基于延缓蠕虫传播的集中控制反馈模型

殷安生
(南京邮电大学 信息网络技术研究所,江苏 南京 210003)

摘 要:蠕虫可以传播自身的副本,并且能够在远端机器上执行代码,是一种智能化、自动化的攻击载体。它会扫描和探测网络上存在服务漏洞的节点主机,一旦渗透成功会自我复制许多副本,通过网络传播从一个节点到另外一个节点。提出一种集中控制网络安全组件的反馈模型,能够自我学习,抵御不断产生的病毒。同时采用延迟机制阻碍蠕虫的传播扩散,保证网络不因蠕虫的爆发而陷入瘫痪。与以往的防病毒软件相比,该模型无需对蠕虫病毒有任何了解,它基于网络的状态而非内容。仿真结果表明,应用了该模型的网络其稳定状态得到保证。

关键词:蠕虫;反馈;控制器;延迟;排队算法
中图分类号:TP309.5 **文献标识码**:A **文章编号**:1673-629X(2010)08-0146-04

Integrate Control Feedback Model Based on Delay Worm Propagate

YIN An-sheng
(Institute of Information Network, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: A copy of the worm can spread itself, and can execute code on the remote machine is a kind of intelligent, automated attack vectors. It will scan and detect the network of service vulnerability exists on the node host, once the penetration will be self-replicating the success of a number of copies transmitted through the Internet from one node to another node. This paper presented a feedback model based on integrate control kinds of security module deployed anywhere in the network. It can be grown up through continuous learning by oneself to resist the attack of new worms. In the meanwhile it delay worm propagate with adopting delay mechanism to protect network avoid collapse because of bursting with worms. Contrast with ordinary anti-virus software, model mustn't be knowlegable to worms any more, it based on state of network not the content. It is indicated that the model maintain steady network state by emulator.

Key words: worm; feedback; controller; delay; queue algorithm

0 引 言

从1988年莫里斯从实验室放出第一个蠕虫病毒以来,计算机蠕虫病毒以其快速、多样化的传播方式不断给网络世界带来灾害[1]。特别是1999年以来,高危蠕虫病毒的不断出现,使世界经济蒙受了轻则几十亿,重则几百亿美元的巨大损失。现有的防治蠕虫措施还只能在蠕虫发作后对其进行有效的清除,但此时网络已受到了破坏。有时即使针对某蠕虫的杀毒工具产生了,但是蠕虫的变种也出现了[2]。并且伴随蠕虫病毒的传播往往导致网络的拥塞。

对此,文中提出了一种集中控制反馈模型(ICFM, integrate control feedback model)。通过网络中的安全组件(SM, security module)收集和整理网络的信息,再通过这些信息产生控制指令控制各SM的行为,更新SM的状态策略,生成一个自我学习的网络。同时相对于目前对于蠕虫突然爆发的束手无策,ICFM采用一个延迟机制,建立延迟队列(DQ, delay queue),通过算法控制延迟的时机和状态,缓解蠕虫快速传播对网络的影响。

1 ICMF的结构和原理

1.1 安全组件

一个网络为了保障自身的安全,一定包含各种工具来抵御外来的侵害。这些工具分布在网络的各处,独立的发挥着各自的功能[3]。

收稿日期:2009-12-08;**修回日期**:2010-03-10
基金项目:国家"十一五"高技术研究发展计划(863)项目(2006AA01Z232)
作者简介:殷安生(1982-),男,博士研究生,研究方向为网络安全。

防火墙 FW(Firewalls):是指一种将内部网和公众网络(如 Internet)分开的方法,它实际上是一种隔离技术,在两个网络通信时执行访问控制,它能允许用户"同意"的人和数据进入网络,同时将用户"不同意"的人和数据拒之门外,最大限度地阻止网络中的恶意访问行为。

入侵检测系统(IDS, Intrusion Detection Systems):对(网络)系统的运行状态进行监视,发现各种攻击企图、攻击行为或者攻击结果,以保证系统资源的机密性、完整性和可用性。蜜罐(honeypot):是一种安全资源,其价值体现在被探测、攻击或者摧毁的时候,可以捕获和分析自动化的攻击。

除了这些还包括路由器防火墙、传感器、分析器、扫描器、探针,有时一个组件能集成多种功能[4]。

1.2 系统原理

ICFM 通过不断的自我学习(如图 1 所示),不断完善自身的安全策略,使系统更安全,流经 ICFM 后的信息和网络中新进入的数据信息一起通过模型,各种安全行为和网络中的实时状况相互比较,再产生新的策略以取代或改进旧的机制[5]。这种闭环模型通过网络事件的不同过程来实现,此时系统还在同时接受新的数据流入,并且和新的数据进行比较,产生一个错误或者告警,系统将根据这个错误或告警决定一个新的措施,例如增加一个蜜罐[6]等等。

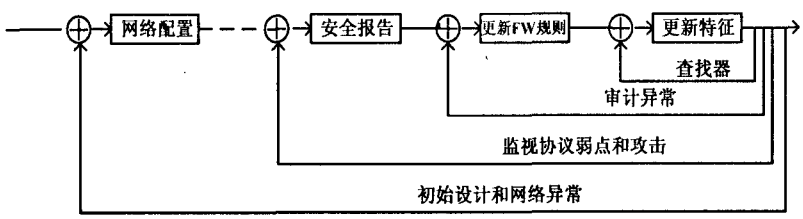

图 1　ICFM 工作原理

首先 ICFM 会根据自己获得的信息对网络进行安全配置,然后通过蜜罐监视网络中的行为,定期产生安全报告,客户机的防火墙审计异常更新防火墙规则,过滤掉不正常或者有危害的数据信息,入侵检测系统查找网络中的入侵行为,更新入侵检测的特征码。

ICFM 是一个整体,通过各种检测的综合达到高效率的查找到蠕虫的目的。例如产生的安全报告显示系统正在被入侵,那么入侵检测系统就需要提取这种入侵的特征码,或者入侵系统报告系统受到了某种病毒的入侵,那么防火墙就要相应的采取某种措施来尽早地抵御病毒的入侵,就可以在路由器的防火墙上直接过滤来自传播病毒的 IP 地址的数据,或某种特征的数据。所有的 SM 由控制器启动或释放。这些组件不断截取网络信息并反馈到系统。

一旦网络受到攻击,ICFM 在阻碍攻击的同时学习攻击的手段和方法,不断更新系统的应对措施,所以 ICFM 受到的攻击越多其系统就越完善。攻击结束后记录攻击的特征码或者特征行为存入 SM 的特征库防止攻击的再次发生。

1.3 系统结构

ICFM 就是在既有的 SM 的基础上,添加一个控制器,控制器和 SM 部署在同一个局域网中,网络中所有的数据(发往本机除外)都会流经控制器,并且控制器能直接和主机联系。和控制器相连的是一个数据分析器,分析器只和控制器连接,不和网络中的其他组件有联系,它分析控制器传递过来的信息,并反馈控制指令[7]。路由器的数据先经过控制器,然后通过交换机进入子网,子网中各个主机都配置了 FW 和 IDS,蜜罐(honeypot)则是完全处于一种最易受攻击的状态,没有任何保护措施。

该控制器了解整个网络的拓扑结构和各个 SM 的配置和策略,并和 SM 进行通信。SM 向控制器提供信息,控制器控制组件(包括主机)的行为。控制器利用自身搜集到的和主机提供的信息通过一系列的算法控制 SM 的工作状态,进而控制网络的状态。

控制器通过搜集的信息控制器更新网络的配置,设置蜜罐,更新防火墙规则,修改入侵检测使用的特征码等等。这种更新采用一种闭环模型,即策略的修改动态的更新。

与以往的系统相比,ICFM 有以下优点:

(1)易于构建,成本低。系统是在集中利用网络现有的 SM 的基础上实现的。

(2)模型更安全,处理速度更快。将核心处理分为控制器(controller)和分析器(analyzer)两个部分。controller 只做一些简单的控制工作,这样流量就不会产生太大的延迟;controller 和 analyzer 采用和节点不同的操作系统(Linux/Unix),在两者之间加锁,并将数据库独立出来。

(3)动态控制。ICFM 能够根据网络的状态采取相应的策略。

(4)智能化。ICFM 无需了解未被发现的蠕虫。被攻击的越多,ICFM 的系统就越完善。

2 延迟机制

2.1 蠕虫的传播

现在流行的蠕虫采用的传播技术一般是尽快地传

播到尽量多的电脑中，蠕虫的扫描策略是这样的：随机选取某一段 IP 地址，然后对这一地址段上的主机扫描。这样随着蠕虫的传播，新感染的主机也开始进行这种扫描，这些扫描程序不知道哪些地址已经被扫描过，它只是简单地随机扫描互联网。于是蠕虫传播的越广，网络上的扫描包就越多。即使扫描程序发出的探测包很小，积少成多，大量蠕虫程序的扫描引起的网络拥塞就非常严重了。

由于网络和宽带的发展以及病毒“免疫”能力的加强。蠕虫传播的速度越来越快，人们已经很难在蠕虫发作时再及时有效地做出反应。从 Red Code 到 Slammer，蠕虫攻陷全球从 12 个小时下降到 10 分钟。

2.2 延迟原理

将 ICFM 中的连接分为三种：接受的连接，延迟的连接，丢弃的连接。应用一个 DQ 来延缓蠕虫病毒的传播，就是限制其连接请求，如图 2 所示。

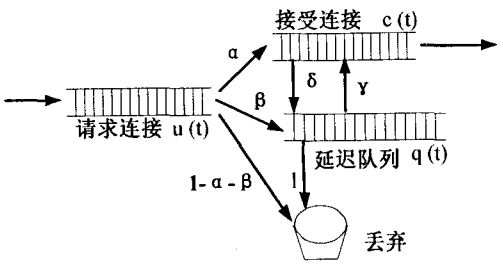

图 2 延迟机制

延迟模型的应用过程：

(1) 网络状态正常，此时参数 $\alpha = 1$，即表示接受全部请求的连接，并一直保持直到网络产生异常。连接请求数为 $u(t)$，接受的连接为 $c(t)$。

(2) 检测到网络上的某个 IP 地址可能发生了蠕虫传播，就为此主机建立一个 DQ，请求连接以参数 β 进入 DQ，DQ 以参数 $\gamma(0 < \gamma < \beta)$ 补偿发送速率，提供尽力而为的服务。$q(t)$ 为队列中的连接数[8]。

(3) 如果接受的连接数量突然急剧增加，说明在短时间内蠕虫爆发了，此时立即以参数 δ 将接受的连接放入 DQ，保证网络的畅通。

(4) 由于存储空间的限制，DQ 的长度是预先计算好的，一旦网络状况继续恶化，DQ 无法装载更多的请求发送的数据，此时就需要调整 α 和 β 的值，让一部分数据不经过 DQ，直接丢弃请求的连接数据。

(5) 为了确保网络的安全，设置了一个丢失率 l。即使连接请求进入了 DQ，也仍然会被直接丢弃。

丢弃时采用一定的策略[9]，例如对于发送大量连接请求的 IP 地址，或者某台主机的 CPU 利用率长时间为 100%；POP3 端口总是被打开，不断发送邮件；传播的数据或者方式非常相似；

总是返回大量的无效服务请求和无效的 IP 地址。那么这样的主机数据将会被大量地丢弃[10]。

3 仿真结果与分析

3.1 延迟仿真

在一定的时间内，数据的传播速度和传播数量是成比例关系的，所以可用某时刻系统接受的连接数来表示同时刻系统的连接速度。这样在 ICFM 实际应用时处理数据能够非常方便。由图 2 得到网络接受的数据流量为 $\alpha u(t) - \delta c(t) + \gamma q(t)$，所以以此为传播速度 V 进行 matlab 仿真。

根据反馈的信息，当速度达到某一个阈值 $v1$ 时，启动延迟器。保持传播速度最多只能以线性增长，当速度突破一个更高的阈值 $v2$ 时，就表示可能会产生网络拥塞，需要丢弃请求的连接和发送的数据，此时系统只信任已建立连接的 IP 地址，对新建立的链接一概丢弃。而且检测到的时间越晚，蠕虫传播的速度越快，那么经过控制器后，速度下降的越厉害。

网络传输过程中，蠕虫的传播是基于病毒源的，通过不断的传播，病毒源越来越多，传播的数量也就越来越多。在一个定时间内，传播的数量和一定时间以前是没有关系的，可以看出传播的数量具有马尔可夫性，同理，对于系统接受的连接也就是允许向外发送的数据也具有无后效性，服从指数分布。

由指数分布概率密度函数 $f(x) = \lambda\exp(-\lambda x)$ 可以看出 λ 值越大，曲线下降的越厉害。在本系统中可以用指数分布来确定参数 α 的值，在网络发生大流量数据传输时，应用 α 值来降低数据发送速度。

总的来说，ICFM 是以牺牲用户的连接请求和发送数据来换得网络的畅通，只能对用户（主机）提供尽力而为的服务，不能保证服务质量（QoS）。

3.2 阈值的计算

由上面的介绍可知，数据包的到达和接受都服从指数分布，对于具有多个处理器的系统来说，可近似为一个 M/M/N 排队模型，所有的处理器共享一个公用的队列。该队列是一个生灭过程模型，其生灭速率为 λ 和 μ。

系统的稳定状态概率如下：

$$\eta_k = \eta_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \eta_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}, k < N \quad (1)$$

$$\eta_k = \eta_0 \prod_{i=0}^{N-1} \frac{\lambda}{(i+1)\mu} \prod_{j=N}^{k-1} \frac{\lambda}{N\mu} = \eta_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{N!N^{k-N}}, k \geqslant N \quad (2)$$

k 为泊松到达流的数目,也就是系统请求的连接数。

λ 为出生率,也就是到达速率。

μ 为死亡率,也就是服务速率。

定义 $\rho = \lambda/(N\mu)$,系统稳定条件为 $\rho < 1$。

由 $\sum_{k=0}^{\infty} \eta_k = 1$ 和式(2),可以得到 η_0(生灭过程在 0 时刻的稳定状态概率) 的表达式:

$$\eta_0 = [\sum_{k=0}^{N-1} \frac{(N\rho)^k}{k!} + \frac{(N\rho)^N}{N!} \frac{1}{1-\rho}]^{-1} \quad (3)$$

系统中的平均请求数量:

$$q = E[Q] = \sum_{k\geqslant 0} k\eta_k = N\rho + \rho \frac{(N\rho)^N}{N!} \frac{\eta_0}{(1-\rho)^2} \quad (4)$$

ρ 为 ICFM 的利用率,表示系统忙的时间比例。称为系统的通信量强度(traffic intensity)。为了确定阈值,需要求得 $E[Q]$,此时需要确知通信量强度 ρ 的值,在南京邮电大学校园网中心通过测试,得知在大流量通过时,网络保持畅通的 ρ 值最大可以达到 99.75%。而网络发生明显延迟时 $\rho = 89.5\%$。

在数据仿真时,可以简化 $M/M/N$ 模型,对 N 的取值可以小一点,再应用以上两个 ρ 值求得 $E[Q]$,再根据网络的结果和状态就可以确定阈值。

以 slammer 为例,在一个小型网络环境下,仿真结果(见图 3)表明 ICFM 基本达到了预期的要求。在同样的网络初始状态下,应用了 ICFM 的网络其状态得到保证。由于采用了延迟机制,可以发现在网络状况恶化时,ICFM 发现的越晚,状态恢复的速度越快。在网络状态到达 201235($v1$) 时,延迟序列启动,使网络在最恶劣的情况下保持线形增长;当网络状态到达 255546($v2$) 时,表明网络已经接近瘫痪的边缘,此时采取直接丢弃策略(连接数量越多,丢弃越多越快),使网络尽快恢复正常。

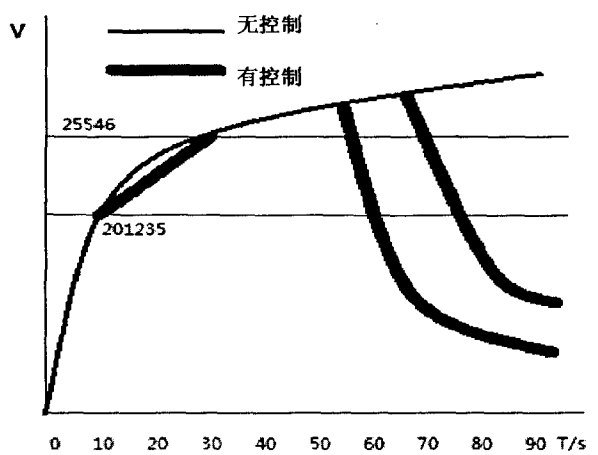

图 3　仿真结果

4　结束语

ICFM 无需对蠕虫有任何了解,且能提供事件日志和 SNMP 陷阱警告,保护网络基础设施架构,基于网络中已有的设备,既不浪费资源,又能持续扩展。不但能应对蠕虫的传播,还能阻止例如 DDoS 这样通过大量数据阻塞网络的攻击。

通过仿真结果可以看出,模型对网络流量的控制和预期的设想是一致的。但是在采用丢弃策略时仍然不能保证用户有效数据的传送[11],这对管理员控制网络很不利。在后续的工作中将重点逐步完善丢弃策略并研究模型如何具体实现[12],使其适应各种网络环境。

参考文献:

[1] Chen S, Tang Y . Slowing down Internet worms[C]//Distributed Computing Systems, 2004. Proceedings. 24th International Conference on, Phoenix, Arizona, USA: [s. n.], 2004:312 - 319.

[2] Kreidl O P, Frazier T M. Feedback control applied to survivability: a host - based autonomic defense system[J]. Reliability, IEEE Transactions on, 2004, 53(1): 148 - 166.

[3] 张仕斌,谭　三,易　勇,等. 网络安全技术[M]. 北京:清华大学出版社,2004.

[4] 林　闯. 计算机网络和计算机系统的性能评估[M]. 北京:清华大学出版社,2001.

[5] 郭　晔. 面向 agent 的蠕虫防御系统研究[D]. 杭州:浙江大学,2007.

[6] 汪　伟. 网络蠕虫检测技术研究与实现[D]. 杭州:浙江大学计算机科学与技术学院,2006.

[7] ZOU C C, GONG Wei2bo, TOWSLEY D, et al. The monitoring andearly detection of Internet worms[J]. IEEE /ACM Trans on Net working, 2005, 13(5): 961 - 974.

[8] 魏宗舒. 概率论与数理统计[M]. 北京:高等教育出版社, 1983:394 - 399.

[9] 张　嵩. AR 模型在预测中的分析[J]. 襄樊学院学报, 2008, 29(5): 13 - 14.

[10] 汪　伟,鲁东明,董亚波,等. 面向内网的网络蠕虫检测系统设计与实现[J]. 计算机工程,2006,32(17):205 - 206.

[11] ZOU C C, GONGWei2bo, TOWSLEY D. Codered worm propagation modeling and analysis[C]/ /Proc of ACM Conference on Computerand Communications Security. Hitton Alexandria Mark Center, Alexandria, VA, U.S.A:[s.n.], 2002: 138 - 147.

[12] 亓俊红,苏　波. 基于蠕虫传播机理的主动防御策略[J]. 山东理工大学学报:自然科学版,2006(6):62 - 65.