协议分析技术在入侵检测系统中的应用①

周 杨

(军事经济学院 基础部计算机教研室，武汉 430035)

摘 要：入侵检测作为一种动态的网络安全技术，是计算机安全不可缺少的组成部分。目前的入侵检测系统大都采用模式匹配算法，针对高速网络环境下此类系统的检测引擎所面临的性能瓶颈问题，介绍了基于协议分析的入侵检测技术的实现原理，提出利用网络协议的高度规则性快速探测攻击的方法，借此减少虚警和误判的可能性，并提高了网络入侵检测系统的性能和效率。

关键词：入侵检测；协议分析；网络安全

Application of Protocol Analysis Technology to Intrusion Detection System

ZHOU Yang

(Computer Teaching and Researching Section, Military Economics Academy, Wuhan 430035, China)

Abstract: The intrusion detection is a new network security technology and Intrusion Detection System forms one of the most important parts of modern computer security. Nowadays most of Intrusion Detection Systems have been using the pattern matching algorithm, this essay introduces the implement principle of the intrusion detection technology based on protocol analysis on the question of the performance bottlenecks of these systems' detect engine at high-speed internet environment. Then we propose the method of detecting attack rapidly by making use of the network protocols' high degree of regularity. In this way, the possibility of false alarm and miscarriage of justice can be reduced. And at the same time, the performance and the efficiency of network Intrusion Detection System can be improved.

Keywords: intrusion detection; protocol analysis; Internet security

入侵检测是一种计算机系统的安全防御措施，它通过从计算机系统或计算机网络中的若干关键点收集信息并对其进行分析，探测系统或网络中是否有违反安全策略的行为和遭到袭击的迹象。从网络安全立体纵深、多层防御的角度而言，入侵检测作为一种积极主动的动态安全技术，能够实时应对内部、外部攻击及误操作，在网络系统受到危害之前拦截并响应入侵。入侵检测系统则能从很大程度上简化管理员的工作，保证网络的安全运行。

1 基于模式匹配的入侵检测系统的局限性

入侵检测通常使用两种技术：异常检测和误用检测。随着对计算机系统弱点和攻击手段的不断收集与研究，入侵特征化描述的方法越来越有效，这也使得误用检测方法的使用越来越广泛。而模式匹配是误用检测技术中最常用的分析方法。

首次提出用模式匹配算法检测入侵的是 Sandeep Kumar 博士，该方法具有分析速度快、误报率较低等优点，其主要工作过程如下[1]：

(1) 从网络数据包头开始与攻击特征进行比较；

(2) 若比较结果相同，则认为检测到一个可能的攻击；

(3) 若比较结果不同，则从网络数据包中下一个位置重新进行比较；

(4) 直到检测到或网络数据包中的所有字节匹配完毕，一个攻击特征匹配结束；

(5) 对于每个攻击特征，重复从(1)开始的比较；

(6) 直至所有攻击特征匹配完毕，对数据包的匹配

① 收稿时间:2010-10-10;收到修改稿时间:2010-12-07

© 中国科学院软件研究所 http://www.c-s-a.org.cn

结束。

由此可见，模式匹配将网络数据包看作无序、随意的字节流，不涉及网络数据包的内部结构，而只是机械化地对网络中传输的数据包逐一进行匹配。这种检测方法有两个最根本的缺陷：一是所需计算量大，二是使用固定的特征模式探测攻击，只能探测出明确、唯一的攻击特征，即使有轻微变换的攻击串都将被忽略。

随着网络用户的增加和多媒体应用的普及，网络流量越来越大，网络速度越来越快，现有实用的入侵检测系统大都基于特征而采用模式匹配算法，在高速网络环境下，这类系统的检测引擎面临着较为严重的性能瓶颈：一是随着网络数据流的高速化，检测引擎需要分析和处理的数据包大大增加；二是伴随网络攻击的多样化和攻击特征数的不断增加，检测引擎需要匹配的特征模式在不断增长，单个数据包的处理效率直线下降。这使得入侵检测系统在应用于高速网络环境时出现严重的丢包现象，表现出较高的漏报率。因此，改善处理速度、解决数据包丢失等问题是提高网络入侵检测系统性能和效率的关键。

2　基于协议分析的入侵检测技术原理

协议分析技术不同于传统的模式匹配算法，它能够智能地“理解”协议，利用网络协议的高度规则性快速探测攻击的存在，从而提高入侵检测的效率。协议分析技术主要包括协议解码与命令解析，它们能够让系统读懂协议，明确在数据包的什么位置获取什么内容，并且能够判断这些内容的真实性。协议解码的实质就是数据包从协议栈由底向上升，同时去掉各层附加报文首部的过程，系统沿着协议栈向上分析，不仅可以使用当前层已知的协议信息，还能够排除本层及上层不属于此协议结构的其他类型的攻击；而解析器则是一个命令解释程序，它能够读取攻击串及其所有可能的变形，从而发掘其本质含义。入侵检测引擎包含多种不同的命令语法解析器，也因此能够对不同高层协议（如 Telnet、FTP、SMTP、SNMP 等）的用户命令进行更为详细的分析。

协议分析作为网络入侵检测系统的基础，其主要作用体现在：第一，协议分析为检测引擎提供输入数据，是检测引擎的基础；第二，协议分析针对数据包的上下文进行分析，直接提高了检测的有效性；第三，在进行应用层协议分析时，字符串的匹配被定位到具体字段，这无疑是提高字符串匹配性能的一个有效手段[2]。

图 1 所示的是一个基于数据包分析的网络入侵检测系统的通用结构。首先，入侵检测器收取所有的通信流量；接着，对于收到的原始数据包，解析出每层协议头部的各个域；然后，进入入侵检测引擎，按照基于特征或基于异常的方法对数据包进行分析，如果发现攻击特征或者异常现象，则发送一个报告给决策模块；最后，决策模块根据报告的严重级别确定所需采取的行动，并通知反应模块，由反应模块对发现的攻击行为做出响应。常用的响应方法包括向控制台发出警报、切断数据包连接、通知防火墙隔离攻击方等。

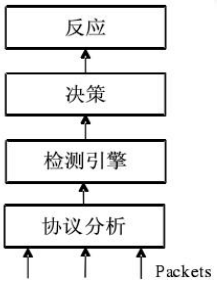

图 1　网络入侵检测通用结构图

3　协议分析的具体过程

TCP 协议和 IP 协议是网络通信的核心协议，在 RFC 的 0791[3]和 0793[4]文档中，分别定义了 TCP 数据包和 IP 数据包的格式。正是因为这种格式定义与网络的结构和类型无关，而只与协议相关，因此使得协议分析具有很广泛的适用性。根据以太网帧结构的定义，在第 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 则为 NOVELL 协议。在 IP 数据包的格式定义中，第 10 字节为第四层协议标识：TCP 为 06，UDP 为 11，ICMP 为 01；而 TCP 数据包的第 3、4 字节为应用层协议标识，即端口号：80 为 HTTP 协议，21 为 FTP 协议，23 为 TELNET 协议。图 2 描述了协议分析的基本流程。

这里结合一个实例说明协议分析的主要过程：

对于网络数据包 BD5%Hy289s820800B9v5yt $0611tbhk76500801293ugdB2*00397e39123789012345345678945601267890123456789012345012345667 89，首先，根据以太网协议规则，在第 13 字节处应有 2 个字节的第三层网络层协议标识，这里得知其值为“0800”，可见以太网帧数据区域中携带的是 IP 协议；接着，根据 IP 协议结构，在第 24 字节处应有 1 个字节的第四层传输协议标识，这里得知其值为“06”，可见 IP 帧中数据区域携带的是 TCP 协议；最后，根据 TCP 协议的第 35 字节处应有 2 个字节的应用层协议标

© 中国科学院软件研究所 http://www.c-s-a.org.cn

识，即“端口号”，这里得知其值为“0050”，转换为十进制的 80，得知其为一个 HTTP 访问。

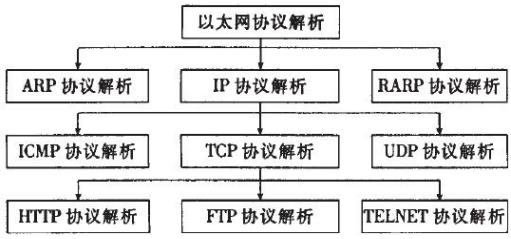

图 2 协议分析基本流程图

将获得的此特征值与特征库内容进行比较，若与特征吻合，则访问被判定为攻击行为。由此可见，协议分析的过程就是一条从根到某个节点或叶子的路径，而每个叶子、节点则是某一种攻击类型的分析机。分析机的功能在于分析某一特定协议的数据，判断其是否具有攻击的可能性。每个分析机的数据结构中都包含协议名称、协议代号及该协议对应的攻击检测函数等信息。协议名称是该协议的唯一标志，而协议代号则是为了提高分析速度所使用的编号。一般情况下，分析机应尽可能地靠近叶子结点，因为越靠近树根部分的分析机，调用的次数越多，而根部附近聚集过多的分析机会严重影响系统的性能。为了提高检测的精确度，可以在树中添加自定义的协议结点，以此来细化分析数据。叶子节点上的协议类型划分得越细，分析机的效率越高。

4 基于协议分析的检测模块设计

4.1 协议分析的算法实现

数据包捕获函数库(Packet Capture Library)作为独立的 API 函数接口，用于对用户层次的数据包进行捕获，并为底层网络监控编程提供一个易于移植的应用框架，这些底层网络应用包括数据收集、安全监控和网络调试等。利用 Libpcap 提供的库函数进行数据采集，通过 pcap-open-live 函数设置网卡的状态为混杂模式，可以提供从链路层直接捕获数据包的功能，并且能够设置数据包的过滤器以捕获指定的数据[5]。

在协议分析过程中，主要实现对 IP 数据包内容的分析，具体过程如下：

```
void ip-protocol-packet (struct pcap-pkthdr *packet-head, char *packet-content)
//packet-head 表示被捕获数据包的头信息，packet-content 表示被捕获数据包的具体内容
{ struct ip-head *ip-protocol //协议变量，一般可以根据 IP 协议格式自行定义
ip-protocol=(struct ip-head*)(packet-content+14)
if(ip-source-add==ip-local-add)//判断源 IP 是否为本机 IP
{if(ip-destination-add&ip-add-broakcast==ip-add-broak-cast) //判断 smurf 和 land 攻击
.....//smurf 攻击分析程序
else
if(ip-destination-add==ip-source-add)
.....//land 攻击程序
}
else
{switch(ip-protocol->ip-protocol)
{ case 6:
//TCP 协议内容分析
tcp-protocol=(struct tcp-head *)(packet-content+14+20);
source-port=ntohs(tcp-protocol->tcp-source-port); //获取源端口
destination-port = ntohs (tcp-protocol-> tcp-destination-port); //获取目标端口
.....
switch(destination-port)
{case 80:
.....//HTTP 协议分析程序
case 21:
.....//FTP 协议分析程序
case 23:
.....//TELNET 协议分析程序
case 25:
.....//SMTP 协议分析程序 }
case 17:
.....//UDP 协议内容分析
case 01:
.....//ICMP 协议内容分析 }}}
```

需要说明的是，在算法的实现上并不一定局限于上述方法，可以针对不同的协议定义不同的函数，而在主函数中利用函数 pcap-loop()循环捕获网络数据包，再逐层调用协议分析函数即可。

4.2 检测模块的具体实现

入侵检测模块从协议分析模块获取刚被捕获的数据包信息，而这些信息协议的解析模块均存放在全局变量中。利用全局变量将这些信息传递到检测模块中，系统便获得了网络捕获数据包的相关信息。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

检测模块所做的工作就是对规则进行匹配，即判断规则中事件定义的真假。由于在规则的入侵特征定义中，入侵表达式均采用“&”连接，因此，当一个入侵特征表达式为假时，事件定义也为假，此时的数据包信息与此条入侵规则不相匹配，于是数据包便开始匹配下一规则。

通常，判断事件定义中一条表达式的真假用函数compare_a_statement()来实现，根据协议变量的值确定捕获的数据包所属的协议类型，从而选择相应协议的规则文件进行匹配。这里，假设捕获的数据包是 IP 数据包，因此需要匹配 IP 规则库中的规则。compare_a_statement()函数的实现过程如下：

```
int compare_a_statement(char* variable[500],char sign1,char *result)
{//判断事件定义中一个事件表达式的真假
int number_result=0;
int variable_value;
number_result=char(result); //将变量类型转换为字符型
variable_value=get_protocol_variable(variable);  //获得协议变量的值
if (variable_value==-1)
  return 0;  //如果协议变量为-1，退出
if(sign1=='=')
{/当符号为“=”时
if(variable_value==number_result)    //判断规则中的协议值与真实的协议变量值是否相等
  return 1;}
if(sign1=='≈')
{ //当符号为“≈”时
  if(variable_value==number_result)    //判断规则中的协议值与真实的协议变量值是否相等
    return 1;}
    if(sign1=='>')
{ //当符号为“>”时
if(variable_value==number_result)    //判断规则中的协议值与真实的协议变量值是否相等
  return 1;}
if(sign1=='<')
{ /当符号为“<”
if(variable_value==number_result)    //判断规则中的协议值与真实的协议变量值是否相等
  return 1;}
  return 0;  //默认返回 0
```

需要注意的是，若一条规则特征定义中用“&”连接的规则表达式全为真时，数据包才匹配整条规则；反之，只要有一个规则表达式为假，数据包将不匹配规则。

5　结语

入侵检测作为一种积极主动的防御技术，与防火墙、漏洞扫描等安全技术共同形成一个动态的安全体系。基于协议分析的入侵检测系统作为当前较为先进的入侵检测系统，具有检测速度快、系统消耗低、误报率低等优点，协议分析技术将成为网络环境下入侵检测的主流技术。

目前对于提高模式匹配效率及协议分析性能的研究还在继续，模式匹配技术与协议分析技术作为入侵检测系统中数据分析模块的重要技术，两者的结合将增加检测的可靠性，提高分析速度，减少虚警和误判的可能性，从而突破网络高速发展带来的性能瓶颈。因此，进一步要做的工作是对网络协议、计算机系统漏洞和入侵行为进行更为详细的研究，以便更准确地定义入侵特征，继续完善检测系统的功能，并力争开发出基于协议分析的通用入侵检测系统。

参考文献

1 Protocol Analysis and Command Parsing vs.Pattern Matching in Intrusion Detection System. http:// www.net workice.com/products/documentation.html,2000.

2 唐正军,李建华.入侵检测技术.北京:清华大学出版社,2004.117–120.

3 Internet Protocol DARPA Internet Program Protocol Specification.http://www.ietf.org/rfc/rfc0791.txt?number=791,1981–09.

4 Transmission Control Protocol DARPA Internet Program ProtocolSpecification.http://www.ietf.org/rfc/rfc0793.txt?number=793,1981–09.

5 杨小平,苏静.基于协议分析的入侵检测技术研究.计算机应用研究,2004,2:108–110.

© 中国科学院软件研究所 http://www.c-s-a.org.cn