可防冲突的专用协议栈①

张 林[1], 吴振强[2]

[1](商洛学院 数学与计算机应用学院, 商洛 726000)
[2](陕西师范大学 计算机学院, 西安 710062)

摘 要: 由于网络一般都使用公共的网络协议, 对于一些特殊的用户, 出于特殊的考虑, 对公共协议的安全性不够放心, 需要量身打造适合自己的专用协议, 以满足安全性和特殊性的需求. 设计了基于 CIPSO 标准改造的专用协议, 实现在现有网络环境下的正常通信, 满足根据安全级别等特定信息对数据流转进行控制的需求. 为避免因相似性造成的协议冲突问题, 提出防冲突标识的概念并设计出防冲突协商机制. 采用移植 LWIP 协议栈的方式实现该专用协议.

关键词: 专用协议; LWIP 协议栈; 防冲突标识; 协商

Conflict-Preventable Proprietary Protocol Stack

ZHANG Ling[1], WU Zhen-Qiang[2]

[1](College of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China)
[2](School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

Abstract: Public network protocol is generally used in network, but some special users, with special consideration, do not trust the safety of the public protocol. It needs customized proprietary protocols, to meet the needs of security and particularity. A proprietary protocols is designed based on CIPSO standard transformation, to realize the existing normal communication under the network environment, and meet other specific information requirements according to the security level to control the data flow. To avoid conflicts due to the similarity by the protocol, the concept of anti-conflict flag is proposed and the anti-conflict negotiation mechanism is designed. With the method of transplantation LWIP stack, the proprietary protocol is implemented.

Key words: proprietary protocol; LWIP stacks; anti-collision flag; consult

随着网络的快速发展, 人们对网络安全和信息安全的关注度不断提高, 网络通信的安全性需求也越来越高, 网络安全已经上升到国家战略的高度, 网络协议防护、安全网络协议也逐渐成为科研领域研究的热门话题.

在网络系统中, 为了保证数据通信双方能正确而自动地进行通信, 需要针对通信过程的各种问题, 制定一整套交互双方必须遵守的规则, 这就是网络系统的通信协议. 从通信的协议表现形式来看, 它规定了交互双方用于"交谈"的一套语义和语法规则, 以规范有关功能部件在通信过程中的操作[1,2]. 在目前的网络中, 有许多公共的协议, 比如最著名的 TCP/IP, 是因特网上大家都需要遵守的. 由于网络的开放性, 数据在传输的过程中, 必然存在安全性问题, 对于不同的用户, 对数据安全的要求不一样, 例如一些特殊机构, 对数据安全就有很高的要求, 他们不希望在公共的协议下来进行数据的传输, 于是就出现了对通信协议的个人定制, 也就是根据用户的安全需求和特殊要求, 制定一套符合个人要求的协议, 个人群体在网上使用, 与别人无关, 但又不能影响别人.

专用协议的定制必须满足个人要求, 但又得符合公共协议的标准, 当越来越多的专用协议出现以后,

① 基金项目:国家自然科学基金(61173190)
收稿时间:2015-01-08;收到修改稿时间:2015-03-02

协议之间不可避免的会发生冲突, 那就得能够及时对冲突进行处理. 文章在现有LWIP协议栈的基础上, 设计一种专用协议, 并给出了避免冲突的方法.

1 网络协议分析

1.1 协议结构

协议的关键在于对数据包结构的有效构造和正确解析, 根据对现有网络环境下多种网络系统的分析[3], 在一个特定的网络系统中, 大致可以将协议分为协议头和数据内容两部分, 结构如图 1 所示.

图 1 协议总体结构

与 OSI 的 7 层模型不同的是, TCP / IP 协议仅分为四层, 分别为应用层、传输层、网络层和网络接口层, 一个数据包的格式如图 2 所示.

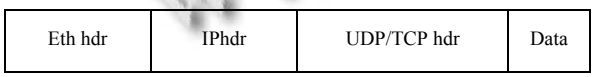

图 2 TCP / IP 协议数据包结构

(1)Eth hdr: 这是网络接口层对应的协议首部, 以 mac 帧头为例, 其中包含了三个字段, 前两个字段分别为 6 字节的目的 mac 地址和源 mac 地址, 第三个字段是 2 字节的类型字段, 说明了上层使用的是什么协议, 以方便接收端把收到的数据部分交给上层协议.

(2)IPhdr: 这是网络层对应的协议首部, 普通 ip 首部长度为 20 字节, 包含 4 位版本号、4 位首部长度、1 字节服务类型、2 字节总长度、2 字节标识、3 位标志、13 位片偏移、1 字节生存时间、1 字节协议、2 字节首部检验和以及 4 字节源 ip 地址和 4 字节目的 ip 地址等字段, 其中版本号指示了该协议是 ipv4 还是 ipv6.

(3)UDP / TCP hdr: 这是传输层对应的协议首部, 以 TCP 首部为例, 其中包含了 2 字节源端口号、2 字节目端口号、4 字节序列号、4 字节确认号、4 位首部长度、6 位保留段、6 位特殊指针标志、2 字节窗口大小、2 字节检验和以及 2 字节紧急指针等字段.

1.2 现有协议的局限性

在现有网络通信协议中, 信息传递主要依靠各层相应的字段来控制, 而这些字段是固定不变的, 无法在数据传输过程中根据特定的要求来控制通信的进行, 难以满足特殊用户特定的通信需求, 如一些军事部门、安全部门对涉密数据的流转控制, 需要根据安全等级、部门级别等信息确定路径上的信息是否允许被传递, 而诸如部门级别这一类的信息无法体现在现有公共网络协议中, 因此设计专用协议以满足特殊个体的特殊需求是有必要的. 若无对此类特有信息的控制, 就有可能导致单位内部信息或一些涉密信息在公共网络环境下流转, 使一些无关人员有可能得到这些信息. 为了保证专用协议能够满足公共网络协议, 文章在 CIPSO 的基础上来设计, 解决了专用协议在公共网络环境下的正常通信.

2 专用协议的设计

文章所设计的专用协议结构在目前国际流行的数据流安全标记绑定技术 CIPSO (Commercial IP Security Option)基础上对其字段含义进行了丰富, 这样设计而来的专用协议既拥有 CIPSO 原有的优势, 又可满足专用协议在现有网络环境下的正常流转.

2.1 CIPSO 简介

CIPSO 标准是基于 IPSO 协议扩展了安全标记的内容和处理办法, 这使得IPSO不仅在美国国防部网络和相关开源社区中被支持, 也使得它能够被应用到政府其他部门和商业环境中成为可能.

CIPSO 引入了解释域的概念, 域的定义由统一的部门负责, 如 DISA. 解释域是由多个安全选项的具体值构成的一个统一的系统, 域的标识被称为DOI标识[4]. 其数据包格式如图 3 所示.

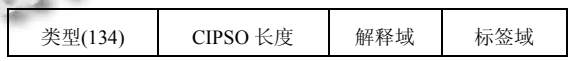

图 3 CIPSO 数据包格式

(1)类型: 该字段为 CIPSO 的类型字段, 用 1 字节表示, 对于 CIPSO 而言, 其值为固定值 134;

(2)CIPSO 长度: 该字段为 CIPSO 的长度字段, 用 1 字节表示, 由于 CIPSO 作为 IP 选项存在, 所以其长度最大值为 20, 最小值为 3;

(3)解释域: 该字段由多个安全选项的具体值组成, 标识安全域的唯一身份, 用 4 字节表示, 域的标识又被称为 DOI 标识, 解释域由 SDRC(Security Domain Registered Center)统一管理;

(4)标签域: 该字段表示数据包的安全标记信息,

包括范畴信息、机密级信息等, 用 0 到 14 字节表示, 可以定义不同的标签类型, 用于表示多种安全标记信息.

2.2 专用协议的结构

专用协议借用 CIPSO 的结构, 改造其标签域为自定义标签而成的, 协议总体结构如图 4 所示.

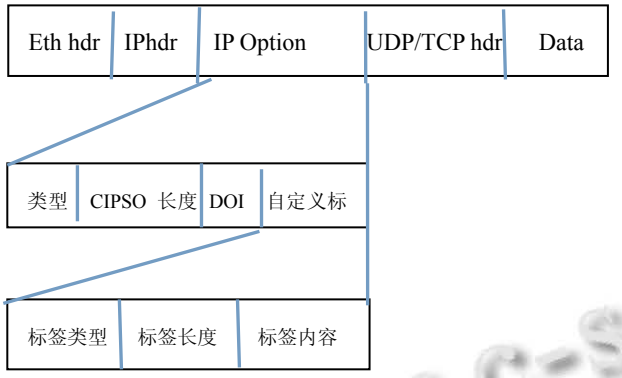

图 4 本文专用协议总体结构

改造后的标识的在整体上依然符合 CIPSO 的标准, 前三个字段与 CIPSO 相同, 第四个字段为自定义标签字段[5]. 在标准 CIPSO 中, 考虑到信息安全的通用性, 同时为了便于封装和解析, 标签域结构包含标签类型、标签长度、标签内容三个字段:

(1)标签类型: 该字段使用 1 字节表示, 其中 0 到 127 的定义是标准的标签格式, 具体格式可以在相应的 RFC 文档中找到, 大于 127 的标签类型则由 DOI 官方管理和定义, 目前 DOI 对标签的定义只有类型 1、类型 2 和类型 5 三种类型;

(2)标签长度: 该字段也使用 1 字节表示, 长度的值是标签内容的长度, 以字节为单位;

(3)标签内容: 在 DOI 官方定义中, 对标签的具体内容格式进行了规定, 仅包含机密级别和范畴等字段. 专用协议的提出即建立在对标签内容字段的具体含义的改造, 标签内容可根据特定部门的特殊需求进行修改和扩充, 不一定仅代表机密级别和范畴.

2.3 专用协议的冲突分析

随着各类不同应用的产生, 各个用户对通信安全性、可靠性要求的不断提高, 导致现有协议难以完全满足通信的要求, 专用协议的使用变得越来越广泛, 而这一情况将无可避免的造成协议在复杂的网络环境中的冲突[6]. 尽管专用协议都有属于自己的一套结构和规范, 但任何一个网络协议都必须符合通用协议的一些基本要求, 因而, 任何两个专用协议之间必然存在一定的相似性, 无论从内容还是格式而言, 这样的相似性都无法避免, 而任何一类相似性的存在都有可能造成通信的冲突, 继而影响到通信的正常进行. 在 CIPSO 基础上改造的专用协议在通信过程中就有可能与现有的 CIPSO 协议发生冲突, 导致通信的中断或者被拦截[7].

2.4 专用协议的识别

为了避免和解决冲突, 提高专用协议与普通协议之间、专用协议与其他专用协议之间以及专用协议自身的兼容性, 提出了防冲突标识(Anti-Collision Flag , ACF)的概念. 当使用专用协议的设备收到数据包时, 对该标识字段进行校验, 校验正确则表示该数据包为我们所需要的包, 反之则按正常包进行处理, 这个标识字段就是我们所说的防冲突标识. 为了提高其解决冲突的能力, 防止被误认为是攻击的数据包, 该标识字段的值不能一直固定, 当专用协议出现冲突后, 为了不影响其他用户使用, 专用协议主动修改自己的格式, 以避免冲突.

防冲突标识既是专用协议的标志, 也是解决冲突的方案. 为了便于组包和解析, 需要在上述提到的自定义标签类型定义的基础上对该类型字段进一步细化, 字段长度扩展为 9 位, 前 8 位代表标签的类型, 其最低位指示了是否有防冲突标识, 若该位值为 1, 则代表自定义标签中含有防冲突标识, 若为 0, 则表示没有. 标签长度字段缩减为 7 位, 由于 CIPSO 作为 IP 选项存在, 其长度最大为 20 字节, 因此 7 位的标签长度已足够表示, 不会影响到标签的正常表示. 在此基础上, 我们即可根据防冲突标识对协议进行识别, 含防冲突标识的 CIPSO 数据包即为我们定义的专用协议格式的数据包, 反之则表示不是.

3 动态防冲突标识协商

3.1 防冲突标识格式

为了保证专用协议的数据包能被正确识别, 我们在数据包中插入防冲突标识字段, 带有 ACF 字段的数据包基本组织结构如图 5 所示.

为了提高防冲突表示得可用性、可靠性和保密性. ACF 字段由标识值、随机数两部分组成.

(1)标识值(value): 该字段为 ACF 的标识值字段, 是不经常变化的固定字符串, 用 4 字节表示;

(2)随机数(rand_num): 该字段是是 ACF 的随机数

字段, 由随机数产生函数生成, 用 1 字节表示.

在客户端组装数据包前, 向标识协商系统请求防冲突标识的初始值, 即该标识的第一个字段, 在本文中我们将第一个字段称为固定属性部分. 然后调用随机数生成函数得到符合条件的随机数与标识值进行异或操作, 再级联上随机数即可作为专用协议的 ACF 字段.

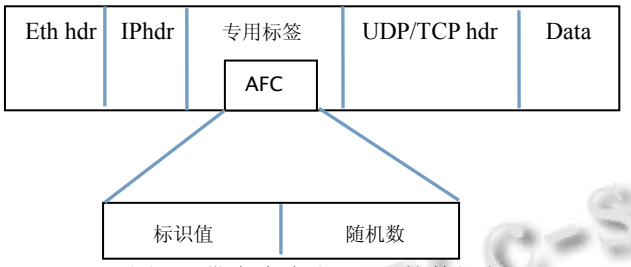

图 5 带专有字段 ACF 的数据包

由于 ACF 字段值会一直变化, 而固定属性部分不常变化, 所以在文章中取 ACF 字段的固定属性值作为专用协议的防冲突标识字段.

3.2 防冲突标识协商过程

专用协议防冲突标识协商是在使用专用协议的设备能够快速、有效识别专用协议格式的数据包的基础上, 针对专用协议发生冲突而做的专用协议防冲突标识动态协商. 协商应达到规避冲突、保证正常通信的效果[8,9].

3.2.1 网络内部协商过程

当单位内使用专用协议栈的设备第一次进行通信时, 设备会首先初始化专用协议防冲突标识字段, 即 ACF 字段固定属性值, 也就是说整个网络初始化所有设备和计算机使用的 ACF 字段的固定属性值部分是相同的[10].

当数据包在内部网络通信中发生冲突时, 我们以下发的形式更新网内所有设备和计算机中 ACF 字段固定属性值部分, 进行冲突的协商, 从而达到解决冲突, 使通信继续正常进行的目的.

3.2.3 跨网协商过程

当专用协议格式数据包在两个或以上网络中进行通信时, 由于权限的问题, 某一网络无法对另外其他任何一个网络下发 ACF 字段固定属性值, 因此, 我们需要通过另外一种方式来初始化和协商该部分的值. 即对客户端到服务端的有效路径上所有设备的 ACF 字段固定属性值部分进行初始化和修改.

当客户端第一次发起通信时, 我们通过路由确定出出一条可用的路径, 当客户端到服务端的路径确定后, 由专用协议防冲突标识协商系统发出一条防冲突标识初始化的命令, 将路径上所有设备和计算机的 ACF 字段的固定属性值部分设置为同一个值.

基于上述已初始化所有防冲突标识的路径, 客户端和服务端开始数据的发送和接收. 若在此过程中, 数据收发出现中断, 将采用以下方法进行路径的检查工作, 确定原因和解决方案:

(1)连通性检查: 通过网络诊断工具进行网络连通性的检查, 若网络本身不通, 则进行物理排查, 反之进行第(2)步操作;

(2)协商操作: 由管理员人为改变或由系统自动重新生成 ACF 字段固定部分, 并通过协商系统发出防冲突标识协商命令, 对上述路径上所有设备和计算机的 ACF 字段固定部分进行协商.

(3)字段升级: 若上述两种方式均不行, 则由单位间管理员私下对 ACF 字段进行协商, 统一升级字段, 以解决问题.

4 基于LWIP协议栈的专用协议的实现

4.1 LWIP 简介

LWIP (Light Weight Internet Protocol) 是一个 TCP/IP 协议簇的小型独立实现[11,12], 该协议簇由瑞士计算机科学院(Swedish Institute of Computer Science)的 Adam 等人开发. 选择该协议栈的理由如下: ①它是开源的, 便于我们根据专用协议的实际需求进行移植和修改; ②它的功能相比于其他开源协议栈是比较全的, 最新版本已经支持 ipv6; ③与硬件平台的无关性, 它提供简单且易于使用的底层硬件 API; ④它需要调用的系统函数的接口容易构造, 即提供了易用的与操作系统的 API. 协议栈整体框架有四层:

(1)传输层: 该层包含 TCP、UDP 两个模块, 实现了数据报文协议和传输控制协议;

(2)网络层: 该层包含 IP、ICMP 两个模块, 实现了 ip 数据包的发送、接收、分片、重组等功能, 同时还提供了传输控制报告、错误信息的功能;

(3)网络接口层: 该层包含了 ARP 模块, 完成了 ip 地址与 mac 地址之间的相互映射, 同时维护了一个 arp 缓存表, 保证网络传输的高效有序进行.

4.2 专用协议在 LWIP 中的实现

4.2.1 自定义标签结构的实现

具体代码如下:

```
struct cipso{
    unsigned char cipso_type ;
    unsigned char cipso_len ;
    unsigned long cipso_doi ;
    struct acf{
    unsigned short tag_type ;
    unsigned char tag_len ;
    unsigned char *data ;
                }
            }
```

4.2.2 套接字接口层

为了标识专用协议的特殊性, 我们对 LWIP 提供的套接字接口进行了封装, 统一在所有 socket 相关函数前添加了一个 s, 即 s_socket, 以示与普通协议的区别, 同时便于用户识别. 在此基础上, 我们还需要在 set_soket_opt()函数中添加入口 SPECIAL_IP, 使其能够传递自定义标签信息, 具体代码如下:

```
 switch ( level )
{
. . .
case IPPROTO_IP:
   switch ( optname )
     {
     . . .
     //以下入口为自定义标签选项特意增加:
case IP_SL_OPT:
        sock -> conn -> pcb.ip -> slopt = * ( ip_sl_opt * ) optval; //得到传入的自定义标签信息
LWIP_DEBUGF ( SOCKETS_DEBUG , ( " LWIP_setsockopt ( %d , IPPROTO_IP ,
            IP_SL_OPT , ..) -> %d \n " , s, sock -> conn-> pcb.ip -> slopt ) ) ;
        break ;
     }
. . .
}
```

4.2.3 传输层改造

给出其中几个主要的函数

①. assem_out_sp_buffer() 该函数为自行构造的函数, 用于构造自定义标签.

函数声明如下:

```
static void assem_out_sp_buffer ( struct tcp_pcb * pcb , struct pbuf  * p ,  ipX_addr_t * src , ipX_addr_t * dest , u8_t ttl , u8_t tos , u8_t proto ) ;
```

相关描述:

功能: 以指定方式将传入的信息构造成自定义标签

输入: @pcb: 存放自定义标签数据及其他信息的 pbuf

@p: 需发送的数据

@src: 源地址(组 ip 包头时所需项)

@dest: 目的地址(组 ip 包头时所需项)

@ttl: 生存时间(组 ip 包头时所需项)

@tos: 服务类型(组 ip 包头时所需项)

@proto: 上层协议(组 ip 包头时所需项)

输出: 无

②. reassem_out_sl_buffer() 该函数为自行构造的函数, 用于解析自定义标签.

函数声明如下:

```
Static void resassem_in_sl_buffer ( struct tcp_pcb * pcb ) ;
```

相关描述:

功能: 根据 tag_type 类型解析专用

输入: @pcb: 存放自定义标签数据及其他信息的控制块

输出: 无

③. tcp_send_empty_ack() 该函数用于发送 ack 信号. 为使其能够发送携带自定义标签的 ack, 需要增加代码.

④. tcp_output_segment() 该函数用户向 ip 层发送数据. 为使其能够发送携带自定义标签的 ack, 需要增加部分代码.

⑤. tcp_listen_input() 该函数用于处理接收到的 syn 数据包, 为使其能够解析携带自定义标签的 syn, 需要在源代码中回应 ack 函数之前增加代码, 该部分代码如下:

```
if  ( ip_data.special_label !=  NULL )
     {
          disassem_tcp_in_sl_buffer ( ( struct  tcp_pcb * ) pcb ) ;
```

```
        npcb -> slopt   =   pcb -> slopt ;
    }
```

根据 ip_data 中存放的 special_label 是否为空, 判断接收到的 syn 数据包是否有自定义标签, 若上述条件满足, 则进行自定义标签信息解析操作, 将得到的 pcb 中的信息赋给发送回应 ack 需要的 pcb.

5 系统测试

在测试中, 由于协议栈尚未以服务的形式嵌入到操作系统中, 因此协议栈的启动需手工进行. 当协议栈启动后, 向协议栈发送 ARP 请求报文, 若可以接收到响应报文, 则表明协议栈的 ARP 模块在功能上是正常的. 最易行的发送 ARP 请求的方式就是通过网络诊断工具, ping 协议栈已配置好的地址. 由于 ICMP 报文是封装在 IP 数据报发送的, 向协议栈发送 ICMP 查询报文, 若能接收到响应报文, 则表明协议栈的 IP 模块功能是正常的. 最便于操作的发送 ICMP 请求的方式也是网络诊断工具, ping 协议栈已配置好的地址. 若 ping 命令得到了响应, 说明收到了 ICMP 回送请求应答报文. 因此, ping 命令测试, 将可测试 ARP、ICMP 和 IP 三个模块的功能是否正常, 并可说明协议栈是否启动成功.

整个测试主要对网内套接字通信测试和异网环境下的数据传输测试. 测试结果证明, 经过改造以后的协议, 其 ARP、ICMP 和 IP 三个模块的功能都一切正常, 说明协议栈启动成功.

6 结语

文章提出了一种专用协议的结构, 并在 LWIP 协议栈中进行了实现, 同时提出了专用协议防冲突标识的概念, 可以有效处理协议之间的冲突. 通过测试, 该协议栈可以正常启动, 并且三个主要模块功能都一切正常. 这种方法完全可以作为一种私人订制的网络协议, 用以满足特殊用户的特殊需求. 当然文章所设计的专用协议还有一些需要改进的地方: 1)由于处理程序的增加, 改造后的协议栈性能相比原来的协议有所下降, 对协议性能的提高将是要进一步研究的内容. 2) 防冲突标识是本文的特色, 但冲突处理协商过程还可以进一步优化, 以提高其安全性和高效性.

参考文献

1 任午令,等.计算机网络技术与应用.杭州:浙江大学出版社, 2006.

2 冯博琴,等.计算机网络.第 2 版.北京:高等教育出版社,2004.

3 Kurose JF, Ross KW.计算机网络:自顶向下方法.陈鸣译.北京:机械工业出版社,2008.

4 Dunkels A. Design and implementation of LWIP TCP/IP stack [Technical Report]. Swedish Institute of Computer Science, Feb, 2001.

5 马相林.基于安全标记的区域边界访问控制技术研究[学位论文].郑州:解放军信息工程大学,2010.

6 ISO/IEC 15408-1-1999. Common Criteria for Information Technology Security Evaluation2.2.2004.

7 马新强,黄羿.基于安全标签的访问控制研究与设计.计算机工程与设计,2008,12:5432–5434.

8 Peyravian M, Jeffries C. Secure remote user access over insecure networks. Computer Communications, 2006, 29: 660–667.

9 Ray I, Kumar M. Towards a location-based mandatory access control model. Computers and Security, 2006.

10 杨玉佳.LwIP 在 μCOS-Ⅱ平台上的移植与应用[学位论文].成都:电子科技大学,2009.

11 Dunkeis A. Design and implementation of the LwIP TCP / IP stack. Sweden. Swedish Institute of Computer Science. 200l: 2l–30.

12 韩德强.基于“C / OS 一Ⅲ的 LWIP 协议栈的移植与实现. 电子技术应用,2013,39(5).