



HTTP 四十问全解析



(/gitchat/author/5c32b561a4078563ee85769b)

axiya (/gitchat/author/...

在大型互联网公司负责业务系统的研发工作，具有十年工作经验。对于 Java 后台研发相关技术有深入的理解与实践。

查看本场Chat



(/gitchat/activity/5de39fc4e8a3a37d46ee95cd)

HTTP 基本概念

1. 什么是 HTTP? HTTP 的作用是什么?
2. 什么是 TCP/IP?
3. TCP/IP 协议族分几层?
4. TCP/IP 协议族分层有什么优点?
5. TCP/IP 分层与 OSI 分层对比
6. 什么是 TCP/IP 通信传输流?
7. TCP/IP 协议族中的 IP 协议
8. TCP/IP 协议族中的 TCP 协议是什么?
9. 请介绍一下 TCP/IP 协议中的 DNS
10. IP、TCP、DNS 和 HTTP 的关系
11. URI 和 URL
12. HTTP 向服务器传递信息的方法
13. 什么是持久连接? 为什么要持久连接?
14. Cookie 的作用是什么? 它是怎样工作的?
15. 什么是 HTTP 报文?
16. HTTP 传输数据的方式有哪些?
17. 怎样发送多种数据的多部分对象集合?
18. 怎样获取部分内容的范围请求?
19. 什么是内容协商? 有哪些类型?
20. 基于 HTTP 的功能追加的协议有哪些?
21. 构建 Web 内容的技术有哪些?
22. HTTP 协议无状态指什么? 怎么才能将状态保存?



- 23. GET 和 POST 的区别是什么?
- 24. HTTP 2.0 与 HTTP 1.1 的区别



HTTP 状态码详解

1. 什么是 HTTP 状态码?
2. 请介绍一下常用的 HTTP 状态码? 并解释一下分别表示什么含义
3. 状态的主要类别有哪几种? 分别表示什么含义?

HTTP 报文解析

1. HTTP 报文首部包含哪些内容?
2. 介绍一下 HTTP 首部字段, 以及构成方式
3. 请介绍一下 HTTP 首部字段的类型有哪几种
3. HTTP 协议首部字段

HTTPS 的使命

1. HTTP 的缺点及解决方案
2. 什么是 HTTPS
3. 什么是相互交换密钥的公开密钥加密技术
4. 请介绍一下 HTTPS 的安全通信机制
5. HTTP 与 HTTPS 的区别是什么?

安全及漏洞全面解析

1. 什么是 SQL 注入?
2. 如何防止 SQL 注入攻击?
3. 什么是 XSS?
4. 如何防止 XSS 漏洞
5. 请介绍一下 CSRF 是什么?
6. CSRF 怎么防御?

HTTP 基本概念

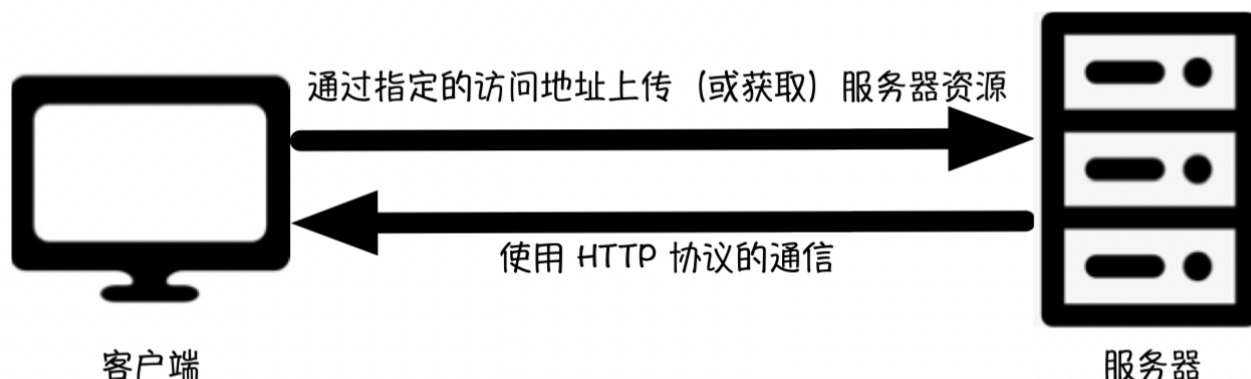
1. 什么是 HTTP? HTTP 的作用是什么?

HTTP 全称: HyperText Transfer Protocol , 超文本传输协议。

HTTP 从客户端到服务器端等一系列运作流程提供规范，是目前互联网上使用的最广泛的一种规范。



1. HTTP 协议用于客户端和服务端之间的通信
2. 通过请求和相应的交换达成通信
3. HTTP 是一种不保存状态的协议
4. HTTP 通过使用 URI 来定位互联网的资源



2. 什么是 TCP/IP?

1. 计算机与网络设备通信，须基于一定的方法规范来进行。确定通信对象、通信语言选择、开始结束通信方式、不同操作系统或者硬件之间如何通信，这些都是需要制定的规则协议。
2. TCP/IP 协议就是由这些多种互联网通信相关协议组合而成，HTTP 为其子集。大部分常用的互联网网络，均通过 TCP/IP 协议族来进行。
3. TCP/IP 协议族常见的协议还包括：TCP、IP、HTTP、FDDI、FTP、DNS、UDP、SNMP 等。
4. TCP/IP 也是指 TCP 和 IP 这两种协议，是在 IP 协议的通行过程中，使用到的协议族的统称。

3. TCP/IP 协议族分几层?

TCP/IP 协议族可以分为 4 层，分别是应用层、传输层、网络层和链路层。

1. 应用层：应用服务之间的通信协议规范，如 FTP、DNS 和 HTTP 都在这层。
2. 传输层：传输层对应用层传输两台计算机之间的数据。传输层主要使用以下两种协议：



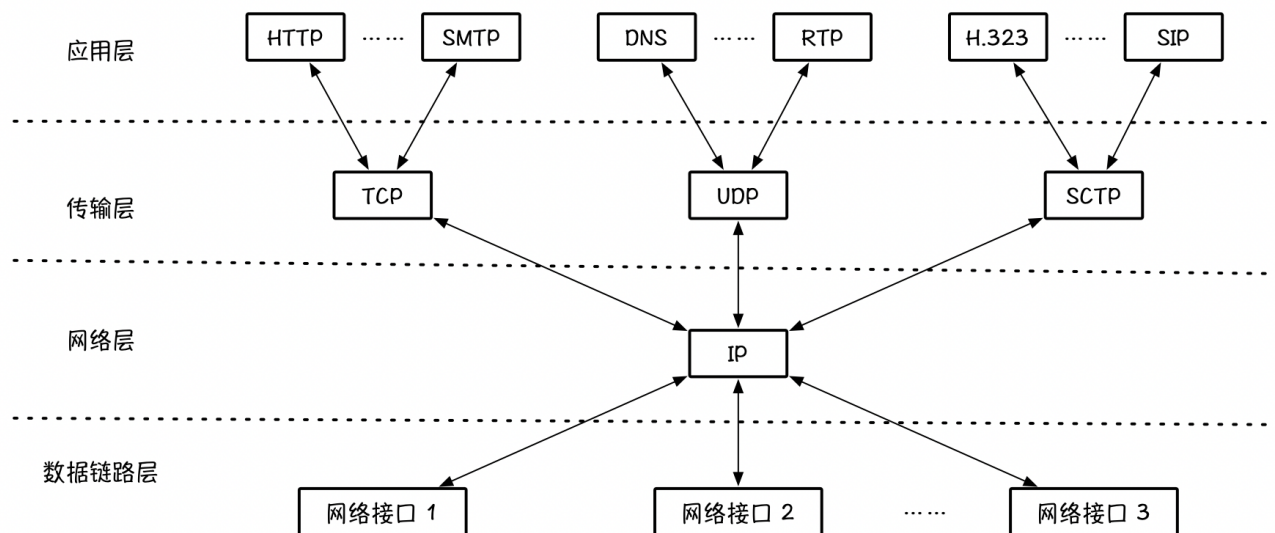
传输控制协议 TCP (数据传输的单位是报文段)

用户数据报协议 UDP (数据传输的单位是用户数据报), 不保证提供交付的可靠性。



3. 网络层: 网络层对传输层提供的数据包进行传送。用来处理网络上流动的数据包, 使用无连接的网际协议 IP 和许多种路由选择协议。网络层还有另一个任务就是选择合适的路由。

4. 链路层 (数据链路层): 硬件上的处理均在链路层的范围内。如: 操作系统、硬件设备的驱动、网卡等。



4. TCP/IP 协议族分层有什么优点?

TCP/IP 协议族分层的优点是:

1. 改动方便: 如果仅使用一个协议, 那么当其中的某一部分发生改变的时候, 就需要把整体全部替换掉。
2. 设计简单: 使用分层时候, 仅需要替换改变的层的内容, 只需要把每层之间的接口部分定义规划好, 那么各层内部就可以随意改变, 更加灵活自由, 在设计上也简单很多。

5. TCP/IP 分层与 OSI 分层对比

TCP/IP 协议族按层次分为以下 4 层: 应用层、传输层、网络层和数据链路层。

OSI 则分为 7 层: 应用层、表示层、会话层、运输层、网络层、数据链路层和物理层。

对应关系如下:



OSI 体系结构

TCP/IP 体系结构



| | |
|-------|-------|
| 应用层 | 应用层 |
| 表示层 | |
| 会话层 | |
| 传输层 | 传输层 |
| 网络层 | 网络层 |
| 数据链路层 | 数据链路层 |
| 物理层 | |

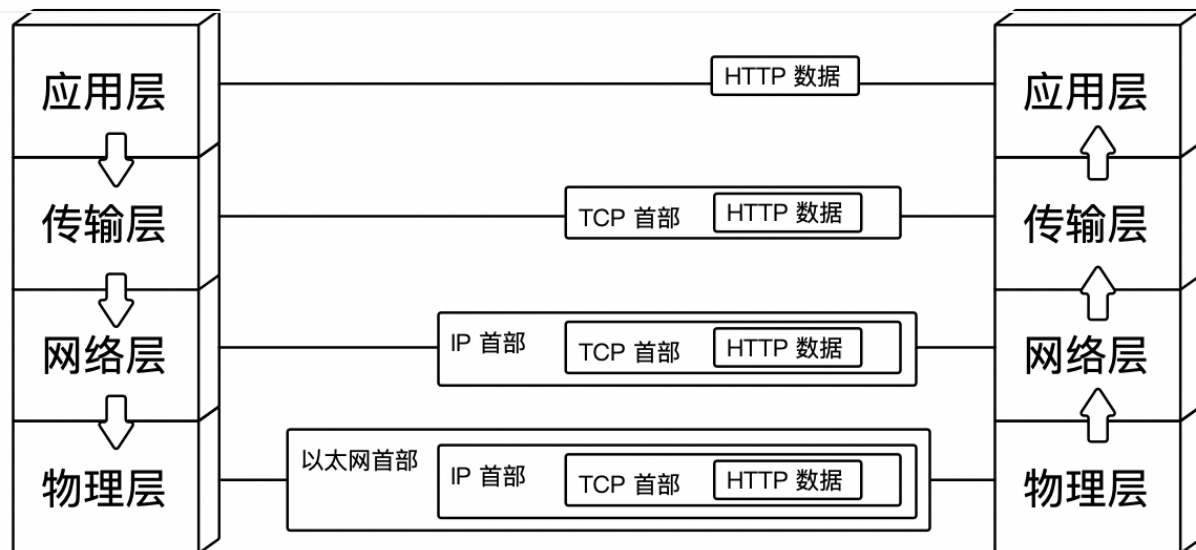
6. 什么是 TCP/IP 通信传输流？

通过 TCP/IP 协议通信方式，会遵循分层的顺序与对方进行通信

- 发送端的顺序是：应用层 -> 传输层 -> 网络层 -> 链路层；
- 接受端的顺序是：链路层 -> 网络层 -> 传输层 -> 应用层。

客户端（发送端）
每通过一层增加首部

服务器端（接收端）
每通过一层删除首部



7. TCP/IP 协议族中的 IP 协议

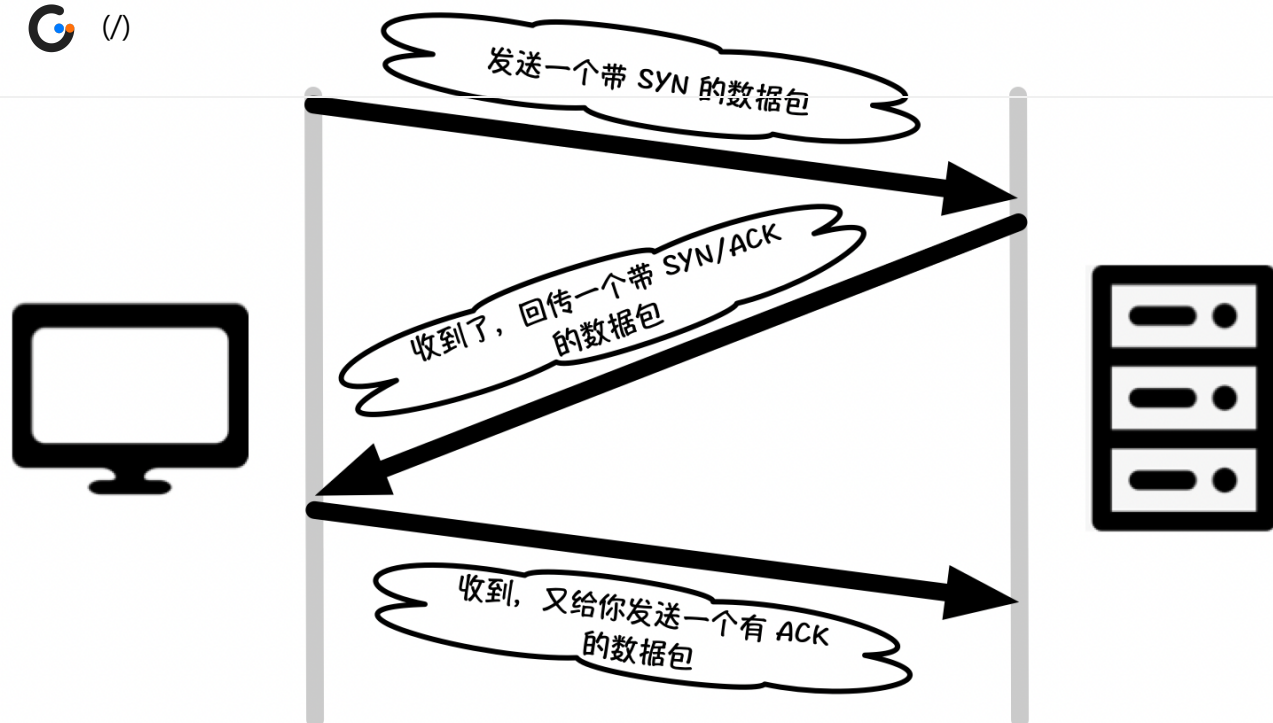
IP 网际协议处于网络层，用于传送数据包。它通过 IP 地址和 MAC 地址将数据包传送到指定的位置。

其中 IP 地址指明了分配给节点的地址，可变化；MAC 地址指明了所属网卡的固定地址，不可变化。

IP 之间的通信是依赖于 MAC 地址的，在网络通信的过程中，根据 ARP（一种地址解析协议）协议，通过 IP 反查出对应 MAC 地址，再通过 MAC 地址来搜索中转目标。

8. TCP/IP 协议族中的 TCP 协议是什么？

TCP 协议提供可靠的字节流服务，主要是通过采用三次握手的策略来确保传输数据的准确性的。



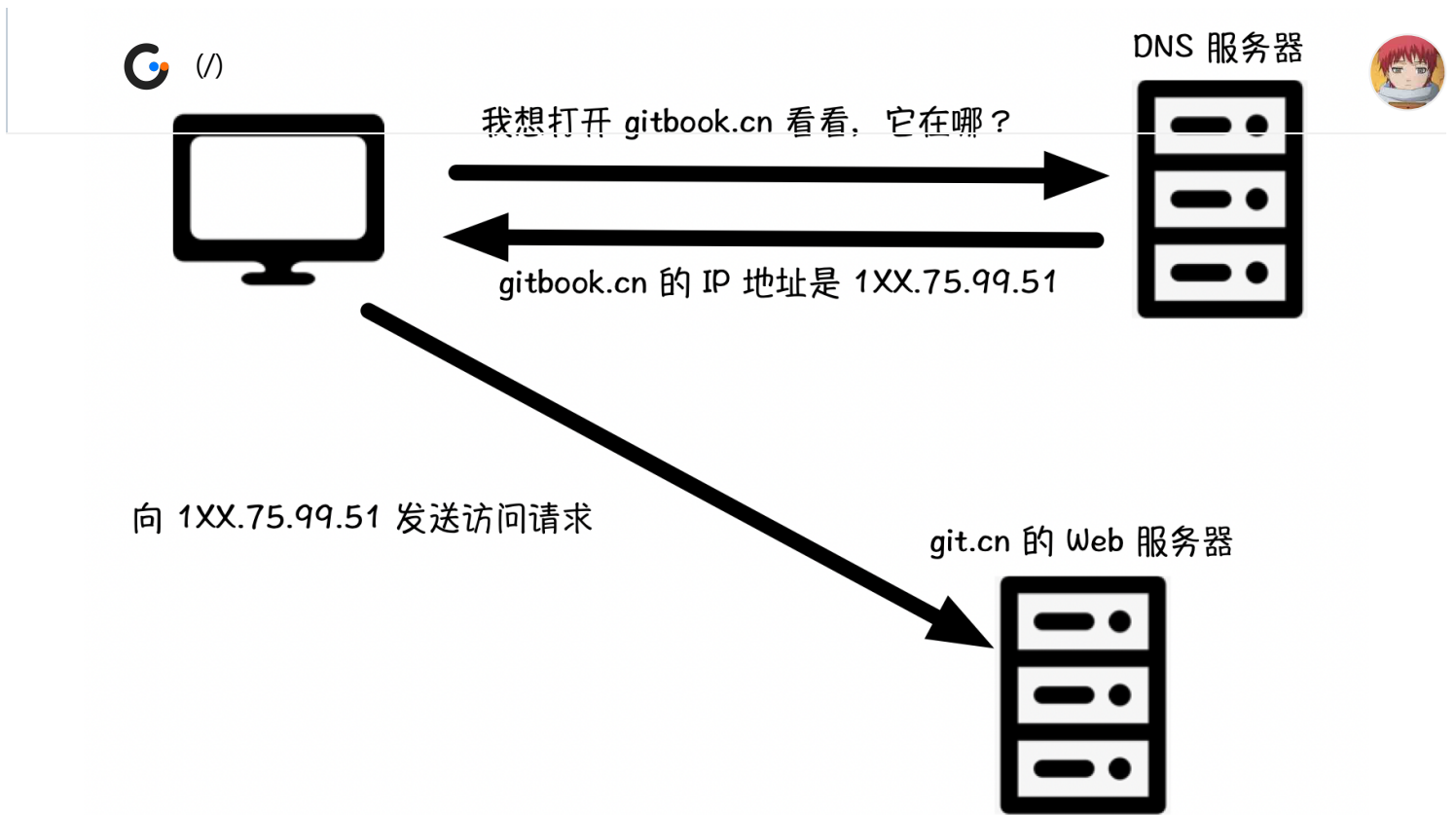
用 TCP 协议将数据包发送之后, 它会向对方确认是否成功送达。握手过程中使用了 TCP 的标志 (flag) ——SYN (synchronize) 和 ACK (acknowledgement) 。

1. 发送端 A 发送 SYN 标志的数据包给信息接收方 B。
2. B 收到后数据包之后, 回传 SYN/ACK 标志的数据包, 表示确认信息。
3. 发送端 A 再回传一个 ACK 标志的数据包, 代表“握手”结束。
4. 若在握手过程中某个阶段莫名中断, TCP 协议会再次按照相同顺序发送相同的数据包。

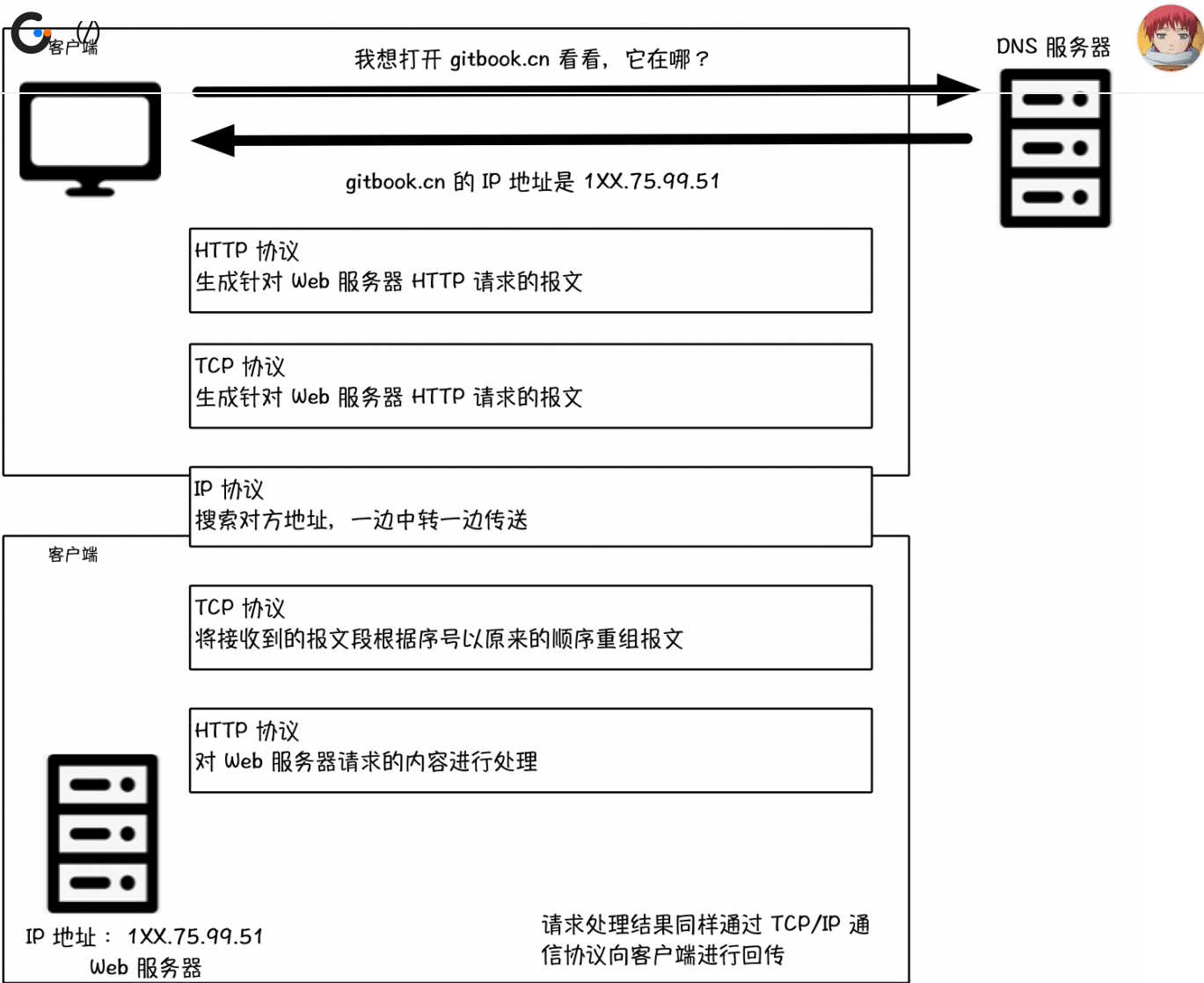
9. 请介绍一下 TCP/IP 协议中的 DNS

DNS 全称为 Domain Name System。

DNS 协议与 HTTP 一样位于应用层, 主要负责将域名和 IP 之间的相互解析。



10. IP、TCP、DNS 和 HTTP 的关系



11. URI 和 URL

- 1. URI（统一资源标识符）：用字符串标识互联网上的某一资源。
- 2. URL（统一资源定位符）：表示网络资源所在的位置。由上可见，URL 是 URI 的一个子集。

12. HTTP 向服务器传递信息的方法

| 方法名称 | 含义 |
|--------------|---|
| GET（获取资源） | 请求访问已被 URI 标识的资源。响应返回经服务器解析后的内容 |
| POST（传输实体主题） | GET 和 POST 都可以传输实体的主题，但一般使用 POST 方法来传输。区别在意 POST 的主要目的并不是获取响应的主体内容。 |



| 方法名称 | 含义 |
|---------------------|---|
| PUT (传输文件) | 用来传输文件。将文件内容放到请求报文的主题之中，然后放到请求的 URI 中 |
| HEAD (获得报文首部) | HEAD 方法和 GET 方法一样，但不返回报文主体的部分。用于确认 URI 是否有效及更新资源的时间等。 |
| DELETED (删除文件) | 用来删除文件，与 PUT 方法相反。DELETED 根据请求删除 URI 内指定的资源 |
| OPTIONS (询问支持的方法) | 查询根据请求 URI 指定的资源支持方法 |
| TRACE (路径追踪) | 让 Web 服务器端将之前的请求返回个客户端的方法 |
| CONNECT (用隧道协议连接代理) | 与代理服务器通信时建立隧道，使用 SSL 和 TLS 协议把加密后的通信内容经网络隧道进行传输。 |

13. 什么是持久连接？为什么要持久连接？

在使用 HTTP 协议建立通信之后，在没有提出要断开连接的时候，TCP 将一直保持连接状态。

持久连接好处是减少了 TCP 连接的重复建立和断开所造成的的额外开销，减轻了服务器端的负载。而且减少重复建立连接的时间可以使 HTTP 请求和相应更早的结束，这样 Web 页面的加载速度也相应提高了。

14. Cookie 的作用是什么？它是怎样工作的？

1. Cookie 技术将 Cookie 写入请求信息和响应报文中，以此来控制和管理客户端的状态。
2. Cookie 是通过由服务器端发出响应报文中的 SetCookie 的首部字段的信息，告知客户端需要保存 Cookie 的。当客户端再次发送请求的时候，会在请求报文中加入 Cookie 值。服务器端在接收到带有 Cookie 值的请求后，就会去查连接请求的来源，对比服务器存储的记录，然后得到之前的状态信息。

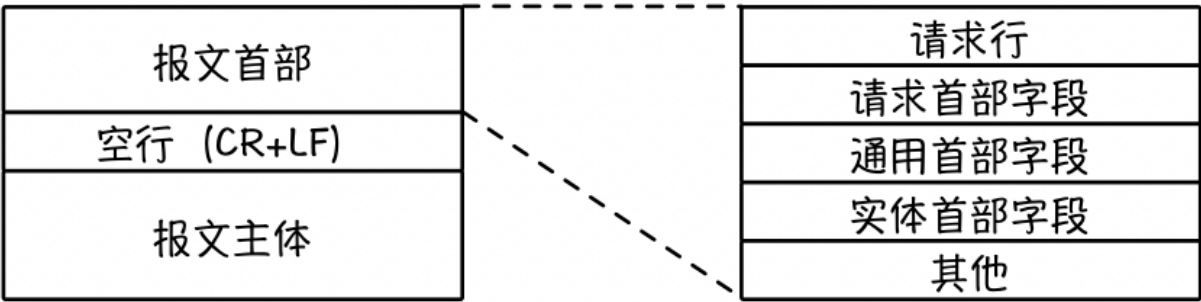
15. 什么是 HTTP 报文？

HTTP 协议交互的信息被称为 HTTP 报文。报文大致可以分为报文首部和报文主体两块，两者由空行 (CR+LF) 来划分，报文主体可以不要。

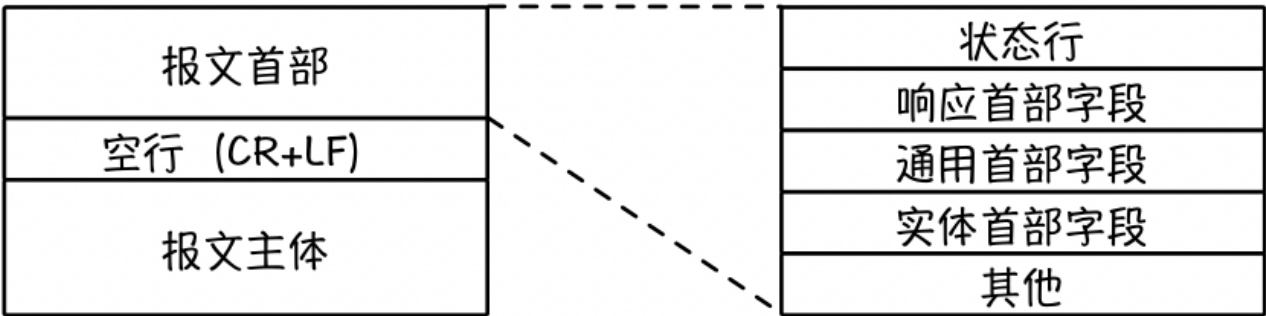
| | |
|------------|---------------------------|
| 报文首部 | 服务器端或者客户端需要处理的请求或响应的内容及属性 |
| 空行 (CR+LF) | CR (回车) + LF (换行) |
| 报文主体 | 应被发送的数据 (可以不要) |



1. 客户端的 HTTP 报文被称为请求报文



2. 服务器端的 HTTP 报文为响应报文



由上可知，请求报文和相应报文都是由请求行、状态行、首部字段和其他组成。

16. HTTP 传输数据的方式有哪些？

- HTTP 可以直接按照数据的原貌进行传输，也可以在传输的过程中对数据进行编码来提升传输的速率。但是在提高传输速率的同时，编码操作会占用更多的 CPU 等资源。
- 可以采用压缩传输内容的编码方式提高传送速率。采用将主题编码分割成块，然后进行编码传输的分块传输编码形式，这种操作可以提高用户的使用体验。

注意：通常报文主体等同于实体主体。但是如果在传输的过程中进行编码操作，实体主体的内容将发生变化，会导致它和报文主体产生差异。



17. 怎样发送多种数据的多部分对象集合？



1. 在 HTTP 报文中使用多部分对象集合时，需要在首部字段里加上 Contenttype。
2. 通过字符串 boundary 来切分各类实体，这些实体是由多部分对象集合指定的。

18. 怎样获取部分内容的范围请求？

可以通过首部字段 Range 来指定资源的 byte 的范围。

1. 1001~2000 字节

```
Range:bytes=1001-2000
```

2. 1001 以后的所有字节

```
Range:bytes=1001-
```

3. 从开始到 1000 字节和 2001~5000 的多重范围

```
Range:bytes=0-1000,2001-5000
```

针对范围请求，响应会返回状态码为 206 的响应报文。而对于多重范围的范围请求，响应会在首部字段 ContentType 标明 multipart/byteranges 后返回响应报文。

19. 什么是内容协商？有哪些类型？

内容协商机制是指客户端和服务端就响应的资源内容进行交涉，然后提供给客户端最为适合的资源。内容协商会以响应资源的语言、字符集、编码方式等作为判断的基准。其内容包含在首部以下字段中：Accept、Accept-Charset、Accept-Encoding、Accept-Language、Content-Language。

内容协商包括：服务器驱动协商、客户端驱动协商和透明协商三种。

1. 服务器驱动协商：由服务器端进行内容协商。以请求的首部字段为参考，在服务器端自动处理。但对用户来说，以浏览器发送的信息作为判定的依据，并不一定能筛选出最优



内容。

2. 客户端缺东协商：由客户端进行内容协商的方式。用户从浏览器显示的可选项列表中手动选择。还可以利用 JavaScript 脚本在 Web 页面上自动进行上述选择。比如按 OS 的类型或浏览器类型，自行切换成 PC 版页面或手机版页面。
3. 透明协商：是服务器驱动和客户端驱动的结合体，是由服务器端和客户端各自进行内容协商的一种方法。

20. 基于 HTTP 的功能追加的协议有哪些？

1. 消除 HTTP 瓶颈的 SPDY 协议
2. 通过浏览器进行全双工通信的 WebSocket
3. 成长了的 HTTP 2.0
4. Web 服务器管理文件的 WebDAV

21. 构建 Web 内容的技术有哪些？

1. HTML：Web 页面几乎都是由 HTML 写成的。
2. 动态 HTML：是指使用客户端脚本语言将静态 HTML 变为动态的 HTML 的技术的总称。例如：客户端脚本语言 JavaScript 和指定于发生动态变化的 HTML 的 DOM 等。
3. Web 应用：如通过 Web 功能提供的应用程序；与 Web 服务器及程序协作的 CGI；因 Java 而普及的 Servlet 等。
4. 数据发布格式及语言：如可扩展标记语言 XML；发布更新信息的 RSS 和 Atom；JavaScript 衍生的轻量级易用 JSON 等。

22. HTTP 协议无状态指什么？怎样才能将状态保存？

HTTP 协议无状态在一个会话里面，不同的两次请求彼此是不了解。

但是通过 Cookie 或者 Session 可以将状态保存，后续访问可能利用到前面的信息。

23. GET 和 POST 的区别是什么？

1. 从服务器获取信息一般使用 GET，想服务器发送信息一般用 POST。
2. GET 和 POST 数据提交方式不同，GET 通过在 URL 请求后面增加 filed=value 的封装形式来进行；POST 则利用协议 BODY 来进行数据的封装。

3. GET 传输数据量比较小，效率也不高；而 POST 可以传输比较大的数据量。
4. GET 不安全，可以被外部看见，造成信息泄露的风险，POST 相对来说安全一些。



24. HTTP 2.0 与 HTTP 1.1 的区别

1. HTTP 2.0 没有采用文本格式，采用的是二进制格式。
2. HTTP 2.0 采用的是完全多路复用机制，而非有序并阻塞的。
3. HTTP 2.0 将报头进行压缩，降低了成本。
4. HTTP 2.0 服务器主动将响应“推送”到客户端的缓存里面。

HTTP 状态码详解

1. 什么是 HTTP 状态码？

HTTP 状态码全称：HTTP Status Code。表示服务器在响应超文本传输协议访问的时候返回的状态 3 位数字代码。例如，当客户端向服务端进行 HTTP 请求的时候，服务器会返回一个代码数据来回应请求，这个代码数据就是：HTTP 状态码。

2. 请介绍一下常用的 HTTP 状态码？并解释一下分别表示什么含义

1. 200：OK，基于 HTTP 协议的访问在服务端被正常处理并返回。
2. 302：临时重定向，表示请求的网页临时移动到其他的 URI。
3. 404：表示服务器上无法找到访问的资料员。
4. 500：表明服务器端访问响应发生了错误。可能是后台 BUG，也可能是机器故障导致。

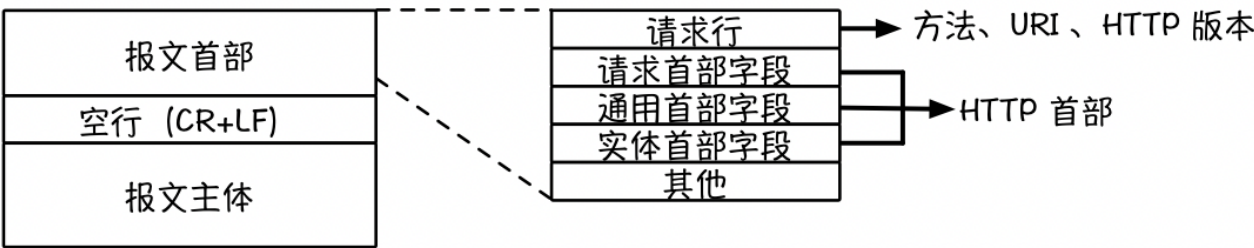
3. 状态的主要类别有哪几种？分别表示什么含义？

1. 1XX Informational（信息性状态码）：服务器正在处理当前的请求。
2. 2XX Success（成功状态码）：请求被服务器正确接收，并正确执行。
3. 3XX Redirection（重定向状态码）：需要再次操作，才能完成整个访问操作。
4. 4XX Client Error（客户端错误状态码）：客户端的请求出现问题，服务端无法响应（例如，访问不存在的资源）。
5. 5XX Server Error（服务器错误状态码）：服务器内部处理访问请求的时候出现异常。

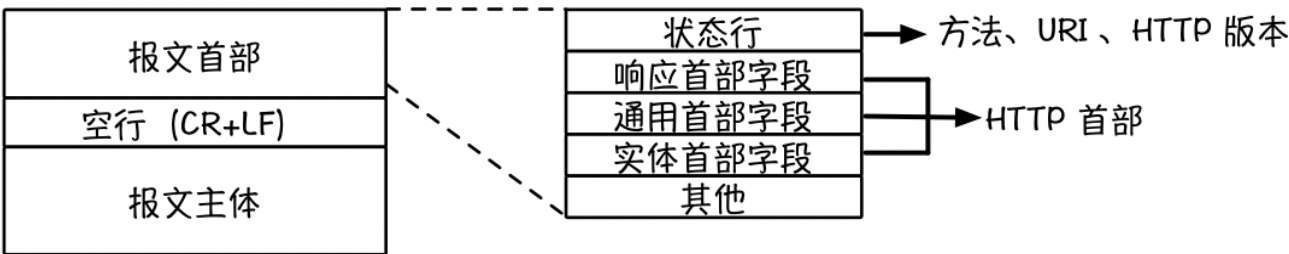
1. HTTP 报文首部包含哪些内容？

HTTP 协议的请求和响应报文中必定包含 HTTP 首部。首部内容为客户端和服务端分别处理请求和响应提供所需要的信息。

在请求中，HTTP 报文由方法、URI、HTTP 版本、HTTP 首部字段等部分构成。



在响应中，HTTP 报文由 HTTP 版本、状态码（数字和原因短语）、HTTP 首部字段 3 部分构成。





2. 介绍一下 HTTP 首部字段，以及构成方式

首部字段的主要作用：给浏览器和服务端提供一些必要信息，如报文主体 SIZE、语言类型、认证方式等内容。它是 HTTP 报文的组成要素之一。

首部字段构成方式：由字段名称和字段值组成，用冒号“:”分隔。例如：Content-type:text/html。首部字段可以有多个值组成。

3. 请介绍一下 HTTP 首部字段的类型有哪几种

首部字段类型总共分为四类。如下所示：

- 1. 通用首部字段 (General Header Fields)
- 2. 请求首部字段 (Request Header Fields)
3. 响应首部字段 (Response Header Fields)
4. 实体首部字段 (Entity Header Fields)



3. HTTP 协议首部字段

| | | |
|-----------------------------------|-------------------|----------------|
| 通用首部字段 (请求报文与响应报文都会使用的首部字段) | Date | 创建报文时间 |
| | Connection | 连接的管理 |
| | Cache-Control | 缓存的控制 |
| | Transfer-Encoding | 报文主体的传输编码方式 |
| 请求首部字段 (请求报文会使用的首部字段) | Host | 请求资源所在服务器 |
| | Accept | 可处理的媒体类型 |
| | Accept-Charset | 可接收的字符集 |
| | Accept-Encoding | 可接受的内容编码 |
| 响应首部字段 (响应报文会使用的首部字段) | Accept-Language | 可接受的自然语言 |
| | Accept-Ranges | 可接受的字节范围 |
| | Location | 令客户端重新定向到的 URI |
| | Server | HTTP 服务器的安装信息 |
| 实体首部字段 (请求报文与响应报文的实体部分使用的首部字段) | Allow | 资源可支持的 HTTP 方法 |
| | Content-Type | 实体主类的类型 |
| | Content-Encoding | 实体主体适用的编码方式 |
| | Content-Language | 实体主体的自然语言 |
| | Content-Length | 实体主体的的字节数 |



HTTPS 的使命

1. HTTP 的缺点及解决方案

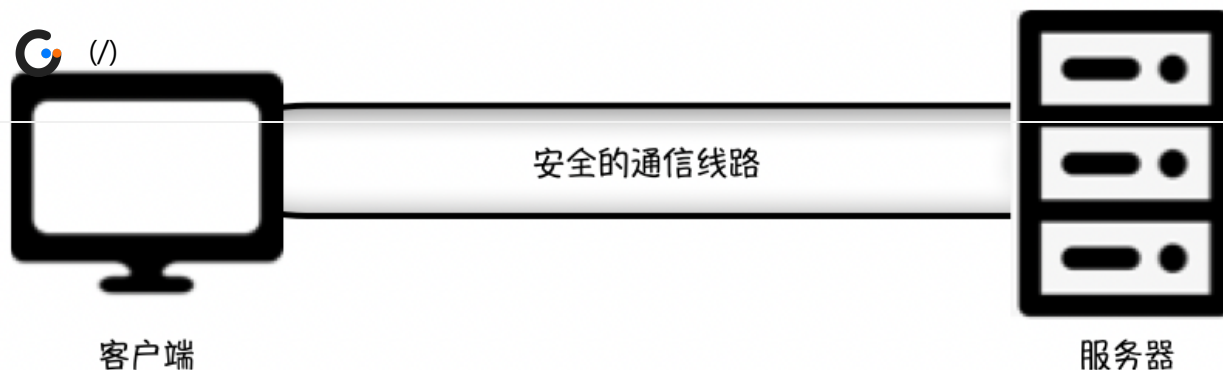
缺点：

1. 明文形式通信（未进行加密操作），极可能被盗取数据。
2. 没有验证访问者的合法身份，会遇到被伪装欺骗可能。
3. 报文完整性无法进行验证，所以内容信息会被篡改的可能。

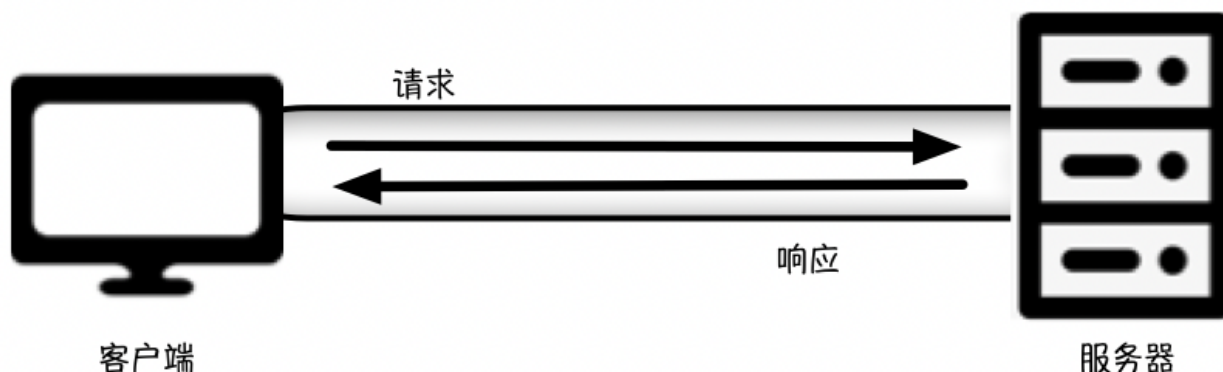
解决方案：

1. 加密处理预防窃听

- 通信加密：HTTP 协议加密机制缺失，但利用 SSL（SecureSocketLayer，安全套接层）或 TLS（TransportLayerSecurity，安全传输层协议）共同作用，加密 HTTP 的传输信息。



服务器与客户端之间建立起安全的通信线路之后开始通信



- 内容加密：HTTP 协议不提供加密操作，因此 HTTP 协议传输的数据本身加密，把 HTTP 报文里所含的数据进行加密操作。但数据传输的过程中仍有数据被篡改的可能。

2. 使用 SSL 可以验证对方身份。SSL 除了具备加密处理能力，还使用了称为证书的方法，可用于确定通信方。

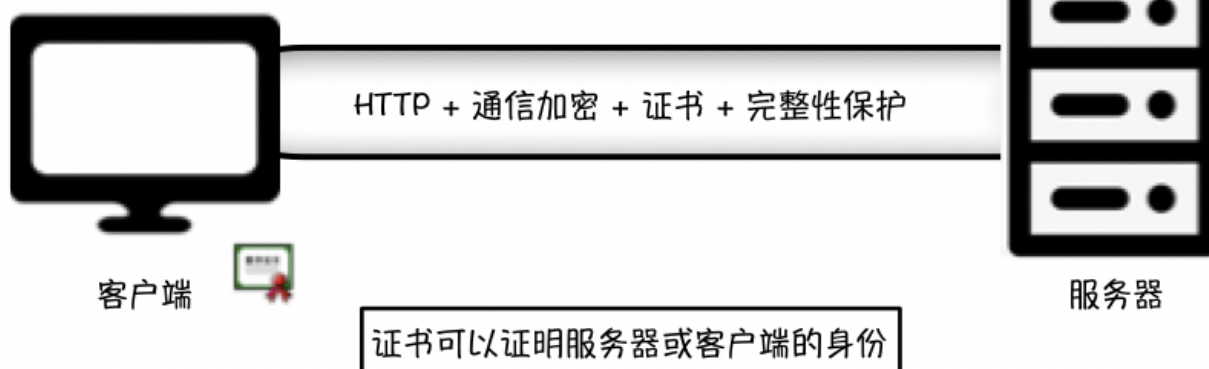
3. 可以使用 MD5 和 SHA1 等散列值校验的方法，以及用来确认文件的数字签名方法。

非常可惜的是，以上的一些方法仍然存在很大的风险，如果想要有效地保证信息的安全性，则需要使用 HTTPS。

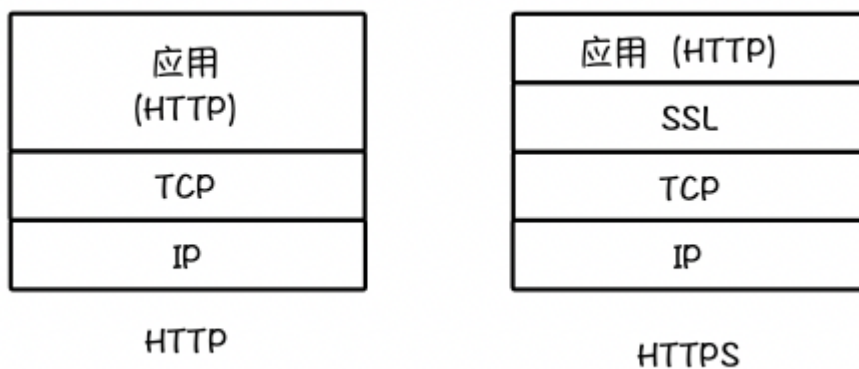
2. 什么是 HTTPS

HTTP + 加密 + 认证 + 完整性保护 = HTTPS

1. HTTP 加上加密处理和认证以及完整性保护后即是 HTTPS



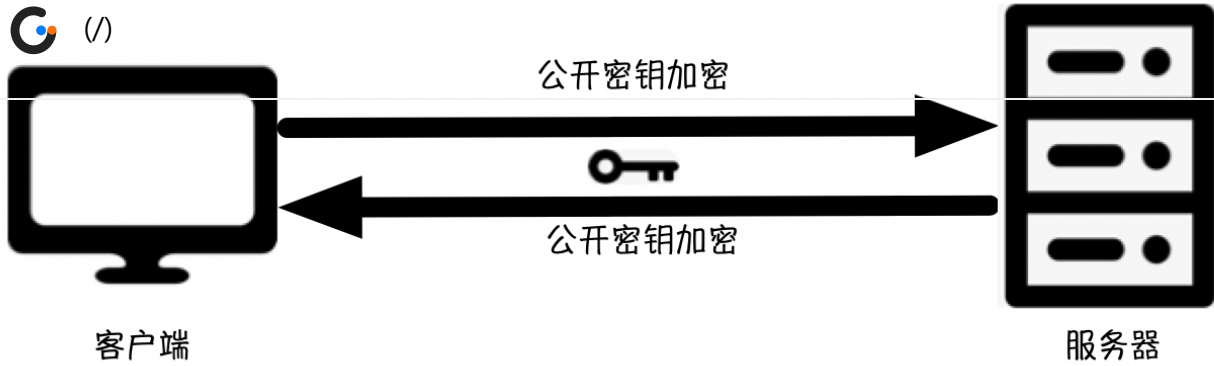
2. HTTPS 是身披 SSL 外壳的 HTTP



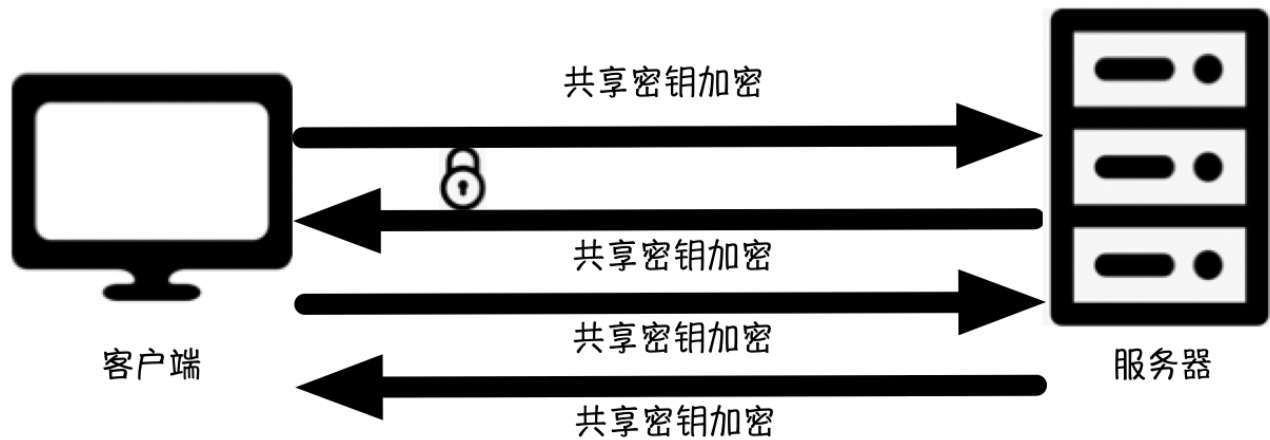
3. 什么是相互交换密钥的公开密钥加密技术

公开密钥加密处理起来比共享密钥加密方式更为复杂，因此若在通信时使用公开密钥加密方式，效率就很低。

1. 使用公开密钥加密方式，安全地交换在稍后的共享密钥加密中要使用的密钥



2. 确保交换的密钥是安全的前提下，使用共享密钥加密方式进行通信。



4. 请介绍一下 HTTPS 的安全通信机制

1. 利用对称密钥原理，服务器端生成对称密钥，私钥自己保存，公钥发送到外部。
2. 客户端向一个权威的服务器检查证书的合法性，如果合法，客户端生成随机数，这个数字就是通信的密钥，用公钥加密这段随机数，然后发送到服务器。
3. 服务器使用密钥解密获取对称密钥，然后，双方就可以安全通信了。

5. HTTP 与 HTTPS 的区别是什么？

1. 安全性质不同：HTTP 是不安全的，而 HTTPS 是安全的。
2. URL 开头不同：HTTP 以 `http://` 开头，HTTPS 以 `https://` 开头。
3. 标准端口不同：HTTP 标准端口是 80，HTTPS 的标准端口是 443。
4. 加密要求不同：HTTP 无需加密，而 HTTPS 对传输的数据进行加密。
5. 证书要求不同：HTTP 无需证书，而 HTTPS 需要 SSL 证书。



1. 什么是 SQL 注入？

SQL 注入是一种注入攻击。攻击者通过将破坏性 SQL 代码进行数据库查询，使攻击者能够完全控制数据库资源。

2. 如何防止 SQL 注入攻击？

1. 不要使用动态 SQL，使用完整的语句和参数化方式来查询。
2. 合理设置数据库的权限。
3. 禁止直接向用户显示数据库错误。
4. 对访问数据库的 Web 服务，使用 Web 应用程序防火墙。

3. 什么是 XSS？

XSS 全称：跨站脚本攻击（Cross Site Scripting）。是将前端脚本代码插入 Web 页面中，当用户浏览页面时，会执行嵌套在 Web 页面里面的脚本代码，从而达到攻击用户的目的。

XSS 类型包括：

- 存储型 XSS：存入了数据库，再取出来时导致的 XSS
- 反射型 XSS：在网址 URL 后输入 XSS 代码，如 `<script> alert(1)</script>`，然后访问时导致 HTML 页面加载这段代码即可达到弹框效果。

4. 如何防止 XSS 漏洞

1. 在信息提交或者 url 参数传递前，对需要的参数进行过滤
2. 过滤用户输入，检查用户输入的内容中是否有非法内容。如 `<>`（尖括号）、`"`（引号）、`'`（单引号）

5. 请介绍一下 CSRF 是什么？

CSRF：Cross-site request forgery 跨站请求伪造。

cookie 是网站利用来识别用户的，用户成功登陆之后浏览器就会得到一个 cookie 来标识其身份，在不关闭浏览器或者退出登录，以后访问这个网站会带上这个 cookie。



1. 登录某一受信任网站 X，并生成本地 Cookie。
2. 如果此时用户也访问了网站 B，访问者在网站 A 的数据就会被 B 使用用户 cookie 假冒更新。

6. CSRF 怎么防御？

1. 验证码与二次验证
2. 对请求的 referer 进行检测
3. 添加随机 token 校验

本文首发于 GitChat，未经授权不得转载，转载需与 GitChat 联系。



71



0

互动评论



说点什么

评论



sinli

7 个月前

不错啊，图是用什么工具画的？



鼓掌



阿加西

9 个月前

图很不错，浅显易懂



鼓掌



Daniel

图例很生动，感谢博主

1 年前



鼓掌



农夫三拳²³³

2 年前

不错，可以考虑再出个tcp/ip的面试题解析~



鼓掌



Pu

2 年前

可



鼓掌



Pu

2 年前

可



鼓掌



We Smile

2 年前

耐思



1



RAY

2 年前

很全面👍



2

查看更多