23 | HTTPS是什么？SSL/TLS又是什么？

2019-07-19 Chrono

《透视HTTP协议》 课程介绍 >

讲述：Chrono

时长 11:57 大小 10.95M

从今天开始，我们开始进入全新的“安全篇”，聊聊与安全相关的 HTTPS、SSL、TLS。

在第 14 讲中，我曾经谈到过 HTTP 的一些缺点，其中的“无状态”在加入 Cookie 后得到了解决，而另两个缺点——“明文”和“不安全”仅凭 HTTP 自身是无力解决的，需要引入新的 HTTPS 协议。

为什么要有 HTTPS?

简单的回答是“**因为 HTTP 不安全**”。

由于 HTTP 天生“明文”的特点，整个传输过程完全透明，任何人都能够在链路中截获、修改或者伪造请求 / 响应报文，数据不具有可信性。

比如，前几讲中说过的“代理服务”。它作为 HTTP 通信的中间人，在数据上下行的时候可以添加或删除部分头字段，也可以使用黑白名单过滤 body 里的关键字，甚至直接发送虚假的请求、响应，而浏览器和源服务器都没有办法判断报文的真伪。

这对于网络购物、网上银行、证券交易等需要高度信任的应用场景来说是非常致命的。如果没有基本的安全保护，使用互联网进行各种电子商务、电子政务就根本无从谈起。

对于安全性要求不那么高的新闻、视频、搜索等网站来说，由于互联网上的恶意用户、恶意代理越来越多，也很容易遭到“流量劫持”的攻击，在页面里强行嵌入广告，或者分流用户，导致各种利益损失。

对于你我这样的普通网民来说，HTTP 不安全的隐患就更大了，上网的记录会被轻易截获，网站是否真实也无法验证，黑客可以伪装成银行网站，盗取真实姓名、密码、银行卡等敏感信息，威胁人身安全和财产安全。

总的来说，今天的互联网已经不再是早期的“田园牧歌”时代，而是进入了“黑暗森林”状态。上网的时候必须步步为营、处处小心，否则就会被不知道埋伏在哪里的黑客所“猎杀”。

什么是安全?

既然 HTTP“不安全”，那什么样的通信过程才是安全的呢?

通常认为，如果通信过程具备了四个特性，就可以认为是“安全”的，这四个特性是：机密性、完整性，身份认证和不可否认。

机密性（Secrecy/Confidentiality）是指对数据的“保密”，只能由可信的人访问，对其他人是不可见的“秘密”，简单来说就是不能让不相关的人看到不该看的东西。

比如小明和小红私下聊天，但“隔墙有耳”，被小强在旁边的房间里全偷听到了，这就是没有机密性。我们之前一直用的 Wireshark ，实际上也是利用了 HTTP 的这个特点，捕获了传输过程中的所有数据。

完整性（Integrity，也叫一致性）是指数据在传输过程中没有被篡改，不多也不少，“完完整整”地保持着原状。

机密性虽然可以让数据成为“秘密”，但不能防止黑客对数据的修改，黑客可以替换数据，调整数据的顺序，或者增加、删除部分数据，破坏通信过程。

比如，小明给小红写了张纸条：“明天公园见”。小强把“公园”划掉，模仿小明的笔迹把这句话改成了“明天广场见”。小红收到后无法验证完整性，信以为真，第二天的约会就告吹了。

身份认证（Authentication）是指确认对方的真实身份，也就是“证明你真的是你”，保证消息只能发送给可信的人。

如果通信时另一方是假冒的网站，那么数据再保密也没有用，黑客完全可以使用冒充的身份“套”出各种信息，加密和没加密一样。

比如，小明给小红写了封情书：“我喜欢你”，但不留心发给了小强。小强将错就错，假冒小红回复了一个“白日做梦”，小明不知道这其实是小强的话，误以为是小红的，后果可想而知。

第四个特性是**不可否认**（Non-repudiation/Undeniable），也叫不可抵赖，意思是不能否认已经发生过的行为，不能“说话不算数”“耍赖皮”。

使用前三个特性，可以解决安全通信的大部分问题，但如果缺了不可否认，那通信的事务真实性就得不到保证，有可能出现“老赖”。

比如，小明借了小红一千元，没写借条，第二天矢口否认，小红也确实拿不出借钱的证据，只能认倒霉。另一种情况是小明借钱后还了小红，但没写收条，小红于是不承认小明还钱的事，说根本没还，要小明再掏出一千元。

所以，只有同时具备了机密性、完整性、身份认证、不可否认这四个特性，通信双方的利益才能有保障，才能算得上是真正的安全。

什么是 HTTPS?

说到这里，终于轮到今天的主角 HTTPS 出场了，它为 HTTP 增加了刚才所说的四大安全特性。

HTTPS 其实是一个“非常简单”的协议，RFC 文档很小，只有短短的 7 页，里面规定了**新的协议名“https”，默认端口号 443**，至于其他的什么请求 - 应答模式、报文结构、请求方法、

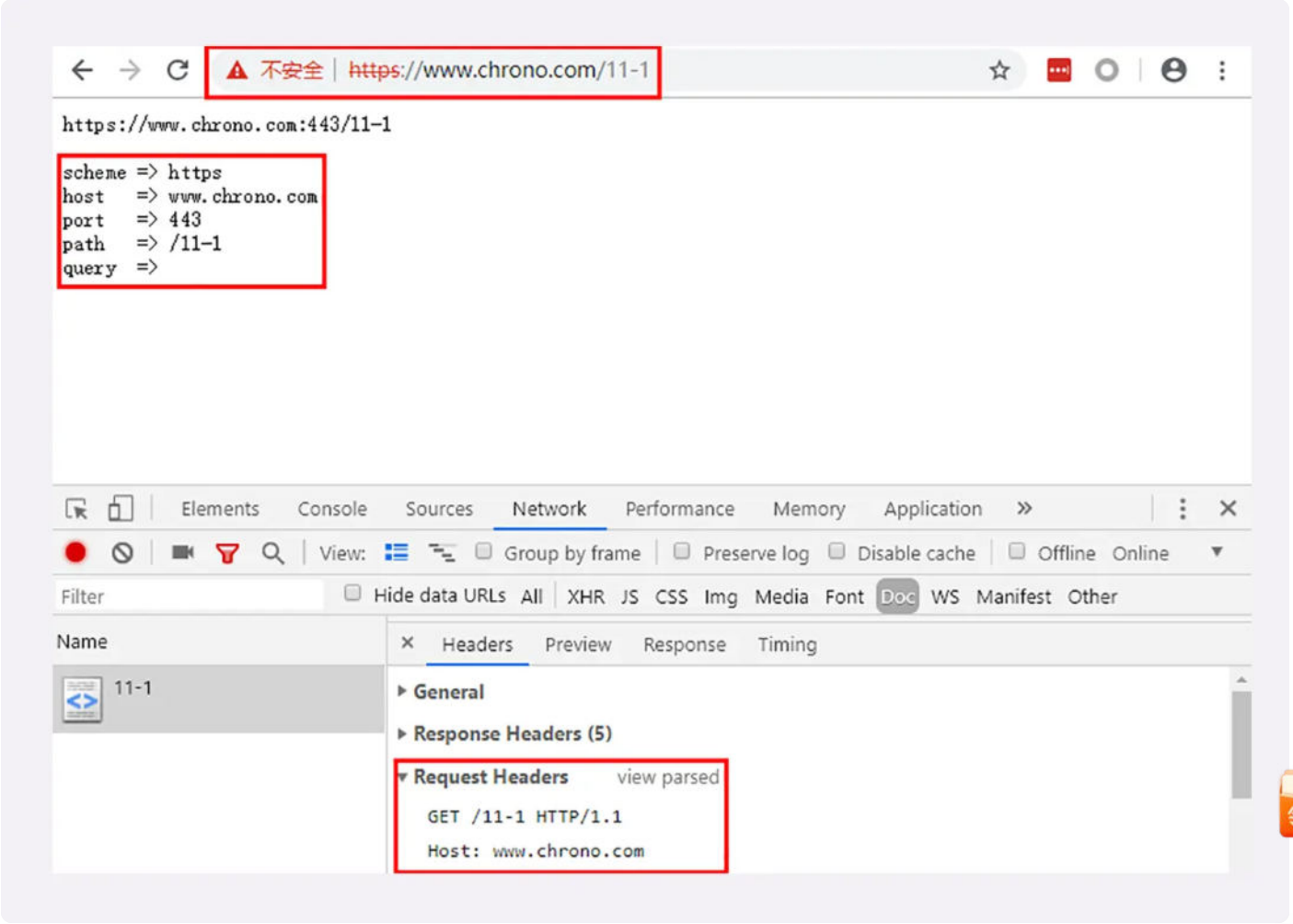
URI、头字段、连接管理等等都完全沿用 HTTP，没有任何新的东西。

也就是说，除了协议名“http”和端口号 80 这两点不同，HTTPS 协议在语法、语义上和 HTTP 完全一样，优缺点也“照单全收”（当然要除去“明文”和“不安全”）。

不信你可以用 URI“https://www.chrono.com”访问之前 08 至 21 讲的所有示例，看看它的响应报文是否与 HTTP 一样。

复制代码

```
https://www.chrono.com
https://www.chrono.com/11-1
https://www.chrono.com/15-1?name=a.json
https://www.chrono.com/16-1
```

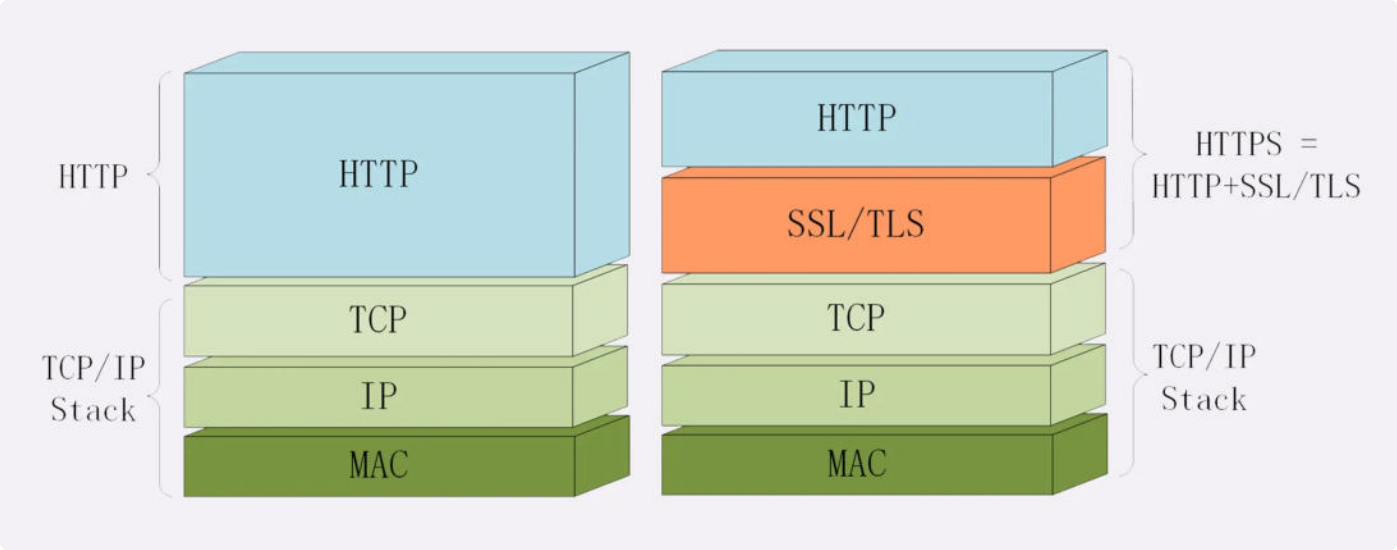


你肯定已经注意到了，在用 HTTPS 访问实验环境时 Chrome 会有不安全提示，必须点击“高级 – 继续前往”才能顺利显示页面。而且如果用 Wireshark 抓包，也会发现与 HTTP 不一样，不再是简单可见的明文，多了“Client Hello”“Server Hello”等新的数据包。

这就是 HTTPS 与 HTTP 最大的区别，它能够鉴别危险的网站，并且尽最大可能保证你的上网安全，防御黑客对信息的窃听、篡改或者“钓鱼”、伪造。

你可能要问了，既然没有新东西，HTTPS 凭什么就能做到机密性、完整性这些安全特性呢？

秘密就在于 HTTPS 名字里的“S”，它把 HTTP 下层的传输协议由 TCP/IP 换成了 SSL/TLS，由“**HTTP over TCP/IP**”变成了“**HTTP over SSL/TLS**”，让 HTTP 运行在了安全的 SSL/TLS 协议上（可参考第 4 讲和第 5 讲），收发报文不再使用 Socket API，而是调用专门的安全接口。



所以说，HTTPS 本身并没有什么“惊世骇俗”的本事，全是靠着后面的 SSL/TLS“撑腰”。只要学会了 SSL/TLS，HTTPS 自然就“手到擒来”。

SSL/TLS

现在我们就来看看 SSL/TLS，它到底是个什么来历。

SSL 即安全套接层（Secure Sockets Layer），在 OSI 模型中处于第 5 层（会话层），由网景公司于 1994 年发明，有 v2 和 v3 两个版本，而 v1 因为有严重的缺陷从未公开过。

SSL 发展到 v3 时已经证明了它自身是一个非常好的安全通信协议，于是互联网工程组 IETF 在 1999 年把它改名为 TLS（传输层安全，Transport Layer Security），正式标准化，版本号从 1.0 重新算起，所以 TLS1.0 实际上就是 SSLv3.1。

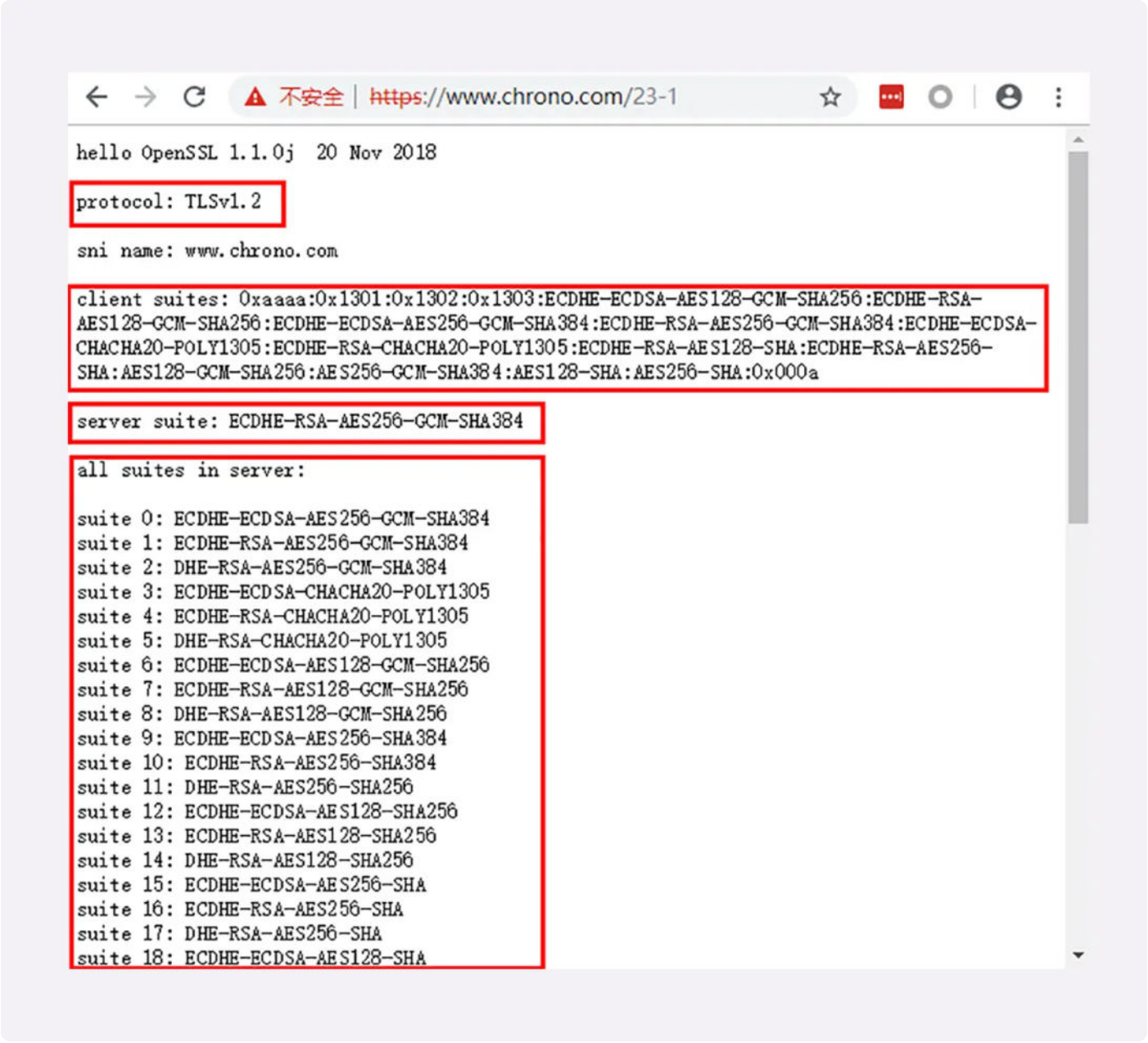
到今天 TLS 已经发展出了三个版本，分别是 2006 年的 1.1、2008 年的 1.2 和去年（2018）的 1.3，每个新版本都紧跟密码学的发展和互联网的现状，持续强化安全和性能，已经成为了信息安全领域中的权威标准。

目前应用的最广泛的 TLS 是 1.2，而之前的协议（TLS1.1/1.0、SSLv3/v2）都已经被认为是不安全的，各大浏览器即将在 2020 年左右停止支持，所以接下来的讲解都针对的是 TLS1.2。

TLS 由记录协议、握手协议、警告协议、变更密码规范协议、扩展协议等几个子协议组成，综合使用了对称加密、非对称加密、身份认证等许多密码学前沿技术。

浏览器和服务器在使用 TLS 建立连接时需要选择一组恰当的加密算法来实现安全通信，这些算法的组合被称为“密码套件”（cipher suite，也叫加密套件）。

你可以访问实验环境的 URI“/23-1”，对 TLS 和密码套件有个感性的认识。



你可以看到，实验环境使用的 TLS 是 1.2，客户端和服务器都支持非常多的密码套件，而最后协商选定的是“ECDHE-RSA-AES256-GCM-SHA384”。

这么长的名字看着有点晕吧，不用怕，其实 TLS 的密码套件命名非常规范，格式很固定。基本的形式是“密钥交换算法 + 签名算法 + 对称加密算法 + 摘要算法”，比如刚才的密码套件的意思就是：

“握手时使用 ECDHE 算法进行密钥交换，用 RSA 签名和身份认证，握手后的通信使用 AES 对称算法，密钥长度 256 位，分组模式是 GCM，摘要算法 SHA384 用于消息认证和产生随机数。”

OpenSSL

说到 TLS，就不能不谈到 OpenSSL，它是一个著名的开源密码学程序库和工具包，几乎支持所有公开的加密算法和协议，已经成为了事实上的标准，许多应用软件都会使用它作为底层库来实现 TLS 功能，包括常用的 Web 服务器 Apache、Nginx 等。

OpenSSL 是从另一个开源库 SSLeay 发展出来的，曾经考虑命名为“OpenTLS”，但当时（1998 年）TLS 还未正式确立，而 SSL 早已广为人知，所以最终使用了“OpenSSL”的名字。

OpenSSL 目前有三个主要的分支，1.0.2 和 1.1.0 都将在今年（2019）年底不再维护，最新的长期支持版本是 1.1.1，我们的实验环境使用的 OpenSSL 是“1.1.0j”。

由于 OpenSSL 是开源的，所以它还有一些代码分支，比如 Google 的 BoringSSL、OpenBSD 的 LibreSSL，这些分支在 OpenSSL 的基础上删除了一些老旧代码，也增加了一些新特性，虽然背后有“大金主”，但离取代 OpenSSL 还差得很远。

小结

1. 因为 HTTP 是明文传输，所以不安全，容易被黑客窃听或篡改；
2. 通信安全必须同时具备机密性、完整性、身份认证和不可否认这四个特性；
3. HTTPS 的语法、语义仍然是 HTTP，但把下层的协议由 TCP/IP 换成了 SSL/TLS；
4. SSL/TLS 是信息安全领域中的权威标准，采用多种先进的加密技术保证通信安全；
5. OpenSSL 是著名的开源密码学工具包，是 SSL/TLS 的具体实现。

课下作业

1. 你能说出 HTTPS 与 HTTP 有哪些区别吗?
2. 你知道有哪些方法能够实现机密性、完整性等安全特性呢?

欢迎你把自己的学习体会写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，也欢迎把文章分享给你的朋友。

课外小贴士

01 一个有趣的事实，当前所有 TLS 的 RFC 文档末尾数字都是“46”（2246、4346、5246、8846）。

02 除了 HTTP，SSL/TLS 也可以承载其他的应用协议，例如 FTP=>FTPS，LDAP=>LDAPS 等。

03 OpenSSL 前身“SSLeay”的名字来源于其作者之一“Eric A. Young”。

04 关于 OpenSSL 有一个著名的“心脏出血”（Heart Bleed）漏洞，出现在 1.0.1 版里。

05 OpenSSL 里的密码套件定义与 TLS 略有不同，TLS 里的形式是“TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384”，加了前缀“TLS”，并用“WITH”分开了握手和通信的算法。

06 另一个比较著名的开源密码库是 NSS（Network Security Services），由 Mozilla 开

发。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独购买本课程，你将得 20 元

生成海报并分享

赞 21　提建议

上一篇　22 | 冷链周转：HTTP的缓存代理

下一篇　24 | 固若金汤的根本（上）：对称加密与非对称加密

精选留言 (42)

写留言

djfhchdh

2019-07-19

机密性由对称加密AES保证，完整性由SHA384摘要算法保证，身份认证和不可否认由RSA非对称加密保证

作者回复: √

共 2 条评论>

88

-W.LI-

2019-07-25

老师好!有个问题，之前调用第三方的支付走https协议都需要本地配置一个证书。为啥最近有个项目也是用的https协议(url里会放token)。直接和http一样调用就好了，不需要本地配置证书了呢?

作者回复: 本地证书是用来做双向认证的，服务器用客户端的证书来验证客户端的证书。

通常我们上网是单向认证，只验证服务器的身份，客户端（也就是用户）的身份不用证书验证。

19

David Mao

2019-07-20

老师，请教一下，我们现在正在申请SSL证书，SSL证书有专门的机构颁发，文中老师提到HTTPS能够鉴别危险网站，防止黑客篡改，这些具体是怎么做到的呢？由专门机构颁发的原因是什么？谢谢老师。

作者回复: 如果网站是http而不是https，那么浏览器就会认为网站不安全，有风险。

如果证书内容不完善，或者被列入了黑名单，那么浏览器也可以提示用户有危险。

这些的关键都是证书，用证书里的信息，来验证网站的有效性、真实性。

因为证书用来证明网站的身份，就像身份证、学位证一样，如果随便颁发，那么它的可靠性就得不到保证，所以必须要由指定信任的专门机构来颁发，由ca来“背书”，保证证书和它关联的网站是安全可靠的。

19

彩色的沙漠

2019-07-19

1、HTTPS相对于HTTP具有机密性，完整性，身份认证和不可否认的特性,HTTPS是HTTP ov

er SSL/TLS,HTTP> HTTP over TCP/IP
2、实现机密性可以采用加密手段，接口签名实现完整性，数字签名用于身份认证

作者回复: √

11

Danpier
2020-01-12

有个疑问，维基百科 OSI 模型图表把 SSL\TLS 归到第6层（表示层），文中说 SSL 属于第5层（会话层），这里是不是写错了？ 附：
https://en.wikipedia.org/wiki/OSI_model#Layer_6:_Presentation_Layer

作者回复: 这个分层没有统一的定论，ssl/tls在tcp/ip里属于应用层，但不能准确对应到osi的某一层，因为它即有会话功能，又有加解密表示。

我们只要会用、理解就行，不要过于拘泥于学术。

7

李海明
2019-11-08

1、https与http协议相比，最重要的是增加安全性，这种安全性的实现主要是依赖于两个协议底层依赖的协议是不同的，https在传输的应用层与传输层协议之间增加了ssl/tls,这就使得http在固有协议之上增加一层专用用于处理数据安全的工具。

2、机密性：数据使用非对称加密传输
完整性：数据用公钥加密，私钥解密，数据生成摘要算法，同步传输

作者回复:
1.正确。

2.机密性主要用对称加密实现，非对称加密虽然也可以，但是效率太低，不实用。

共 2 条评论>

5

纳兰容若
2020-11-04

老师您好
一直以来不太明白openssl的各版本，我看官网上还有2.0和3.0的，还有后面还有t、h、j字母

跟在后面，这些大概有什么区别，正常使用不知道选择什么版本好，老师有什么建议么
感谢老师回复

作者回复: OpenSSL的官网上对各个版本写的很清楚，最早它是由ssleay发展来的，所以就沿用了0.98的版本号，后来又有好几个系列，比如1.01、1.10等等，小版本号用字母表示。

目前OpenSSL准备跳过2.0，直接出3.0，我们用最新的1.11就好了，之前的版本都将不再维护。

4

lesserror
2019-12-11

老师，以下问题，麻烦解答：
1. 这就是 HTTPS 与 HTTP 最大的区别，它能够鉴别危险的网站?这个仅仅从浏览器弹出不安全的提示来说的嘛？或者说怎么个鉴别法?
2. 网站是否真实也无法验证。加了https的网站也有可能是钓鱼网站吧？也没法验证啊?

作者回复:
1.如果网站的证书不可信（过期、失效、被废除、伪造），那么就可以说明网站是不安全的，而http不能对网站有任何的认证措施。

2.证书有dv、ov、ev三种，能从ca的层面证明网站的所有者。

3.当然，如果网站故意作恶，https也无法制止，它只能证明网站确实是如证书所声明的，不是假冒的。

4

业余草
2019-08-05

老师，我的个人网站：https://www.xttblog.com 在mac上的谷歌浏览器最新版中控制台总是会报一个错误，而我已经是https了，这个问题，空扰了我很久

作者回复: 什么错误，说出来看看。

共 6 条评论>

4

何用
2019-07-22

P-256 是 NIST（美国国家标准技术研究所）和 NSA（美国国家安全局）推荐使用的曲线。而密码学界不信任这两个机构，所以 P-256 是有可能被秘密破解但出于政治考虑而未公开?

作者回复: 是的，可能有这个隐患，就跟des一样。

4

mini💝

2021-02-03

请问老师 端口的作用是什么呢？为什么http和https的默认端口是不一样的

作者回复: 这个属于tcp/ip层次的知识了，我简单说一下吧。

互联网上的机器都用ip地址来标记，但只有ip还不够，一台机器上会有很多不同的服务，为了区分，就要再加上端口，这样才能完整地标记一个网络服务。

比如有台主机的地址是10.1.1.1，它在端口22上开了ssh服务，21上开了ftp服务，80是http，443是https，这样客户端就可以用地址加上不同的端口去访问主机上的服务了。

http和https的默认端口都是国际标准化组织分配的，比如etcd就用了2379。

共 2 条评论>

3

蒋润

2019-09-24

老师你好 https能有效防止抓包然后篡改报文数据,防止xxs攻击吗

作者回复: https传输的内容是加密的，所以抓包后看不到明文，是无法窜改数据的。

但xss属于内容攻击，报文本身是合法的，所以它不能防止。

https只能保证数据传输安全，但在链接的两端不能提供保护。

共 2 条评论>

3

zhangdroid

2021-04-11

HTTPS：即HTTP over SSL/TLS，用来解决HTTP明文传输导致的不安全问题。流程大致为：
使用对称加密算法加解密报文，保证机密性；使用摘要算法保证数据完整性；使用证书CA来进行身份认证;而不可否认则由非对称加密算法来实现。由于非对称加密算法耗时比对称加密算法长，所以用非对称加密算法来加解密给报文加密的对称算法的秘钥：即使用公钥对对称加密算法秘钥进行加密，私钥用来相应地解密。

作者回复: good。

2

WL

2019-07-19

请问一下老师我这边用WireShark抓包，发现两个TLS请求和响应之间和两个HTTP请求和响应之间有很多个TCP的包，请问一下这些TCP的包是一个HTTP的响应没有发完后续一致在通过TCP包发HTTP响应的responseBody吗?

作者回复: 应该在wireshark里看一下这些tcp包的端口、发送方向，应该不是https相关的包，可以过滤一下试试。

https必须在ssl/tls握手之后才能发送http报文。

2

火车日记

2019-07-19

1 明文、不安全vs四个特性，端口80vs端口443，无加密解密流畅性vs一定的性能消耗
2 对称加密算法保证机密性，散列值算法保证完成性和安全性

作者回复: √

2

爱编程的运维

2021-09-07

像一般的web网站存储用户的密码，密码存在数据库表中都是加密后的，这个加密跟https中的加密有啥区别?

作者回复: 我们通常所说的加密和密码学里的加密不是一个概念。

密码的正式名称应该叫password，存在数据库里实际上是做了摘要hash，比如md5、sha1，不是密码学里的用密钥算法加密。

而https里的加密是真正的密码学，里面有对称算法、非对称算法、摘要算法、证书等等，非常复杂。

可以再看后面的课程来加深理解。

海绵薇薇

2021-06-29

老师好，我想问下，在HTTPS协议上传输的报文，是怎么被缓存的，因为传输的内容应该都是加密的，那么如何做到If-Match或者If-Modified-Since判断的呢？CDN可以解析加密过后的内容吗？

作者回复:

1.https传输过程加密，但在两边是解密的，所以可以看到头字段，然后再走正常的判断处理流程。

2.如果cdn有证书和私钥，就可以解密数据，否则就只能做透明代理，在tcp层面转发。

1

钱

2020-03-30

1：你能说出 HTTPS 与 HTTP 有哪些区别吗?

正如文中所言HTTPS比HTTP多了一个S，这个S代表安全，是基于SSL/TSL实现的，SSL/TSL是专门用于安全传输的，具体咋实现的比较复杂还没弄明白，主要就是各种加密算法的应用，后面继续看。

2：你知道有哪些方法能够实现机密性、完整性等安全特性呢?
这个问题不知如何回答，不会，看答案如下。
机密性由对称加密AES保证
安全性由SHA384摘要算法保证
身份认证由RSA非对称加密算法保证
不可否认由RSA非对称加密算法保证
符合以上四点的才算是安全的通信方式，实现安全性看样子很不容易啊！
这些加密算法，他的发明者是否比较容易破解呢？还是说加密之后即使是发明者也无能为力，那如果解密的东西丢啦咋弄?

作者回复: 加密算法设计出来就是要任何人都无法破解的，否则就是有后门。

这个是密码学的基本原则，必须有密钥才能解密，至于如何保管就是另外的事情。

1

锦

2019-07-19

老师好，有几个问题请教下：“收发报文不再使用 Socket API，而是调用专门的安全接

口。”这个安全接口是什么呢？另外SSL/TLS运行在第五层，通讯不走下层TCP/IP的话，怎么把消息发到交换机呢？

作者回复: 可以看一下https的协议栈，它的下面还是tcp/ip。

拿OpenSSL来说，它提供了一系列的接口函数，比如SSL_read、SSL_write，加密后封装成tls记录，再交给tcp传输。

1

忧天小鸡

2021-12-23

HTTP是在TLS协议之后发起的协议？
以前我一直理解成HTTP后发起TLS协议，我记得看到的文章也是这样的，我以前是错误的？

作者回复: 看看后面的tls握手就会明白是怎么回事了。