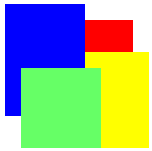


第6章 网络安全加固

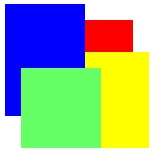
传送层安全加固

董建阔

南京邮电大学

关键知识点

- 传送层安全加固技术是基于TCP协议的端到端数据安全传送技术，它可以直接为两个网络应用(即用户)提供数据安全传送服务。
- SSL协议包括SSL记录协议和SSL握手协议。
- SSL记录协议主要作为SSL协议簇的数据传送协议。
- SSL握手协议主要作为SSL协议中用于安全连接建立的信令协议

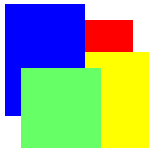


主要内容

- SSL协议概述
- SSL记录协议
- SSL握手协议

传送层安全加固技术

- 传送层安全加固技术是指在端到端的传送层(TCP协议提供的服务)进行安全加固的技术。
- 传送层安全技术是直接面向网络应用的安全加固技术。它是得到成功而广泛应用的一类网络安全加固技术。
- 现在广泛使用的传送层安全协议是指安全套接层(SSL)协议和在基于SSL协议进行标准化的传送层安全(TLS)协议，是对传输控制协议(TCP) 应用编程接口-套接字接口的安全加固，SSL协议最初是为安全访问万维网(WWW)应用而设计的。

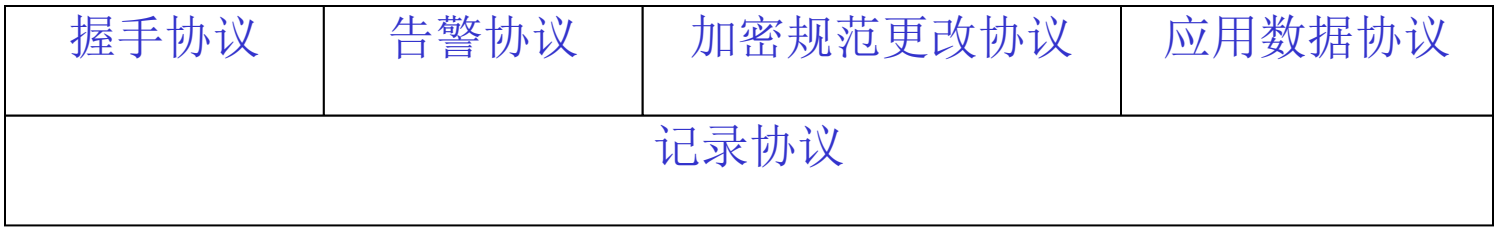


SSL协议概述

- 安全套接层(SSL)协议是一种在互联网TCP服务之上“端到端”安全加固协议。
  - 这是对TCP服务接口—套接层的安全加固
- 该协议允许客户端和服务器端应用在能够防范窃听、干扰和假冒报文的安全方式下进行通信。
- 现在通常使用的SSL协议的版本是SSL 3.0。
- SSL协议的主要优点是独立于应用协议，任何一个应用协议可以透明地加载到SSL协议之上。
  - 遵循了互联网的“端到端透明传送”基本原则

SSL协议的两个功能子层

- SSL协议包括两个层次：
  - 记录协议，直接基于某个可靠传送协议(TCP)之上, 用于封装SSL上层协议；
  - 握手协议、告警协议、加密规范更改协议、以及应用数据协议，被封装在SSL的记录协议内。



SSL协议分层结构

SSL协议的功能特征

- SSL协议提供的安全连接具有以下3种特征：
  - 连接是秘密的。SSL协议经过初始握手，协商了通信双方的共享密钥之后，就可以利用SSL记录协议对所有应用数据进行加密传递。SSL协议采用对称密钥加密算法。
  - 对等方的身份是可以验证的。SSL协议利用 公钥加密算法进行应用层通信双方的身份真实性验证。
  - 连接是可信任的。在SSL连接上传递的报文包括了报文验证码，可以用于报文完整性检查。

SSL协议发送和接收报文过程

- SSL协议发送和接收报文的处理过程。

图6.17 SSL协议发送和接收报文的处理过程

SSL协议之间的关系

- SSL协议完成握手协议交互，建立SSL安全连接之后，就可以传递应用层协议的报文。
- SSL协议发送和接收应用协议报文的过程，实际上就是SSL记录协议处理应用协议报文的过程。
- SSL记录协议也采用同样处理流程，对SSL自身的上层协议，包括握手协议、告警协议和加密规范更改协议，进行类似的处理。

SSL记录协议的功能和原理

- SSL记录协议利用SSL会话和连接中协商的加密算法、身份验证算法以及相应的密钥，对上层协议报文进行加密。
- SSL会话和连接必须通过SSL握手协议创建。SSL握手协议是SSL协议中关键的信令协议。
- SSL记录协议主要作为SSL协议中关键的数据传递协议。

SSL记录协议的报文格式

- 采用类似于C语言描述SSL记录协议报文字段格式 如下：

```
struct {
        uint8   major, minor;
    } ProtocolVersion;      //定义了兼容的协议版本范围
 enum {
        change_cipher_spec(20), alert(21), handshake(22),
        application_data(23), (255)
    } ContentType;          //定义了SSL记录协议的内容类型
```

SSL记录协议的报文格式(续1)

- SSL记录协议的明文报文格式：

```
struct {
    ContentType      type;       //已定义的内容类型
    ProtocolVersion  version;    //已定义的版本结构
    uint16           length;     //报文长度
    opaque fragment[SSLPlaintext.length];
} SSLPlaintext;
```

SSL记录协议的报文格式(续2)

- SSL记录协议的密文报文格式：

```
struct {
    ContentType       type;                 //已定义的内容类型
    ProtocolVersion   version;              //已定义的版本结构
    uint16            length;               //报文长度
    select (CipherSpec.cipher_type) {
        case stream:    GenericStreamCipher; //流密文类型
        case block:     GenericBlockCipher;  //块密文类型
    } fragment;
} SSLCiphertext;
```

SSL记录协议的报文格式(续4)

- 对于流加密数据应该采用以下结构类型：

```
stream-ciphered struct {
    opaque    content[SSLCompressed.length];
    opaque    MAC[CipherSpec.hash_size];
} GenericStreamCipher;
```

- 采用流加密算法进行加密的数据，包括压缩之后的数据content字段和报文验证码MAC字段，这种定义说明了必须先计算MAC值，然后再进行加密。
  - 注意：这种设置有可能遇到DOS攻击

SSL记录协议的报文格式(续5)

- 对于块加密数据应该采用以下结构类型：

```
block-ciphered struct {
    opaque    content[SSLCompressed.length];
    opaque    MAC[CipherSpec.hash_size];
    uint8     padding[GenericBlockCipher.padding_length];
    uint8     padding_length;
} GenericBlockCipher;
```

- 采用块加密算法进行加密的数据，包括压缩之后的数据content字段、报文验证码MAC字段、填充padding字段、以及填充长度padding_length字段。

SSL记录协议的处理过程

- 发送过程：SSL记录协议接收到应用层协议报文之后需要进行分段、压缩、生成报文验证码(MAC)、加密然后调用TCP这类可靠传送协议传递；
- 接收过程：SSL记录协议接收到TCP协议传递的SSL报文之后，进行解密、验证报文身份和完整性、解压缩和合段，然后提交给相应的应用层协议处理。

SSL记录协议的处理过程(续2)

- SSL记录协议处理应用报文的过程：

图6.17 SSL协议发送和接收报文的处理过程

SSL握手协议概述

- SSL握手协议的交互过程可以简单概括为：

(1) SSL客户端和服务器协商一个协议版本，

(2) 选择加密算法，

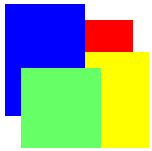
(3) 彼此验证身份(可选项)，

(4) 并且利用公钥加密技术生成共享保密字。

SSL的会话

- SSL会话是两个应用实体完成了身份真实性验证之后建立的一种关联，这种关联类似于互联网安全关联与密钥管理协议(ISAKMP)中定义的第一阶段ISAKMP安全关联。
- SSL会话协商了以下SSL会话属性：

（1）会话标识符，由服务器选择的一个任意字节序列，用于标识一个活跃的、可继续使用的会话。



SSL的会话(续1)

（2）对等方证书，存放应用协议交互的对等方公钥的X.509v3证书，该属性可能为空。

（3）压缩方法，SSL记录协议压缩数据使用的压缩算法。

（4）加密及双方身份验证规范。

（5）主保密字，在客户端和服务器之间共享的48字节的保密字。

（6）可继续标志，指示该会话是否可以用于发起一条新连接的标志。

SSL的连接

- SSL连接是在SSL会话之上，进一步协商应用实体交互双方采用的身份验证保密字、加密密钥、加密算法初始向量的一条安全连接。
- SSL连接类似于ISAKMP协议在第二阶段建立的安全协议关联。
- 一个SSL会话可以包括多条SSL连接，而服务器和客户机之间可以同时建立多个SSL会话，满足不同应用的安全通信需求。

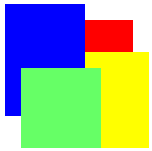
SSL连接的属性

SSL连接具有以下属性：

（1）服务器和客户机随机数，服务器和客户机为每个连接选择的一个字节序列，相当于服务器和客户机的一次性数。

（2）服务器写MAC保密字，服务器生成MAC时采用的保密字 。

（3）客户机写MAC保密字，客户机生成MAC时采用的保密字。



SSL连接的属性(续)

（4）服务器写密钥，服务器对数据加密的密钥；

（5）客户机写密钥，客户机对数据加密的密钥；

（6）初始向量，在CBC模式下使用块加密时，需要为每个密钥维护一个初始向量。

（7）顺序编号，对于每个连接，服务器和客户机都维护两个不同的报文顺序编号，用于服务器和客户机发送和接收报文。

SSL握手协议的交互过程

SSL握手协议的交互过程可以分成：

(1) 交互问候报文，协商会话和连接属性；

(2) 交互证书报文，验证对方身份；

(3) 交互加密规范更改报文，通知对方连接成功建立，开始传递应用数据。

SSL握手协议交互问候报文

SSL握手协议的问候报文交互过程如下：

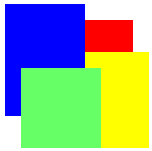
（1）SSL客户端发送客户端问候报文ClientHello；

（2）SSL服务器收到后必须用服务器问候报文ServerHello响应，否则SSL协议产生“致命错误”，SSL连接失败。

客户端问候报文

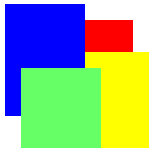
- 客户端发出的ClientHello报文的格式如下所示：

```
struct {
    ProtocolVersion   client_version;
    Random            random;
    SessionID         session_id;
    CipherSuite       cipher_suites<1..2^16-1>;
    CompressionMethod compression_methods<1..2^8-1>;
} ClientHello;
```



客户端问候报文(续)

- client_version表示客户端期望在该会话中使用的SSL版本；
- random表示客户端产生的随机数；
- session_id表示期望采用已经建立的会话标识符，如果该字段为空，则说明客户端期望新建立一个会话；
- cipher_suites表示一组可选的加密算法组合；
- compression_methods表示一组可选的压缩算法。



服务器响应报文

- 服务器响应的ServerHello报文格式如下所示：

```
struct {
    ProtocolVersion     server_version;
    Random              random;
    SessionID           session_id;
    CipherSuite         cipher_suite;
    CompressionMethod   compression_method;
} ServerHello;
```

服务器响应报文(续)

- server_version字段存放客户端建议最低的SSL版本号和服务器支持的最高SSL版本号；
- random表示服务器产生的随机数，不能与客户端产生的随机数相同；
- session_id表示该连接对应的会话标识符，如果客户端session_id为空，则服务器生成新的session_id；如果客户端session_id不为空，则服务器查找会话缓存，确实是否与该会话标识符匹配的、可继续使用的会话；
- cipher_suite表示服务器选择的一种加密算法组合，如果是继续使用的会话，则采用原来会话协商组合；
- compression_method表示服务器选择的一种压缩算法，如果是继续使用的会话，则采用原来会话协商值。

SSL握手协议交互证书报文1

- SSL握手协议的证书报文交互过程如下：

（1）在问候报文之后，如果需要验证服务器的身份，则服务器利用证书报文Certificate向客户端发送它自己的证书。

（2）如果服务器没有证书，或者它的证书只用于签名，则发送一个服务器密钥交换报文ServerKeyExchange，用于身份验证。

SSL握手协议交互证书报文2

（3）服务器也可以向客户端发送证书请求报文CertificateRequest，请求客户端返回证书。随后，服务器就可以发送服务器问候结束报文ServerHelloDone，指示服务器的问候报文握手阶段已经结束，服务器等待SSL客户端的响应。

（4）收到服务器证书请求后，SSL客户端必须发送证书报文Certificate，用于客户端的身份验证。

SSL握手协议交互证书报文3

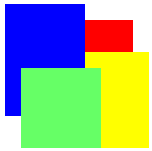
（5）如果客户端没有证书，则发送无证书告警报文no_certificate alert，随后发送一个客户端密钥交换报文ClientKeyExchange，用于客户端的身份验证。

（6）客户端也可以发送一个验证自己证书的签名报文CertificateVerify。

SSL握手协议交互证书报文4

（7）SSL客户端完成对服务器身份验证之后，就可以发送加密规范更改报文ChangeCipherSpec，启用与服务器约定的加密规范。

（8）客户端以新的加密算法和密钥发送完成报文Finished。



SSL握手协议的交互过程

M1: A → B: ClientHello

M2: B → A: ServerHello[, Certificate] [, ServerKeyExchange]

[, CertificateRequest], ServerHelloDone

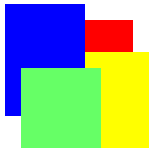
M3: A → B: [Certificate,] ClientKeyExchange[, CertificateVerify],

ChangeCipherSpec, Finished

M4: B → A: ChangeCipherSpec, Finished

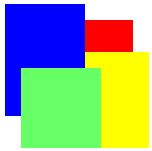
M5: A → B: ApplicationData

M6: B → A: ApplicationData



SSL握手协议的交互过程(续)

- 这里假定A表示SSL客户端，B表示SSL服务器；报文M2、M3和M4实际上包括了多个SSL报文，其中方括号“[ ]”内的SSL报文是可选报文。以上假定客户端先发送应用层数据。
- SSL客户端也可以利用已经建好的SSL会话创建新的SSL连接，这样，就不需要传递证书报文、密钥交换报文和证书验证等报文进行双方的身份验证，简化了SSL握手协议的交互过程。



握手协议的简化交互过程

- 简化的SSL握手协议：

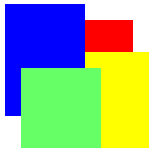
M1: A → B: ClientHello

M2: B → A: ServerHello, ChangeCipherSpec, Finished

M3: A → B: ChangeCipherSpec, Finished

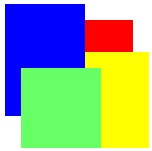
M4: A → B: ApplicationData

M5: B → A: ApplicationData



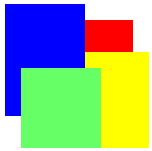
SSL握手协议的说明

- 我们在前面介绍客户端问候报文和服务器问候报文时已经介绍，利用已有的SSL会话创建的SSL连接只能沿用原来SSL会话协商的加密算法组合和压缩算法。
- 在新的SSL连接中可以利用新的随机数协商新的身份验证和加密密钥。



本节重点回顾

- SSL协议的概述
  - SSL协议在端到端的传送层进行安全数据传递。
- SSL记录协议
  - SSL记录协议可以作为SSL协议中关键的数据传递协议。
- SSL握手协议
  - SSL握手协议可以作为SSL协议中关键的信令协议。
  - SSL会话和连接的建立过程



思考题

(1) SSL协议提供哪些安全功能？SSL协议主要是针对哪类网络应用而设计的安全协议？

(2) SSL协议由几个协议构成？为什么说SSL协议可以透明地支持所有应用协议？

(3) SSL记录协议的主要功能是什么？它对于上层报文通常需要经过几个步骤的操作？

(4) SSL握手协议如何创建SSL的会话和连接？