非对称密钥。私有密钥+公开密钥。
* 优点：相对安全
* 缺点：加解密速度慢，计算量大
* 常用：RSA

1. **HTTPs机制**

* **本质**

HTTPS采用共享密钥加密和公开密钥加密两者并用的混合加密机制。HTTP+SSL(表示层)

* **基本流程**

在交换密钥环节使用公开密钥加密方式，之后的建立通信交换报文阶段则使用共享密钥加密方式

* **具体步骤**

S1.【服务器】服务器运营人员向数字证书认证机构提出公开密钥申请。数字证书认证机构分配签名密钥，并将公开密钥放入公钥证书后绑定在一起。

S2.【客户端】认证机构公开密钥植入客户端。

S3.【服务器】将公开密钥证书发送给客户端。

S4.【客户端】使用认证机构的公开密钥认证数字证书，取出服务器公钥。

S5.【客户端】用服务器公钥对生成随机数(pre-master-secret)加密，发送给服务器。

S6.【服务器】用服务器私钥破解出随机数，作为对称密钥用来进行数据传输。

1. **HTTPs缺陷**

* 通信慢：加解密耗时
* 处理速度慢：耗费CPU
* 负载增强

1. **确认访问用户身份的认证**
2. **认证的本质：**核对信息

密码，动态令牌，数字证书，生物认证，IC卡等

1. **HTTP使用的认证方式**
2. **BASIC认证（基本认证）**

* 通过header字段(WWW-Authenticate等)携带认证信息

1. **DIGEST认证（摘要认证）**

* 通过header字段(WWW-Authenticate等)携带**加密**后的认证信息

1. **SSL客户端认证**

* 借由HTTPS的客户端证书完成认证

1. **FormBase认证（基于表单认证）**

* 借助session，cookie，salt等

1. **基于HTTP的功能追加协议**
2. **HTTP的瓶颈**

* 一条连接智能发送一个请求
* 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。

1. **Ajax(Asynchronous JavaScript and XML, 异步js和xml)**

* 局部更新。请求过来了，进行局部更新

1. **Comet**

* 延迟应答。请求过来了，服务器先挂着，当服务器有更新时，再返回该响应

1. **SPDY**

* 会话层加入。特点：多路复用，赋予请求优先级，压缩HTTP首部，推送功能，服务器提示功能

1. **WebSocket**

* 参考笔记 websocket

1. **HTTP2.0**

* 新的二进制格式
* 多路复用：连接共享，一个request对应一个id，一个连接上可以有多个resquest（真正的并行）
* Header压缩
* 服务器推送：把客户端所需资源伴随index.html一起响应到客户端

1. **常见web攻击技术**

**！！！参考《白帽子讲web安全》**

1. **攻击分类**

* **以服务器为目标的主动攻击**

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；代表是SQL注入攻击和OS命令注入攻击

* **以服务器为目标的被动攻击**

指利用圈套策略执行攻击代码的攻击模式，攻击者不直接对目标web应用发起攻击；代表是跨站脚本攻击和跨站点请求伪造

1. **跨站脚本攻击(Cross-Site Scripting, XSS)**

* 指在浏览器内运行非法的HTML标签或JavaScript进行的一种攻击

1. **SQL注入攻击(SQL Injection)**
2. **OS命令注入攻击**

* 指通过web应用执行非法的操作系统命令达到攻击的目的；只要能调用shell函数的地方就存在被攻击的风险

1. **HTTP首部注入攻击**

* 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

1. **HTTP响应截断攻击**

* 指攻击者通过在响应首部字段内插入两个换行，作出首部与主体分割所需的空行的攻击

1. **邮件首部注入攻击**

* 攻击者通过向邮件首部To或Subject内任意添加非法内容发起的攻击

1. **目录遍历攻击**

* 指对本无意公开的文件目录，通过非法阶段其目的路径后，达到访问目的的一种攻击

1. **远程文件包含漏洞**

* 脚本需要调用其他文件读入时，攻击者通过改变调用脚本，执行攻击

附：

**HTTP**：HyperText Transfer Protocol 超文本传输（转移）协议

**URI**：Uniform Resource Identifier 统一资源标识符

**URL**：Uniform Resource Location 统一资源定位符

**HTTP无状态协议**：自身不对请求和响应之间的通信状态进行保存