基于网络事件和深度协议分析的入侵检测研究

朱映映，吴锦锋，明仲
(深圳大学 计算机与软件学院，广东 深圳 518060)

摘 要：针对制约 NIDS（基于网络的入侵检测系统）的问题，提出了基于网络事件和深度协议分析的入侵检测模型 MIDM，实现了对入侵的分析与综合。扩展了 ABNF 范式形式化定义网络事件，基于所提出模型重新实现了入侵检测系统。实验证明与当前主流 NIDS 相比，新模型有效降低了误检率和特征库冗余；具有随网络流量和特征库快速增长，而 CPU 占用率维持低水平增长的特性，能更好地适应高速网络环境；同时还具有一定的特征泛化和检测未知入侵的能力。

关键词：入侵检测；协议分析；模式匹配；异常检测；高速网络
中图分类号：TP393.08 **文献标识码**：A **文章编号**：1000-436X(2011)08-0171-08

Research on intrusion detection based on network events and deep protocol analysis

ZHU Ying-ying, WU Jin-feng, MING Zhong
(Computer and Software College, Shenzhen University, Shenzhen 518060, China)

Abstract: The problems for restricting NIDS were investigated. Based on network events and deep protocol analysis, a new model MIDM analyzing and integrating network intrusion was proposed. After extending ABNF to describe network events, a new NIDS was built based on MIDM. Experimental results proved that, comparing to the current mainstream NIDS, the model MIDM can work effectively with less false positive rate and less redundancy of rule base. And if network stream and rule base were extended quickly, the CPU utilization of new model's would remain low growth, which makes MIDM better adapt to high-speed network. And it's also able to detect some unknown attacks and sustain rule generalization.

Key words: intrusion detection; protocol analysis; pattern matching; statistical anomaly detection; high-speed network

1 引言

入侵检测系统(IDS, intrusion detection system)依检测方式的不同可被划分成基于异常检测的 IDS 和基于特征检测的 IDS 两大类[1]；依检测对象的不同又划分成基于网络的 IDS(NIDS, network-based IDS)和基于主机的 IDS(HIDS, host-based IDS)2 种。有多种方法被引入到入侵检测中，如基于知识库的专家系统(IDES)[2,3]、基于统计学的[4,5]、基于机器学习的、基于模式识别和数据挖掘理论[6,7]的方法等。目前，NIDS 面临着如何把误报率(FPR, false positives rate)和漏报率(FNR, false negatives rate)降到最低；如何减少规则冗余；如何进行快速规则匹配以适应高速网络环境等重要问题。

收稿日期：2010-09-06；**修回日期**：2010-12-31
基金项目：国家自然科学基金资助项目(60703112)；深港创新圈基金资助项目(ZYB200907060012A)；广东省自然科学基金资助项目(10351806001000000)
Foundation Items: The National Natural Science Foundation of China (60703112); The Shenzhen - Hongkong Innovation Zone Project (ZYB200907060012A); The Natural Science Foundation of Guangdong Province (10351806001000000)

美国国家计算机安全中心使用PBEST[3]制作了一套多方入侵检测预警系统。该系统具有一定的数据驱动和推理功能，但是当规则库很大时，系统性能会严重下降。在法国防御机构的 MIRADOR 项目中，F. Cuppens 等[8]建立了 CRIM 模型来降低误报和漏报现象，使用 Snort 和 e-Trust 2 种 NIDS 来生成规则，经过警报关联后，系统误报率有所下降，但是它不能提高系统的检测速度。J.O. Nehinbe[7,9]基于 Snort 提出了 Expert-track 模型，自动对规则进行优化调整，较大幅度地优化了规则空间。B.D. Cabrera[5]等针对 DoS 攻击建立了基于泊松分布模型的网络异常检测方法。仿真实验表明模型的正确率在 60%~85%之间，并且没有漏报现象。李秀婷[10]改进了 Snort，依自动机理论提出了“状态协议分析法”检测 DDoS 攻击。为了适应高速网络环境，Sarang[11]、陈一骄[12]、黄建[13]等分别基于 Snort 规则集提出了基于硬件实现的快速多模式匹配算法，相比于单模式匹配算法，这些方法更具优势。协议分析技术是近年来应用于入侵检测的新技术。Abbes T[14]和 Holger[15]将这项技术引入到他们的研究中，但是只局限在个体报文并且只在非应用层协议上做分析，具有一定的局限性。

为解决 NIDS 面临的困难，本文做了 3 点创新：提出了基于事件和深度协议分析的入侵检测系统模型；扩展了 ABNF 范式来形式化定义网络入侵事件，实现了规则的强联系；在前 2 点工作的基础上实现了一个新的入侵检测系统。实验证明该系统与当前主流的 NIDS 相比具有更高的准确性、更低的规则冗余、更快的检测速度、更适应高速网络，并具有一定的特征泛化和检测未知入侵的能力。

2 基于网络事件和深度协议分析的入侵检测系统建模

如果把基于特征检测的规则空间称为特征域，记为 R，把基于异常检测的异常空间称为异常域，记为 S，把所有报警的集合所构成的空间称为报警域，记为 A，把所有真实入侵构成的空间成为入侵域，记为 I，那么所有在入侵检测中的研究可归结为寻求建立在空间 R、S、A 上的映射 F，如图 1 所示。

F 可以分解成如式(1)所示。

$$F:\begin{cases} f_1:(R,S)\to(R,S,A) \\ f_2:A\to I \end{cases} \tag{1}$$

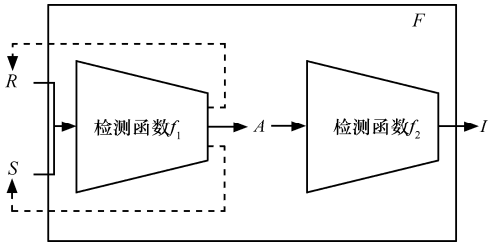

图 1 入侵检测研究的映射过程

如图 1 所示，虚箭头体现在对规则域和异常域的学习和优化上。检测函数 f_2 是在报警已经产生的情况下为了降低误报率(FPR)和漏报率(FNR)而进行的修正过程。这样，A-I 就是误报空间，而 I-A 则是漏报空间。由于 f_1 的检测未必全部正确，在 f_2 过程进行修正是必要的。但如果误报和漏报现象过多，反而会使得 f_2 占用大量 CPU 时间从而抑制检测引擎的速度。因此，要提高系统性能，着重要解决 f_1 的映射过程和构造合理的 R、S 空间。

在对大量实际的网络攻击进行研究的基础上，本文对 NIDS 重新建模，并将该模型称为基于网络事件和深度协议分析的入侵检测数学模型(mathematical intrusion detection model based on network events and deep protocol analysis)，简称 MIDM 模型。把网络上为完成某一目的而发生的所有行为的集合，称为一个网络事件(network event)。MIDM 模型将网络事件自顶向下分为 4 层：会话层(dialog layer, 又称事件层)、语句层(sentence layer)、应用数据层(applied data layer)和词汇层(word layer)，如图 2 所示。

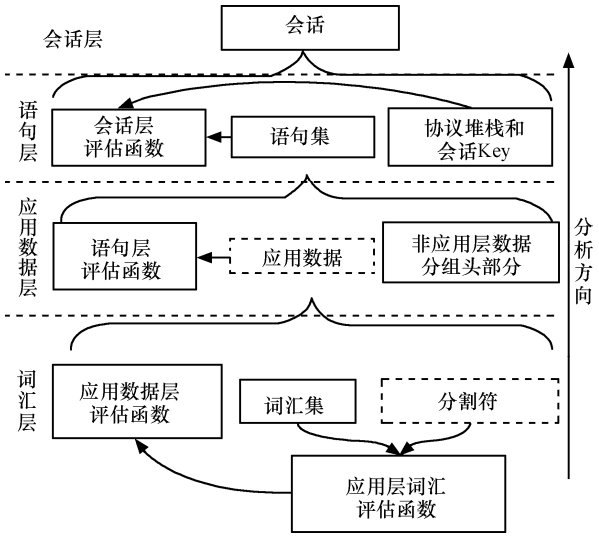

图 2 MIDM 模型

MIDM 模型实现了一个自顶向下、从整体到局部的划分，但具体的检测过程却是自底向上、由局部到整体、从分析到综合的过程。基于协议分析的方法被认为是入侵检测的新方法，但如何实现协议分析，研究者们至今还没有统一意见，之前的研究[14,15]都将协议分析局限在单个数据报文中，并且只对非应用层协议做分析。本文的研究方法与它们最大的不同，体现在本文不仅将协议分析抽象到基于事件的整体性上，而且还将协议分析深入到具体的应用层数据里面，故而称为深度协议分析。下面介绍 4 个重要定义。

定义 1　在数据报文里具有为特征模式提供相对参考位置的特殊码字或若干连续的特殊码字，称为分割符，记为 F。

分隔符具有为特征匹配提供参考位置的重要特性。例如在对 Web 服务器的 CGI 攻击报文中，0x0d0a 就为攻击模式提供参考位置。分隔符的引入具有提高检测速率和检测准确率的作用。

定义 2　网络上的 2 台计算机进行通信时，在一个报文里可能包含一项或多项重要信息，这些信息被称为词汇或词，记为 W。词是协议分析的基本单元，网络通信协议通过词来认知信息。一个词可以通过一个属性向量来描述：

$$W=(F, N_F, W_r, W_n, W_l, W_m, W_{ml}, \delta_w) \tag{2}$$

其中，F 为分隔符，可为空值；N_F 是分隔符计数；W_r 为词的特征串；W_n 描述一个词在所属的通信语句中是否为必需的属性；W_l 是长度属性，描述词包含的字节数；W_m 为匹配模式属性，它分为定点模式和区间模式 2 种，在定点模式下，匹配在指定的位置进行，在区间模式下，允许在区间内滑动匹配；W_{ml} 为匹配长度属性；δ_w(>0)为词入侵疑似度属性，它用于表征一个词在入侵事件中的重要程度。数据报文被分析时首先调用词语评估函数，得出相应的疑似度 Δ_W，词语评估函数分为精确评估和模糊评估 2 种，假定 W_r'为被评估的模式。在精确评估的情况下，Δ_W的取值有 0 或者 δ_w 2 种取值：

$$\Delta w(W_r, W_r') = \begin{cases} \delta_w, & W_r = W_r' \\ 0, & \text{其他} \end{cases} \tag{3}$$

在模糊评估的情况下，$\Delta_W \in [0, \delta_w]$：

$$\Delta_W(W_r, W_r') = \rho\delta_w \tag{4}$$

其中，ρ 这样计算：假设模式 W_r 的长度为 N，模式 W_r'与模式 W_r 在按比特进行比较时相等比特数的统计量为 N_1，则

$$\rho=N_1/N \tag{5}$$

模糊评估的引入有一个好处，它允许特征与真实模式出现偏差，但又不会导致整体失效，能有效支持特征泛化。

定义 3　网络上通信的 2 台计算机之间的一次信息交互过程，即网络中传输的一个数据报文，称为一个通信语句，记为 S。它也可以通过一个属性向量来描述：

$$S=<I_s, P_s, I_d, P_d, P, S_l, S_t, W_n, \Delta_s> \tag{6}$$

其中，I_s、P_s、I_d、P_d 分别表示通信的源 IP 地址、源端口、目的 IP 地址和目的端口。P 是指协议堆栈，它是一个向量。S_l 表示语句的长度。S_t 表示语句的产生时刻，或者被捕获时刻。W_n 描述语句所具有的语义词的数量。Δ_s 表示一个语句的入侵疑似度，通常采用以下函数：

$$\Delta_s = \sum_{j=0}^{w_n-1} \Delta_{wj} \tag{7}$$

其中，Δ_{wj} 表征语句中第 j 个词的入侵疑似度，这样 Δ_s 就表征了一个语句相对于某个被检测的攻击事件所具有的疑似程度。

定义 4　在网络上捕捉到 $j(j \geq 2)$ 台计算机 $C_1, C_2, \cdots, C_j$ 在从事关联通信时产生的一组数据报文序列$\{s_{t_1}, s_{t_2}, s_{t_3}, \cdots, s_{t_n}\}$，称为 $C_1, C_2, \cdots, C_j$ 之间的一组会话，记为 D。$s_{t_1}, s_{t_2}, \cdots, s_{t_n}$ 分别表示在 t_1，t_2，…，t_n 时刻捕获到的报文，即语句，其中 $t_1<t_2<\cdots<t_n$。

用会话入侵疑似度 Δ_D 来描述会话属于某种入侵的疑似程度。除此之外，时间序列会话还包括协议堆栈 P、数据报文数量 N、通信时间长度 T、IP 集合 S_I、端口集合 S_P 等属性。会话用一个属性向量表示

$$D = <P, S_I, S_P, N, T, \Delta_D> \tag{8}$$

为了标识不同的会话，引入会话 Key 概念，记为 DKey，DKey 是能正确区别会话唯一性的一组关键字。在防护目标为 TargetIP 的情况下，DKey 可表述为

DKey=TargetIP/[Mash]+[TargetPort]+[OtherIP/[Mash]]+[OtherPort]+[PS]+[OtherOptionalFlags] (9)

其中，“+”表示连接，“[]”表示可选项。例如，超越用户权限的远程 telnet 攻击是攻击一对一的，取 DKey=TargetIP+TargetPort(23)+OtherIP+OtherPort+PS 可以唯一表示一个这类攻击的事件。又如，分布式对 Http 服务器的 DDos 攻击，取 DKey = TargetIP+TargetPort(80)可唯一区别这类攻击。

会话入侵疑似度 Δ_D 采用如下函数：

$$\Delta_D = \sum_{i=0}^{n-1} \Delta_{si} \tag{10}$$

把式(7)代入，得：

$$\Delta_D = \sum_{i=0}^{n-1} \sum_{j=0}^{w_n-1} \Delta_{wj} \tag{11}$$

一些网络层攻击事件，通常在短时间内集中爆发。以 SYN Flood[16]为列，当攻击发生时，网络中迅速产生大量的 SYN 报文而只有很少的 ACK 报文。由于正常情况下 SYN 和 ACK 报文总是成对出现的。因此，像 SYN 这种可能产生攻击的报文具有攻击性，称为正模式(positive mode)；其对应的 ACK 报文具有抵消攻击性，称为负模式(negative mode)。为了消除因长时间统计累积而造成的误判和增强在短时间内检测的敏感性，引入如下函数对会话入侵疑似度进行评估：

$$\Delta_D = \Delta_i(t, s_i^t) = \begin{cases} \Delta_0, & t = nT \\ (1+\eta)\Delta_{i-1}, & \text{如果} s_i^t \text{是正模式} \\ (1+\eta)^{-1}\Delta_{i-1}, & \text{如果} s_i^t \text{是模式且} \Delta_i > \Delta_0 \end{cases} \tag{12}$$

其中，s_i^t 为 t 时刻捕获到的数据报文，i 为从 t=0 开始的报文计数器，Δ_0、η 为常数。Δ_D 为关于时间 t 和计数器 i 的函数，由于在时间维度上以 T 为周期，因此能有效消除因时间累积造成的误判，而乘数 $(1+\eta)$、$(1+\eta)^{-1}$ 相对于 i 采用的幂乘式则能在较短的周期 T 内有效增强检测敏感度。

设定阈值 γ，MIDM 模型可以采用 IF—THEN 结构来判断入侵：

IF $\Delta_D > \gamma$,THEN Alert or Alert_and_TakeSomeAction

3 扩展 ABNF 范式描述攻击事件

作为能描述上下文无关文法的一种形式语言，BNF 已经被广泛应用于表示编程语言的语法规则。互联网技术规范也经常需要定义一种形式化语法，可以自由地使用作者认为有用的任何符号[17]。多年来，ABNF 范式，即扩充的 BNF 范式，已经在许多互联网规范中流行，在早期的 ARPA 网络中，每一个网络规范都包含了一个自己的 ABNF 范式，并且已形成了 ABNF 描述网络协议的公共引用。ABNF 增加了自有的核心规则[18]，使之更适用于网络协议的定义。

ABNF 从规则命名、规则操作符上都对 BNF 进行了改进和增强。ABNF 操作符有连接(RULE = RULE1 RULE2)、选择(RULE=RULE1\RULE2)、增试选择(RULE=\RULE)、值域选择(RULE=% C##-##)、有序组((RULE1 RULE2))、循环(<a>*<b> element)、指定循环(<n>element)、可选序列([Rule])和注释(;)等。

为了使 ABNF 范式适用于入侵检测，本文增加了一条排除(~)操作。RULE1~RULE2 表示在 RULE1 出现时 RULE2 不出现。经上述扩展，MIDM 模型所定义的攻击事件，就可以使用扩展的 ABNF 范式来描述。

例如，对于 Http 服务器的攻击在分类中是属于应用层攻击事件的一种。仅仅利用目录 cgi-bin 就可以进行多种攻击[19]。黑客利用该目录下的 cgi 可执行程序，就可以执行多种针对 Web 服务器的攻击，甚至可以直接穿透现代 Web 架构中的 cache 机制进行攻击。在研究中，把这类攻击统事件称为 CGI_AE。假设受攻击的服务器 IP 为 DestIP，使用 80 或 8080 端口提供服务。下面利用扩展的 ABNF 范式定义 CGI_AE 在 Get 或 Post 请求时进行攻击的部分：

CGI_AE = 1* CGI_Attack_Pck; 用"Pck" 标识报文

CGI_AE_Pck=Pre_Part IPHeader TCPHeader AppData;用"Part"标识分析时不关心的部分

IPHeader = Ver 4BIT 3OCTET 4BIT Tcp 6OCTET DestIP[Post_Part]

Key = DestIP Dest_Port

Ver = %b 0.1.0.0; %Http 版本号，IPv4

Tcp = %b 0.1.1.0; %d6

TCPHeader = (2*2OCTET) (Dest_Port) (16*16OCTET)

Dest_Port=%x00.50/0x1f.90; 端口 80 或 8080.

AppData=(GorP_Features SP Post_Part)/(Pre_Part CRLF Ref_Features CRLF PostPart); "Features" 标识复合规则

N1_Sep = "=" ; "Sep"标识分隔符

N2_Sep = "/"

N1_Feature = "cgi-bin/" / "cgi-bin/" ;"Feature"标识单一规则

GorP_Features

=*LWSP("Get"/"Post")N1_Feature[*(VCHAR~

N1_Sep~LWSP) N1_Sep Optional_Features]

Optional_Features =[Optional1_Feature]\[Optional2_Feature]\…

Optional1_Feature = "./" / "../" / ".../"　;企图获取超级权限

Optional2_Feature = "`" ;"`" 非法请求

Optional3_Feature=/1*VCHAR("ls%20"/"rm%20"); 超越权限

Optional4_Feature=/1*VCHAR ("%20"/"% 00"); 非允许请求

又如，在网络级攻击 SYN Flood 中，攻击者通常使用大量的虚假 IP 不断向服务器发出 SYN 请求，在收到服务器的 SYN-ACK 报文后却不做应答，造成服务器花费大量时间和空间去处理虚假的半连接列表，因缓冲区溢出而拒绝服务，甚至导致系统崩溃。检测这种网络层的攻击需要短时间内进行正确的统计。由于正常三握手链接中 SYN 请求(正模式)和 ACK 回应(负模式)成对出现，攻击事件在短时间段内集中爆发，对该事件的评估函数可以使用式(12)。同时研究人员对大量的正常三握手报文做统计发现，正常 ACK 报文的 Window size 都是 65 535，而建立连接后不出现 Windows size 为 65 535 的 ACK 报文。这样对一个目标服务器 IP 为 DestIP 的 SYN Flood 事件，令 Key=DestIP，用扩展的 ABNF 范式定义如下：

SYN_Flood_AE = 1*((m* Three_HS_SYN_Pck) /(n* Three_HS_ACK_Pck));当攻击发生时，m 远大于 n

Three_HS_SYN_Pck = IP_Header TCP_Header1

Three_HS_ACK_Pck= IP_Header TCP_Header2

IP_Header = Version 4BIT 3OCTET 4BIT Tcp 6OCTET Key[Post_Part]

Version = %b 0.1.0.0

Tcp = %b 0.1.1.0

Key = DestIP

TCP_Header1 = 12OCTET 10BIT Flag1_Feature Post_Part

TCP_Header2 = 12OCTET 10BIT Flag2_Feature Window_size_Feature Post_Part

Flag1_Feature = %b 0.0.0.0.1.0

Flag2_Feature = %b 0.1.0.0.0.0

Windows_size_Feature=16%b1;通常在三握手 ACK 报文中为 65 535

4　基于 MIDM 模型的入侵检测系统 ERIDS

图 3 是基于 MIDM 模型实现的入侵检测系统原理图，称为事件—规则入侵检测系统(ERIDS, event- rules intrusion detecting system)。形式化的入侵事件定义经过规则警报模块(RAF, rule alert function)和事件报警模块(EAF, event alert function)后，被翻译成一组由 C++语言编写的原子规则和入侵事件类，并将事件类作为关联原子规则的整体存入知识库。

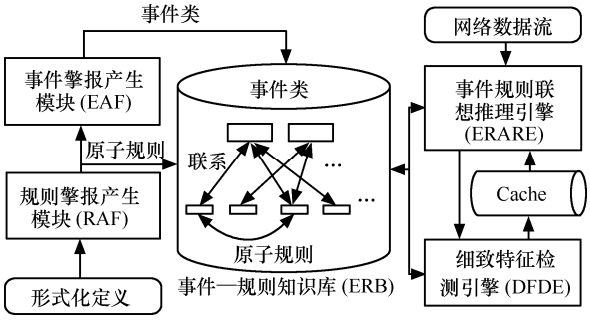

图 3　事件—规则生成和入侵检测系统(ERIDS)

事件—规则知识库(ERB, event-rule base)不仅存储松散的规则，还存储着事件类以及事件实例与规则之间的联系，它为检测引擎提供了高效的规则联系。事件类按协议分层存储在一个称为事件树的结构里。不同事件类可共享子规则，进而有效降低知识库的特征冗余。

事件—规则联想推理引擎(ERARE, event-rule associating and reasoning engine)是检测部分的核心模块之一。它把可能是入侵的事件都记录在一个实例事件列表中。在检测时，ERARE 先对报文的 Key 做散列，试图将它定位到具体的实例事件中。运算过程可能引发 2 种行为：存在实例事件 Key 与检测 Key 匹配时，ERARE 依据实例事件记录的联系进行推理，得出可能关联的子规则；在不存在实例事件与 Key 相匹配的情况下，依据知识库中的事件类—规则树进行推理，得出关联规则；也可能不存在关联规则，则报文安全通过。

细致特征检测模块(DFDE, detail feature detection engine)对报文执行详细检测。在 ERARE 推理的第一种情况下，DFDF 依据得出的原子规则调用相应的评估函数进行检测，并把结果和统计量记录到实例事件列表中，调用会话评估函数判断是否应

该报警。第二种情况下，DFDF 依据得出的原子规则调用评估函数进行检测，在出现可能入侵时，*D* 将在实例事件列表中创建可疑事件类的实例，把结果和统计量记录到实例事件中，并判断是否应该报警。ERARE 和 DFDF 之间有一个高速缓存空间。用于存储入侵疑似度超过某一阈值的事件关键信息。ERARE 在检测 Key 时首先与 Cache 中的 Key 做比较，有效提高了检测速度。

5 实验结果

实验环境为多台实验室内计算机组成的吉比特网络。其中的 Linux 系统装有 Web 服务器 Ftp 服务器等，模拟被攻击的服务器。实验使用了 3 份数据，其中一份数据来自 MIT 林肯实验室，较具权威性。其余 2 份数据是研究人员获取和制作的，各种主要属性列在表 1 之中。

表 1 实验数据

数据集名称	来源	所含攻击	用途	大小
DataSetA	MIT 林肯实验室	Web 入侵、DDos、TCPScan	检测样本	11.2GB
DataSetB	省际主干网	未知	检测样本、压力测试样本	142.0GB
DataSetC	本文研究者在实验室内的模拟攻击	SYN Flooding、TCP Scan、CGI 攻击、木马、蠕虫	部分数据作为规则提取样本、全部数据做检测样本	1.4GB

当前许多研究都基于 Snort[20]进行，但 Snort 模型结构自身的一些缺陷，使得那些研究无法很好地解决 NIDS 面临的主要问题。因此将 MIDM 模型首先与 Snort 进行比较是必要的。

图4是ERIDS与Snort2.8.0进行比较的一个结果。实验中用了 IBM 的 Purify 进行性能测试。它分别列出了在检测不同数量的攻击事件类时，Snort 需要的规则数和 ERIDS 时的事件数量和规则数量，ERIDS 生成的规则数大约只是 Snort 的 1/2。显然，ERIDS 通过增强规则联系，有效降低了规则冗余。Snort 的某些柱状图没有给出，因为在那些情况下它已无法正常运行。显然，ERIDS 的 CPU 利用率并不像 Snort 那样随着规则数量和流量的增长而显著增长，因此更适用于高速网络。

表 2 显示了当阈值 γ 取不同值时的误报率(FPR)。从实验数据中可以看出，当 α=0.6 时，可以准确检测出攻击而且使误报率下降到 0。

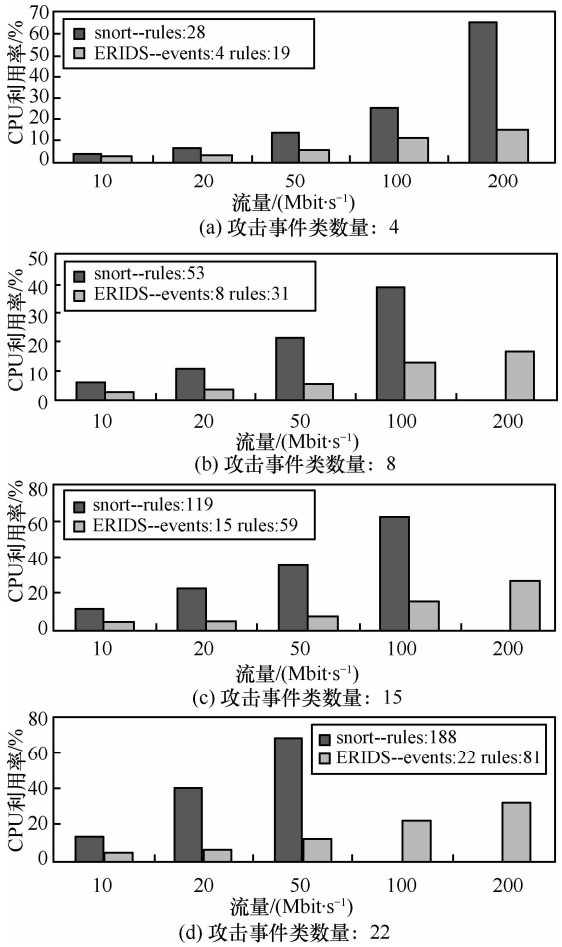

图 4 ERIDS 与 Snort 的对比测试

表 2 γ 取不同阈值时 ERIDS 的误报率

γ	SYN flooding	TCP Scan	CgiAttacks
0.4	0.037 9	0.025 5	0.061 3
0.5	0.007 2	0.007 7	0.007 4
0.6	0.000 0	0.000 0	0.000 0
0.8	0.000 0	0.000 0	0.000 0
平均查全率	0.992 0	0.992 8	1.000 0

本文还将 ERIDS 与 Snort 和李秀婷[10]基于 Snort 的“状态协议分析”改进法就检测 DDOS 进行了比较。实验采用分布式 SYN_Flooding 作为被检测的攻击，并将 Linux 服务器 tcp_max_syn_backlog 参数的值设置为 4 096，网络中其他计算机在 2s 内对其完成攻击。结果如表 3 所示。

可以看出，ERIDS 的有效检测率优于 Snort 及其改进方法。表中的剩余预警时间项表示在系统受到警报后可以采取行动的剩余时间，显然，ERIDS 为系统争取了更多的剩余预警时间。

表 4 分别使用 Snort 和 ERIDS 检测木马、蠕虫

表 3　　ERIDS 与 Snort 及其改进方法检测 DDoS 的对比

检测工具	实际攻击数	截获攻击数	有效检测率	攻击完成时间/s	剩余预警时间/s
原版 Snort 2.8.0	4 096	3 214	0.784 6	2.00	0.82
状态协议分析法	4 096	4013	0.979 7	2.00	1.16
ERIDS	4 096	4 069	0.993 4	2.00	1.35

及其变种时出现的误检率(FPR)和漏报率(FNR)的情况，表中的原版是指用于生成规则的版本。通过对比可以看出， Snort 误检率较高，而 ERIDS 则具有高准确性。还可以看出，ERIDS 有效支持特征泛化，对木马和蠕虫还具有较强的检测变种的能力。

表 4　使用 Snort 和 ERIDS 检测木马和蠕虫及其变种的比较

木马/蠕虫	Snort 误检率	Snort 漏检率	ERIDS 误检率	ERIDS 漏检率
木马 1 原版	0.246	0.000	0.000	0.000
木马 1 变种	0.382	0.154	0.000	0.010
木马 2 原版	0.294	0.000	0.000	0.000
木马 2 变种	0.453	0.194	0.000	0.009
蠕虫 1 原版	0.148	0.000	0.000	0.000
蠕虫 1 变种	0.340	0.892	0.081	0.291

实验表明：ERIDS 具有较好的特征泛化能力和更低的误检率和误报率；大幅减少了特征冗余；提高了检测速度和准确性，更适应高速网络环境。

6　结束语

为了解决 NIDS 面临的主要问题，本文在总结前人研究的基础上，对基于网络的入侵检测系统重新建模，提出了基于事件和深度协议分析的入侵检测系统数学模型(MIDM)。与现有只是将协议分析局限在个体报文，并且只对非应用层数据做协议分析的方法不同，MIDM 模型不仅将网络事件作为入侵检测中的综合整体，对网络事件进行层次分析；同时还将协议分析的方法深入到应用层的细节之中。它综合特征检测和异常检测两大类方法的优点，通过层与层之间的评估函数实现对网络数据的分析与综合。本文还对 ABNF 范式进行了扩展，对入侵事件进行了形式化定义。在上述研究的基础上，实现了事件—规则联系的入侵检测系统 ERIDS。通过与当前主流入侵检测系统及其改进方法的比较，实验结果表明基于 MIDM 模型构建的 ERIDS 具有更高的检测精度和检测速度，大幅降低了规则库冗余，更适合于高速网络环境，同时还具备一定的检测未知入侵的能力。

本文的重点是为了说明 MIDM 模型的和基于该模型构建的入侵检测系统的优越性，因此系统中只使用了和 Snort 同量级的 BM 单模式匹配算法。后续的研究将会基于 MIDM 模型结合多模式匹配算法[11~13]进一步提高 ERIDS 系统的检测速度和准确度。

参考文献：

[1] WHITMA N, MMICHAE L, MATTORD H. Principles of Information Security[M]. Canada: Thomson, 2009. 290 -301.

[2] JAN N Y, LIN S C, TSENT S S, *et al*. A decision support system for onstructing an alert classification model[J]. Journals of Expert Systems with Applications, 2009, 36(8): 11145-11155.

[3] LINDQIVST U, PORRAS P A. Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST)[R]. IEEE Symposium on Security and Privacy, Oakland, 1999. 146-161.

[4] 徐明, 陈纯, 应晶. 一个两层马尔可夫链异常入侵检测模型[J]. 软件学报, 2005,16(2):276-285.
XU M, CHEN C, YING J. A two-layer Markov chain anomaly detection model[J]. Journal of Software, 2005,16(2):276-285.

[5] JO O B D, RAVICHANDRAN B. Statistical traffic modeling for network intrusion detection[A]. Eighth IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'00)[C]. San Francisco, 2000.

[6] WUU L C, HUNG C H, CHEN S F. Building intrusion pattern miner for Snort network intrusion detection system[J]. Journal of Systems and Software, 2007,80(10):1699-1715.

[7] NEHINBE J O. Automated technique for debugging network intrusion detection systems[A]. IEEE 2010 International Conference on Intelligent Systems, Modelling and Simulation (ISMS)[C]. Liverpool, 2010. 362-367.

[8] CUPPENS F, MIEGE A. Alert correlation in a cooperative intrusion detection framework[A]. Proceedings of IEEE Symposium on Security and Privacy[C]. Berkeley, 2002.

[9] NEHINBE J O. A simple method for improving intrusion detections in corporate networks[A]. International Conference on Information Security and Digital Forensics[C]. London, 2009.

[10] 李秀婷. 基于 Snort 的入侵检测系统实现及其改进研究[D]. 西安: 西安电子科技大学, 2008.
LI X T. Research on Implementation and Improvement of Instruction

Detection System Base on Snort[D]. Xi'an: Xi'an Electronic and Science University, 2008.

[11] DHARMAPURIKAR S, LOCKWOOD J W. Fast and scalable pattern matching for network intrusion detection systems[J]. IEEE Journal on Selected Areas in Communications, 2006,24(10): 1781-1792.

[12] 陈一骄. 网络入侵检测系统高速处理技术研究[D]. 长沙：国防科技大学, 2007.
CHEN Y J. Researches on High-Speed Processing for Network Intrusion Detection Systems[D]. Changsha: National University of Defense Technology, 2007.

[13] 黄建. 入侵检测系统中字符串匹配算法与实现[D]. 武汉：华中科技大学, 2008.
HUANG J. Algorithms and Implementation of String Match in Intrusion Detection System[D]. Wuhan: Huazhong University of Science and Technology, 2008.

[14] ABBES T, BOUHOULA A, RUSINOWITEH M. Protocol analysis in intrusion detection using decision tree[A]. Proeeedings of the Inierna-tional Conference on Information Technology: Coding and Comput-ing(ITCC' 04)[C]. Los Alamitos, USA, 2004. 404-409.

[15] DREGER H, FELDMANN A, MAI M. Dynamic application-layer protocol analysis for network intrusion detection[A]. 15th USENIX Security Symposium[C]. Vancouver, 2006. 257-272.

[16] EDDY W, VERIZON. TCP SYN flooding attacks and common mitigations[EB/OL]. http://tools.ietf.org/html/rfc4987,2007.

[17] CROCKER D, OVERELL P. ITEF Network Working Group, Augmented BNF for Syntax Specifications: ABNF (January 2008)[S]. RFC Editor, 2010.

[18] Augmented backus-naur form[EB/OL]. http://en.wikipedia.org/wiki/ABNF, 2010.

[19] Fingerprinting Port 80 Attacks: a look into Web server, and Web application attack signatures[EB/OL].http://www.cgisecurity.com/papers/fingerprint-port80.txt, 2001.

[20] ALDER R, CARTER E F, FOSTER J C. Snort: IDS and IPS Toolkit[M]. Burlington, Canada: Syngress Publishing, 2007.

作者简介：

朱映映（1976），女，山东临沂人，博士，深圳大学副教授、硕士生导师，主要研究方向为多媒体信息处理、图像处理、语音信号处理、网络安全。

吴锦锋（1983），男，广东清远人，深圳大学硕士生，主要研究方向为入侵检测、网络安全。

明仲（1967），男，江西宁都人，博士，深圳大学计算机与软件学院教授、硕士生导师、副院长、主要研究方向为本体论、语义Web、面向对象软件工程和形式化方法。