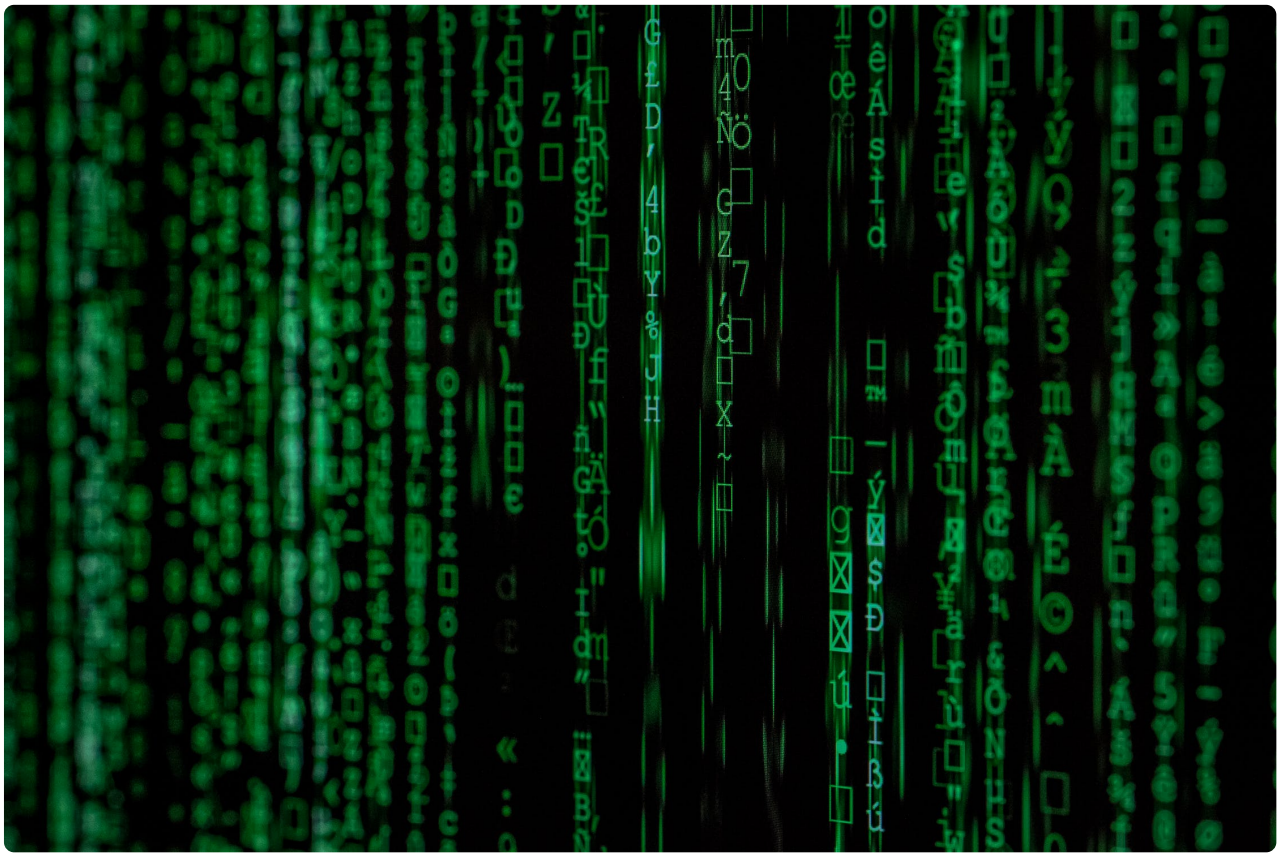
16 HTTPS 当家做主

更新时间：2020-01-16 20:32:15



> 世上无难事,只要肯登攀。——毛泽东

前言

上一篇文章中，我们简单的介绍了加密相关的基础概念，这些内容是理解HTTPS的基础。我们在这篇文章中就给大家展示一下HTTPS是如何通过加密算法来保证通信的安全，以及在Nginx中如何部署一个HTTPS网站。

TLS/SSL加密

互联网加密通信协议的历史，是伴随着互联网产生和发展的。

最开始，网景公司NetScape设计了SSL(Secure Sockets Layer)协议1.0版本，但是该版本并未发布。

接着， NetScape公司发布SSL 2.0版，很快发现有严重漏洞。

真正得到大规模普及使用的是SSL 3.0版。

随着互联网对安全性能的要求越来越高，互联网标准化组织ISOC接替NetScape公司，发布了SSL的升级版 TLS 1.0(Transport Layer Security)版。

在2006年和2008年， TLS进行了两次重大升级，分别为TLS 1.1和TLS 1.2。

TLS 1.0通常被标示为SSL 3.1， TLS 1.1为SSL 3.2， TLS 1.2为SSL 3.3。所以我们可以认为TLS和SSL是相同的，只不过二者的对应的版本号不一样而已。

HTTPS就是使用了TLS的HTTP，可以保证我们传输的信息安全。

HTTPS加密流程

HTTPS 使用 TLS/SSL 进行加密。在真正进行通信之前，客户端和服务器会进行三次通信(类似 TCP 的三次握手)，用于协商后续通信过程中使用的加密算法。我们简单的介绍一下:

> 第一回合： 客户端向服务器发送一个请求，这个请求有一个专业的名称叫做 ClientHello ,顾名思义，这是客户端向服务器打招呼呢。客户端会告诉服务器自己支持哪些加密算法，支持的加密协议版本，并且会发送给服务器一个随机数(记住这个随机数，它是用来生成密钥的)。
>
> 第二回合： 服务器收到客户端的 ClientHello 请求之后，要非常礼貌的回一句 Hello ，这就是 ServerHello 。服务端会看一下和客户端支持的协议版本是否相同，然后 确定 加密算法，并且给客户端发送一个随机数（记住：这是第二个随机数）。最重要的是，服务端会把自己的证书发送给客户端。
>
> 第三回合： 客户端收到服务端的服务端的证书之后，会验证证书是否是真实的（大家真的不用关心如何验证，我们不用关心）。然后再次给服务器发送一个随机数。（划重点了：这是第三个随机数）
>
> 第四回合： 这是服务器发送给客户端的最后一个消息(协商阶段的最后一个消息)，这个消息中包含了前两次服务器发送给客户端的所有内容的 hash 值。

到此为止，双方协商的阶段就结束了。客户端和服务器都可以通过上面说过的三个随机数来生成一个密钥，随后的所有通信都是使用密钥进行加密传输了，具体的流程就和 HTTP 通信的流程相同了。

Nginx 如何配置 HTTPS

第一步：编译 Nginx

如果我们想使用 Nginx 提供 HTTPS 服务，那么我们必须在编译 Nginx 的时候支持 SSL 功能。通过 -V 命令可以查看 Nginx 编译时候使用的参数。

```
[root@80d62660b37d key]# /usr/local/nginx/sbin/nginx -V
nginx version: nginx/1.16.1
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
configure arguments: --prefix=/usr/local/nginx
```

可以看到我们之前编译的 Nginx 并没有包含 SSL 功能，所以我们要重新编译 Nginx 。

```
[root@80d62660b37d nginx-1.16.1]# ./configure --prefix=/usr/local/nginx/ --with-http_s
sl_module
checking for OS
 + Linux 4.9.184-linuxkit x86_64
checking for C compiler ... found
 + using GNU C compiler
 + gcc version: 4.8.5 20150623 (Red Hat 4.8.5-39) (GCC)
checking for gcc -pipe switch ... found
checking for -Wl,-E switch ... found
checking for gcc builtin atomic operations ... found
checking for C99 variadic macros ... found
checking for gcc variadic macros ... found
```

紧接着 make 和 make install 就行了。

第二步：生成证书

大家可以使用 openssl 生成一对证书，这个过程就不再说了，可以借助搜索引擎查询。

```
[root@80d62660b37d key]# ll
total 8
-rw-r--r-- 1 root root 720 Jan 16 10:01 server.crt
-rw-r--r-- 1 root root 887 Jan 16 09:53 server.key
```

第三步：配置 conf 文件

第五步：启动 Nginx

常规启动 Nginx 就可以访问 HTTPS 了。

总结

这一章节原理性东西偏多，大家要仔细理解 SSL 通讯过程。

}

15 加密你的通话记录：从 HTTP 到 HTTPS

17 什么是正向代理？什么又是反向代理？