doi:10.3772/j.issn.1002-0470.2014.10.002

可生存系统的自主可识别性机制研究①

赵国生②* 刘海龙③* 王 健**

(*哈尔滨师范大学计算机科学与信息工程学院 哈尔滨 150025)
(**哈尔滨理工大学计算机科学与技术学院 哈尔滨 150001)

摘 要 提出了一种可生存系统的自主可识别性机制。按此机制,可识别性通过定义面向生存性的可识别性参数实现,自主性通过自主识别单元控制进程实现。首先,定义了若干可识别性参数,依据可识别参数的累计分布函数确定了动态可变的阈值约束;然后,基于可识别参数的参考基点,给出了服务连接可识别性检测的计算方法,并通过自主识别单元来控制实现服务连接的断开和系统资源的释放;最后,在 Emulab 环境下使用实时的 ClarkNet WWW 服务器的访问数据进行了仿真性能实验,结果显示上述方法可以有效地识别出非法连接并且重新分配系统资源给合法用户,提高了系统可生存性的自主认知能力和服务承载能力。

关键词 可生存性,可识别性,自主识别单元,累计分布函数(CDF)

0 引 言

任务关键系统现有的安全状况是无论怎么加强入侵防御和攻击检测,关键服务进程仍然会遭到恶意破坏,失效不可避免。如何在受到入侵攻击、遭到破坏时尽可能保证现有关键服务请求的持续、及时完成,提高系统的可生存性(survivability),已成为急需解决的问题。可生存性研究是下一代网络安全的核心目标,代表着网络安全研究发展的新方向。根据 Westmark[1] 和 Ellison[2] 的定义,生存性可以通过系统的可抵抗性(resistance)、可识别性(recognition)和可恢复性(recovery)(简称 3R 属性)三个方面来描述,这 3R 属性代表了生存性的 3 个热点研究方向:防护、监视和响应。其中,可抵抗性反映了系统基本的生存性需求;可识别性反映了系统对自身状态和环境的认知能力;可恢复性则描述了系统被攻击后的自适应和自修复能力。但是,目前现有研究较多关注的是生存性定义[1,2]、仿真模型分析[3,4]、生存性定性定量评估[5-7]、形式化建模[8-10]、生存性保护[11,12]、应急恢复[13,14]等,着重在可抵抗性、可恢复性两方面进行的工作,而对可识别性机制的研究仅局限在卡内基梅隆大学(CMU)、美国计算机紧急事件响应小组协调中心(CERT/CC)、美国国防先进研究项目局(DARPA)等几个研究机构,且大部分研究仍然停留在模拟和仿真阶段。因此,本文在承接前期生存性的研究成果[15]的基础上,将着重从可生存性的可识别性方面研究主动、自适应性的生存性可识别机制。

目前学术界对生存性的可识别性的典型认识是:发现正在进行的攻击或攻击的前奏,识别当前系统生存状态的能力[16]。生存性范畴内的可识别性更强调对整个系统内外部环境状态的监控和识别,我们可以将系统状态迁移的识别依次转化为对基本服务性能下降的识别、对攻击事件情景集的识别和对各种攻击事件的识别。攻击识别是指系统识别攻击的能力。在面临攻击时响应或识别的能力主要集中在系统能否从攻击中生存下来,攻击的识别是提高系统可生存能力的一项重要内容[17]。可识别性是系统识别攻击或者攻击扫描的能力,这种面临入侵的反应和适应能力是系统应对攻击的核心能力[18]。可识别性反映了军事信息网络对自身状态

① 国家自然科学青年基金(61202458),高等学校博士学科点专项科研基金(20112303120007)和黑龙江省教育厅科学技术项目(12521146,12511099)资助项目。
② 男,1977 年生,博士,副教授;研究方向:可生存系统,认知网络,自律计算;E-mail: zgswj@163.com
③ 通讯作者,E-mail: hsdlhl@sina.com
(收稿日期:2014-05-13)

和环境的监视能力,可从系统对事件识别率和对事件的识别时间 2 个方面来分析[19]。

本文认为,对于一个具有高可生存性的系统的可识别性研究,应从系统内部的服务状态和外部的攻击两个角度入手。对系统内部运行状况的识别包括服务进程状态监控,操作系统的协议堆栈、数据完整性检测,以及维护服务进程各部分间的信任模型等,这方面的研究现有学术机构多有所涉及。而面向生存性的外部服务攻击的识别研究主要侧重研究主动、自适应性的生存性可识别机制,以及在检测到攻击时及时进行自调节和自配置的可生存性规避机制,以使系统持续提供服务。这方面研究还未见报道。

1 可识别性参数

可识别性研究可以从访问用户和系统当前服务连接状态参数作为参考,在本文中以下可识别参数指标被用来对可生存系统的外部服务连接状态进行可识别性检测。

1.1 可识别参数的定义

(1) Uptime(工作时间)和 Downtime(不工作时间)

Uptime 代表访问用户从服务请求连接开始到断开服务连接所经历的时间间隔。Downtime 代表访问用户从断开服务连接到再次请求服务连接的时间间隔。在没有入侵发生的情况下,系统可以随机选择正常访问用户的 Uptime 和 Downtime 参数值,形成参数值集合{Uptime}、{Downtime},然后就可以计算获得正常访问的用户基于 Uptime 和 Downtime 两个参数的累计分布函数(cumulative distribution function,CDF)。

(2) Request rate(请求速率)和 Download rate(下载速率)

Request rate 表示可访问用户在单位时间内发出的服务连接请求次数。Download rate 表示可访问用户的服务下载次数。系统通过随机统计不同可访问用户在单位时间内的 Request rate 和 Download rate,可以计算获得正常访问用户基于 Request rate 和 Download rate 两个参数的累计分布函数(CDF)。

(3) Browsing behavior (浏览行为)

用户 Browsing behavior 取决于两个方面,一个是网站的层级结构,二是用户的浏览习惯。一般来说,一个网站是由很多个内容页面构成的,这些内容页面通过有层级结构的超链接彼此关联。用户的浏览习惯一般代表着可访问用户更愿意浏览哪些页面,有哪些超链接会被正常访问用户直接或间接访问。

(4) Page access rate(页面访问率)和 Page popularity(页面喜好度)

对一个网站来说,每个页面都有不同的 Page access rate,一些页面高频率被人浏览而另一些页面却是很少有人访问。研究指出,一个 Web 网站仅有 20% 的内容被客户高频访问(约 80%),而且文件被访问的概率呈 zipf 分布[20]。这些参数对于判断系统是否遭到入侵攻击具有重要意义,因为一个入侵者是不具备一个正常可访问用户的上网行为。

Page popularity 计算公式为

$$p_i^t = a_i^t / \sum_{j=1}^{N} a_j^t \tag{1}$$

其中 P_i^t 代表页面 i 的用户喜好度,t 代表访问时间间隔,N 代表可访问网站的页面总数量,a_i^t 代表一个页面 i 在时间间隔 t 内的用户访问率。

每一个页面依据喜好度可以被分成若干等级,如 low class、medium class 和 high class。每一等级还可以细分成更小的等级类,如 very low class、very very low class、very high class…。对每一个等级类,系统就可以描述出用户对该等级类内各页面的喜好度或访问率的 CDF。系统依据 CDF 就可以为每一个等级类定义一个可访问阈值,超过阈值的访问者被认为是可疑用户或加入黑名单。

(5) Hyperlink click number(链接点击数)

Hyperlink click number 假设一个可访问页面有 i 个超链接。在正常服务或无攻击发生情况下是统计计算出这 i 个超链接被用户点击的百分比率。这样就可以设定一个点击阈值 j,如果当前对某个超链接的点击数 z 和阈值 j 比较有较大的偏差,则这个用户是可疑用户的概率就越大。

(6) Hyperlink depth(链接深度)

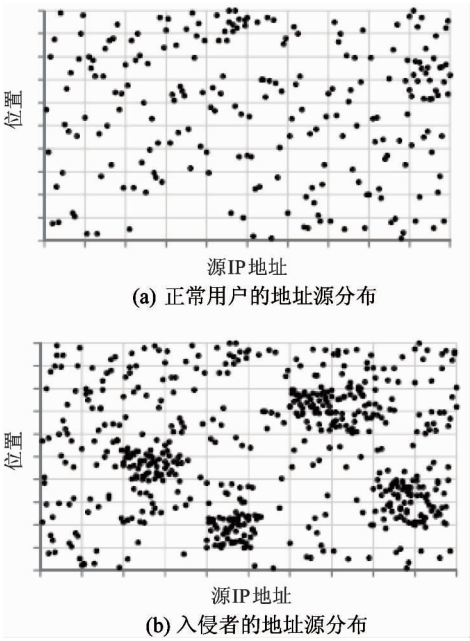
Hyperlink depth 表示可访问用户逐层点击有序链接页面的多少。同理,系统也可以为这个参数计算 CDF。

还有一些其他的访问行为类可识别参数,比如页面分类、用户分布、用户到达率和过期页面等,可以参考文献[21]。

(7) Source IP address distribution(源 IP 地址分布)

正常合法可访问用户的源 IP 地址一般都会广

泛地分布在服务网络上。而入侵者的 IP 地址源分布大多数情况下都会集中在一个 IP 地址集束内,因为入侵者为了控制方便会捕获大量在一个 IP 地址集束内僵尸肉鸡后(一般是同一局域网)才开始对服务网络进行入侵攻击。图 1(a)可以看出合法用户的地址源分布比较均匀且分散,图 1(b)中则出现 4 个较明显地址密集区,说明这些地址密集区中 IP 很有可能就是入侵者电脑。



(a) 正常用户的地址源分布

(b) 入侵者的地址源分布

图 1　可访问用户地址源分布图

1.2　可识别参数有效性分析

Source IP address distribution 参数与 Request rate 和 Download rate 是两个矛盾的参数。假设一个攻击者想要避免源 IP 地址集中在一个区域内的情况发生,那么攻击者就会减少僵尸肉鸡的数量。在一个局域网内,攻击者尽量只选取一部分僵尸肉鸡,从而让服务器看起来僵尸肉鸡的分布分散在各个不同的网段。而僵尸肉鸡的减少意味着要想完成一次有效的攻击就要提高僵尸肉鸡的请求速率和下载速率。而过高的请求速率和下载速率与一个合法访问用户的请求速率和下载速率显然是有差别的。所以攻击者很难通过模仿合法用户来逃避服务器的综合判断。

对于 Hyperlink click number 和 Hyperlink depth 两个参数,攻击者也会遇到同样的问题。因为这两个参数之间也存在矛盾。首先任一网页的 Hyperlink click number 都有一个阈值,而攻击者是无法猜测到这个阈值的。一个攻击者可以在每个网页上尽量少地点击超链接,这样就会保持在服务器设置的阈值之下,但是一个攻击者无法获知每个网页的 Hyperlink click number 的阈值是多少,就算他知道阈值是多少,但是如果想达到一个理想的攻击的话,每个网页的 Hyperlink click number 越少的话,就意味着 Hyperlink depth 会增加,那么可能他就会超过 Hyperlink depth 这个参数所设定的阈值。

综上所述,本节定义的可识别参数对判断一个连接是否是攻击者发起的是有效的,因为对于一个外部的攻击者来说要想收集到这些参数的阈值是十分困难的。

2　自主识别机制

2.1　可识别性检测模型

首先,假设一个服务连接请求为 v,那么这个连接 v 就会有 i 个可识别参数 A_i^v, 参数 A_1^v 可以代表 Request rate, A_2^v 可以代表 Download rate$\cdots$ A_n^v。$S(A_i^v)$ 就是连接 v 对应参数 A_i 的可识别值。可以对一个连接 v 的所有可识别参数的识别值求和:

$$S(v) = \sum_{i=1}^{n} S(A_i^v) \tag{2}$$

其中,n 表示可识别参数的数量。

下面给出一个可识别性参数的可识别检测值的建模与计算过程。这些参数包括 Uptime, Downtime, Request rate, Download rate, Page popularity 和 Hyperlink click number 等。假设 $y = f_i(x)$, f_i 表示可识别参数 A_i 的累积分布函数(CDF),自主识别单元会为每一个可识别参数初始确定一个数对 (x_b, y_b) 作为可识别参数 A_i 的基础参考基点,其中 $y_b = f_i(x_b)$。自主识别单元会通过回馈控制进程选取一个合适的基础参考基点。假设连接 v 的可识别参数 A_i 用 x 来表示,那么我们给出一个计算已识别参数的检测分值公式:

$$S(A_i^v) = \begin{cases} 0, & x_i^v \leqslant x_b \\ -1 \times k^{\mathrm{div}(\frac{x_i^v - x_b}{\Delta x})} \times \dfrac{x_i^v - x_b}{\Delta x}, & x_i^v > x_b \end{cases} \tag{3}$$

其中 k 表示一个初始几何定值,例如 1,2,$\cdots$,n。$\mathrm{div}(p/q)$ 表示 p 和 q 取商,Δx 表示一个恒定的比例因子。x_i^v 对于基础参考基点的偏离量越大,那么在这个公式中所得到的 $S(A_i^v)$ 值就越低。系统可以根据入侵的攻击频率选择适当的定值 k 和初始 Δx 值。

但是,对于 Downtime 参数的可识别性的检测分值计算则是与其他参数的计算方法相反,计算公式如下:

$$S(A_i^v) = \begin{cases} 0, & x_i^v \leqslant x_b \\ -1 \times k^{\mathrm{div}(\frac{x_b - x_i^v}{\Delta x})} \times \dfrac{x_b - x_i^v}{\Delta x}, & x_i^v > x_b \end{cases} \tag{4}$$

对于地址源分布,Source IP address distribution 参数通常来说是个定值。如果系统经过统计后可初步猜测这个 IP 地址是一个入侵者的 IP 地址概率较大的话,那么系统就会给出一个定值,例如 -1,否则定为 0。

2.2 自主识别单元

自主识别单元的引入一是为了实现系统可生存性的自主增强,二是防止系统的有限服务资源被恶意攻击消耗殆尽,导致系统无法持续、及时提供正常的关键服务。自主识别单元首先为系统的每一种可量化分析的服务资源定义 T_1、T_2、T_3 三种阈值划分,并且分别标记为黑色、棕色和白色三种预警状态,而 T_1、T_2 和 T_3 值的确定可通过统计分析和先验知识求得,通常设定为资源总量的 90%、80% 和 60%。当系统的资源消耗超过 T_1 时,设定系统当前的资源使用状况为黑色预警,若系统当前的资源使用状况处于 T_2 和 T_3 之间时,将系统资源状态设定为棕色预警,而当系统处于 T_3 以下时,将系统的资源状态设定为白色(正常状态)。

自主识别单元会定期地检查系统的各种量化资源状态,例如 bandwidth occupancy rate、cpu、memory usage、context switches、System call、Interrupts、Available bytes、current disk queue length 等。当系统服务资源占用率超过 T_1 时,自主识别单元的控制进程被触发,系统被动进入黑色警戒模式,这时系统就不再允许新访问用户的连接服务请求。然后,自主识别单元利用公式(3)和公式(4)计算现有每个服务连接对于所有可识别参数的一个综合检测分值,如果分值低于阈值的话,就强行阻断连接释放所占资源,直到系统可用资源恢复到 T_2 下。当资源状态不是黑色时,系统脱离报警模式且可接受新的可访问用户的服务连接请求。当资源状态是棕色时,如果有新的用户服务连接请求 v,那么自主识别单元首先会判断 v 的 IP 地址是否在黑名单中。如果没有在黑名单中,则需要计算下 v 的 Downtime 参数的可识别性检测分值 S_d^v。然后将新检测分值 S_d^v 与现有系统中服务连接的可识别性的检测分值 $\{S_d^1, S_d^2 \cdots\}$ 比较,如果 $S_d^v > \min\{S_d^1, S_d^2 \cdots\}$,那么就将最小的这个 $\min\{S_d^1, S_d^2 \cdots\}$ 连接断开并释放资源后,接受新的服务连接请求 v。

现有服务连接 v 因系统资源占用处于预警状态需被断开,则需遵循三条规则:

(1) $S(v) = 0$ 的服务连接不能被断开;

(2) $S(v) \geqslant T_0$,自主识别单元就给用户发送一个 puzzle 测试;

(3) $S(v) < T_0$,连接 v 被放入可断开连接的候选队列中。

其中,T_0 为系统初始阈值。

2.3 阈值的自主调整模式

阈值的确定可以先根据以往的攻击情况和系统状态来设定一个初始阈值,随着系统资源占用的变化和服务连接动态剥夺调整的需要,这个阈值可以随着可识别参数初始参考基点的动态变化而自主地动态调整。

如果至少有一种系统可用资源占用处于黑色警戒状态并且没有任何现有连接的可识别检测分值低于初始阈值,那么自主识别单元就需要重新对可识别若干参数设定一个新的参考基点。参考基点调节如下:假设在一个可识别参数的累积分布函数中,(x_b, y_b) 和 (x_b', y_b') 分别是当前参考基点和下一个参考基点。选取一个正量的几何量值 Δy,Δy 的取值可以取为0.05或者0.1,每次减少一个 Δy,可以得到一个新的坐标:$y_b' = y_b - \Delta y$,$x_b' = f^{-1}(y_b - \Delta y)$。所以调整后的新参考基点坐标是 $(x_b', y_b') = (f^{-1}(y_b - \Delta y), y_b - \Delta y)$。

对于 Downtime 参数,因为攻击者总是希望尽可能快速攻破服务器或一个服务系统,因此当防御系统阻挡住一次攻击后,入侵者大概率事件会马上发起下一次探测攻击,因此 Downtime 参数的下一基点的计算方法略有不同,它的值是不断变大,新的数对计算:$(x_b', y_b') = (f^{-1}(y_b + \Delta y), y_b + \Delta y)$。

Source IP address distribution 参数因为不存在累积分布函数和初始参考基点,因此不需要做出阈值调整策略。

2.4 自主识别单元的工作过程

(1) 自主识别单元为每个可识别参数定义一个初始参考基点,初始参考基点值的选取要能够使误判最小化,一般设为较大的值,使最初阶段的误判为 0。

(2) 自主识别单元定时对现有服务连接计算给出一个当前可识别检测分值。

(3) 自主识别单元会定期断开一些可识别检测

分值较低的连接,直到没有一个负值的连接存在,或系统资源恢复到 T_2 以下。

(4) 若系统资源状态是处于 T_2 或 T_3,那么就会终止预警模式。

(5) 若所有的系统资源状态处于 T_3,那么自主识别单元的进程就会被终止。

(6) 若至少有一项系统资源状态处于 T_1,并且没有任何连接的可识别检测分值低于阈值 T_0,那么自主识别单元按照 3.3 节介绍重新设定一个适当的初始基点。

(7) 返回到(2),这个循环一直持续到步骤(5)的条件满足后终止。

自主识别单元控制过程补充说明:(1)不是在每个循环迭代中,所有参数的基点值都会改变。有时,控制单元可能只改变某一些参数的基点值。(2)当前连接被中断时,其 IP 地址和检测分值将被记录在一个称为黑名单的 2 维列表中。(3)因为合法连接在系统资源紧张时也有可能得到一个负的检测值,因此系统可以选择一个合适的"降低门槛"策略。此时,基点可以取值合法连接得到的最大负分的绝对值即可。由此这些合法连接才会被系统接纳,而不是被当作非法连接断掉。

3 仿真与结果分析

基于可信虚拟机,我们在校园网内搭建了一个仿真实验环境,该网络对外只提供一种 WEB 关键服务,在实验中我们测试了两周内所有对于该实验网络系统的所有 http 请求。图 2 给出了可识别参数的 CDF 图。

图 2(a)中大约 50% 的可访问用户 Uptime 值小于 90s,80% 的访问用户 Uptime 值小于 300s,而只有 5% 左右的用户 Uptime 值大于 960s。所以我们选取(10,0.9)作为 Uptime 参数的初始参考基点。在图 2(b)中大约 54% 的访问用户的 Downtime 值大于 8h,80% 的访问用户的 Downtime 都超过 4h,而只有 3% 的访问用户的少于 1h。所以我们选择(3,0.1)作为 Downtime 参数的初始参考基点。

在图 2(c)中 50% 的访问用户 Request rate 小于 0.066 次/s,每 15s 会访问一次 web 服务器。大约 10% 的访问用户每 2s 发送一个以上的请求。因此 Request rate 参数的初始参考基点设置为(0.5,0.9)。在图 2(d)中超过 51% 可访问用户的 Download rate 低于 1000b/s。大约超过 75% 的访问用户

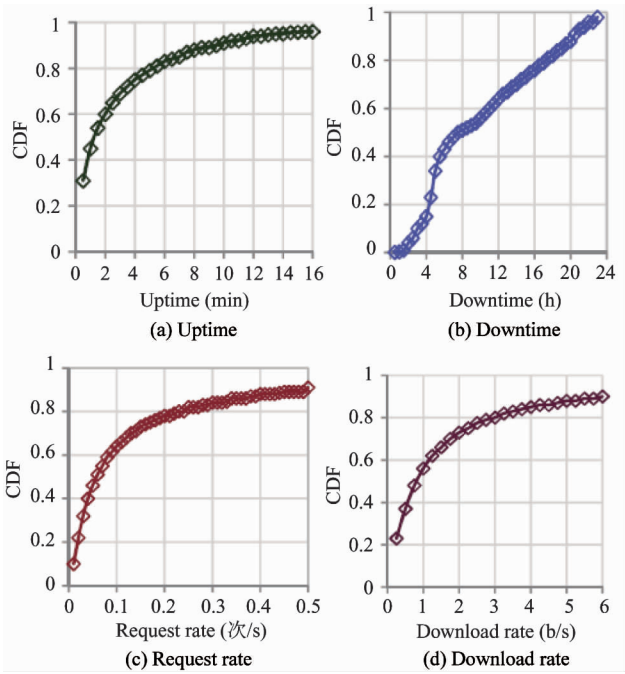
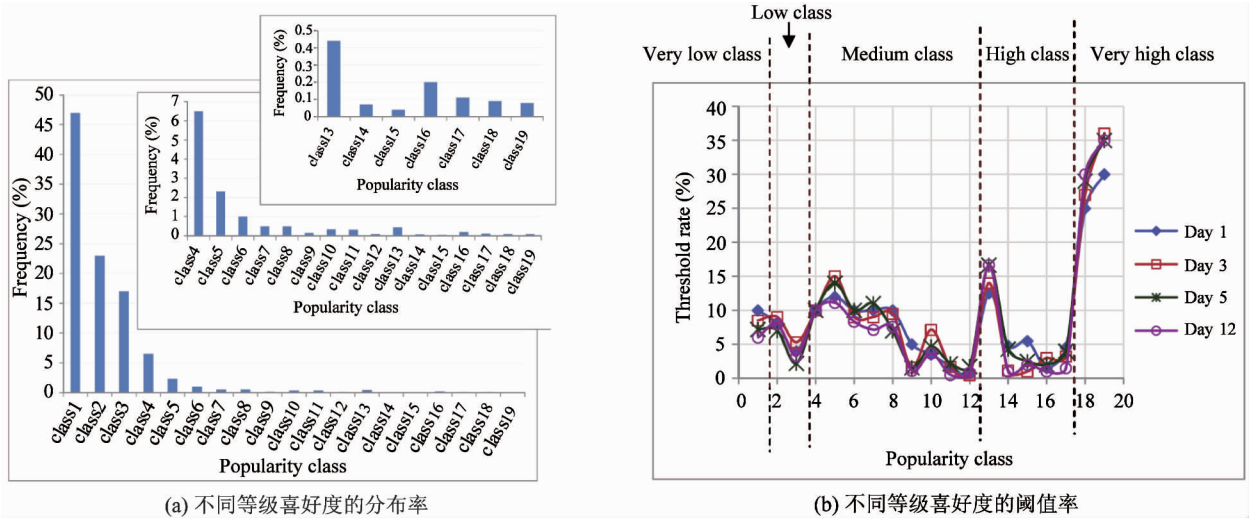


图 2 可识别参数的 CDF 图

的下载速率低于 2500b/s,而只有 9% 的访问用户的 Download rate 超过 5000b/s。所以我们选择(5000,0.9)作为 Download rate 初始基点。

图 3(a)中给出的是两周内每天 10:00 – 15:00 网页喜好度的分布图。试验中把喜好度分为 5 个等级:very low class、low class、medium class、high class 和 very high class。在图 3(a)中 very low class 由 class1 组成,而 low class 细分为 class2-class3,medium class 由 class4-class12 组成,high class 由 class13-class17 组成,very high class 则细分为 class18-class19。图 3(b)显示的是第 1、3、5 和第 12 天基于 Page popularity 的 CDF 得到的不同等级喜好度的阈值率(threshold rate)。可以看出,虽然日期不同,但是每一个喜好度等级类的阈值率都大致保持在一个相似的范围值内。因此,我们可以确定一个喜好度等级类的上限阈值率 T_0 和日期无关的一个固定阈值率 T_0'。

为了验证所提方法对系统生存性的可识别性能力检测的有效性,我们在 EMulab 环境下建立了 Clarknet www 服务的仿真试验。时间跨度两周,共计产生 7532 大小不同的可访问页面然后,在 15:00 前将它们上传到 web 服务器。每次攻击开始时间定为 14:03,系统 Band width 设定为 10^6Byte/s。

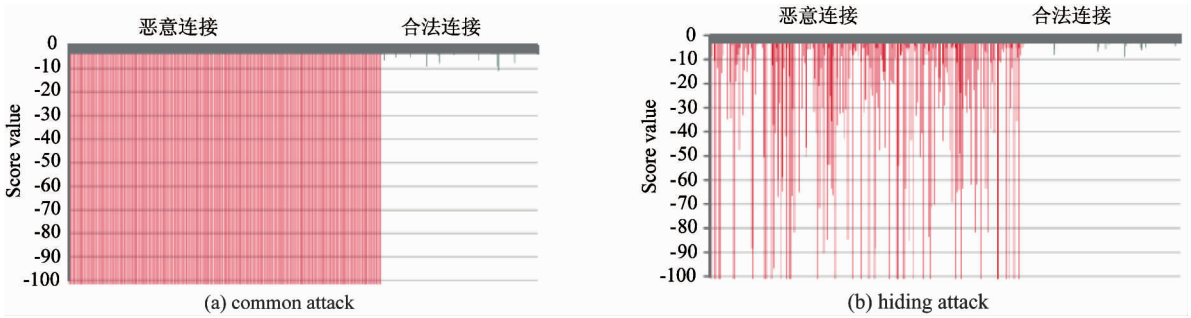


(a) 不同等级喜好度的分布率　　(b) 不同等级喜好度的阈值率

图 3　喜好度的分布和阈值率

在试验中我们可以得到:(1)所有连接的可识别性检测分值;(2)系统服务从一次攻击中恢复的速率 R_C;(3)误判率(FP)和漏报率(FN);(4)合法连接可能误判而被断掉的百分率(PFP);(5)合法连接得到负分值的概率(R_L)和恶意连接却没有得到负分值的概率(R_A)。实验中我们采用了两种攻击方式:common attack 和 hiding attack。对于 common attack 我们使用了 200 台主机作为攻击者的角色,采用 Netsky. Q、Trojan Sientok 和 BlueCode. Worm 这 3 种病毒程序向 web 服务器发送 get 请求,这三种病毒程序的请求速率分别是 300ms、250ms 和 130ms。而对于 hiding attack,300 台攻击主机以合法用户相同的请求速率和下载速率来向 web 服务器发出 TCP 连接请求。

图 4 显示的是入侵发生时的系统生存性的可识别性检测分值(score value)分布图,红色条表示入侵者非法连接的得分,而绿色条代表合法用户正常连接的得分。大约有 7.5% 的合法用户得到了一个负的检测分值,但绝对值很小(最大 11)。在 common attack 中,大多数的恶意连接都得到了一个较低(负数)的综合检测分值,这主要是因为攻击者会使用一个较高的 Request rate 连接服务器,所以 Request rate 这个参数的可识别检测分值就非常低。在 hiding attack 中,不是所有恶意连接会得到负的检测分值,而且这个负的检测分值的绝对值有可能还要低于 common attack 检测分值的绝对值。



(a) common attack　　(b) hiding attack

图 4　入侵发生时的可识别性检测值分布

图 5 表示的是在 hiding attack 下合法流量和攻击流量占用的带宽比,对于 common attack 有相似的结果。在 14:03 发起恶意 hiding attack,此时系统进入警戒模式,攻击发生一分钟后(R_C = 1),由于自主识别单元断开了部分得到负值的连接,服务器的资源占用降到了 threshold2 以下。在 14:04 服务器关闭警戒模式,开始允许新的用户连接到服务器。控制单元继续断开较低检测分值的连接,直到服务器可用服务资源处于 threshold3 以下。当服务器的可用资源恢复到 threshold3 以下时,服务器终止控制单元的进程。可以看到,在 14:03 和 14:04 间,由于一些合法用户的离开而导致所占用的带宽缓慢地

降低。

为了计算误判率(FP)、漏报率(FN)、合法连接可能误判而被断掉的百分率(PFP)、R_L 和 R_A,我们在不同时间重复做了 30 次试验。在 hiding attack 中,即使有相同的请求速率,但由于有源 IP 地址分布参数约束,平均仅有 6.3% 的恶意连接没有得到负分值,而在 common attack 中所有的恶意连接都得到了负的检测分值,由此 FN 和 R_A 的值为 6.3%。实验中平均有 7.5% 的合法连接得到了负的检测分值,即 FP 和 R_L 的值为7.5%。最后,平均有 2% 的合法连接得分值低于 -10(断开连接的阈值为 -10),这时若系统应急参数的阈值得到向下调整,那么此合法连接成为候选合法连接而等待下次调度,否则系统将会断开此连接,即 PFP 为 2%。本文在此把系统的可生存能力简单定义为合法用户得到服务的概率和屏蔽非法用户服务请求的能力。由此,我们可以粗略地计算出系统当前的可生存能力为 84.2%。

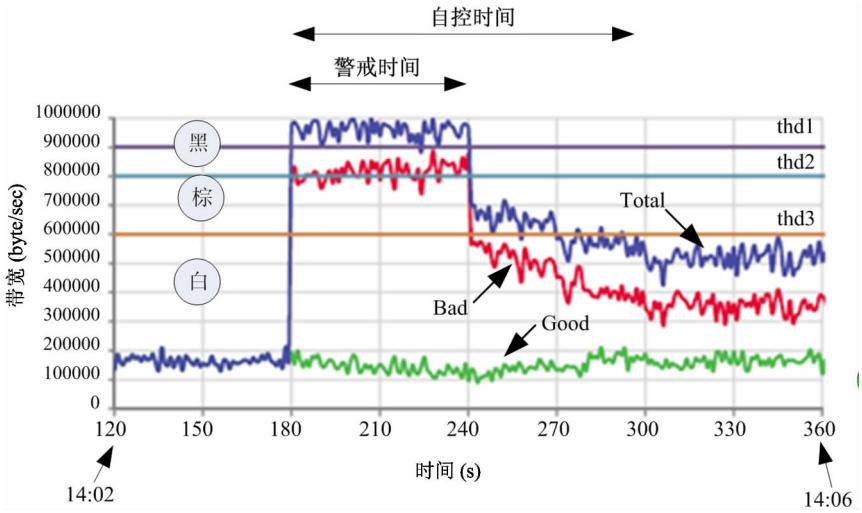


图 5　合法流量和攻击流量占用的带宽

4　结　论

本文从生存性的自主可识别性方面入手,侧重研究了可识别性检测参数定义、自主识别模式以及阈值可变方式等,以提高系统生存性,使系统持续提供服务。提出了一种可生存系统的自主可识别性检测机制,在 Emulab 环境下使用实时的 ClarkNet WWW 服务器的访问数据进行了性能仿真,实验结果显示本文方法提高了系统可生存性的自主认知能力和服务承载能力。

参考文献

[1] Westmark V R. A Definition for Information System Survivability. In: Proceedings of the 37th Hawaii Internal Conference on System Sciences, Track 9, Washington, IEEE Computer Society Press, 2004. 2086-2096

[2] Ellison R, Linger R, Longstaff T. Survivability network system analysis: A Case Study. *IEEE Software*, 1999, (04):70-77

[3] Ellison R J, Moore A P. Trustworthy Refinement through Intrusion-Aware Design (TRIAD): An Overview. In: Proceedings of the 3rd Annual High Confidence Software and Systems Conference, Baltimore, MD, 2003. 1-10

[4] Richard C L, Howard F L, John M, et al. Life-Cycle Models for Survivable Systems. CMU/SEI-2002-ESC-TR-026, Boston: Carnegie Mellon University, 2002

[5] 赵国生,王慧强,王健. 一种基于灰色关联分析的网络可生存性态势评估方法. 小型微型计算机系统,2006,27(10):1861-1864

[6] 王健, 王慧强, 赵国生. 基于序列蒙特卡罗的网络生存态势跟踪评估. 哈尔滨工业大学学报, 2008,40(5):802-806

[7] Wang J, Wang H Q, Zhao G S. A situation assessment method for network survivability. *Wuhan University Journal of Natural Sciences*, 2006,11(6): 1785-1788

[8] Wang J, Wang H Q, Zhao G S. Formal Modeling and Quantitative Evaluation for Information System Survivability Based on PEPA. *The Journal of China Universities of Posts and Telecommunications*, 2008, 15(2): 88-96

[9] 王健,王慧强,赵国生. 分布式任务关键系统生存性自

动分析与验证. 高技术通讯, 2009,19(6): 572-579
[10] Zhao G S, Wang H Q, Wang J. A Novel Formal Analysis Method of Network Survivability Based on Stochastic Process Algebra, *Tsinghua Science & Technology*, 2007, 12(1):175-179
[11] Zinky J A, Bakken D E, Schantz R. Architectural Support for Quality of Service for CORBA Objects. *Theory and Practice of Object Systems*, 1997, 3(1):55-73
[12] Matti A H, Richard D S, Carlos A U, et al. Survivability through Customization and Adaptability: the Cactus Approach, In: DARPA Information Survivability Conference and Exposition (DISCEX 2000), Hilton Head, SC, USA, 2000. 294-307
[13] Xie W, Hong Y, Trivedi K. Analysis of a Two-Level Software Rejuvenation Policy. *Reliability Engineering and System Safety*, 2005,87(1):13-22
[14] Patterson D, Brown A, Broadwell P, et al. Recovery Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies. http://roc.cs.berkeley.edu/papers/ROC_TR02-1175.pdf: Berkeley USA, 2002
[15] 赵国生. 任务关键系统生存性应急增强技术研究[博士学位论文]. 哈尔滨:哈尔滨工程大学,2009. 10-150
[16] 余智华,林思明,陈海强. 网络安全-可生存性研究及网络建模. 北京, 2012
[17] 软件可生存性报告. http://www.doc88.com/p-147660727345.html: 北京:2012
[18] 赵淦森,王维栋,张伟等. 云计算平台生存性研究. 电信科学学报, 2011,9,52-59
[19] 蔡均平,肖治庭,李雪东. 基于云模型的军事信息网络可生存性评估. 武汉理工大学, 2010,32(20),11-15
[20] Xie Y, Yu S. A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors, *IEEE/ACM Transactions on Networking*,2009,17(1): 54-65
[21] Beitollahi H, Deconinck G. ConnectionScore: A Statistical Technique to Resist Application-layer DDoS Attacks. http://www.esat.kuleuven.be/electa/publications/fulltexts/pub 2313.pdf. University of Leuven, Belgium: 2012

Study on the autonomous recognition mechanism for survivable systems

Zhao Guosheng*, Liu Hailong*, Wang Jian**
(*College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025)
(**School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150001)

Abstract

An autonomous recognition mechanism for a survivable system was presented. According to the mechanism, the recognition can be achieved by definition of the recognition parameters referred to the survivability, and the autonomy can be realized through the process for control of the autonomous recognition unit. Firstly, a number of recognition parameters were defined, and according to the cumulative distribution function, the dynamic and variable threshold constraints were determined; Then, based on the reference mark of the recognition parameters, a calculation method for recognition detection of service connections was given, and the autonomous recognition unit was used to implement service connection, disconnection and system resource release; Finally, in the Emulab environment, the accessing data of the real-time ClarkNet WWW server was used to carry out the simulation experiment. The results showed that the proposed mechanism effectively recognized the illegal connections and reassigned the system resources to legitimate users, with the improvements of the survivable system's service capacity and autonomous cognitive ability.

Key words: survivability, recognition, autonomous recognition unit, cumulative distribution function (CDF)