基于主动防御模式下蠕虫病毒特征码的提取模型及算法设计

刘国柱 尚衍筠
(青岛科技大学信息科学技术学院 山东 青岛 266061)

摘 要 从一般的网络攻击技术出发,分析现有网络安全技术的弱点,采用主动防御的新思想来应对当前反病毒技术远远落后病毒技术发展的局面。重点对基于动态陷阱生成技术的蠕虫病毒特征码的自动提取进行研究,对收集的新型蠕虫病毒特征扩充 IDS 的特征库,弥补入侵检测系统对未知病毒攻击无法识别问题,完善防御系统。

关键词 主动防御 动态陷阱 蠕虫 特征码

ON ACTIVE DEFENSE MODE BASED EXTRACTION MODEL OF WORM SIGNATURE CODE AND ITS ALGORITHM DESIGNING

Liu Guozhu Shang Yanjun
(*College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, Shandong, China*)

Abstract Proceeding from general network attack technologies, in this article we analyze current weakness of network security technologies, and propose to adopt a new idea of active defense to deal with the situation that the anti-virus technology far lags behind the development of the virus technologies at present. This paper focuses on the automatic extraction of the worm signature codes of which it is based on dynamic trap generation technology. The research aims at making up the deficiency of the intrusion detection system in its not able to recognize unknown viruses' attack as well as improving the defense system by collecting new virus signatures of worm to expand the IDS characteristic library.

Keywords Active defense Dynamic trap Worm Signature codes

0 引 言

Internet 的快速发展把人类社会带入 21 世纪的信息时代。网络信息系统在政治、军事、金融、商业、交通、电信、文教等方面的作用日益扩大,社会对网络信息系统的依赖也日益增强。随之而来的网络入侵事件也日益增多,社会危害性也越来越大,现有保护网络和信息的安全技术如防火墙、入侵检测系统等,一般是基于规则和特征匹配的方式,只能防范已知攻击。伴随网络技术的进步,黑客攻击手段也不断出新,现有防护技术普遍对新型攻击识别不足,使新型攻击在一定时间内给国家和个人造成重大损失。

自从 1988 年第一例蠕虫病毒出现以来,网络蠕虫病毒在互联网上的危害越来越大。2001 年 Nimda (尼姆达) 蠕虫被发现,对 Nimda 造成的损失评估数据从 5 亿美元攀升到 26 亿美元后继续攀升,到现在已无法估计。目前蠕虫病毒爆发的频率越来越快,尤其是近两年来,出现越来越多的蠕虫病毒如"冲击波"(Worm. Blaster)、"振荡波"(Worm. sasser)、SQL 蠕虫王、网络天空(Worm. W32/ Netsky2P)、"W32/ Zafi2B"等。据估计,2004 年(小规模和大规模的)蠕虫袭击使北美服务供应商为防治蠕虫病毒耗费高达 2145 亿美元。

目前蠕虫病毒的研究工作还主要集中在对已有蠕虫的检测与防范上,蠕虫的防治措施处于一种非常被动的状态。所以研发能早期检测与预警新蠕虫的主动防御技术,提前对未知蠕虫病毒的检测和早期的预警,对减少蠕虫的大规模爆发,降低危害有重要意义。

当前国际上对主动防御模式的研究机构是"蜜网研究联盟",该联盟通过不断研究新的蜜罐(Honeypot)系统组织成虚拟蜜网,通过虚拟蜜网的强大功能实现入侵检测和病毒捕获,为主动防御提供可靠的保证。

本文通过研究和实践,整合现有的陷阱技术、防火墙技术、入侵检测等网络安全技术构建一个综合的动态安全防御体系,这对发展主动防御的网络安全技术有一定的探索实践意义。在整体防御的安全框架下重点研究蠕虫病毒的防治,实现了蠕虫的陷阱捕获和特征码的自动提取。

1 陷阱技术分析

1.1 陷阱系统

陷阱系统是一种网络资源,它对黑客来说是个有吸引力的目标,但实际上是一个可以收集信息的系统。它的价值就在于被攻击和探测,它的资料和数据可能是伪造的,使它看上去更像一台真正提供重要服务的主机,这样使它对黑客更具有诱惑力[1,2]。在被攻击的同时,检测记录整个入侵和破坏过程。由

收稿日期:2008 - 11 - 09。刘国柱,副教授,主研领域:计算机网络与安全。

于在陷阱系统上并不存放真正有价值的数据,因此,任何对它的访问都被视为可疑行为甚至是黑客攻击。陷阱系统不仅会消耗访问者的资源,还会记录访问者的活动或行为信息,安全人员可以针对这些信息进行分析,从而了解网络黑客所使用的新技术和新工具,提前采取防御措施,网络陷阱的含义也由此而来。陷阱系统不是一种独立的防御工具,部署它并不会使网络变得更加安全,它仅是对现有网络安全防御体系的补充,通过使用陷阱系统,可以改变现有防御体系的被动性,达到相对的主动防御。

1.2 蜜罐系统

蜜罐常见的定义如下:蜜罐是一个布署在网络上旨在让黑客进行无授权访问和非法使用的信息系统资源。它具有正式的IP 地址,但没有域名、没有合法的用户、没有授权的服务[3]。蜜罐其实就是个诱饵,就如同使用蜂蜜来引诱蜜蜂到来一样,是一个设计用来引诱入侵者的系统。因此可以看出蜜罐系统是一个特定的陷阱系统类。

一般有两种方法对蜜罐进行分类:一是按照部署目的分为产品型蜜罐和研究型蜜罐[4];二是按照交互程度分为低交互蜜罐和高交互蜜罐[5]。

产品型蜜罐具有事件检测和欺骗功能,可以帮助一个组织和部门减轻安全风险。它通常放置在一个部门的内部网络环境中,由于它对攻击者更具有吸引力,可以诱惑或欺骗攻击者把时间和资源都用于这个蜜罐上,使他们远离实际的工作网络,同时通知管理员系统,使其有所防备。研究型蜜罐的目的是为了获得恶意黑客的信息,它不会直接带来安全效益,但通过它可以研究系统所面临的威胁,以便更好地抵抗这些威胁。低交互蜜罐的交互能力非常有限,一般仅仅模拟了操作系统和网络服务,因此低交互蜜罐的部署简单并且危险小。高交互蜜罐提供了一个可用的真实的操作系统和服务,对黑客更有吸引力,遭受攻击的可能性就更大,我们就有可能收集更多的信息。

2 陷阱捕获蠕虫病毒技术分析

陷阱技术可以捕获和分析蠕虫病毒,从而获得最新的病毒特征,更新 IDS 的病毒特征库,使得 IDS 具有了对新型蠕虫病毒的防御能力。用陷阱捕获蠕虫的原理很简单,蠕虫要传播,首先通过扫描确认攻击对象,然后发起攻击,在确认攻击成功,获得主机控制权后,从网上下载自制自身的代码到新的目标机器。陷阱检测到蠕虫扫描后,就利用虚拟陷阱主机回应欺骗它,让其相信已经成功入侵,这样在它把自身的代码复制到陷阱机,这就捕获了蠕虫。

3 主动防御系统的关键技术与实现

主动防御系统主要是建立在开源 Honeyd[12] 项目上。Honeyd 提供了丰富的脚本语言,利用该脚本语言编写模拟种类操作系统或者不同拓朴结构的虚拟主机网。首先利用脚本语言开发出各种类型的陷阱点的配置模板文件。然后利用物理主机中的脚本语言,如 Windows 下的 VBscript、Linux 下的 Perl 语言,根据特定规则的算法启动 Honeyd 进程,加载新配置模板的 Honeyd 进程,规定时间段结束后,终止 Honeyd 进程,再启运另一种配置模板的 Honeyd 进程。就是通过编写操作系统中的脚本程序,然后在一个算法下动态加载,以此技术来实现动态陷阱的生成。其数据捕获机制层次结构如图 1 所示。

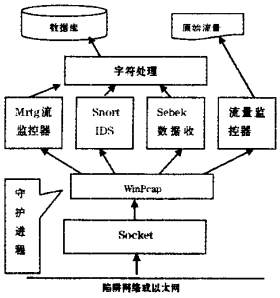

图 1 数据捕获结构层次图

Mrtg 工具提供了对网络流的监控功能,它能够提供关于网络流更具体的信息,如网络流开始和结束时间、网络流传输的字节数等。Snort 入侵检测系统对符合入侵检测特征的攻击数据包发出对应的报警信息,从而标识网络流中存在的已知的攻击事件。Sebek 则主要用于捕获攻击者在蜜罐主机上的系统行为。Sebek 能够捕获的系统行为包括攻击进程树、进程所打开或读写的文件、进程执行的命令、进程所关联的网络流以及 Shell 进程所产生的键击记录等[6]。守护进程接受这些数据源的输入,进行解析并写入数据库中,供进一步分析使用。此外,在陷阱网络上部署的 Tcpdump 流量记录器将在外出接口上监听全部流入流出陷阱网络的网络流量,并抓取到本地的 pcap 文件中,提供最具体的原始流量数据。

4 蠕虫病毒特征码自动提取模型

本模型参考开源陷阱网络系统 Honeyd 的框架结构以及开源蠕虫检测系统 EarlyBird[7] 的原理,设计了一个基于陷阱网络 Honeynet 蠕虫特征码自动提取模型。该模型对 EarlyBird 自动蠕虫特征码提取的主要算法进行改进,即采用能够进行快速字符串匹配的 Karp-Rabin 算法[9] 代替了原来效率较低的 Rabin Fingerprint[10] 算法,同时将 EarlyBird 划分字符子串区间由原来的 30-40 字节变化为 150-200 字节之间。用本模型实现了蠕虫病毒数据包的捕获,然后对蠕虫病毒特征码进行后台自动提取。由陷阱机 Honeypot 组成的陷阱网络 Honeynet 是一种安全资源,陷阱网络的设计目的是吸引黑客与蠕虫的扫描、攻击、攻陷。所有进入或发出陷阱网络的流量都可能预示着扫描、攻击、攻陷。由于蠕虫病毒只是程序,不具备智能分析能力,因此采用低交互陷阱网络就能完成蠕虫的捕获任务,系统重点分析流入陷阱网络的数据包。该原型系统是在开源陷阱网络系统 Honeyd 的基础上实现,以软件插件的形式集成到 Honeyd 中,利用钩子技术截获进入 Honeyd 系统的网络数据包,然后自动特征码生成引擎负责提取新特征码并存入数据库。图 2 是模型的结构示意图。

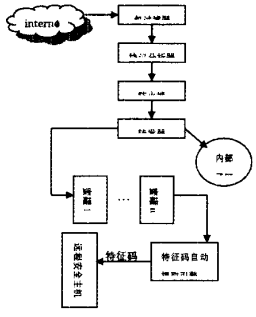

图 2 蠕虫特征码自动提取模型结构示意图

该模型通过基于 Karp-Rabin 的算法对网络数据包进行过滤,实现对网络重复数据的剔除,降低系统的误报率。其中,Karp-Rabin 算法首先需要一个 Hash 函数,此 Hash 函数需要两个特性:(1)高效的计算性;(2)对字符串有较高的识别能力。对于一个长度为 m 的字符串序列 $c_i, c_{i+1} \cdots c_{i+k-1}$,定义 hash 函数如下:

$$\text{Hash}(c_i, c_{i+1}, \cdots c_{i+k-1}) = \text{mod}(2^{K-1}\ asc[c_i] + 2^{K-2}\ asc[c_{i+1}] + \cdots + 2^0\ asc[c_{i+k-1}], q)$$

其中 $asc[c_i]$ 为字符 c_i 的编码,q 为足够大的素数。然后利用防火墙对数据进行进一步过滤,将攻击数据通过转发器发送到陷阱网络,特征码自动提取引擎作为插件的形式嵌入到 Honeyd 系统中,经过数据分析将特征码传送到远程安全主机。

Honeyd 通过虚拟拓扑组件实现多个 IP 地址的虚拟主机,系统从真实环境接收到的数据经包分发器进行处理,Honeyd 系统利用配置数据库中的模板驱动个性引擎与服务子系统来模拟不同的操作系统和不同的服务。特征码生成引擎利用 Hook 技术截获包分发器接收的网络数据包,经分析后提取蠕虫的特征码,最后输出、发布。并将特征码添加到入侵检测系统的规则库中,实现入侵检测系统的主动防御。

5　算法设计

特征码生成算法的主要思想如下:因为蠕虫感染陷阱网络后,短时间内陷阱机会流入大量携带蠕虫机器码的网络数据包,因此当接收流入数据包的缓冲区在 t 时间段内充满时就会生成一条特征码。首先将每个数据包负载划分为变长字符串因子,然后利用贪婪式字符串匹配算法 RKR-GST(the Running Karp-Rabin Greedy String-Tiling,简称 RKRGST)[11]求解相邻两个字符串的最大公共子串来计算字符串之间的相似度,其中字串 A 和 B 的相似度

$$\text{Sim}(A,B) = 2 \times \text{MatchLen} / (|A| + |B|)$$

$$\text{MatchLen} = \sum \text{match}(i,j,\text{length}) \in \text{titles length}$$

$|A|$、$|B|$分别为子串 A、B 的字符长度;$match(i,j,length)$ 为在 A 中起始位置为 i,在 B 中起始位置为 j,长度为 $length$ 的公共子串;$tiles$ 为公共子串集合[8]。相似度小于 d 的字符串因子就编为一个候选因子队列。最后对候选因子按出现频率进行排序,应用贪婪法选择高频候选因子组成特征码。具体描述如下:

P 表示数据包负载 p_i 的集合,R 表示字符串因子 r_i 的集合,S 表示候选因子 s_i 的集合,Q 表示与候选因子 s_i 相似的字符串因子队列,C 表示特征码 c_i 的集合,m,M 为字符串因子大小阈值,d 为相似度阈值,l 为特征码最大长度。伪代码如下:

```
S = ϕ
foundFlag = false
do
  {
   If( size( P ) >0 and DelayTime < t)
   /* 对字符串因子序列进行赋值 */
   {
     for( i = 1 to n)
     {
       r_i ←( p_i divide into small block)
       if( length( r_i ) [ m, M ] )
       R = R + r_i
     }
     for( i = 1 to n)
     {
       do
  }
     forEach s_i ∈ S
     {
  /* 候选因子和字符串因子比较 */
     if( sim( r_i , s_i ) < d)
     {
     Add( Q_i , r_i )
     foundFlag = true
  }
  }
} while( foundFlag = false)
if( foundFlag = false)
S = S + r_i
}
Sort( S)                                    //候选因子按出现频率排序
forEach s_i ∈ SortedS
ci = ci + s_i
  Until length( c_i ) >1
C = C + c_i
} while( size( C ) > maxsize)
```

6　实验数据和算法评价

为了对 EarlyBird 算法与改进的特征码提取算法的运行效率进行比较,本文设计了一个实验。取三种蠕虫样本对本系统进行模拟攻击,如表 1 所示。Karp-Rabin 算法中的 m(字符子串的长度)分别取 40、60、80、100、120、140、160,对每个长度、每个算法都运行多次,对运行时间取较稳定的值,评估结果如图 3 和图 4 所示。

表 1　实验中使用的三种蠕虫病毒

蠕虫名称	总长度	攻击长度	协议	目标端口
Sapphire	376 Bytes	376 Bytes	UDP	1434
CodeRedII	3,8KBytes	3,8KBytes	TCP	80
Welchia	10KBytes	1,7KBytes	TCP	135

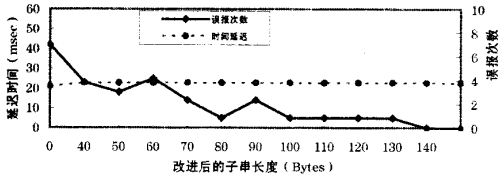

图 3　改进算法实验结果

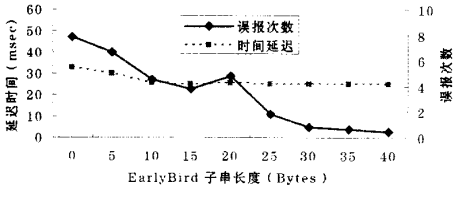

图 4　EarlyBird 系统实验结果

从图 3 中可以看出前期之所以误报率高是因为那些非蠕虫的网络数据包中所包含的微小子串所引发的误报,后期因为子串长度变大,那些非蠕虫的不相关的网络数据包包含大量的相似数据包是不可能的,因此误报率不断降低。当字符串的长度超过 140 的时候,误报率达到了 0,这表明我们的方法对蠕虫爆发没有误报。鉴于大部分的蠕虫的大小都超过 150 字节(从表 1 可以看出),因此采用本文中的方法在识别蠕虫时就不会产生误报。我们也看到了检测延迟与字符串的长度之间的依赖关系可以忽略,因为它的走势在大部分区域都是一条平行于 x 轴的直线,这是因为所有的正确报警都是由属于蠕虫病毒的特征子串产生的,对于同一特征子串的检测延迟是一定的。

通过图 3 和图 4 比较可以看出改进的算法提取的特征码使得入侵检测系统的时间延迟和误报率明显低于 EarlyBird 系统,从而达到了我们的预期。

7 总 结

本文提出的模型经原型系统的实验表明,对于未知蠕虫攻击的检测有较好的检出率。陷阱网络模块运行正常,性能稳定。特征提取引擎在较短的时间内可生成蠕虫程序副本代码特征片断并能够以正确的 Snort 规则格式表示。使整个系统可以更好地应对未知的蠕虫威胁。

参 考 文 献

[1] Mark Pickett. A Guide to the Honeypot Concept[R]. 2003 - 06 - 09. http://www.giac.org/practical/GSEC/Mark_Pickett_GSEC.pdf.

[2] 王利林,许榕生. 基于主动防御的陷阱网络系统[J]. 计算机工程与应用,2002,17:177 - 179.

[3] Lance Spitzner. Honeypots:Definitions and Value of Honeypots. 2002 - 05 - 17. http://www.enteract.com/ ~lspitz.

[4] Chtmming Rong,Geng Yang. Honeypots in Blackhat mode andits implications[C]. USA:Proceedings of the Fourth International Conference on Parallel and Distributed Computing,Applications and Technologies,2003:185 - 188.

[5] Iyad Kuwatly,Malek Sraj,Zaid Al Masri,et al. A dynamic Honeypot design for intrusion detection[C]. USA:Proceedings of the IEEE/ACS ICPS,2004:95 - 104.

[6] Maximillian Dornseif,Thorsten Holz,Christian N Klein. NoSE-BrEaK-Attacking Honeynets[C]. USA:Proceeding of IEEE Workshop on Information Assurance and Security,2004:123 - 129.

[7] Singh S,Estan C,Varghese G,et al. Automated Worm Fingerprinting[C]. Proceedings of 6th Symposium on Operating System Design and Implementation (OSDI),USENIX,2004:45 - 60.

[8] Prechelt L,Malpohl G,Philippsen M. Finding Plagiarisms Among a Set of Program s with JPlag[J]. Journal of Universal Computer Science,2000,8:1016 - 1038.

[9] Karp R M,Rabin M O. Efficient randomized pattern matching algorithms. IBM J. of Research and Development,1987,31(2):249 - 260.

[10] Rabin M. Fingerprinting by random polynomials[R]. Center for Research in Computing Technology-Harvard University,Tech. Rep. 1981:15 - 81.

[11] Wise,Michael J. Running Karp - Rabin Matching and Greedy String Tiling[R]. Basser Department of Computer Science Technical Report 463,1993.

[12] The Honeynet Project. http://www.honeynet.org/misc/project.html.

(上接第 270 页)

所示,协作模块有利于分享结果。每个 IDS 在自己的范围内分析用户的行为然后通过跟 P2P 网络类似的形式(所有 IDSs 都是对等的)分享给其它 IDSs。

本实验中采用了真实的实验数据进行实验,数据采集于实验室局域网。实验分两个步骤进行,第一个步骤是入侵检测代理收集相关数据;第二个步骤是对相关数据进行分析并产生报警。在相同的规则下,报警数量显示报警数量越多,丢包量越少,丢包率越低。实验结果如图 6 所示。

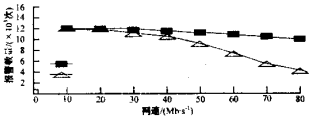

图 6 不同情况下的报警情况

图 6 中的数据是 3 次重复测试的平均值,其中,A 表示 8 个入侵检测代理,8 个入侵检测服务器;B 表示 2 个入侵检测代理,4 个入侵检测服务器的情况。实验显示,在较低流量情况下,该 GIDA 可以很好地检测网格入侵,丢包率低,但随着流量的增大,GIDA 中需要相应地增加信息收集代理和入侵检测代理,动态合理地分配流量,才能使丢包率相对下降,提高系统的检测效果。

5 结 语

在 GIDA 模型中,检测代理运行在每个节点上检测本地入侵,跟其它代理协同追查入侵源并作出响应。GIDA 超越了网格环境中传统基于网络或基于网络的入侵检测系统,体现出了该模型的优越性。通过使用判别分析法来准备分析数据,提高了检测入侵的准确性。不同资源和异构 IDSs 的信任关系是下一步要研究的重点。异构 IDSs 和相互间的信任关系对于协同模块来说需要更复杂的算法,这也是需要更深入的研究。

参 考 文 献

[1] Foster. I,Kesselmen C. (eds.). The Grid:Blueprint for a New Computing Infrastructure. Morgan Kaufmann,1999.

[2] Foster. I,Kesselmen C. Globus:A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications,1997.

[3] Lewis. M,Grimshaw A. The Core Legion Object Model. In proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing,1996.

[4] Lizkow M,Livny M,Mutka M. Condor-a hunter of idle workstations. In proceedings of the 8th International Conference on Distributed Computing Systems,1998.

[5] Nagaratnam N,Janson P,Dayka J,et al. The Secuirty Architecture for Open Grid Services. Open Grid Service Architecture Security Working Group (OGSA-SEC-WG). Global Grid Forum,2002.

[6] Murshed M,Buyya R,Abramson D. GridSim:A Grid Simulation Toolkit for Resource Management and Scheduling in Large-Scale Grid Computing Environments. 17th IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2002),April 15 - 19,2002,Fort Lauderdale,FL,USA.

[7] Tolba M,Abdel-Wahab M,Taha I,et al. Distributed Intrusion Detection System for Computational Grids. Second International Conference on Intelligent Computing and Information Systems,March 2000.

基于主动防御模式下蠕虫病毒特征码的提取模型及算法设计

作者： 刘国柱， 尚衍筠， Liu Guozhu， Shang Yanjun
作者单位： 青岛科技大学信息科学技术学院,山东,青岛,266061
刊名： 计算机应用与软件ISTIC
英文刊名： COMPUTER APPLICATIONS AND SOFTWARE
年，卷(期)： 2009,26(6)

参考文献(12条)

1. Mark Pickett A Guide to the Honeypot Concept 2003

2. 王利林;许榕生 基于主动防御的陷阱网络系统[期刊论文]-计算机工程与应用 2002(17)

3. Lance Spitzner Honeypots:Definitions and Value of Honeypots 2002

4. Chtmming Rong;Geng Yang Honeypots in Blackhat mode andits implications 2003

5. Iyad Kuwatly;Malek Sraj;Zaid A1 Masri A dynamic Honeypot design for intrusion detection 2004

6. Maximillian Dornseif;Thorsten Holz;Christian N Klein NoSE-BrEaK-Attacking Honeynets 2004

7. Singh S;Estan C;Varghese G Automated Worm Fingerprinting 2004

8. Prechelt L;Malpohl G;Philippsen M Finding Plagiarisms Among a Set of Program s with JPlag 2000

9. Karp R M;Rabin M O Efficient randomized pattern matching algorithms 1987(02)

10. Rabin M Fingerprinting by random polynomials 1981

11. Wise Michael J Running Karp-Rabin Matching and Greedy String Tiling 1993

12. The Honeynet Project

本文读者也读过(8条)