第 21 卷 第 4 期
2003 年 11 月

吉 林 大 学 学 报 (信 息 科 学 版)
JOURNAL OF JILIN UNIVERSITY (INFORMATION SCIENCE EDITION)

Vol. 21 No. 4
Nov. 2003

文章编号: 1671-5896 (2003) 04-0412-05

基于模式匹配的网络入侵检测系统

田大新[1], 刘衍珩[1], 魏 达[1], 张树伟[2]

(1. 吉林大学 计算机科学与技术学院, 吉林 长春 130012; 2. 吉林省交通规费征收管理局 白城分局, 吉林 白城 137300)

摘要: 设计了一个基于模式匹配专家系统的网络入侵检测系统, 它的所有组件都是用 LINUX 下的 C 语言编写的。该系统一方面能够抓取计算机网络中数据链路层的所有数据包并记录下来; 另一方面能够用一基于专家系统的检测引擎对抓取的数据包进行实时分析, 以检测各种入侵。该专家系统采用模式匹配原理, 主要采用存在模式和规则表示模式。当检测到入侵时, 系统在发出警报的同时还将数据包的详细内容记录下来, 以发现入侵者的详细信息。

关键词: 入侵检测系统; 专家系统; 数据包; 模式匹配

中图分类号: TP393.08 **文献标识码**: A

引 言

随着网络信息技术的发展和应用, 网络安全已经成为一个非常热门的话题, 而入侵检测 (ID: Intrusion Detection) 则是这两三年来在网络安全领域比较热门的技术。传统的安全策略如防火墙或 VPN 主要注重防护, 而防护相对进攻来说总是滞后的。因为计算机系统和应用软件总是包含无法预知的漏洞, 一种漏洞的发现或者攻击手段的发明与相应的防护手段的采用之间总会有个时间差, 而检测是弥补这个时间差的必要手段。入侵检测是指发现未经授权非法使用计算机系统的个体, 或合法访问系统但滥用了他们使用权限的个体[1]。入侵检测系统具有发现入侵行为, 同时根据入侵的特性采取相应动作的功能[2]。1980 年, Anderson 在报告“计算机安全威胁的监视 (Computer Security Threat Monitoring and Surveillance)”中提出, 必须改变现有的系统审计机制, 以便为专职系统安全人员提供安全信息。此文被认为是有关 ID 的最早论述。1984～1986 年, Dorthy Denning 和 Peter Naumann 研究和发展了一种命名为入侵检测专家系统 (IDES: Intrusion Detection Expert System) 实时模式的入侵检测系统。自从 Denning 于 1986 年发表论文“入侵检测模型 (An Intrusion Detection Model)”[3]以来, 大量的 IDS 被开发出来, 包括研究的原型和商业化的产品[2]。根据数据来源的不同, 主要有基于主机的 IDS 和基于网络的 IDS[4]。基于主机的系统获取数据的依据是系统运行所在的主机, 保护的目标是系统运行所在的主机; 基于网络的系统获取的数据来源是网络传输的数据包, 保护的目标是网络的运行。按照数据分析方法的不同, IDS 分为异常检测模型 (ADM: Anomaly Detection Model) 和滥用检测模型 (MDM: Misuse Detection Model)[5], ADM 的特点是首先总结正常操作应该具有的特征, 在得出正常操作的模型后, 对后续的操作进行监视, 一旦发现偏离正常统计学意义上的操作模式, 即进行报警; MDM 的特点是收集入侵行为的特征, 建立相关的特征库, 在后续的检测过程中, 将收集到的数据与特征代码进行比较, 得出是否具有入侵的结论。笔者所讨论的入侵检测系统为基于网络的滥用检测模型。

1 系统设计

1.1 系统结构图

系统构成如图 1 所示。图 1 中抓包模块的作用是采用数据提取技术收集网络中传输的数据包, 为

收稿日期: 2003-03-13
作者简介: 田大新 (1980—), 男, 河北唐山人, 吉林大学硕士研究生, 主要从事计算机通信与安全研究, Tel: 0431-5196707, E-mail: daxin222@sina.com; 刘衍珩 (1958—), 男, 吉林松原人, 吉林大学教授, 博士生导师, 主要从事计算机通信与网络研究, Tel: 0431-5168355, E-mail: yhliu@email.jlu.edu.cn。

入侵检测系统进行入侵分析提供数据来源。

知识模块的作用是对输入的专家知识进行处理，将其转换成能被检测引擎使用的包含特定警告程度信息的规则。

检测引擎负责对抓包模块抓取的数据包进行同步、整理、组织、分类，并把这些数据转换成规范的格式，将格式化的事件记录与知识库的内容匹配，并根据比较结果决定程序的下一步动作。如果记录表明是一个入侵，系统就产生一次报警。

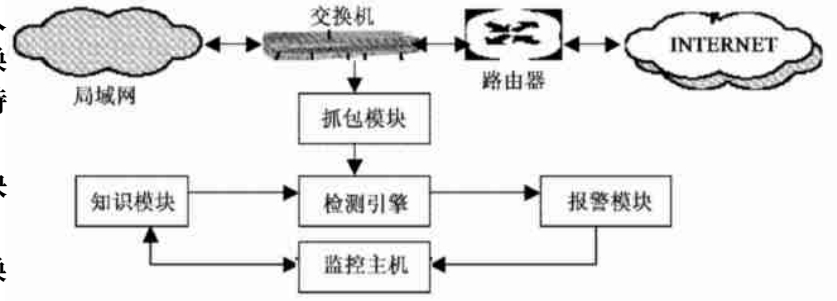

图 1　系统结构图

Fig. 1 System construction

报警模块负责将检测引擎提供的警报信息传递给监控主机。

1.2　详细设计

1.2.1　抓包模块

在计算机网络系统中，局域网普遍采用的是基于广播机制的 IEEE802.3 协议[6]，即以太网(Ethernet)协议。该协议保证传输的数据包能被同一冲突域内的所有主机接收[6]，基于网络的入侵检测正是利用了以太网的这一特性。以太网卡通常有正常模式(normal mode)和杂收模式(promiscuous mode)。在正常模式下，网卡每收到一个到达的数据包，就会检查该数据包的目的地址，如果是本机地址或广播地址则接收，否则丢弃；在杂收模式下，网卡可以接收本网段内传输的所有数据包。

该系统利用网卡的杂收模式，获得经过本网段的所有数据信息，从而对数据包进行检测。它所使用的抓包库是 Libpcap，Libpcap 是 Unix 和 Linux 平台下应用最为广泛的数据包截取库，它所使用的主要函数如下。

char * pcap_lookupdev (char * errbuf)

此函数用于查找合适的网络设备，返回的指针指向的设备号将用于函数 pcap_open_live () 和 pcap_lookupnet ()。如果出错将返回 NULL，并将错误信息存于 errbuf 中。

pcap_t * pcap_open_live (char * device, int snaplen, int promisc, int to_ms, char * errbuf)

此函数用于得到一个抓获包的描述信息，以便分析网络层中的数据包信息，主要参数含义如下：

device 表示已知的网络设备的字符串；

snaplen 确定抓获包的最大字节数；

promisc 确定是否工作在杂收模式 (非 0 为杂收模式)；

to_ms 确定读取包的时间限制；

errbuf 返回错误信息。

int pcap_lookupnet (char * device, bpf_u_int32 * netp, bpf_u_int32 * maskp, char * errbuf)

此函数用于获得网络设备的网络地址和掩码，主要参数含义如下：

device 指定的网络设备；

netp 用以存储获得的网络地址；

maskp 用以存储获得的掩码；

int pcap_loop (pcap_t * p, int cnt, pcap_handler callback, u_char * user)

此函数用于将抓取的数据包传给分析函数，主要参数如下：

p pcap_open_live () 返回的指针；

cnt 指定抓取多少数据包，-1 表示抓取所有的数据包直到有错误产生；

callback 为一指向函数的指针，抓取的数据包将传到此函数中。

1.2.2　检测引擎

检测引擎为一个基于模式匹配原理的专家系统[7]。专家系统主要采用存在模式和规则表示模式两种匹配模式。存在入侵信号表示只要存在这样一种审计事件就足以说明发生了入侵行为或入侵企图，它所对应的匹配模式称为存在模式。存在模式可以理解为在一个固定的时间对系统的某些状态进行检查，

并对系统的状态进行判定。规则表示模式是指用一种扩展的规则表达式方式构造匹配模式，规则表达式由用AND、OR 逻辑表达式连接一些描述事件的原语构成。适用这种模式的攻击信号通常由一些相关的活动组成，而这些活动间没有什么事件顺序的关系。知识表示的一般形式为

RULE * IF〈前提〉THEN〈行为〉

RULE 为一规则链表，* 表示指向此链表中一节点的指针;

前提的一般形式为

(AND〈条件-1〉〈条件-2〉…〈条件-n〉)

它表示条件-1，条件-2，…，条件-n 之间是合取关系，其中每个条件既可以是一个简单条件，也可以是具有OR 关系的复合条件，例如对于条件-i 可为:

(OR〈条件-i1〉〈条件-i2〉…〈条件-im〉)

行为部分由专门表示动作的行为函数表示。知识的数据结构为:

```
struct ether_arp {
    struct arphdr ea_hdr;                /* 固定字节头 */
    u_int8_t arp_sha[ETH_ALEN];          /* 源端的物理地址 */
    u_int8_t arp_spa[4];                 /* 源端的协议地址 */
    u_int8_t arp_tha[ETH_ALEN];          /* 目的端的物理地址 */
    u_int8_t arp_tpa[4];                 /* 目的端的协议地址 */
}; /* 以太网地址 */
struct ip {
    unsigned int ip_hl:4;                /* IP 头长度 */
    unsigned int ip_v:4;                 /* 版本号 */
    u_int8_t ip_tos;                     /* 服务类型 */
    u_int8_t ip_len;                     /* IP 包长度 */
    u_int8_t ip_id;                      /* 标识 */
    u_int8_t ip_off;                     /* 分段偏移 */
    u_int8_t ip_ttl;                     /* 生命期 */
    u_int8_t ip_p;                       /* 协议 */
    u_short ip_sum;                      /* 头部校验和 */
    struct in_addr ip_src;               /* 源端地址 */
    struct in_addr ip_dst;               /* 目的端地址 */
}; /* IP 数据报 */
struct tcphdr {
    u_int16_t th_sport; /* 源端口 */
    u_int16_t th_dport; /* 目的端口 */
    tcp_seq th_seq; /* 顺序号 */
    tcp_seq th_ack; /* 确认号 */
    u_int8_t th_x2:4; /* 保留 */
    u_int8_t th_off:4; /* 数据偏移 */
    u_int8_t th_win; /* 窗口大小 */
    u_int8_t th_sum; /* 校验和 */
    u_int8_t th_urp; /* 紧急指针 */
}; /* TCP 数据片 */
struct udphdr {
    u_int16_t uh_sport; /* 源端口 */
    u_int16_t uh_dport; /* 目的端口 */
    u_int16_t uh_ulen; /* 包长度 */
    u_int16_t uh_sum; /* 校验和 */
}; /* UDP 数据片 */
```

1.2.3 知识模块

知识模块主要完成专家知识的录入及生成检测引擎可用的检测树。系统采用检测树来表示问题，一棵检测树构成了对一类入侵的完整描述。例如: 对于检测拒绝服务 (Dos) 攻击的检测树如图 2 所示。检测树的搜索策略采用深度优先的左序遍历，具体算法如下。

树的存储结构采用孩子表表示法。

```
struct child_tree {
    char * data;
    int number;
    struct chile_tree * next;
    };
struct tree {
   struct chile_tree first;
   struct tree * next;
};
```

递归算法如下[8]。

```
function alarm_tree (struct tree * head)
{
  getdata (head→data);                    /* 访问根节点数据 */
  alarm_tree (heak→next→number= = left); /* 遍历 head 的最左子树 */
  alarm_tree (heak→next→number= = ! left); /* 遍历 head 除去最左子树剩下的子树 */
}
```

1.2.4　报警模块和监控主机

报警模块将检测引擎检测到的入侵及时反映到监控主机并记录下来，同时负责专家知识的录入工作。

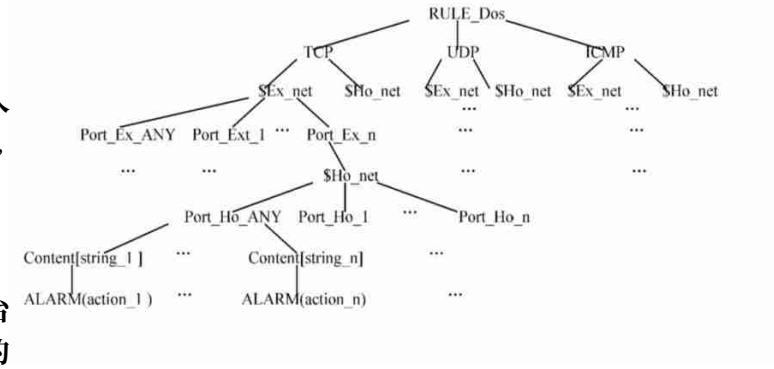

图 2　检测树

Fig. 2 Detection tree

2　实验及结果

将该系统安装到局域网中的一台以 Red Hat Linux 7.3 为操作系统的 586 计算机上，测试了系统的丢包率（见表 1）和检测能力（见图 3）。

对丢包率采用先规定好抓包数 i（i= 10, 50, 100, 300, 500），然后看实际抓到的包数 j，采用多次测量取平均值的方法，最后求出总的丢包率，如表 1 所示。

表 1　丢包率测试结果

Tab. 1 Test results of discard packet rate

包数/个	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	均值	丢包率/%
10	7	9	9	9	9	8	6	9	9	9	8.4	16.0
50	41	44	44	42	41	35	38	39	39	38	40.1	19.8
100	71	71	68	85	79	77	85	71	87	77	77.1	22.9
300	249	246	227	124	255	228	220	239	274	224	228.6	23.8
500	353	314	325	340	335	365	346	370	300	330	372.0	25.6


```
03/06/03[12, 24, 31]—ICMP!!!  —SUCCESS IN: Geting ethernet packet, ip packet and icmp packet
[Ethernet]:
0: 0: E8: D3: B8: D3 →0: E0: 4C: 73: 4F: 89.
Ethernet type: 0x800, length: 0x62
IP:
202. 198. 27. 210 →202. 198. 27. 126.
IP version: 67108864, header length: 0x5000000, type: ICMP, TOS: 0x0, ip length: 0x54, ID: 0x4A5B, OFF: 0x4000, TTL: 0, SUM: 0xE33E, DF.
[ICMP]:
TYPE: 0 |code0-ECHO REPLSY, ID: 1964, SEQ: 148, SUM: 46867.

03/06/03[12, 24, 31]—ICMP!!!  —SUCCESS IN: Geting ethernet packet, ip packet and icmp packet
[Ethernet]:
0: E0: 4C: 73: 4F: 89 →0: 0: E8: D3: B8: D3
Ethernet type: 0x800, length: 0x62
IP:
202. 198. 27. 126 →202. 198. 27. 210.
IP version: 67108864, header length: 0x5000000, type: ICMP, TOS: 0x0, ip length: 0x54, ID: 0x0, OFF: 0x4000, TTL: 0, SUM: 0x6D9A, DF.
[ICMP]:
TYPE: 0 |code0-ECHO, ID: 1964, SEQ: 148, SUM: 44819.
```

图 3　入侵检测测试结果

Fig. 3 Test result of intrusion detection

丢包率随抓包数目上升而有小幅升高的原因是测试时要求把所有抓获的包的详细信息都记录并存储起来，记录的包越多占用系统时间就越多。而在真正安装时只对那些入侵数据进行记录，正常数据

包将丢弃。据此可以推测出系统的丢包率在10%～20%之间。

对于检测能力，笔者从局域网中的一台计算机对另一台计算机发起了ping攻击，结果入侵检测系统发现了该攻击，并记录了下来(见图3)。

图3表明，入侵检测系统发现了此次攻击并记录下了攻击方和被攻击方的物理地址、网络地址、攻击时间和数据包的详细信息。

3 结 论

笔者系统地阐述了入侵检测系统的设计及其实现。它使用的Libpcap包捕获库保证了系统运行的高速和稳定。在检测引擎中使用了基于模式匹配的专家系统，能对网络数据链路层的数据包进行实时分析，在发现入侵行为时进行报警并记录下入侵者的详细信息。目前，该入侵检测系统能够检测出端口扫描、拒绝服务攻击、缓冲区溢出等攻击形式。

参考文献:

[1] Mukherjee B, Levitt T L. Network intrusion detection [J]. IEEE Network, 1994, 8 (3): 26—41.

[2] 胡亮, 康健, 赵阔, 孟凡二 (HU Liang, KANG Jian, ZHAO Kuo, MENG Fan-er). 入侵检测系统 (Intrusion detection systems) [J]. 吉林大学学报 (信息科学版) (Journal of Jilin University (Information Science Edition)), 2002, 20 (4): 46—53.

[3] Denning D E. An Intrusion-Detection Model [J]. IEEE Transaction on Software Engineering, 1987, 13 (2): 222—232.

[4] 蒋建春, 马恒太, 任党恩, 卿斯汉 (JIANG Jian-chun, MA Heng-tai, REN Dang-en QING Si-han). 网络安全入侵检测: 研究综述 (A survey of intrusion detection research on network security) [J]. 软件学报 (Journal of Software), 2000, 11 (11): 1 460—1 467.

[5] Spafford E, Zamboni D. Data collection mechanisms for intrusion detection systems [R]. CERIAS Technical Report, Center for Education and Research in Information Assurance and Security, West Lafayette: Purde University, 2000.

[6] Andrew S. Tanenbaum. Computer Networks 3rd ED [M]. [s. l]: Prentice Hall, 1996. 275—286.

[7] Kumar S, Spafford E H. A pattern matching model for misuse intrusion detection [A]. In Proceedings of the 17th National Computer Security Conference [C]. Baltimore MD USA: NIST National Institute of Standards and Technology/National Computer Security Center, 1994. 11—21.

[8] Patterson D W. Introduction to Artifical Intelligence and Expert Systems [M]. New Jersey: Prentice Hall, 1990.

Design and implementation of network-intrusion detecting system

TIAN Da-xin[1], LIU Yan-heng[1], WEI Da[1], ZHANG Shu-wei[2]

(1. College of Computer Science and Technology, Jilin University, Changchun 130012, China;
2. Baichen's Bureau of Traffic of Jilin, Baichen 137300, China)

Abstract: This is a network-intrusion detection system based on expert system. All the sections of the intrusion detection system are developed in C program under Linux. The system can get all the packets from data link layer and record them, perform real-time traffic analysis and detect intrusions using a special expert system. The expert system in the detection engine uses the Match Pattern Theory, adopts existing pattern and regulation expression pattern. If it finds an intrusion, it will make an alarm and write down the data packets in order to find the detailed information.

Key words: Intrusion detection system; Datagram; Expert system; Match pattern