第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

针对 SSL/TLS 的典型攻击

张 明 许博义 郭艳来

(信息系统安全技术重点实验室北京系统工程研究所 北京 100101)

摘 要 SSL/TLS 是网络上广泛使用的一个安全协议,它在 TCP 层和使用 TCP 的应用程序之间提供安全服务,能保证消息的保密性和完整性。SSL/TLS 协议的标准在不断地完善,但是针对 SSL/TLS 的攻击也在不断地出现。首先对 SSL/TLS 协议进行了介绍,其次重点分析了各种典型的针对 SSL/TLS 的攻击。针对 SSL/TLS 协议的攻击被分为 3 类:与机制有关的攻击、与实现有关的攻击、与信任模型有关的攻击。在每类攻击下,都给出了几个具体的实例。

关键词 SSL/TLS,攻击,机制,实现,信任模型

中图法分类号 TP393.08 文献标识码 A

Review of Typical Attacks on SSL/TLS

ZHANG Ming XU Bo-yi GUO Yan-lai

(National Key Laboratory of Science and Technology on Information System Security,
Beijing Institute of System Engineering, Beijing 100101, China)

Abstract SSL/TLS is a cryptographic protocol widely used on the Internet. It works on behalf of the underlying transport layer and encrypts the data of network connections in the application layer to provide confidentiality and integrity guarantees. The protocol standards of SSL/TLS are constantly improved, but there are also increasing attacks. We first introduced some basic knowledge of SSL/TLS, and then analyzed the typical attacks on SSL/TLS. Attacks are divided into three categories: attacks related to mechanisms, attacks related to implementations, and attacks related to trust models. For each category, several specific instances were presented.

Keywords SSL/TLS, Attack, Mechanism, Implementation, Trust model

1 引言

互联网无疑是 20 世纪最伟大的发明之一,它的出现给人类的生活带来了巨大的便利。早期的互联网仅仅是一个能进行文件传输的小型网络。进入 21 世纪,互联网的规模越来越大,互联网的应用也呈现爆炸式的增长,浏览新闻、网上购物、预订机票、查阅邮件等已渗入到我们的日常生活当中。随着对互联网依赖的加深,我们才意识到,互联网存在很多不安全的因素,互联网上的信息往往是透明的,它甚至对信用卡账户和密码都无法提供可靠的保护。为了保证网络上信息的安全性,Netscape 公司最先提出了安全套接层(Secure Sockets Layer,SSL)的概念。SSL 在 TCP 和使用 TCP 的应用程序之间提供安全服务,它为通信的双方提供加密等保护,确保消息的保密性和完整性。目前最常使用的是 SSL 的第三个版本——SSLv3,它对以前版本的 SSL 的缺陷进行了修复和完善[1,2]。传输层安全协议(TLS:Transport Layer Security)是在 SSLv3 的基础上经 IETF 标准化的版本,它与 SSLv3 非常接近,仅存在一些细小的差别,本文主要讨论 SSLv3 和 TLS,在不引起歧义的情况下,我们直接用 SSL 指代 SSLv3。SSL/TLS 最初的设计目标是保护 Web 会话的安全性,但是后来它也被广泛地用于保护电子邮件的传输、VPN 的认证和加密等多种类型的网络应用中。毫无疑问,SSL/TLS 已是目前互联网上使用得最广泛的安全性协议。虽然 SSL/TLS 为网络的传输提供了安全性保护,但这并不意味着互联网的安全问题已经得到了解决,事实上,保护互联网安全的协议还有很多,例如 IPsec、DNSsec、SET 等,但是即便这样,互联网的安全问题也不容乐观,针对安全协议的攻击层出不穷。本文首先对 SSL/TLS 的工作原理进行了介绍,其次重点讨论了针对 SSL/TLS 的典型攻击。

本文第 2 节对 SSL/TLS 协议的工作原理进行了介绍,重点介绍了 SSL/TLS 握手协议和 SSL/TLS 记录协议;第 3 节详细讨论了针对 SSL/TLS 的典型攻击,我们将针对 SSL/TLS 的攻击分为 3 类:与机制有关的攻击、与实现有关的攻击、与信任模型有关的攻击,对于每类攻击我们都给出了几个典型的实例;最后对全文的工作进行了总结。

2 基础知识

SSL/TLS 协议不是简单的单个协议,而是一个两层协议,它包含的协议规范有:SSL/TLS 握手协议、SSL/TLS 修改密码规范协议、SSL/TLS 警报协议、SSL/TLS 记录协议。SSL/TLS 握手协议用于协商加密算法、交换密钥等;SSL/TLS 修改密码规范协议用于更新 SSL/TLS 连接使用的密码

张 明(1990—),男,硕士生,主要研究方向为网络安全、入侵检测,E-mail:Mingle-Cheung@yeah.net;许博义(1964—),男,硕士,研究员,主要研究方向为网络安全;郭艳来(1989—),男,硕士生,主要研究方向为网络安全。

组，协议由一个仅包含一个字节（值为 1）的消息组成，用于通知对方改变连接状态；SSL/TLS 警报协议用于向对方传递与 SSL/TLS 相关的警报，它的消息由两字节组成，其中一个字节表示警报的级别，另一个字节表示警报的内容；SSL/TLS 记录协议对应用层数据进行压缩、加密等操作后交给 TCP 层进行传输。它们的层次关系如图 1 所示。

图 1 SSL/TLS 协议栈

从图 1 可以看出，SSL/TLS 握手协议、SSL/TLS 修改密码规范协议、SSL/TLS 警报协议在 SSL/TLS 记录协议之上，它们与 HTTP 协议的级别相同，它们的消息可由 SSL/TLS 记录协议处理后交给 TCP 层。但是，值得注意的是，SSL/TLS 握手协议的消息可以直接由 TCP 进行传输，因为在 SSL/TLS 握手协议完成之前，SSL/TLS 记录协议还不知道采取什么样的措施对上层数据进行处理。SSL/TLS 握手协议和 SSL/TLS 记录协议是最重要的两个协议。

SSL/TLS 握手协议在数据传输之前进行，由客户端和服务器之间交互的一系列消息组成，消息是一个简单的三元组格式：（类型，长度，内容）。握手协议完成后，表明双方已经为安全通信做好了准备，包括客户端和服务器之间的相互认证、协商加密消息所用的会话密钥、生成消息认证码（Massage Authentication Code，MAC）的算法。SSL/TLS 握手协议的整个过程由 4 个阶段组成：阶段 1，建立安全能力；阶段 2，服务器认证和密钥交换；阶段 3，客户端认证和密钥交换；阶段 4，完成。其中，阶段 1 用于协商建立安全连接的参数，阶段 2 和阶段 3 进行认证和密钥交换，阶段 4 验证认证过程和密钥交换是否成功。SSL/TLS 握手协议消息交换的详细过程如图 2 所示。

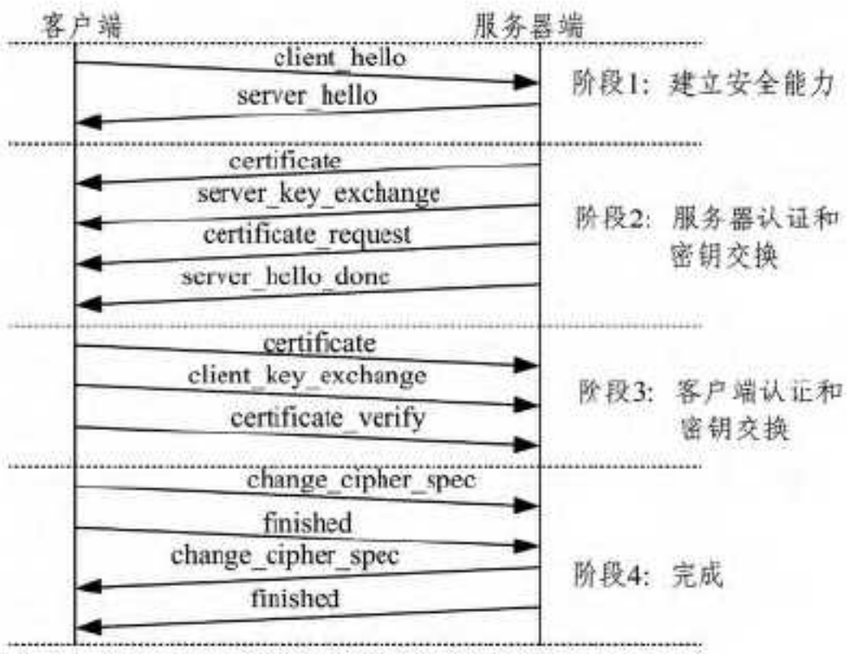

图 2 SSL/TLS 握手协议消息交换过程

SSL/TLS 记录协议指明了在传输应用层消息之前将对消息进行何种操作和格式处理。图 3(a)说明了 SSL/TLS 记录协议的整个操作过程。来自应用层的数据首先会被分段，然后进行压缩（也可以选择不压缩，主要取决于握手协议协商的结果），其次加上 MAC 并加密，最后加上 SSL/TLS 记录头。经过 SSL/TLS 记录协议处理后的最终数据单元会被放入一个 TCP 段中，然后经过网络传输到达接收方，接收的数据经过解密、验证、解压、重组的逆向操作后传递给应用层的用户，至此，一个完整的传输过程已完成。

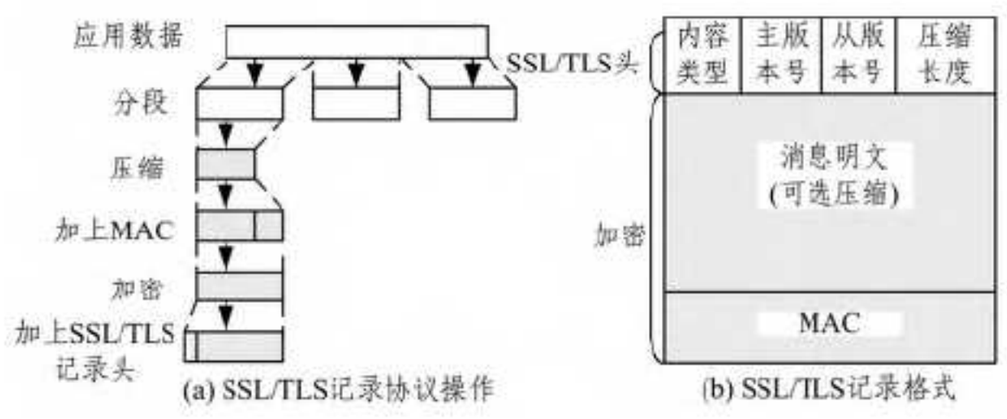

图 3 SSL/TLS 记录协议

应用层的数据经过 SSL/TLS 记录协议处理后格式如图 3(b)所示，处理后的数据可分为两个部分：SSL/TLS 头和加密部分。SSL/TLS 记录头由内容类型、主版本号、从版本号、压缩长度 4 个字段组成。

3 针对 SSL/TLS 的典型攻击

针对 SSL/TLS 的攻击非常多，这些攻击要么欺骗客户端去获取用户的敏感信息，要么欺骗服务器去进行非法的访问，但是在这里我们不是简单地将 SSL/TLS 攻击分为针对客户端的攻击和针对服务器的攻击，而是将 SSL/TLS 的攻击分为 3 类：与机制有关的攻击、与实现有关的攻击、与信任模型有关的攻击。我们认为这样的划分比较合理，SSL/TLS 协议与其他的互联网协议一样，也是由一组规范和机制组成，再完美的机制也可能存在薄弱环节，无可避免地给攻击者留下可趁之机，把从 SSL/TLS 的设计机制上找漏洞进而进行攻击的一类行为称作与机制有关的攻击；但是，尽管 SSL/TLS 的某些设计机制可能不存在缺陷，但是在实际实现中，由于某些策略暂时无法实现，进而采取了折衷的策略，攻击者就有可能从具体的实现上找漏洞进行攻击，我们把这类攻击称作与实现有关的攻击；SSL/TLS 协议是建立在信任模型（即依靠第三方机构给客户端或服务器签发证书）的基础之上的，因此，如果信任模型存在问题，整个 SSL/TLS 的安全性就无从谈起。糟糕的是，很多攻击者意识到从 SSL/TLS 的机制上进行攻击存在一定的难度，如果把目光转向信任模型可能会达到事半功倍的效果，他们进行了诸如伪造证书等攻击，我们把这类攻击称作与信任模型有关的攻击。

3.1 与机制有关的攻击

本节讲述了 3 个与机制有关的攻击，分别为：SSLstripping Attack、Cross-Protocol Attack 和 Renegotiation Attack。具体细节如下。

3.1.1 SSLstripping Attack

在 2009 年的 Blackhat 会议上，一种新的针对 SSL/TLS 的攻击被纰漏出来，它就是 SSLstripping Attack[3]。简言之，SSLstripping Attack 将 HTTPS 协议的“S”剥除（strip）掉，使用户的敏感信息通过 HTTP 协议传输，最终达到窃取用户信息的目的。

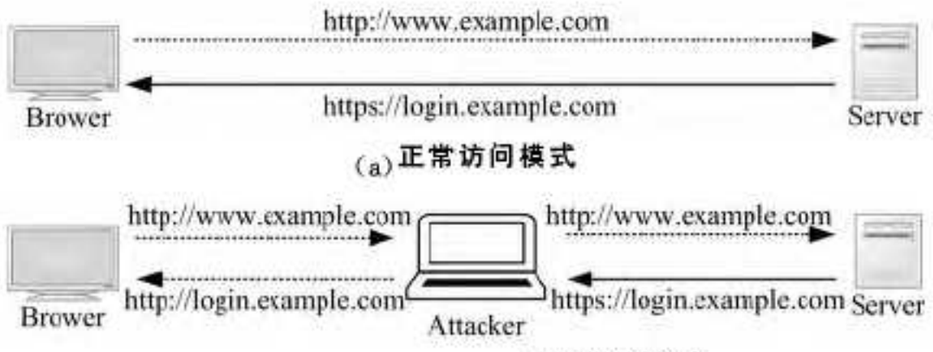

图 4 SSLstripping Attack 示意图

图4通过对比的方式展示了SSLstripping攻击的过程。在正常模式下(见图4(a)),用户访问某个站点,例如http://www.example.com,如果需要登录等操作,服务器会返回一个通过HTTPS协议加密过的页面,例如https://login.example.com,接下来用户会填写用户名密码等,由于整个消息是通过HTTPS协议传输的,因此用户的敏感信息会得到保护。而在SSLstripping Attack模式下(见图4(b)),用户与服务器之间的交互信息都会被攻击者截断,攻击者会将用户的请求发送给服务器,一旦服务器返回以HTTPS协议加密的页面时,攻击者就会将该页面劫持,然后通过HTTPS协议给用户发送一个相同或相似的页面,接下来的情形可想而知,当用户在http页面上填写信息时,这些信息都赤裸裸地暴露给了攻击者。

通过以上过程可以看出,SSLstripping Attack是中间人攻击(Man in The Middle,MITM)的一种模式,它在技术上并没有很高的要求,而是通过用户的访问习惯等将用户引入了攻击页面。如果用户在访问敏感页面时都会手动键入以https开头的地址,或者观察地址是否以https开头,那么这种攻击就不会得逞;相反,如果用户常常通过导航等链接访问敏感页面时,往往会被引入攻击的陷阱。文献[4]设计了一种通过视觉线索(其称之为SSLight)的方法来帮助用户判断所访问的页面是否具有不安全因素,进而抵御SSLstripping攻击。Fairweather等[5]对SSLight进行了实现,他们设计了一个基于SSLight的Chrome插件来保护用户免受SSLstripping的攻击。文献[6]指出用户受SSLstripping攻击的一个重要原因是他们经常依靠服务器导向https页面,而不是显式地访问https页面,文中利用浏览器的历史记录对经常访问的https站点建立访问模式,如果用户某次的访问行为与访问模式不相符,则认定为可能受到了SSLstripping攻击,很显然,这种方法有一定的局限性,如果浏览器没有存储历史记录,或者用户经常访问新的https站点,则无法建立访问模式。

3.1.2 Cross-Protocol Attack

文献[7,8]描述了一种适用于所有TLS版本的Cross-Protocol攻击,它可视为针对SSLv3的Wagner and Schneier Attack[1]的扩展。从SSL/TLS的工作原理可以看出,客户端和服务器可能支持不同种类的密钥交换算法,Cross-Protocol Attack通过对密钥交换参数进行错误的解释来达到攻击的目的。一种典型的情形是,如果服务器支持ECDH(Elliptic Curves Diffie-Hellman)密钥交换算法,客户端支持DH(Diffie-Hellman)密钥交换算法,攻击者会模仿服务器的行为,将ECDH参数解释为普通的DH参数,从而取得客户端的信任。但是这种情形对开源的服务器并不适用,因为它们不支持ECDH密钥交换算法[7]。从以上分析可以看出,Cross-Protocol攻击通过模仿服务器的特征来取得客户端的信任,进而以服务器的身份与客户端通信来达到窃取信息的目的。文献[9]中描述的Multi-Protocol Attack也是利用不同协议之间交互引起的漏洞来进行攻击,与Cross-Protocol Attack类似。

3.1.3 Renegotiation Attack

在SSL/TLS协议中,通信双方在已建立安全连接的情况下,可以重新协商密钥参数,然后使用新的密钥来加密消息进行通信,密钥协商过程必须在之前建立的安全通道中进行,但是这样并不能保证整个重协商过程不会被攻击者利用。重协商攻击(Renegotiation Attack)[10-12]就是利用重协商的漏洞来模仿客户端的特征,然后去欺骗服务器的一种攻击行为。

Renegotiation Attack的原理如图5所示。攻击者首先会通过SSL/TLS握手协议向服务器发起建立SSL/TLS安全连接的请求,在安全通道建立之后,攻击者可以与服务器之间进行一些常规的通信。当攻击者嗅探到客户端将要向服务器发起建立安全连接的请求时,它会劫持客户端发送的握手包,然后通过自己先前建立的安全通道发送给服务器。由于握手包是通过攻击者的安全通道发送的,服务器在接收到握手包后,就会误认为这个握手包是由攻击者发送的,即认为攻击者发起了一个重协商的请求。由于握手包本来来自客户端,这样以来,服务器就会误认为攻击者具有客户端的某些权限,接下来,客户端发往服务器的流量就会被攻击者插入具有某种企图的流量,而服务器又盲目地相信这些流量都是合法的请求,进而返回一些攻击者期望的内容。

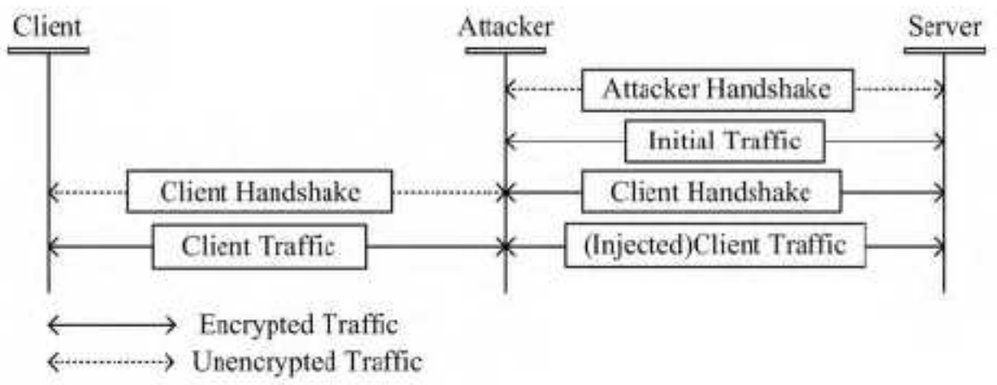

图5 Renegotiation Attack原理

RFC 5746[13]对两种抵抗Renegotiation Attack的方法进行了标准化,它们是Signaling Cipher-Suite Value(SCSV)和Renegotiation Information Extension(RIE)。文献[14]对SCSV/RIE的安全性进行了证明。

3.2 与实现有关的攻击

本小节给出了两个攻击实例,PRNG Attack和OpenSSL Heartbleed Attack,它们都是从实现的漏洞上对SSL/TLS进行攻击的。

3.2.1 PRNG Attack

SSL/TLS协议的设计多处使用了随机数,但在实际实现中尚不能得到真正意义上的随机数,目前通常采用伪随机数发生器(Pseudo Random Number Generator,PRNG)生成近似随机的数。伪随机数由于缺乏随机数不可预测的性质,常常成为攻击的目标。我们把利用一定的手段对PRNG生成的随机数进行预测的行为称作PRNG攻击。早期版本(低于1.22)的Netscape浏览器使用的PRNG就经常成为攻击的目标,攻击者可以成功预测SSL/TLS的会话密钥[15,16]。2008年,Debian操作系统的升级导致其OpenSSL所使用的伪随机数的随机性大大降低,结果使得预测签发TLS证书的私钥成为可能[17-19]。Heninger等[20]对整个Internet范围内使用SSL/TLS的服务器进行了调查,结果发现使用伪随机数导致脆弱性的现象非常普遍,0.75%的SSL/TLS证书由于密钥生成时熵空间不足而出现了共享密钥的情况,0.50%的使用SSL/TLS的主机的RSA私钥可以被恢复。

3.2.2 OpenSSL Heartbleed Attack

OpenSSL是目前互联网上使用得最广泛的开源的SSL/TLS软件。披露于2014年4月的OpenSSL Heartbleed攻击引起网络安全界的轰动,很多人认为SSL/TLS真的已不再安全。OpenSSL Heartbleed Attack充分利用了OpenSSL的一

个扩展功能——OpenSSL Heartbeat Extension[21]的一个漏洞。Heartbeat Extension 作为 SSL/TLS 的一个扩展功能，已被写入 RFC 6520[22]。Heartbeat Extension 用于测试当前的 SSL/TLS 是否处于活跃状态，避免了使用重协商的机制。Heartbeat Extension 的原理非常简单，通信的某一方发送一个 Heartbeat Request 消息，消息由负载和负载长度两部分组成，负载通常是一个文本字符串，负载长度作为附加字段对整个负载的长度进行说明；接收方收到 Heartbeat Request 消息后，原则上应该将该消息的负载原封不动地返回给发送方，但是接收方在返回负载时不是根据负载的实际大小返回内存中的内容，而是根据消息中负载长度字段指明的大小返回内存中的内容[23]。Heartbleed Attack 巧妙地利用了 Heartbeat Extension 的一个漏洞，如果发送方在发送 Heartbeat Request 消息时，设定一个与负载实际长度不相符的负载长度字段，比如远远大于负载的实际长度，那么接收方在根据负载长度字段的值返回内容时，就会将内存中的额外内容返回给发送方，而额外内容中可能包含用户名、密码等敏感信息。图 6 形象地展示了 Heartbleed 攻击的原理。

图 6 Heartbleed 攻击原理

虽然 Heartbleed[24-26]在披露后很快得到了修复，但是它影响的范围非常广，充分说明一个细微的疏忽很可能带来非常大的危害。

3.3 与信任模型有关的攻击

本节讲述了 3 个与信任模型有关的攻击，它们是：Visual Spoofing Attack、Compelled Certificate Creation Attack 和 MITM Attack with Forged Certificate。具体细节如下。

3.3.1 Visual Spoofing Attack

为了对用户的隐私和敏感数据进行保护，除了可以采用技术手段外，还可以通过应用程序给用户必要的安全提示。日常应用中，我们最常见的是浏览器对 https 站点提供的安全性提示。当访问一个 https 站点时，浏览器通常都会有以下两点提示：

- URL 以 https 开头；
- 在浏览器的某个位置（最常见的位置是地址栏）会显示一个挂锁图标，当点击这个挂锁时，会弹出与证书相关的信息。

可能某些浏览器还有其他的安全提示，例如，有的浏览器可能会将网址与对应的机构或公司挂钩，并将机构或公司的名字显示在浏览器的某个位置上[15]。

安全提示给用户判别网站的真伪提供了帮助，但是另一方面它也引入了安全隐患。攻击者可能通过伪造安全提示来欺骗用户，这就是所谓的视觉欺骗攻击（Visual Spoofing Attack）。进行视觉欺骗的手段很多，例如，攻击者可以通过图片的形式伪造一个挂锁或是证书，对 URL 的字母进行微小的变化以使用户访问的网址发生变化等[27]。

从以上分析可以看出，视觉欺骗攻击充分地利用了用户警惕性不够或是缺乏安全知识这一软肋。要达到视觉欺骗的目的，必须把用户引诱到某个伪造的站点，通常将引诱用户访问的行为称作 Mounting Attack[28]，引诱通常通过链接或广告投放等方式进行。通过以上分析可以看出，Visual Spoofing Attack 不是利用技术缺陷来进行攻击的，而是利用了用户的浏览行为或心理状态，所以防范这种攻击并不是十分的困难。文献[29]指出，可以在浏览器上设定一个信任域（Trusted Credentials Area），集中展示信任信息，但是应意识到，信任域也可能被攻击者做手脚。文献[30]指出，对于专业的使用者，可以通过浏览器的查看源代码的功能查询所访问站点的源代码，通过检查代码的关键位置来判定是否受到了 Visual Spoofing 攻击，但是这种方法局限性太大，例如，攻击者可以利用 JavaScript 改变浏览器按钮的功能，把用户引到一个仍然包含欺骗信息的页面。

3.3.2 Compelled Certificate Creation Attack

SSL/TLS 的一个重要的环节是认证，尤其是对服务器的认证。对服务器的身份进行认证意味着认证成功则客户端就会完全相信服务器并与之协商会话密钥。而认证通常采用的方式是由 CA（Certificate Authority）为服务器签发证书。在证书信任模型（Trust Model）中，根证书机构（Root CA）是被浏览器默认相信的 CA，它们不仅可以签发证书，还可以授权其他机构成为 CA，这些新授权的 CA 进一步也可以授权新的机构成为 CA，只要整个信任链不出问题（信任关系最终可以导向根证书机构），授权就可以持续进行。在信任模型中，任意一个 CA 都可以为任何一个网站签发证书。浏览器对 CA 完全的信任也为互联网的安全带来了隐患，试想，如果一个 CA 变节转而为不合法的网站签发证书，那么势必会有很多用户上当受骗。文献[31]讲述了一种强制 CA 生成不合法证书的攻击（Compelled Certificate Creation Attack），很多机构会强制 CA 为某些网站签发证书，这样用户就会消除戒心而随意地浏览这些网站，然而，情报机构可能早已经在监听或拦截用户的通信。

3.3.3 MITM Attack with Forged Certificate

通过伪造证书进行中间人攻击[32]是 SSL/TLS 面临的一类常见的攻击。一个典型的攻击场景如图 7 所示。

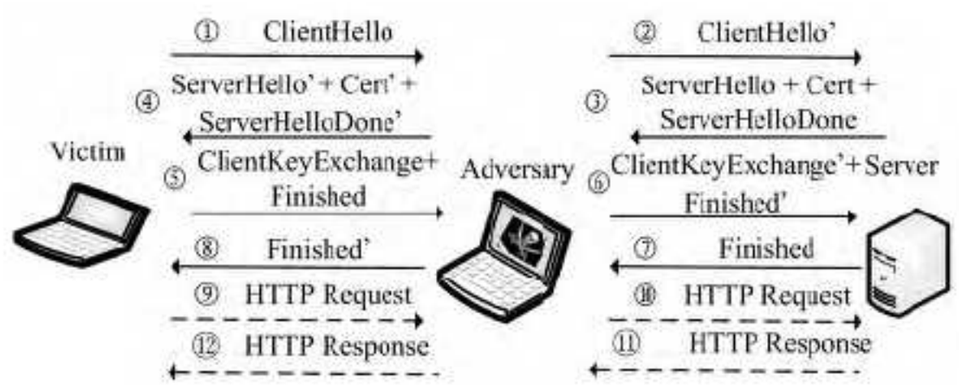

图 7 利用伪造的证书进行中间人攻击

图 7 中的攻击过程如下：攻击者首先将自己安插在 Victim 和 Server 的链路之上；当 Victim 向 Server 发起建立 SSL/TLS 连接的请求时（过程①），攻击者会对请求进行拦截，并用一个伪造的证书（记作 Cert'）对 Victim 进行回应（过程④）；如果伪造的证书成功地欺骗了 Victim，接下来就会完成

SSL/TLS 连接的建立(过程⑤和⑧),至此,攻击者已经成功地伪装成了 Server;同时,攻击者还会与 Server 建立 SSL/TLS 连接,把自己伪装成 Victim(过程②③⑥⑦)。从以上过程可以看出,攻击者与 Victim 和 Server 分别建立了一条 SSL/TLS 连接,而 Victim 和 Server 却以为他们之间只有一条连接,但是它们之间的通信早已经被攻击者监听、拦截或者是修改。

文献[33]指出,SSL/TLS 协议采用第三方机构签发证书本身就具有一定的复杂性,用户也只能求助第三方机构对证书进行检查来防止中间人攻击,文章提出了一种不借助于第三方机构而直接对证书进行检查来防止中间人攻击的方法——DVCert(Direct Validation of Certificates)。Holz 等[34]设计了一种检测中间人攻击的工具——Crossbar。Crossbar 首先在 Internet 上部署很多的测试点(hunters),然后让这些测试点与使用 SSL/TLS 的服务器握手,并将握手时获得的证书和产生的路由信息记录下来,最后交给汇总服务器(Central Server)分析,汇总服务器通过对证书进行统计分析以判断哪些证书是伪造的,并可通过路由信息判断攻击服务器的位置。现在有很多服务器都采用自签名的证书以节省使用 PKI 的开销,使用自签名证书必须建立在 TOFU(Trust On First Use)的基础之上,即浏览器第一次访问自签名的服务器时,必须信任该服务器发送的自签名证书,不难看出,使用自签名证书很容易受到中间人攻击。文献[35]采用 DoubleCheck 方法来解决这个问题,DoubleCheck 在获取自签名证书时分两个阶段,首先,远程服务器(Remote Server)从多个途径获取自签名证书,如果多途径获取的自签名证书信息都一致,则信任该证书,其次通过远程服务器将该证书发送给客户端,可见,DoubleCheck 方法是借助远程服务器来验证自签名证书的合法性的。

结束语 作为互联网上广泛使用的安全协议,SSL/TLS 一直是攻击者青睐的目标。本文对 SSL/TLS 所遭受的典型攻击进行了分析,这些攻击或与设计机制有关,或与实现细节有关,亦或与 SSL/TLS 使用的信任模型有关。攻击者最常采用的做法是从设计机制上找漏洞,然后设计攻击算法,然而这类攻击往往具有一定的难度;于是,很多攻击者转而从 SSL/TLS 的具体实现上找漏洞,然后发起攻击;由于 SSL/TLS 协议的设计遵循了 X.509 标准,因此信任模型也常常成为攻击的着眼点,这类攻击往往具有易于实现,且后果严重的特点。要防御针对 SSL/TLS 的攻击,除了不断完善 SSL/TLS 协议的标准外,还应向用户普及安全知识。

参 考 文 献

[1] Wagner D, Schneier B. Analysis of the SSL 3.0 protocol[C]// The Second USENIX Workshop on Electronic Commerce Proceedings. 1996:29-40

[2] Paulson L C. Inductive analysis of the Internet protocol TLS [J]. ACM Transactions on Information and System Security (TISSEC), 1999, 2(3):332-351

[3] Marlinspike M. New tricks for defeating SSL in practice [J]. BlackHat DC, February, 2009

[4] Shin D, Lopes R. An empirical study of visual security cues to prevent the SSLstripping attack[C]// Proceedings of the 27th Annual Computer Security Applications Conference. ACM, 2011:287-296

[5] Fairweather D, Shin D. Demo: A Chrome Extension to Prevent the SSLstripping Attack[OL]. http://cups.cs.cmu.edu/soups/2012/demo/demo03.pdf

[6] Nikiforakis N, Younan Y, Joosen W. HProxy: Client-side detection of SSL stripping attacks[M]// Detection of Intrusions and Malware, and Vulnerability Assessment. Springer Berlin Heidelberg, 2010:200-218

[7] Mavrogiannopoulos N, Vercauteren F, Velichkov V, et al. A cross-protocol attack on the TLS protocol[C]// Proceedings of the 2012 ACM Conference on Computer and Communications Security. ACM, 2012:62-72

[8] Jakovljevic A. Exploring cross-protocol attacks on the TLS protocol[J]. Katholieke University Leuven, 2012, 27(2):11-38

[9] Cremers C. Feasibility of multi-protocol attacks[C]// The First International Conference on Availability, Reliability and Security (ARES 2006). IEEE, 2006:8

[10] Rescorla E. Understanding the TLS Renegotiation Attack[J]. Educated Guesswork, 2009, 11(1):13-28

[11] Kurmus A. TLS renegotiation vulnerability (CVE-2009-3555) [J]. Common Vulnerabilites & Exposures, 2009, 35(5):35-55

[12] Zoller T. TLS/SSLv3 renegotiation vulnerability explained[J]. α-Secc University of Luxembourg, 2011, 1(1):7-13

[13] Rescorla E, Ray M, Dispensa S, et al. Transport layer security (TLS) renegotiation indication extension[J]. Internet Engineering Task Force(IETF), 2010, 18(5):3-5

[14] Giesen F, Kohlar F, Stebila D. On the security of TLS renegotiation[C]// Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. ACM, 2013:387-398

[15] Clark J, van Oorschot P C. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements[C]// 2013 IEEE Symposium on Security and Privacy (SP). IEEE, 2013:511-525

[16] Goldberg I, Wagner D. Randomness and the Netscape browser [J]. Dr Dobb's Journal-Software Tools for the Professional Programmer, 1996, 21(1):66-71

[17] Bello L, Bertacchini M, Hat B. Predictable PRNG in the vulnerable Debian OpenSSL package: the what and the how[C]// the 2nd DEF CON Hacking Conference. 2008

[18] Ahmad D. Two years of broken crypto: debian's dress rehearsal for a global PKI compromise[J]. Security & Privacy, IEEE, 2008, 6(5):70-73

[19] Yilek S, Rescorla E, Shacham H, et al. When private keys are public: results from the 2008 Debian OpenSSL vulnerability[C]// Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. ACM, 2009:15-27

[20] Heninger N, Durumeric Z, Wustrow E, et al. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices[C]// USENIX Security Symposium. 2012:205-220

[21] Seggelmann R, Tuexen M, Williams M. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension [J]. IETF draftietf-tls-dtls-heartbeat-00 (June 2010), 2012, 26(4):1-9

[22] Seggelmann R, Tuexen M, Williams M. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension [J]. IETF draftietf-tls-dtls-heartbeat-00 (June 2010), 2012

(下转第 419 页)

5 GIS 中数据偏向性保护实例分析

电网 GIS 平台作为提供电网资源空间信息管理及电网空间信息服务的企业级公共信息服务平台，是 GIS 系统的具体应用之一，电网 GIS 涉及的地理范围广阔，功能强大，该平台提供图形浏览、查询定位、矢量图形、电网专题图、空间分析、电网拓扑分析、瓦片地图、电网基础等服务。对认证、发电、输电、变电、配电、用电数据采用偏向性数据加密方法来保护，分析方法的效率如表 1 所列。

表 1 偏向性数据加密效率分析

数据类型	加密算法	时间/s
认证	RSA	20
输电	DES	32
变电	DES	39
配电	3DES	76
用电	AES256	45
总体数据包	DES	251

从表 2 可以看出，GIS 中数据的偏向性加解密方法所用时间明显小于传统的加解密方法，偏向性数据加密方法分类对数据处理，细化了数据类型，在加密过程中可以做到有的放矢，提高效率，并且有利于过程控制。

表 2 数据解密效率分析

数据类型	解密算法	时间/s
认证	RSA	20
输电	DES	31
变电	DES	40
配电	3DES	76
用电	AES256	48
总体数据包	DES	254

结束语 安全和高效是 GIS 提供服务的前提，GIS 的数据复杂多变，要求分门别类，为了在保证安全性的基础上，尽量提高数据处理效率，本文研究了一种偏向性数据加密方法，针对不同的应用，划分数据的分类，细化数据的流程，选择不同的加密方法，可以柔性地满足不同的要求。经过实际的应用分析，证明该方法在加密效率方面存在优势。

随着信息技术的发展和应用，人们在强调信息技术的应用的同时，更多地开始关注稳定和安全的保障，所以对数据加密的研究一直很炙手，只要有足够的时间保证，像 DES 加密算法都可以破解[7,8]，其他加密算法也一样，偏向性数据加密和保护方法在研究加密算法的同时更多地关注数据保护的技巧和方法，倡导一种模块化的数据加密和保护思想，优化数据处理流程，减少无用的数据处理。对偏向性数据保护思想在其他系统中的扩展需要继续延伸。

参 考 文 献

[1] 刘爱龙，张东，陈涛，等. 地图数据网络分发的混合加密算法[J]. 计算机工程，2008，34(18)：186-188

[2] 贾培宏，马劲松，史照良，等. GIS 空间数据水印隐藏与加密技术方法研究[J]. 武汉大学学报：信息科学版，2004，29(8)：747-751

[3] 蔡乐才. 应用密码学[M]. 北京：中国电力出版社，2005

[4] 谈娴茹. 基于 DES 和 RSA 的网络数据安全系统[J]. 中国民航学院学报，2003，21(A02)：133-136

[5] 林柏刚. 网络与信息安全教程[M]. 北京：机械工业出版社，2004

[6] 杨德保. 工科概率统计(第 3 版)[M]. 北京：北京理工大学出版社，2007

[7] 王立胜，王磊，顾训穰. 数据加密标准 DES 分析及其攻击研究[J]. 计算机工程，2003，29(13)：130-132

[8] Zadeh J A. Review of a Mathenmatical Theory of Evidence [J]. Al Magazine，1984，5(3)：81-83

(上接第 412 页)

[23] Wikipedia. Heartbleed[EB/OL]. [2014-6-14]. http://en.wikipedia.org/wiki/Heartbleed

[24] Durumeric Z, Kasten J, Adrian D, et al. The matter of Heartbleed[C]//ACM Internet Measurement Conference(IMC). 2014

[25] Momani E M H, Hudaib A A Z. Comparative Analysis of OpenSSL Vulnerabilities & Heartbleed Exploit Detection[J]. International Journal of Computer Science and Security (IJCSS), 2014, 8(4): 159

[26] Mpofu T P, Elisa N, Gati N. The Heartbleed Bug: An Open Secure Sockets Layer Vulnerability[J]. International Journal of Science and Research(IJSR). 2012, 2319(7064): 1470-1473

[27] Ye E, Yuan Y, Smith S. Web spoofing revisited: SSL and beyond [J]. Dartmouth Computer Science Technical Report, 2002, 417(36): 1-15

[28] Adelsbach A, Gajek S, Schwenk J. Visual spoofing of SSL protected web sites and effective countermeasures[M]// Information Security Practice and Experience. Springer Berlin Heidelberg, 2005: 204-216

[29] Herzberg A, Gbara A. Protecting (even) naive Web users, or: preventing spoofing and establishing credentials of Web sites [J]. Bar Ilan University, 2004, 7(18): 1-26

[30] Felten E W, Balfanz D, Dean D, et al. Web spoofing: An internet con game[J]. Software World, 1997, 28(2): 6-8

[31] Soghoian C, Stamm S. Certified lies: Detecting and defeating government interception attacks against ssl(short paper)[M]// Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2012: 250-259

[32] Ornaghi A, Valleri M. Man in the middle attacks Demos [EB/OL]. [2014-6-14]. http://www.smarttech.ie/wp-content/uploads/2013/12/bh-us-03-ornaghi-valleri.pdf

[33] Dacosta I, Ahamad M, Traynor P. Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties[M]// Computer Security-ESORICS 2012. Springer Berlin Heidelberg, 2012: 199-216

[34] Holz R, Riedmaier T, Kammenhuber N, et al. X.509 Forensics: Detecting and Localising the SSL/TLS Men-in-the-middle[M]// Computer Security-ESORICS 2012. Springer Berlin Heidelberg, 2012: 217-234

[35] Alicherry M, Keromytis A D. Doublecheck: Multi-path verification against man-in-the-middle attacks[C]// IEEE Symposium on Computers and Communications(ISCC 2009). IEEE, 2009: 557-563