协议分析在入侵检测系统中的应用①

The Implementation Of The Protocol Analysis in The Intrusion Detection Systems

费洪晓 谢文彪 郭球辉 戴宏伟 裘方敏
(中南大学信息科学与工程学院 湖南长沙 410075)

摘要:本文在分析现有入侵检测系统基础上,设计了网络入侵检测系统框架,主要探讨了其中网络数据包捕获模块和网络协议解析模块的设计思想与实现过程。在数据包捕获部分设计中主要讨论了 Linux 下的 BPF 机制和 Libpcap 函数库,利用它们实现网络数据包的捕获功能;在协议分析模块中详细讨论了以太网、IP、TCP、UDP、ICMP 等协议的解析过程。测试结果表明能够对捕获的 TCP/IP 包进行有效地解码。

关键词:入侵检测 TCP/IP 协议 包捕获 协议分析

1 引言

当今在全球范围内,对计算机及网络基础设施的攻击行为,已经成为一个越来越严重和值得关注的问题。对网络安全问题的一种实用解决方法是:对已经建设的信息系统,按照一定的安全策略建立相应的安全辅助系统。IDS(Intrusion Detection System,入侵检测系统)就是这样一种系统。对入侵检测系统的研究,最核心的部分就是对入侵检测技术的研究。现在的入侵检测技术应用最多的是数据包模式匹配检测方法,但此方法已经暴露出很多不足。现在人们开始探讨协议分析技术在入侵检测中的应用,它是下一代入侵检测系统的关键技术之一[1]。它可以解决 IDS 领域长期以来的应用瓶颈问题:检测准确性以及大流量应用网络环境下的系统性能。新一代 IDS 将凭借此项最新技术,融合传统特征模式匹配技术的优点,开发出更加完善、优秀的入侵检测与防护系统[2]。

本文首先介绍了网络入侵检测系统基本框架,接着介绍了结合高速捕包技术的协议分析技术。主要分析了网络数据包的捕获及协议分析方法基本原理及解码方法。并且对其进行测试,结果证明能够对捕获的网络数据包进行较详细的解码。

2 网络入侵检测系统结构

网络入侵检测系统(network intrusion detection system,NIDS)通过监听共享网段,捕获该网段上的数据包并对其进行分析,从而剥离出一些与入侵特征相关的标志然后再将这些标志同现有的入侵特征进行模式匹配,从而检测出存在于该网络中的入侵活动,并且利用一些响应手段向网络管理员发出告警信息并采取相应的行动。设计网络入侵检测系统的几个关键因素在于数据包的捕获和正确的分析解码,对入侵特征的模式匹配以及确保入侵特征的及时更新等[3]。

网络入侵检测系统总体设计大致如图 1 所示。由图可以看出整个系统可以分为三个大的模块,根据其所处的位置可以将其命名为底层模块、中层模块及上层模块。本文主要对中层模块进行设计实现。

3 网络数据包的捕获及过滤

设计采用 libpcap(Packet Capture Library)来实现对网络数据包的捕获。用 BPF(Berkeley Packet Filter,伯克利数据包过滤器)来对捕获的数据包进行过滤。

libpcap 是一个与实现无关的访问操作系统所提供的分组捕获机制的分组捕获函数库,用于访问数据链路层。下面介绍 libpcap 的一些重要函数及基本调用。

① 基金项目:国家自然科学基金资助(60173041),湖南省自然科学基金资助(05JJ30119)

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

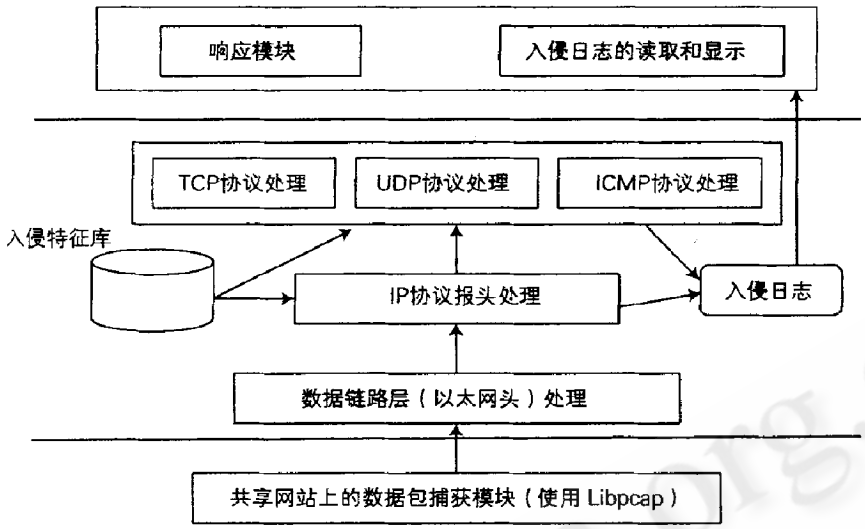

图 1

//设置 PCAP 程序库，实现对网卡的设置，如果返回 NULL 则打印出错信息

if((pkt = pcap_open_live(argv[1], DEFAULT_SNAPLEN,1,1000,ebuf)) == NULL)

{(void)fprintf(stderr,"%s",ebuf);

exit(1);}

//循环取数据包，改变参数 -1 为其它值，可确定取数据包的个数，这里为无限个 if(pcap_loop(pd, -1,packet_proce,NULL) <0) //出错时返回 -1，否则调用包处理函数

{(void)fprintf(stderr,"pcap_loop:%s\n",pcap_geterr(pd));

exit(1);}

//关闭函数库

pcap_colse(pkt);

exit(0);

上述函数及其调用基本上可以完成对数据包的捕获环境设置及实现。接下来对数据包进行过滤。

BPF 有两个主要部件：Network tap 和 Packet filter。其过滤过程是：当一个数据包到达网络接口时，链路层驱动程序将其提交到系统协议栈。如果 BPF 正在接口监听，驱动程序将首先调用 BPF。BPF 将数据包发送给过滤器 filter，过滤器对数据包过滤，并将数据提交给过滤器关联的上层应用程序。然后链路层驱动程序重新取得控制权，将数据包提交给上层的系统协议栈处理。其实现有 Pcap_compile 和 Pcap_setfilter 等函数。

4 协议分析

采用协议分析方法的优越性是建立在协议高度规则的基础上，要对数据包进行协议分析就必须详细了解协议结构[4]。

4.1 协议基本结构

所有 TCP，UDP，ICMP，IGMP 数据都以 IP 数据报格式传输，其基本结构如下表 1 所示。

(1) 以太网帧封装格式包括目的 MAC 地址(6 个字节)、源 MAC 地址(6 个字节)、协议类型(2 个字节)以及数据(46 - 1500 个字节)。

表 1　IP 数据报格式

以太网首部	IP 首部	TCP 首部	数据部分	以太网尾部
14 个字节	20 个字节	20 个字节		4 个字节

表 2　IP 协议数据报

版本	报头长度	服务类型	总长度	
标志符			标志字节	段偏移
生存有效时间	协议		报头校验和	
源 IP 地址				
目标 IP 地址				
选项				填充
数据				

(2) IP 数据报的格式，及其首部中 20 个字节的具体各字段如图 2 所示(斜体部分为首部，通常为 20 个字节，除非含有选项字段)。

(3) TCP 数据封装在 IP 数据报中，包括 IP 首部(2 字节)、TCP 首部(20 字节)和 TCP 数据三个部分。其中 TCP 首部的数据格式如图 3 所示(斜体部分为首部，通常为 20 个字节)。数据进入协议栈时的封装情况如表 3 所示。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

表 3　TCP 协议数据报格式

源端口								目的端口
序号								
确认序号								
首部长度	保留字段	URC	ACK	PSH	RST	SYN	FIN	窗口字段
校验和								紧急指针
TCP 选项 + 填充								

4.2　协议分析的实现

这里以以太网为例来说明协议分析的流程。首先按照协议解码技术进行上层协议分析，根据协议的不同(TCP/UDP/ICMP)及对应协议端口的不同，转向不同的解析程序进行数据包的分析，具体分析过程可以描述如下：根据以太网的帧结构的定义，在以太帧的第 13 字节处包含了两个字节的第三层协议标识，0800 为 IP 协议，0806 为 ARP 协议，8138 为 NOVELL 协议等。在 IP 数据包的格式定义中，第 10 个字节为第四层协议标识，如：TCP 为 06，UDP 为 11，ICMP 为 01 等。而 TCP 数据包的第 3、4 个字节为应用层协议标识。如 80 为 http 协对以太网数据包解码分析。

协议解码具体实现主要包括数据链路层协议解码(以太网头)和高层协议解码。数据链路层帧解码函数就是将捕获的数据包进行数据链路级的拆解，并且把结果存放在 Packet 数据结构中。高层数据包解码函数将从数据链路层帧解码函数传递的数据包进行解码，并将 Packet 结构剩下的属性域内容补完。

以太网头解码

函数 void DecodeEthPkt(Packet * p, struct pcap_pkthdr * pkthdr, u_int8_t * pkt)

参数：p→解码的数据结构指针

pkthdr→Pcap 捕获的数据包的指针

pkt→拆解后数据包(网络层)的指针

返回：无

流程：

(1) p 指向的 Packet 结构内容清空并将 p 中 pkth 和 pkt 属性指针赋值

(2) 处理包长度属性，并且检验包的大小

(3) 从 pkthdr 中分离出 Ethernet 协议帧头部，保存到 p 的相应域

(4) 提取数据包类型并计数，通过 switch – case 分支结构决定高层(网络层)数据包的解码函数，如若为 IP 协议，则调用 DecodeIP( )函数解码。议、21 为 FTP 协议、23 为 TELNET 协议等。

对 IP 数据包解码分析

函数名：DecodeIP

原型：void DecodeIP(u_int8_t * pkt, const u_int32_t len, Packet * p)

参数：pkt→指向要拆解的数据包(网络层)的指针

len→保存要拆解的数据包(网络层)的长度

p→指向解码的数据结构指针

返回：无

调用源：数据链路层帧解码函数和一些高层数据包解码函数

流程：

(1) 获取 IP 头

(2) 检验包的大小

(3) 排除非 IP 数据包

(4) 数据包长度设置，并且检测包头长度

(5) IP 数据包校验和计算

(6) IP 选项测试，并且调用 DecodeIPOptions( )函数给 IP 选项解码

(7) 处理分段数据包

(8) 根据协议不同，调用相应高层函数解析

(9) 清空缓存并返回

5　测试及结果

在 Linux 下显示抓获经过协议分析的包的细节如下所示。由此可见较好的实现了对数据包头的解码。

05/25 – 15：32：49.358281 219.133.49.163：8000 – > 192.168.1.13：4000

UDP TTL：53 TOS：0x0 ID：47249 IpLen：20 DgmLen：92

Len：72

05/25 – 15：32：51.294271 61.177.95.227：80 – > 192.168.1.13：1249

TCP TTL：51 TOS：0x0 ID：37914 IpLen：20 DgmLen：40 DF

* * * A * * * F Seq：0xD2E5AF2E Ack：0x954E6859 Win：0x1920 TcpLen：20 (下转第 42 页)

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

(上接第 38 页)

6 小结

本文基于协议分析原理,对入侵检测的系统框架及数据包捕获,协议解码进行了深入研究。测试说明能较好实现对网络数据包的解码,为入侵检测的数据分析部分做好了准备。

参考文献

1 Kenneth D. Reed. Protocol Analysis, WB77. 0 [M], WestNet. Inc, 2001. 118 – 122.

2 Next Generation Intrusion Detection Expert System (NIDES) [EB/OL]. http://www.sdl.sri.com/projects/nides/index.html

3 张俊安,网络入侵检测系统研究与实现[D],[硕士学位论文],成都 西南交通大学,2000.

4 索红光、石乐义、梁玉环,TCP/IP 协议分析器的设计开发[J],计算机工程与应用,1999(11):80 – 83。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn