计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
50(3): 498-508, 2013

基于上下文验证的网络入侵检测模型

田志宏　王佰玲　张伟哲　叶建伟　张宏莉
(哈尔滨工业大学计算机科学与技术学院　哈尔滨　150001)
(tianzhihong@hit.edu.cn)

Network Intrusion Detection Model Based on Context Verification

Tian Zhihong, Wang Bailing, Zhang Weizhe, Ye Jianwei, and Zhang Hongli
(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin* 150001)

Abstract　Network intrusion-detection systems (NIDSs) are considered an effective second line of defense against network-based attacks directed to computer systems. Because of the increasing severity and likelihood of such attacks, the NIDSs are employed in almost all large-scale IT infrastructures. The Achille's heel of NIDSs lies in the large number of false positives. However, today's NIDSs often try to detect not only intrusions, but also successful intrusion attempts. This is because it can be difficult for an NIDS to determine the result of an intrusion attempt. A popular approach of verifying intrusion attempt results is to let an IDS be aware of the environment and configuration of the systems under attack. Based on the above idea, in order to eliminate the negative influence on IDS stability caused by non-relevant alerts, a network intrusion detection model is designed based on context verification. With the combination of environment context, weakness context, feedback context and anomaly context, our model constructs an effective, stable, integrated, and extendable non-relevant alerts processing platform which focuses on context verification and integrates multiple security techniques. It achieves the automatic validation of alarming and automatic judgments of their effectiveness to eliminate the non-relevant alerts, and thus it establishes the reliable foundation for alerts association.

Key words　intrusion detection; context; non-relevant positives; false positives; context verification

摘　要　大量误报引发的可信问题一直是入侵检测研究领域所面对的具有挑战性的未解技术难题之一.为了提高入侵检测系统的确定性和准确性,必须对其告警信息加以区分,滤除无效攻击导致的虚警,从而自动准确地识别有效攻击.由此,提出了一种基于上下文验证的网络入侵检测模型,结合环境上下文、弱点上下文、反馈上下文和异常上下文等多种上下文信息,构建了一个以上下文为中心、多种验证技术相结合的高效、稳定、完整、易管理、可扩充的虚警处理平台,实现了告警的自动验证以及攻击行为能否成功地自动判定,从而达到滤除虚警的目的,使入侵检测系统起到真正的预警作用.

关键词　入侵检测;上下文;虚警;误报率;上下文验证

中图法分类号　TP393

对入侵检测系统的研究始于1980年Anderson为美国空军作的一份题为"Computer Security Threat Monitoring and Surveillance"的技术报告,之后迅速发展出多个分支,近年来更是得到长足的发展.然

收稿日期:2011-04-01;**修回日期**:2011-12-05
基金项目:国家自然科学基金项目(60903166);中央高校基本科研业务费专项资金项目(HIT.NSRIF.2010041)

而,大量误报引发的可信问题却一直是入侵检测研究领域所面对的具有挑战性的未解技术难题之一,产生此问题的原因很多,主要可分为2类:第1类原因是攻击特征描述不完善或IDS自身存在算法或分析方法的缺陷造成的,它可以通过提高攻击特征描述及检测算法的准确性加以解决;第2类是情况发生于攻击者发动了对具体攻击目标并不构成任何危害的某种攻击时,例如:CodeRed蠕虫病毒利用Microsoft IIS缓冲区溢出漏洞进行传播,而IIS服务只能运行于Windows操作系统环境,而Linux类操作系统并不存在,因此CodeRed蠕虫对Linux主机的攻击不会产生任何威胁.然而,由于运行中的IDS并不清楚自己保护的服务器的具体操作系统类型,于是一个正在扫描运行Apache服务的Linux Web服务器的CodeRed蠕虫病毒就会被IDS判定为攻击并产生虚警(non-relevant positives).虚警显然并不为安全管理员所关心,但大量存在的虚警势必影响IDS的精确度、有效性和可用性,并最终引发安全管理员的信任危机.

1 问题描述

网络安全是是一个整体,因此不能放弃网络的整体安全状况,孤立地看待IDS的告警结果.由于网络具有动态变化的特性,用户随时都有可能升级软件版本、下载漏洞补丁或者更新防火墙配置策略,而这些都会对IDS告警正确性产生影响.文献[1]指出,一般情况下,漏洞、攻击和补丁符合如图1所示的时序关系,即漏洞公布和出现相应攻击的相隔时间极其短暂,补丁和IDS特征的出现往往由于开发、测试等环节而略微滞后,但是真正等到IDS应用规则检测相应攻击时,大部分主机都已打过补丁,此时IDS检测到的攻击肯定是无效的,虚警由此产生.

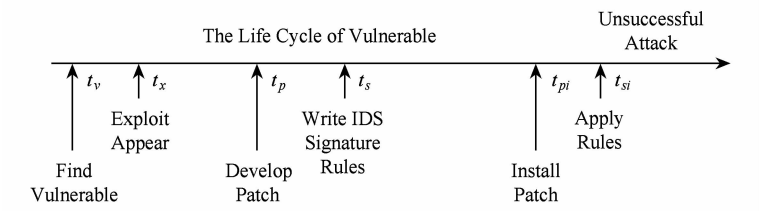

Fig. 1 Time line of vulnerability, exploit, patch, and signature.

图1 漏洞、攻击和补丁的时序关系图谱

显然,如果IDS能够在输出告警的同时,判定攻击是否成功,就不会存在上述问题.然而,当前的IDS大都以网络报文探测引擎和主机日志探测引擎为主要事件检测来源,这种技术路线的局限性就是对攻击事件的有效性缺乏控制管理,因此容易产生虚警,影响检测效能.

为了提高IDS告警信息的确定性和准确性,必须对IDS的告警信息加以区分,自动准确地识别有效攻击并滤除虚警,从而使IDS告警真正起到预警的作用.由此,本文侧重于虚警的研究并提出相应解决方案.

2 相关工作分析

作为IDS领域的热点研究问题,如何优化体系结构和最大限度地提高检测算法的效率一直是学者们竞相追逐的目标.但对于虚警,则少有人问津.

为了提高网络入侵检测系统Bro的检测能力,Sommer等人[2]最早提出了上下文特征(context signature)的概念,并利用功能更为强大的正则表达式(regular expression)替代传统的结构简单的攻击特征描述方法,提供了多规则关联和规则与策略脚本交互的高级特性,从而将攻击的上下文信息融入到攻击特征中.用户可根据实际需求手工编写相应脚本来定制告警的关联方法,但脚本语言过于复杂,需具备专业的知识方可熟练掌握.

文献[3-5]则采用告警验证的思想,为安全管理员提供了告警过滤和事后验证功能,实现了更为高级的安全管理方案.此类方法利用风险评估软件,如Nessus[6],预先对网络中所有信息资产的弱点进行扫描检查,分类整理后存入环境资产数据库,并通过有效的策略互动和检测事件过滤等方法,以CVE-ID[7]作为索引对IDS告警进行实时验证,如果被监控主机不存在某个攻击可以利用的弱点,IDS将抑制告警的产生,从而有效地去除了由于弱点不匹配造成的虚警.

Almgren 等人[8]通过检查“404 Not Found”信息来确认 CGI 脚本不存在的方法,来检测针对 Web 服务器的失效攻击. Zhou 等人[9]则采用一种轻量级协议分析的方法分析攻击对应用协议产生的结果,以此来区分攻击的有效性. WebSTAT[10]提出一种针对 Web 服务器的缓冲区溢出攻击验证方法,他们认为成功实施的攻击不会在服务程序日志中留下任何记录,因此通过关联攻击行为和日志文件内容来验证缓冲区溢出攻击是否有效.

蒋建春等人[11]在分析了攻击各阶段的特点及其相互依赖关系之后,提出了一种基于攻击上下文的有效攻击验证算法,并同时指出,没有通用的上下文能解决所有的网络攻击,而应根据网络环境所面临的攻击威胁选用相应的攻击上下文. 相应实验结果表明,该算法在充分利用攻击环境、攻击操作前后关联等上下文信息的基础上,能够有效地检测真实的入侵行为,同时提高 IDS 的准确性.

针对虚警的研究还处于起步阶段,上述方法均只采用了一些尝试性的技术手段,缺乏面向滤除虚警的系统化研究. 基于上述考虑,吸取上述方法的各自优点,在已有研究工作的基础上[12-14],本文提出了一个基于上下文验证的网络入侵检测模型(context verification based network intrusion detection model, CVNIDM),结合环境上下文、弱点上下文、反馈上下文和异常上下文等多种上下文信息,CVNIDM 构建了一个以上下文为中心,多种验证技术相结合的高效、稳定、完整、易管理、可扩充的虚警处理平台,实现了告警的自动验证以及有效攻击的自动判定,从而达到滤除虚警的目的,使告警起到真正的预警作用.

3 CVNIDM 的上下文定义

网络活动不能孤立地理解,必须根据用户的意图、网络环境的现有状态及用户行为历史记录来解释. 因此对于 IDS 来说,攻击行为是上下文相关的,传统方法对告警的反应特性已经越发无法满足目前的安全需求,不能片面地根据网络活动的局部信息就判断网络用户行为是否具有入侵性,而应充分地利用具体的上下文信息,对于不同的攻击手段采取不一样的处理措施. 也就是说,对于网络行为非法性判定必须受到上下文的约束. 例如:我们常常难以确定某个用户访问某个敏感的服务器是否是入侵行为. 但是,如果根据用户访问时间、访问的源 IP 地址、服务器的安全策略等上下文信息,就能够有助于判定入侵行为. 假设用户所对应的上下文信息是“超级用户、源 IP 地址未经授权、访问时间又是凌晨 1 点,且服务器的安全策略是禁止超级用户远程登录”. 那么我们就能够准确地识别出该用户的远程访问动作是入侵行为. 下面给出上下文的相关定义[11]:

定义 1. 上下文是指攻击行为与相关信息的综合体,主要包括攻击行为对象、攻击时间、攻击发生外部事件、攻击环境、攻击活动的前后关系、攻击活动的网络情况以及攻击行为效果等.

根据上下文的概念,以适应不同类型攻击的检测需求、同时提高 IDS 的确定性和准确性为目的,CVNIDM 扩展了过滤规则,通过分析攻击与攻击环境之间的依赖关系,根据攻击动作及其前后关联的上下文信息,在告警产生的同时进行相应的攻击上下文验证,并以此为基础来发现有效的网络攻击行为.

因此,上下文信息是 CVNIDM 的核心概念,是判定虚警的主要依据,CVNIDM 的上下文主要包括环境上下文、弱点上下文、反馈上下文和异常上下文 4 种:

1) 环境上下文主要指 CVNIDM 所监控的网络环境,具体包含网络的拓扑信息(HOP 数和路径的 MTU)、路由器或者防火墙的配置规则情况、各主机 TCP/IP 协议栈的参数设置、所运行的服务以及操作系统的类型等信息;

2) 弱点上下文则涵盖所监控主机或者网络设备的所有已知弱点信息,主要依靠风险评估软件的主动扫描获得;

3) 反馈上下文则描述某些攻击如果无法成功作用于目标服务,目标服务将会产生的反馈信息,它主要用于判定攻击是否失效. 例如:攻击者利用 Microsoft IIS CGI 文件名错误解码漏洞(CVE-2001-0333)企图调用 Windows 命令行解释器 cmd. exe,如果攻击成功,IIS 服务器一般会返回“200 OK”,反之,IIS 服务器一般会返回“4XX”的 HTTP 响应码来表示访问错误. 这里的响应码“4XX”就是相应攻击的反馈上下文;

4) 异常上下文的定义基于如下思想,“成功作用于主机的攻击会引发主机行为的某种异常[15],因此,异常上下文实际上就是受监控主机的流量、连接数等信息的统计简档,它的作用是用来判定主机活动是否异常,从而推断攻击是否生效.

4 CVNIDM 的有限状态机模型

作为理论基础,下面给出 CVNIDM 框架的形式化描述,CVNIDM 可以抽象为一个有限状态机模型,相关的符号定义参见表 1 所示:

Table 1 Symbol Definnation for CVNIDM State Machine

表 1 CVNIDM 有限状态自动机的符号定义

Symbol	Definition
$Event=\{e_1,e_2,\cdots,e_n\}$	Network events set
$Signature=\{s_1,s_2,\cdots,s_m\}$	Signature set for misuse detection
$Profile=\{p_1,p_2,\cdots,p_i\}$	Normal profile set for anomaly detection
$Alert=\{a_1,a_2,\cdots,a_j\}$	Alerts set
$Context=\{c_1,c_2,\cdots,c_k\}$	Attack context set

基于上述符号定义,CVNIDM 可用自动机$\langle S,U,T,s_0\rangle$来表示,其中 S 为有限状态空间的集合,输入变量集合 $U=Event$,状态迁移函数集合 $T=\{t_0,t_1,\cdots,t_k\}$,$s_0\in S$ 表示初始状态.

CVNIDM 的状态转换如图 2 所示,其中 s_1 表示检测状态,s_2 表示告警状态,s_3 和 s_4 为终态,分别表示虚警和有效告警状态. 与传统的 IDS 不同,CVNIDM 的终态并非 s_2 标识的告警状态,这主要是由于 CVNIDM 需要对告警进行上下文验证,实际上,完全可以认为 CVNIDM 是对传统 IDS 功能的扩展.

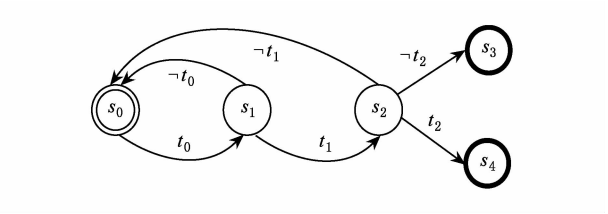

Fig. 2 CVNIDM state machine model.

图 2 CVNIDM 的有限状态机模型

此外,所有状态迁移函数均为真值表达式,$t_0=\{U_i=\varnothing\}$描述是否有待分析的网络事件,$t_1=\{U_i=(Signature\cup\overline{Profile})\}$表示输入事件 U_i 是否具有攻击特征或是异常行为,$t_2=\{Alert\Leftrightarrow Context\}$是上下文验证函数,利用上一节描述的 4 种上下文信息对处于 s_2 状态的告警进行验证,最终依据验证结果判定是否为虚警. 显然,上下文验证函数 t_2 是 CVNIDM 的核心.

5 基于上下文验证的网络入侵检测

与有限状态机模型相符,CVNIDM 的系统结构设计如图 3 中虚线框所示. 其中,检测引擎和传统的 IDS 功能相同,主要负责判别捕获到的网络数据是否与攻击特征库的特征相匹配或者偏离行为模式库中的正常行为轨迹,从而产生相应的告警数据流. 而上下文验证模块则调用上下文验证函数 t_2,利用各种上下文信息对告警流进行验证.

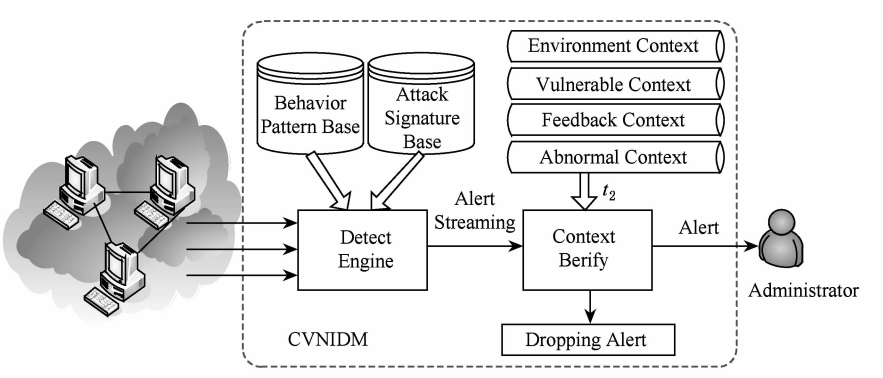

Fig. 3 Structure model of CVNIDM.

图 3 CVNIDM 的系统结构模型

根据验证方式的不同,上下文验证函数 t_2 主要分为 4 类,分别是基于多属性映射的静态上下文验证(multi-attribute mapped static conetxt verify, MMCV)、基于弱点关联的主动上下文验证(vulnerability correlation based active conetxt verify, VCCV)、基于攻击反馈的被动上下文验证(attack feedback based passitive conetxt verify, AFCV)以及基于 Chi-square 检验的异常上下文验证(chi-square abnormality conetxt verify, CACV). t_2 的返回值为布尔型,CVNIDM 根据 t_2 返回值的逻辑或运算结果决定虚警,下面对上述 4 种验证函数加以讨论.

显然,CVNIDM 的优势体现在其掌握了大量的目标系统背景知识,如主机的弱点情况、网络的拓扑信息及系统状态等.对于复杂的大规模网络系统,大量主机将导致需关联分析的数据剧增,为适应大规模网络系统入侵检测的需求,CVNIDM 支持分布式协作的部署方式.可根据网络规模决定部署 CVNIDM 的数量.此外,4 种上下文验证函数采用插件式设计,可根据 CVNIDM 的负载自由设定上下文验证函数的组合方式.采用分治策略降低数据处理规模.

5.1 基于多属性映射的静态上下文验证

传统 IDS 并不掌握被检测目标的诸如弱点缺陷和业务应用情况等具体信息,其事件检测能力主要依赖于特征库的完备性和协议报文分析能力,事件检测引擎的策略并没有针对性,对于只能在特定目标存在弱点或特定应用服务环境下才能有效的攻击事件,很容易产生虚警,例如 Win95 上的 DOS 事件、第三方应用 gftpd 的远程溢出事件或者 Windows 办公环境中的 Unix 类远程溢出事件等.为了解决目标主机和传统 IDS 在理解网络信息时所产生的分歧,CVNIDM 融合扫描技术和 CVE 弱点库对被监控网络的安全状况预先进行扫描检查,收集环境上下文和弱点上下文信息,生成多属性映射(multi-attribute mapping),并依此对每个告警信息执行一系列的验证操作.

多属性映射描述了告警和攻击上下文之间的某种对应关系,在 CVNIDM 中,这种对应关系以谓词的形式来表述.所谓谓词就是以上下文信息为输入参数,如果关系成立则返回 True,否则返回 False.例如,谓词 ExistHost(172.16.9.158)决定了目标网络是否存在 IP 地址为 172.16.9.158 的这台主机;而谓词 VulnerableBufferOverflow(172.16.9.158) 则决定了主机 172.16.9.158 是否存在缓冲区溢出漏洞.另一方面,仅仅使用一个谓词未必能够完整地描述出攻击与上下文的对应关系,因此,为了适应某些复杂的上下文表示,我们将谓词扩充为谓词的逻辑表达式形式,并用符号 AND、OR、NOT 分别表示与、或、非 3 种条件.于是,谓词"ExistService(172.16.9.158,111) AND VulnerableSadmind(172.16.9.158)"则表示主机 172.16.9.158 在 111 端口是否开放 Portmap 服务,而且同时具有相应的 Sun Solaris 操作系统 Sadmind 服务远程执行命令漏洞.

下面对由于环境上下文和弱点上下文的因素导致虚警的情况各举一例,分别说明 MMCV 的具体验证过程:

情景 1. 一般来讲,IDS 会在路由器或者防火墙之外,而受监控主机却在路由器和防火墙的内部,因此就会导致外部网络数据报文到达 IDS 和受保护主机的跳数不同,因此,数据报文中的生存时间选项(TTL)就决定了此报文能否到达最终的目标主机.于是,攻击者就会利用这一点,通过精心构造的具有不同 TTL 值的数据报文致使 IDS 产生告警,但目标主机却不会收到这些报文,以此来愚弄安全管理员,以期达到造成其麻痹心理,降低警觉性的目的.

然而,对于 CVNIDM 来说,情况则完全不同.由于事先已经通过扫描等操作了解了网络拓扑以及路由器或者防火墙的配置规则情况,CVNIDM 会采用相关的谓词操作对告警信息进行上下文验证,因此并不存在上述问题.例如,目标主机 172.16.9.158 与 CVNIDM 的跳数相差 1,也就是说,该主机和 CVNIDM 之间存在一台路由器,假设攻击者与 CVNIDM 的跳数相差 5,则如果攻击者向目标主机发动某种攻击,而数据报文的 TTL 选项却设置为 5,显然这些数据报文在到达主机 172.16.9.158 之前就会被路由器丢弃.对此,CVNIDM 会首先产生相应的告警,同时,对谓词 ExistHost(172.16.9.158)进行判断,如果谓词返回值为 False,说明受保护网络内部并无此 IP 地址,直接将该告警丢弃;接下来,根据谓词 EqualHops(172.16.9.158)来判断攻击者的源 IP 地址(根据告警信息内容可得)与主机 172.16.9.158 以及 CVNIDM 之间的跳数是否相等,如果该谓词返回为 False,说明该告警为虚警并予以丢弃.上述一系列谓词操作可表示成如下逻辑形式:ExistHost(172.16.9.158) AND EqualHops(172.16.9.158),只有当该逻辑谓词表达式的结果为 True,此告警才会作为有效告警报告给安全管理员.

情景 2. 最常见的虚警大多是由于攻击目标与攻击行为本身不相符合造成的,例如,操作系统的类型不一致或者目标主机并不存在相应弱点甚至并没有运行相应服务等.

以 CodeRed 蠕虫病毒为例,如果其扫描了 IP 地址为 172.16.9.158 的这台主机并触发了相应的告警信息,CVNIDM 会首先对谓词 ExistHost(172.16.9.158)进行判断,如果谓词返回值为 False,说明受保护网络内部并无此 IP 地址,直接将该告警丢弃;进而根据谓词 ExistServiceIIS(172.16.9.158)判断主机上是否运行有 Microsoft IIS 服务进程,如果谓词返回 True,则根据谓词 VulnerableIISExploit

(172.16.9.158)判断是否存在 IIS 缓冲区溢出漏洞.上述一系列谓词操作可表示成如下逻辑形式:"ExistHost(172.16.9.158) AND ExistServiceIIS (172.16.9.158) AND VulnerableIISExploit (172.16.9.158)",只有当该逻辑谓词表达式的结果为 True,此告警才会作为相关告警报告给安全管理员.

显然,MMCV 除需要预先收集被监控网络的相关上下文信息之外,并不需要任何其他额外的操作.攻击所触发的告警只需经过几次谓词判断之后,即可立即上报给安全管理员,因此具有延迟低、开销小的优势.然而,由于多属性映射是静态生成的,而多属性映射本身却又完全依赖于动态的网络环境,因此,如果使这种静态的处理方式能够适应网络环境实时变化的需求,必须频繁地更新多属性映射信息,但是这是以耗费大量人力和物力资源为代价的.

5.2 基于弱点关联的主动上下文验证

据著名的计算机安全事件响应小组 CERT/CC 的统计数据表明,弱点的数量有逐年升高的趋势,图 4 给出了 CERT/CC 每年报告的弱点数量,从中可以看出,截止至 2005 年 12 月,共报告弱点总数已达 22716 个.此外,ISS 公司的 X-Force[16] 安全专家小组在总结长期安全实践工作的基础上,提出十种使用最为普遍但风险最高的弱点(分别为 DoS、脆弱的帐号、数据库、电子商务 Web 应用、邮件系统、文件共享、RPC,BIND,Linux 缓冲区溢出和 IIS),并统计得出目前的攻击事件中有 80%都是出自上述这 10 种弱点.由此可见,CVNIDM 中的弱点上下文完全可以作为验证攻击有效性的有力依据.

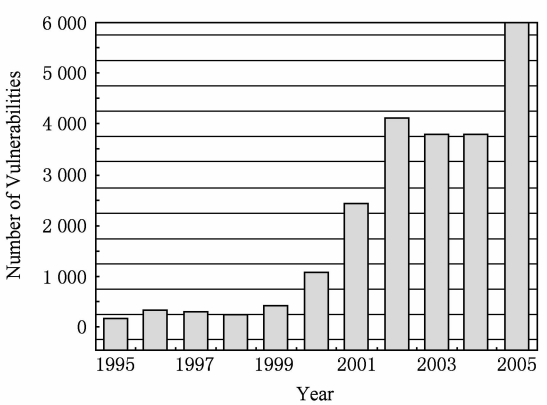

Fig. 4 Number of vulnerabilities reported by CERT/CC.

图 4 CERT/CC 接报的弱点数量

与 MMCV 上下文验证方法不同,VCCV 提供了一种动态的、无需事先收集上下文相关信息的验证方式,其基本思想是将攻击特征与被利用弱点相结合,通过实时检测恶意攻击所利用的弱点是否存在来验证该告警信息,从而断定攻击行为能否成功.表 2 列出了攻击特征和攻击所利用弱点之间的所有关联关系,其中√表示存在相应的属性,×则表示不存在相应属性.VCCV 正是利用其中的第 2 类结果,通过弱点关联的方法识别虚警,从而减少 IDS 的虚警率.

Table 2 Relations of Signature and Alert

表 2 攻击特征及所用弱点之间的对应关系

Category	Attack	Vulnerable	Alert	Result
1	√	√	√	True
2	√	×	√	False
3	√	√	×	Negative
4	×	√	√	False
5	×	×	×	Abnormal
6	√	×	×	Negative
7	×	√	×	Abnormal
8	×	×	×	Abnormal

VCCV 以检测插件的形式实现,一旦 IDS 的检测规则被触发,VCCV 的相应检测插件会被立即调用,执行适宜的 Nessus NASL[17] 脚本动态地检验目标主机是否存在相应弱点,假如存在弱点,输出相应的告警信息,表明该攻击是一次有效攻击;否则为无效攻击,不会产生虚警.

相比于静态方法,VCCV 的优点是无需事先收集上下文信息,而是采用一种实时验证目标主机弱点的方法.因此,主机弱点状态的动态变化不会影响验证结果的准确性,能够真实地反映攻击是否有效;但是,由于 VCCV 在验证弱点时,需要扫描目标主机,这就会带来额外的流量,对网络的负载造成影响,甚至会影响到目标主机的相应服务.

5.3 基于攻击反馈的被动上下文验证

VCCV 虽然能够利用目标主机的弱点上下文实时地滤除虚警,然而,仅仅通过弱点上下文并不能够完全解决告警的验证问题.考虑下面 2 种情况:1)攻击者发动缓冲区溢出攻击,但是因为设置了错误的指针偏移量,该攻击并未成功,此时,如果目标主机恰好存在相应的弱点,VCCV 将产生误报;2)针对 LiOn 蠕虫[18] 编写的 Cheese 蠕虫[19],它利用 LiOn 蠕虫留下的后门(10008 端口的 rootshell)进行传播.攻击成功后,它会自动修补系统漏洞并清除掉 LiOn 蠕虫留下的所有痕迹.在这种情况下,尽管 VCCV 能够识别 Cheese 蠕虫的攻击特征,但却无法

验证已经被打过补丁的漏洞,最终导致误报.与VCCV不同,AFCV无需向网络注入验证数据,而是被动的监听网络数据报文,将反馈上下文作为攻击是否生效的判断依据,通过攻击的反馈信息识别弱点服务的状态改变,从而避免VCCV的误报问题.

显然,AFCV的验证范围并不涵盖所有的攻击,其准确性依赖于那些具有明显反馈特征的弱点,这里称为AFCV弱点.因此,AFCV设计的一个重要环节是如何表示AFCV弱点特征.在研究了大量的CERT组织发布的弱点描述之后,我们采用了如下的特征表示方法:

定义2. 一个AFCV弱点特征S是一个三元组(*name*, *pre-action*, *post-action*),其中:1) *name*是能够对所对应弱点进行功能性描述的名称;2) *pre-action*定义了试图利用弱点服务进行攻击的恶意攻击行为特征;3) *post-action*定义了上述弱点服务对于恶意攻击所产生的预期的"输出",AFCV利用该输出进行进一步的验证.

每一个AFCV弱点特征都能够描述一类弱点服务. *name*能够对弱点信息给出清晰的概括描述; *pre-action*则对应着恶意攻击的可检测特征; *post-action*则描述了如果攻击成功后,区别于失效攻击的弱点服务的反馈信息.根据上述定义,AFCV能够精确的识别恶意事件攻击弱点服务的整个过程.图5给出了典型的AFCV弱点特征实例,该实例清晰地刻画出实际的攻击过程:攻击者向微软的IIS服务器发送了4个分别包含*pre-action*域的恶意URL串,如果服务器返回了HTTP响应码"200 OK"(*post-action*被触发),则表明该服务器上存在相应的弱点服务,攻击者发起的此次攻击就会生效.

name: WEB-IIS register.asp access
pre-action: uricontent:"/register.asp"
post-action: 200 OK

name: WEB-IIS /pcadmin/login.asp access
pre-action: uricontent:"/pcadmin/login.asp"
post-action: 200 OK

name: WEB-IIS /exchange/root.asp attempt
pre-action: uricontent:"/exchange/root.asp?acs=anon"
post-action: 200 OK

name: WEB-IIS query.asp access
pre-action: uricontent:"/issamples/query.asp"
post-action: 200 OK

Fig. 5 AFCV vulnerability signatures of WEB-IIS.
图5 WEB-IIS的AFCV弱点特征

显然,上面例子中4个*post-action*是完全相同的,而*pre-action*则完全不同.事实上,通过分析大量的攻击,我们发现许多类型的攻击,例如一些蠕虫,利用的往往是相同的弱点,显然它们具有相同的*post-action*.因此,假如一个未知攻击试图溢出一个已知弱点服务,我们仅仅需要关注该攻击的攻击特征,即*pre-action*,而无需考虑其*post-action*.可见如此设计可以使AFCV的处理过程得到简化.此外,相比于前几种方法,AFCV的优点是在上下文验证的同时不会附加额外的网络负载.

5.4 基于Chi-square检验的异常上下文验证

前面几种验证方法均需要对已知的环境上下文、弱点上下文和反馈上下文的特征模式进行形式化描述,并以特征库的形式提供给CVNIDM,其有效性依赖于上下文特征描述的准确性.这几种方法的优势在于可以有针对性的建立高效的上下文验证机制,精确度高,但是主要缺陷是必须经常动态更新相应的特征才能适应新的上下文信息,否则就无法验证未知的上下文信息.不同于上述验证方法,CACV的实现基于如下思想:"成功作用于目标主机的攻击会引发目标主机诸如流量、TCP连接数、带宽、流量分布、报文分布等属性的异常".

例如:对于常见的DoS攻击来说,如果对某台服务器的DoS攻击成功,这台服务器就不会响应任何客户的请求,于是,相比于未受攻击之前,此服务器对外发出的数据报文数量会明显减少;再比如蠕虫攻击,遭到蠕虫感染的主机和感染之前的主要区别在于感染之后主机主动向外部发出的TCP连接会明显增多.

基于上述思想,CACV首先建立被监控主机的正常活动行为模式,一旦遭到攻击,CACV会利用Chi-square检验[20]方法继续审计其后续活动情况,通过判断是否存在异常上下文来判定攻击是否生效.

以计算目标主机向外发出的TCP连接总数为例,说明CACV发现异常上下文的处理过程:在训练阶段,需要预先定义总类数,设为n,并为每类估算其具体分布值.设抽样时间间隔为0.5 h,抽样结果中最大的连接数量为10 000个.若总共分10类,即$n=10$,则平均分到每类的数量为1 000个,于是每类具体分布值分别为(1~999),(1000~1999),…,(9000~∞).

接下来的检测阶段中,利用如下的频度分布向量FDV来存放每次抽样得到的结果数据:

$$f=(f_1,f_2,\cdots,f_{10}), \tag{1}$$

设接下来 5 个抽样周期分别得到 1 433,659,1 233,3 780 和 10 043 个连接数,则更新式(1)中的相应类的数值,即得到新的 FDV 为

$$f=(1,2,0,1,0,0,0,0,0,1), \tag{2}$$

扩展到对多个属性的检测,设为 m,则得到下面的频度分布矩阵 $\boldsymbol{F}$:

$$\boldsymbol{F}=\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}\begin{pmatrix} f_{11} & & f_{1n} \\ & \ddots & \\ f_{m1} & & f_{mm} \end{pmatrix}. \tag{3}$$

考虑到用户行为和网络环境的动态变化,应该使用某一段历史时期的数据,而不宜使用全部历史数据来描述网络行为. 因此,每隔一段时间对 $\boldsymbol{F}$ 进行一次老化操作,即:

$$\boldsymbol{F}_{\text{new}}=\gamma\boldsymbol{F}_{\text{old}}, \tag{4}$$

其中,γ 为衰减因子,取值范围在(0,1)之间,它的大小决定了频度分布矩阵的老化速度,取值越小,CACV 对抽样值的遗忘速度就越快.

定义长期简档 LP 来描述与异常情况相区别的历史行为,同时定义短期简档 SP 描述每次 Chi-square 检测的观测值. 显然 LP 的更新周期 γ_l 应该远大于 SP 的更新周期 γ_s,CACV 设定 LP 的更新周期为抽样频率的 100 倍,也就是说每执行 100 次抽样,LP 更新 1 次;而 SP 的更新周期为抽样频率的 4 倍. 此外,每次更新操作的同时还需要加上本次的抽样值,于是 LP 的更新操作为

$$LP_{i+1}=\gamma_l LP_i+M_i, \tag{5}$$

SP 的更新操作为

$$SP_{k+1}=\gamma_s SP_k+M'_k. \tag{6}$$

最后,以 LP 表征统计量的数学期望,SP 表征观测值,则可根据 Chi-square 检测来评判 LP 与 SP 之间的偏差:

$$T_i=\sum_{j=1}^{n}\frac{\left(s_{ij}-\sum_{k=1}^{n}s_{ik}\dfrac{l_{ij}}{\sum_{h=1}^{n}l_{ih}}\right)^2}{\sum_{k=1}^{n}s_{ik}\dfrac{l_{ij}}{\sum_{h=1}^{n}l_{ih}}}, \tag{7}$$

式(7)在 SP 每次老化操作时候均会执行一次. 其中,s_{ij} 和 l_{ij} 分别表示 SP 和 LP 矩阵中的元素,T_i 表示第 i 个属性的 Chi-square 检测值,如果 CACV 检测的属性有 m 个,则总的检测结果为 $Score=\sum_{k=1}^{m}T_k$. $Score$ 越大,表明 SP 和 LP 的差异越大,目标主机出现异常的可能也就越大,可以通过预先设定的阈值 T,来衡量 $Score$.

不难看出,与上面提到的其他 3 种验证方法不同,利用异常上下文进行告警上下文验证的 CACV 无需频繁更新上下文特征,具有迅速适应未知攻击上下文信息的优势. 然而由于 CACV 无法预知目标主机正常的突发事件,因此存在误报的可能性. 但对于 CVNIDM 来说,同时采用上述 4 种验证方法可以起到互相弥补、相互协作的作用,真正地构建了一个以上下文验证为中心,多种安全技术相结合的高效、稳定、完整、易管理、可扩充的虚警处理平台,实现了告警的自动验证以及攻击行为能否成功的自动判定.

6 融合决策分析

根据 4 种上下文验证方法的特点,CVNIDM 还设计了融合决策分析模块. 较直观的融合方法是采用(K/N)投票表决算法,当 N 种检测方法的检测结果中 K 个判定为入侵行为,就认为攻击有效. 但这种决策方法的最大的检测率依赖于 K 值的正确选择. 为获得更好的判别结果,我们采用朴素贝叶斯分类器完成验证结果融合.

设 X 是样例集,V 是样例的类别集合. 对于样例 $\forall x\in X$,由属性值 $a_1,a_2,\cdots,a_n$ 组成,则样例 x 属于类别 $v\in V$ 的概率为:$P(v_j\mid a_1,a_2,\cdots,a_n)$. 应用贝叶斯公式可得:

$$V_{\text{map}}=\arg\max_{v_j\in V}P(a_1,a_2,\cdots,a_n\mid v_j)P(v_j), \tag{8}$$

V_{map}为目标值. 为简化计算,假设样例的属性相互独立,则有:

$$P(a_1,a_2,\cdots,a_n\mid v_j)=\prod_{i=1}^{n}P(a_i\mid v_i).$$

由此可得朴素贝叶斯分类模型:

$$V_{\text{NB}}=\arg\max_{v_j\in V}P(v_j)\prod_{i=1}^{n}P(a_i\mid v_j), \tag{9}$$

当对 4 个上下文验证函数生成的检测结果进行基于朴素贝叶斯分类器的决策融合时,将检测到攻击标记为 1,反之标记为 0. 则当 4 个验证函数同时工作时的检测结果组成 4 维向量. 在利用上述分类器对检测结果向量进行融合决策分析时,只需通过比较 4 维向量属于正常的后验概率和属于入侵的后验概率的大小,即可判断系统是否真的受到入侵攻击.

7 实验数据与结果分析

为了进一步检验 CVNIDM 的有效性,我们在 Linux RedHat 7.3 上以 Snort 为基础,实现了 CVNIDM 的原型系统,并进行了相关测试.实验床共有 4 台机器,其中 2 台模拟攻击者,1 台作为被监控主机,另外 1 台安装了 CVNIDM,测试环境的主机为均采用曙光服务器(CPU 是 PⅣ 2.4 GHz×2,内存是 4 GB).

此外,Snort 和 VCCV 采用的 Nessus 的版本分别是 2.1.0 和 2.0.10,所支持的检测规则数量为 2 164 个,其中可以通过 CVE-ID 关联 NASL 脚本的规则数量为 514 个,通过 Bugtraq-id 关联的规则数为 315 个,通过 nessus-id 关联的规则数为 315 个.

在该实验环境中,我们利用一些攻击工具产生了 2 组攻击数据,一组具有明确的 CVE 编号,而另一组虽然没有对应的 CVE 编号,但有明显的反馈上下文.所有攻击在表 2 和表 3 中列出.大部分反馈上下文集中于利用 HTTP-IIS-UNICODE 解码漏洞和 ASP-BackDoor 漏洞的攻击,实际上二者都是利用 IIS 服务器的相应弱点,它们的总数量共有 110 个,对应于 Snort 的 web-iis.rules 规则文件.它们的反馈上下文是相同的,均为 HTTP 的状态响应码,但由于数量众多,我们在实验中各选一种进行测试.

Table 2 Attack Data 1

表 2 第 1 组攻击数据

Attacks	CVE-ID
WEB-IIS Unicode directory traversal attempt	CVE-2000-0884
SMTP sendmail 8.6.9 exploit-	CVE-1999-0204
EXPLOIT x86 Linux samba overflow-	CVE-1999-0811
EXPLOIT x86 Linux mountd overflow-	CVE-1999-0002
WEB-IIS ISAPI .ida access-	CVE -2000-0071

Table 3 Attack Data 2

表 3 第 2 组攻击数据

Attacks	Feedback context
HTTP-IIS-UNICODE bug for Http Malicious request	200OK
ASP_BackDoor	200OK
HTTP_SqlCmdShell	200OK
Wollf backdoor	Wollf Remote Manager
fluxay	Fluxay Core Encyption Version

此外,为方便起见,我们为被监控主机打上了所有上述攻击的补丁.也就是说对于被监控主机来说,上述 8 种攻击都将是无效攻击,产生的告警均应视为虚警.显然这样的测试结果并不影响实验的有效性.

首先我们分别采用 Snort 和 CVNIDM 对第 1 组攻击进行检测,测试结果显示 Snort 对上述攻击全都产生告警.由于被监控主机并不具备上述攻击所要利用的弱点,CVNIDM 并未给出任何告警信息.

对于第 2 组攻击,Snort 同样生成了全部攻击的告警信息,对 CVNIDM 的测试过程如下,对于前 3 种对应 HTTP 服务的攻击,我们首先在目标主机上开启 httpd 服务进程,开启 HTTP 服务,这种情况下,攻击会成功实施,而且由于目标服务器会陆续返回 HTTP 状态码 200OK,因此 CVNIDM 会产生告警.接下来,我们关闭 httpd 服务进程,这时,由于 CVNIDM 不会检测到相应的反馈上下文信息,因此得出该告警为虚警的正确结论.后 3 种攻击均属于后门程序,首先关闭服务器端的 Wollf 进程,流光扫描代理进程以及 WINSHELL 后门进程,CVNIDM 由于没有发现相应的反馈上下文信息,从而得出告警为非相关告警的正确结论.随后将上述服务器端进程开启,CVNIDM 就会立即产生针对上述 3 种攻击的告警.

接下来,我们在攻击主机上模拟了著名的 IRC 后门病毒,对被监控主机进行攻击.IRC 后门病毒在成功感染主机之后自动尝试连接指定的 IRC 服务器,接受攻击者的文件拷贝指令,后门病毒则在本地执行操作,向攻击者所在主机传送被监控主机上的重要文件.

检测到 IRC 后门病毒的攻击特征后,CVNIDM 继续审计被监控主机的输出数据,并得到了如图 6 所示的 Chi-square 检测值分布情况.

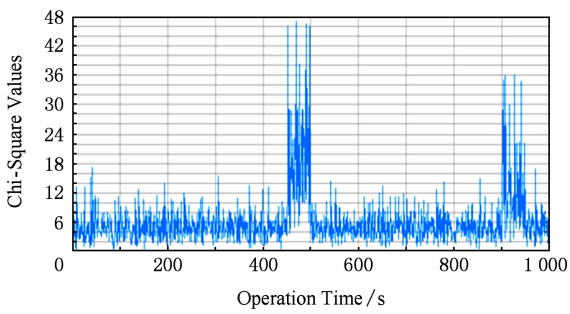

Fig. 6 Chi-square value of IRC backdoor virus.

图 6 IRC 后门病毒的 Chi-square 检测值分布

从图 6 可以清晰地看出,在 450 s 到 500 s 以及

900 s 和 1 000 s 的两个时间段各出现了一次 Chi-square 检测值的突变,而该时间恰好与 IRC 后门病毒向攻击者传送目标主机文件的时间相互吻合,CVNIDM 依此可以判定检测到的 IRC 后门病毒为有效攻击.

接下来,我们对 CVNIDM 和 Snort 在报文处理的平均运行时间和 CPU 占用率方面的性能进行了比较,背景流量选用教育网出口录制的 80 端口的 HTTP 协议数据,总计大小为 20 GB,相应的测试结果分别由图 7 和图 8 给出.

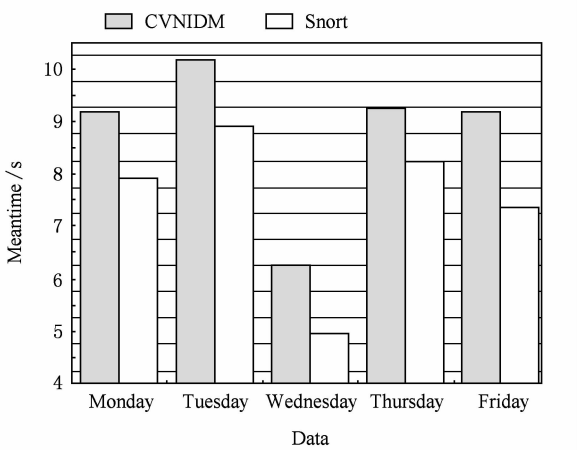

Fig. 7 Comparison of run time for packets processing.

图 7 报文处理运行时间对比

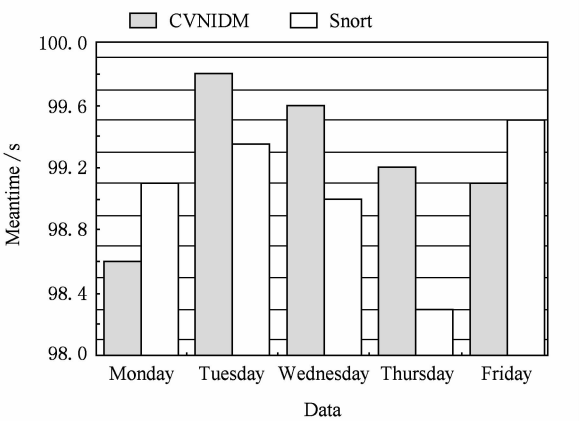

Fig. 8 CPU usage comparison.

图 8 CPU 利用率的对比结果

在报文处理的平均运行时间上二者的主要差异是由于 CVNIDM 在检测到攻击特征之后,还需要进一步追踪上下文信息.显然,这个延迟是不可避免的,但是此开销是可以通过优化措施尽量降低,可以看出 CVNIDM 和 Snort 在平均运行时间上的差距是相对较小的,只有周五的时间差距相对较大,为 1.3 s.另一方面可以看出,由于输入的数据报文数量巨大,两个系统的 CPU 的利用率均达到 90% 以上,基本处于过载状态.

最后,为验证 CVNIDM 的融合决策分析效果,我们在实验室出口网关处部署了原型系统,具体网络环境包括 PC 机 65 台,曙光服务器 13 台(分别包括邮件服务器、网页服务器、ftp 服务器).为了突出实验效果,在网关处采用镜像的方式将流量对称地分配到两台 CVNIDM 上,一台部署了融合决策分析模块,另一台采用简单的(K/N)投票表决算法.从稳定运行 45 d 的测试结果统计分析数据来看,采用融合决策分析的 CVNIDM 比另一台检测准确率平均提高 25%~30%左右.

8 结 论

本文提出了一种基于上下文验证的网络入侵检测模型——CVNIDM,结合环境上下文、弱点上下文、反馈上下文和异常上下文等多种上下文信息,CVNIDM 构建了一个以上下文为中心,多种验证技术相结合的高效、稳定、完整、易管理、可扩充的告警处理平台,实现了告警的自动验证以及攻击行为能否成功的自动判定,从而达到滤除虚警的目的,使告警起到真正的预警作用.为提高 IDS 告警信息的确定性和准确性提供了可靠保障.

参 考 文 献

[1] Lippmann R, Webster S, Stetson D. The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection [C] //Proc of the 15th Int Symp on Recent Advances in Intrusion Detection. Berlin: Springer, 2002: 307-326

[2] Sommer R, Paxson V. Enhancing byte-level network intrusion detection signatures with context [C] //Proc of the 10th ACM Conf on Computer and Communications Security. New York: ACM, 2003: 262-271

[3] Kruegel C, Robertson W. Alert verification: Determining the success of intrusion attempts [C] //Proc of the 1st Workshop on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA). Berlin: Springer, 2004: 2622-2628

[4] Gula R. Correlating IDS Alerts with Vulnerability Information [M]. Englewoocl Cliffs, NJ: Prentice Hall, 2003

[5] Desai N. IDS Correlation of VA Data and IDS Alerts [M]. Englewoocl Cliffs, NJ: Prentice Hall, 2005

[6] Nessus Vulnerability Scanner [EB/OL]. 2001. [2011-01-08]. http://www.ness-us.org

[7] Common Vulnerabilities and Exposures [EB/OL]. 2003. [2011-01-08]. http://www.cve.mitre.org

[8] Almgren M, Debar H, Dacier M. A lightweight tool for detecting Web server attacks [C] //Proc of Network and Distributed Systems Security (NDSS 2000) Symp. San Francisco: Morgan Kawfmann, 2000: 157-170

[9] Zhou J, Carlson A, Bishop N. Verify results of network intrusion alerts using lightweight protocol analysis [C] //Proc of the 21st Annual Computer Security Applications Conf (ACSAC). Los Alamitos, CA: IEEE Computer Society. 2005: 117-126

[10] Vigna G, Robertson V, Kemmerer R. A stateful intrusion detection system for world-wide Web servers [C] //Proc of the 19th Annual Computer Security Applications Conf (ACSAC). Los Alamitos, CA: IEEE Computer Society, 2003: 82-96

[11] Jiang Jianchun, Qing Sihan. Network intrusion detection based on attack context [C] //Proc of NetSec 2005. Beijing: China Institute of Communications. 2005: 28 - 34 (in Chinese)
(蒋建春, 卿斯汉. 基于攻击上下文的网络入侵检测[C] 全国网络与信息安全技术研讨会. 北京: 中国通信学会, 2005: 28-34)

[12] Tian Zhihong, Fang Binxing, Zhang Hongli. Design and implementation of network intrusion detection unit based on semi-polling driven [J]. Journal of Communications. 2004, 25(7): 146-152 (in Chinese)
(田志宏, 方滨兴, 张宏莉. 基于半轮询驱动的网络入侵检测单元的设计与实现[J]. 通信学报, 2004, 25(7): 146-152)

[13] Tian Zhihong, Zhang Weizhe, Zhang Yongzheng. Attack scenarios reasoning, hypothesizing and predicting based on capability transition model [J]. Journal of China Institute of Communications, 2007, 28(12): 78-84 (in Chinese)
(田志宏, 张伟哲, 张永铮. 基于权能转换模型的攻击场景推理、假设与预测[J]. 通信学报, 2007, 28(12): 78-84)

[14] Tian Zhihong, Zhang Yongzheng, Zhang Weizhe, et al. An adaptive alert correlation method based on pattern mining and clustering analyzing [J]. Journal of Computer Research and Development, 2009, 46(8): 1304-1315 (in Chinese)
(田志宏, 张永铮, 张伟哲, 等. 基于模式挖掘和聚类分析的自适应告警关联[J]. 计算机研究与发展, 2009, 46(8): 1304-1315)

[15] Bolzoni D. ATLANTIDES: An architecture for alert verification in network intrusion detection systems [C] //Proc of the 21st Large Installation System Administration Conf (LISA'07). Berkeley: USENIX. 2007: 141-152

[16] Internet Security System [EB/OL]. 2006. [2011-01-08]. http://xforce.iss.net

[17] The Nessus Attack Scripting Language Reference Guide [EB/OL]. 2004. [2011-01-08]. http://www.virtualblueness.net/nasl.html

[18] Lion Worm [EB/OL]. 2005. [2011-01-08]. http://www.sans.org/y2k/lion.htm

[19] CERT® Incident Note IN-2001-05 [EB/OL]. 2001. [2011-01-08]. http://www.cert.org/incident_notes/IN-2001-05.html

[20] Iguchi M, Goto S. Network surveillance for detecting intrusions [C] //Proc of Internet Workshop. Los Alamitos: IEEE Computer Society, 1999: 99-106

Tian Zhihong, born in 1978. Associate professor. Member of China computer federation. His current research interest is computer network and network security, intrusion detection and forensic.

Wang Bailing, born in 1978. Associate professor. Member of China computer federation. His current research interest is computer network and network security.

Zhang Weizhe, born in 1975. Associate professor. Member of China computer federation. His current research interest is computer network and network security.

Ye Jianwei, born in 1978. Lecturer. His current research interest is computer network and network security.

Zhang Hongli, born in 1973. Professor. Member of China computer federation. His current research interest is computer network and network security.