doi:10.3969/j.issn.1000-436x.2013.12.016

基于消息篡改的端信息跳变技术

林楷，贾春福
(南开大学 计算机与控制工程学院，天津 300071)

摘 要：研究了端信息跳变技术在应用中存在的理论和技术问题，提出了基于消息篡改的跳变技术，并在此基础上建立了跳变栈模型，分别给出了跳变栈模型 3 种实现方案的工作原理及其优缺点分析。通过实验验证了基于消息篡改的端信息跳变技术的应用价值。
关键词：网络安全；拒绝服务攻击；端信息跳变技术；数据消息篡改
中图分类号：TP393.08 **文献标识码**：B **文章编号**：1000-836X(2013)12-0142-07

End hopping based on message tampering

LIN Kai, JIA Chun-fu
(College of Computer & Control Engineering, Nankai University, Tianjin 300071, China)

Abstract: The main theoretical and technical problems were studied in the application of end hopping, and the idea of message tampering was proposed and the model of end hopping stack was built upon it. Three feasible implementations for the end hopping and discuss their advantages and disadvantages were respectively provided. And the experiment result shows the potential application of the message tampering and the end hopping stack model in practice.
Key words: network security; denial of service attack; end hopping technology; message tampering

1 引言

网络和信息安全问题日益严峻，网络安全事件正以爆炸式的方式增长。Prolexic 安全工程与响应组织（PLXsert）在其 Q1 2012 安全报告[1]中指出，相比于 2011 年同季度，2012 年第一季度的总攻击数增长了 25%，其中针对金融服务企业的攻击流量正以每季度 3 000%的速度增长，分布式拒绝服务攻击（DDoS，distributed denial of service attack）总流量达 9.5 petabyte，共 408 万亿个网络分组。

DDoS 攻击是网络安全领域的始作俑者，通过发送出远超出服务器的网络带宽或计算资源的数据流量，使得服务器始终处于超负荷状态，而无法向合法用户提供服务。然而，基于单个或少数攻击节点的拒绝服务攻击往往难以形成超过服务器服务性能的数据流量，为此，攻击者往往会选择僵尸网络[2~4]作为 DDoS 的辅助工具。僵尸网络是指大量的、受控制的、分布式的、能够执行命令的计算机群体，能够为攻击者提供隐蔽的、安全的、能够自我扩张的、具备超级攻击能力的攻击平台。

在抵抗 DDoS 攻击中，传统的网络防护技术（如防火墙、入侵检测技术等）存在明显的不足，其抵御网络攻击的方法主要是检测并过滤具有异常特征或行为的数据流量[5,6]，以降低攻击效果。然而，这些传统的防护技术在面对来自于僵尸网络的 DDoS 攻击时形同虚设：由于僵尸网络的规模在理论上没有上限，即可以形成无限强度的攻击流量，以非常有限的资源来检测和过滤无限的攻击流量本身就构成了拒绝服务服务攻击；攻击者利用僵尸机可以进行真实的服务请求，使数据流量变得合

收稿日期：2012-07-04；修回日期：2013-03-19
基金项目：国家自然科学基金资助项目（60973141, 61272423）；国家重点基础研究发展计划（“973”计划）基金资助项目（2013CB834204）；高等学校博士学科点专项科研基金资助项目（20100031110030）
Foundation Items: The National Natural Science Foundation of China (60973141, 61272423); The National Basic Research Program of China (973 Program)(2013CB834204); The Specialized Research Fund for the Doctoral Program of Higher Education of China (20100031110030)

法，从而直接穿透防火墙等设备。

网络攻防对抗是一种敌众我寡、敌暗我明的不均衡对抗。如果防御者仅利用传统的网络防护技术，不足以帮助其扭转这种不均衡的局势。因此，急需研究更为高级复杂的网络防护技术，如蜜罐技术[7,8]、端信息跳变技术[9~11]等，这些防护技术能够大幅帮助防御者提升其自身的防御能力，扭转不利局面，使其在网络对抗中处于均衡，甚至有利地位。

端信息跳变技术（下文中简称跳变技术）是指在网络服务过程中不断改变通信所使用的端信息（IP 地址、端口和协议等）。在跳变技术中，将服务器隐藏在跳变系统之中，使攻击者无法获得真实的服务端信息，从而将网络攻击扼杀在初始阶段。类似于攻击者将其自身隐藏在僵尸网络之中，使防御者难以逆向追踪到自己。相关研究表明，跳变技术能够很好地保护服务器[9,10]。

一般情况下，所有的网络服务都默许一个工作准则：服务端信息一旦设定好就固定不变，如 Web 服务默认为 80 号端口、FTP 服务默认为 21 号端口。只有通过重新启动服务才能够修改端口。而且，现有的服务软件、服务平台都根据这个准则进行了程序编码和算法优化，如果贸然在服务过程中修改服务端信息势必会导致服务器系统的紊乱甚至崩溃。因此，如何在不破坏这个准则的前提下，将跳变技术的应用于现有网络服务是亟待研究的课题之一。

在研究跳变技术的应用问题之前，有必要明确跳变技术作为一种网络防护技术应该具备的基本特性，主要包括：

① 安全性，能够有效抵御网络攻击；

② 高效性，能够高效地完成所有网络任务；

③ 透明性，能够透明地应用于网络服务；

④ 兼容性，能够兼容软硬件的更新或扩充。

本文根据跳变技术的基本特性和数据消息的处理流程，提出了基于消息篡改的端信息跳变技术，建立了跳变栈模型；然后给出了跳变栈模型中各层跳变的详细方案和指标对比；最后利用实验验证了基于消息篡改的端信息跳变技术的应用价值。

2　基于消息篡改的跳变技术

2.1　数据消息的处理过程

网络通信是建立在数据消息传输的基础之上的，这里的数据消息指的是包含源目端信息（数据消息的来源和目的）和有效数据内容的二进制连续块。在网络通信过程中，发送者调用 *send* 函数将需要发送的数据封装为数据消息并填充源目端信息，接收者调用 *recv* 函数接收指定通信端信息的数据消息，工作原理如图 1 所示。其中箭头代表了数据消息的传输方向，多层空间的划分是根据操作系统的分层设计及可编程扩展的环境而设定的。

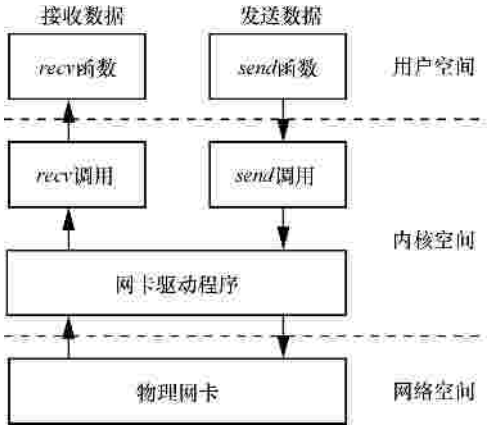


图 1　层次结构的数据消息收发过程

在发送数据的过程中，网络程序调用 *send* 函数来发送数据消息，*send* 函数调用了一个专门用于发送的 *send* 调用来发送数据消息，这个系统调用会将网络程序的缓冲区中的数据复制到一个内核空间的结构体，然后由网卡驱动程序负责发送该数据消息。在接收数据的过程中，网络程序调用 *recv* 函数来接收网络数据消息，*recv* 函数调用了一个专门用于接收的 *recv* 调用，每当接收到数据消息时，网卡会将数据消息复制到网卡驱动程序中，并存储为一个内核空间的结构体中，然后将结构体中的数据消息复制到网络程序的缓冲区中。

2.2　数据消息篡改

为了简化分析过程，定义端信息为 IP 地址和端口的二元组(a, p)，其中 a 为 IP 地址，p 为端口。考虑一种简单的网络服务：服务器在端信息(a_s, p_s)上提供网络服务，客户端绑定了端信息(a_c, p_c)，服务过程如图 2 所示。

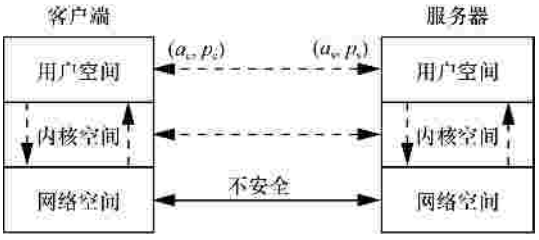


图 2　简单的网络服务

在数据消息由用户空间传输到网络空间的过程中进行篡改，篡改协议如表 1 所示，其中(a_s^i, p_s^i)指的是动态变化的端信息,其中 i 代表第 i 个跳变阶段。

表 1 数据消息的篡改协议

客户端和服务器	图 3~图 5 中的标识	方向	数据消息	数据消息（篡改后）
客户端	①	发送	$(a_c, p_c)\rightarrow(a_s, p_s)$	$(a_c, p_c)\rightarrow(a_s^i, p_s^i)$
	②	接收	$(a_c, p_c)\leftarrow(a_s^i, p_s^i)$	$(a_c, p_c)\leftarrow(a_s, p_s)$
服务器	③	发送	$(a_s, p_s)\rightarrow(a_c, p_c)$	$(a_s^i, p_s^i)\rightarrow(a_c, p_c)$
	④	接收	$(a_s^i, p_s^i)\leftarrow(a_c, p_c)$	$(a_s, p_s)\leftarrow(a_c, p_c)$

将服务端信息(a_s, p_s)设置为无法直接访问的内部端信息。这样，从服务器和客户端角度来看，客户端绑定本地端信息(a_c, p_c)同服务器(a_s, p_s)进行通信。从攻击者角度来看，客户端与不断改变通信端信息的服务器(a_s^i, p_s^i)进行通信。

2.3 篡改环境

本文暂不考虑服务器和客户端的主机系统安全问题，而主要考虑来自于网络的威胁。跳变技术必须在数据消息进入不安全的网络环境之前对数据消息进行篡改。考虑到网络编程提供了 3 种可选的方案：用户空间的原始套接字或 Winpcap 等、内核空间的 NDIS 驱动或 Kernel Module 等和网络空间的嵌入式编程等，跳变技术可以安全地分别在用户空间、内核空间和网络空间中对数据消息进行篡改。

3 跳变技术的跳变栈模型

定义端信息跳变技术的跳变栈模型为至上而下的层叠栈状结构：用户空间、内核空间和网络空间。

3.1 用户空间层的跳变技术

用户空间层的跳变技术（下文中简称用户层跳变）是指将在用户空间对数据消息篡改，工作流程如图 3 所示。

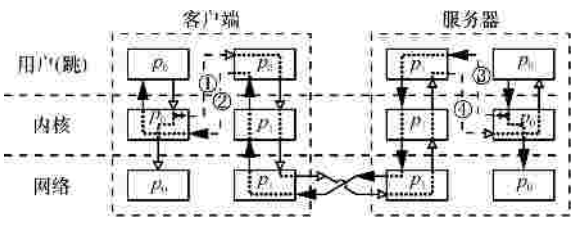


图 3 跳变技术（用户空间）

其中，p_0 和 p_1 分别代表原始的数据消息和篡改后的数据消息；带编号的虚线连接箭头代表数据消息的篡改过程，分别对应表 1 中的规则。

客户端发送数据消息时（图中空白箭头），客户端的跳变技术监听并得到需要篡改的数据消息 p_0 的拷贝，将 p_0 的目的端信息(a_s, p_s)篡改为端信息(a_s^i, p_s^i)以形成 p_1，然后将 p_1 送上网络空间；当 p_1 传输到服务器时，服务器的跳变技术监听并得到数据消息 p_1 的拷贝，将 p_1 的目的端信息(a_s^i, p_s^i)篡改为端信息(a_s, p_s)以形成 p_0。同理，服务器发送数据消息时（图中实心箭头），也遵循上述流程，不同的是篡改的字段是数据消息的源端信息。

利用操作系统提供的原始套接字或者 Winpcap 开发包可以方便地实现用户层跳变。客户端和服务器透明地进行网络通信，而攻击者去无法确定服务器的通信端信息。然而，由于操作系统没有提供用户空间的数据消息拦截功能，因此 p_0 也会传输到网络空间，这样会存在大量的冗余数据消息 p_0，即每一个数据消息都会有一个无效的拷贝。

用户层跳变的优点包括：兼容性好，能够适用于大多数操作系统；实现简单，以软件安装方式就可以部署。不足包括：不能够拦截客户端发出数据消息，这会给客户端和服务器带来没有必要的开销；而且在传输过程中，需要多次的空间切换，如发送过程中的内核空间的 p_0 到用户空间的 p_1，再到内核空间的 p_1。

3.2 内核空间层的跳变技术

内核空间层的跳变技术(下文中简称内核层跳变)指的是在内核层空间对数据消息进行篡改，工作流程如图 4 所示。

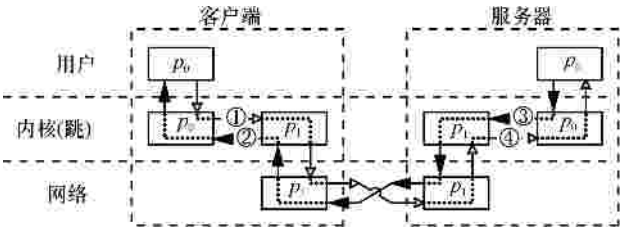


图 4 跳变技术（内核空间）

其中 p_0 和 p_1 分别代表原始的数据消息和篡改后的数据消息；带编号的虚线连接箭头代表数据消息的篡改过程，分别对应表 1 中的规则。

客户端发送数据消息时（图中空白箭头），客户端的跳变技术在内核中拦截到需要篡改的数据消息 p_0，直接将 p_0 的目的端信息(a_s, p_s)篡改为端信息(a_s^i, p_s^i)以形成 p_1，然后将 p_1 送上网络空间；当 p_1 传输到服务器时，服务器的跳变技术拦截到数据

消息 p_1，直接篡改 p_1 的目的端信息(a_s^i, p_s^i)为端信息(a_s, p_s)以形成 p_0。同理，服务器发送数据消息时（图中实心箭头），也遵循上述流程，不同的是篡改的字段是数据消息的源端信息。

不同于用户层跳变，内核层跳变运行在操作系统的内核空间中，直接在操作系统的网络协议栈上对数据消息进行篡改，不需要空间切换，能够节省大量的内存和处理器资源。

操作系统为了方便开发人员对系统本身进行深度的扩展，往往会提供一些涉及内核的编程接口，例如，Windows 为开发人员提供了基于 NDIS 的中间层驱动。

内核层跳变的优点包括：透明度高，用户层软件无法感知跳变技术的存在；直接在内核中对数据消息进行篡改具有更好的效率。不足包括：内核层跳变需要严格操作系统的版本对应，不同版本的操作系统需要不同的代码实现。

3.3　网络空间层的跳变技术

网络空间层的跳变技术（下文中简称网络层跳变）指的是在主机系统与网络交换机之间，用独立的跳变设备对数据消息进行篡改，工作流程如图 5 所示。

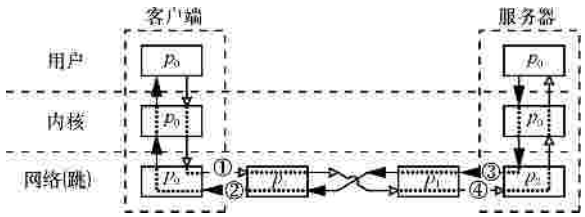


图 5　跳变技术（网络空间）

其中，p_0 和 p_1 分别代表原始的数据消息和篡改后的数据消息；带编号的虚线连接箭头代表数据消息的篡改过程，分别对应表 1 中的规则。

跳变设备是拥有 2 个网卡接口的、具有独立处理能力的嵌入式设备。客户端发送数据消息时（图中空白箭头），客户端的跳变设备拦截到需要篡改的数据消息 p_0，直接将 p_0 的目的端信息(a_s, p_s)篡改为端信息(a_s^i, p_s^i)以形成 p_1；当 p_1 传输到服务器时，服务器的跳变设备拦截到数据消息 p_1，直接篡改 p_1 的目的端信息(a_s^i, p_s^i)为端信息(a_s, p_s)以形成 p_0。同理，服务器发送数据消息时（图中实心箭头），也遵循上述流程，不同的是篡改的字段是数据消息的源端信息。

网络层跳变，相对于用户层跳变和内核层跳变，脱离了主机系统的空间范畴，在独立的跳变设备上实现跳变技术。其优点包括：完全透明，主机系统完全独立于跳变技术；无性能开销，所有消息篡改的任务由跳变设备完成，不需要消耗主机系统的任何资源。不足包括：需要提供一定费用购置独立的跳变设备。

4　跳变栈模型的分析

4.1　安全性分析

主机的系统安全已经超出了本文的研究范围，暂不考虑主机系统的安全问题，在本文中认为主机系统是绝对安全的。

由于跳变技术中的通信双方都是可信的（可以采用第三方认证的方式），因此在同服务器进行网络通信之前，客户端可以通过可信的安全渠道获得同步策略中跳变算法所使用的同步密钥。然而，攻击者作为未知的不可信“用户”，不能通过安全渠道获取同步密钥，因此无法掌握跳变规律。

因而，攻击者只能够尝试侦听捕获篡改后的数据消息（如图 3、图 5 中的 p_1）。然而，从存在海量噪声（其他网络应用所产生的数据消息）的网络上准确捕获 p_1 是极其困难的。如果攻击者试图通过侦听 p_1 来攻击跳变技术，结论将如文献[10]中关于跳变技术抗窃听攻击能力的研究表明跳变技术有效地分散了网络流量，使得攻击者完整解析出数据报文的复杂度大大增加。

因此，跳变栈模型能够很好地保证跳变技术的安全性。

4.2　效率分析

消息篡改技术是指在数据消息离开主机系统之前仅对其源或目的端信息进行修改并重新计算校验和的过程。为了提高效率，消息篡改技术采用增量式校验和修改策略：在原校验和的基础之上，对数据消息中需要修改的字段进行增量式修改。由于仅仅需要修改源或目的端信息（最多包含 6 字节），消息篡改是非常高效的。

4.3　各层性能对比

在阐述跳变栈模型的过程中，客户端与服务器的跳变技术都在同一层空间中：都为用户层跳变、内核层跳变或网络层跳变。然而，在实际实现跳变技术时，客户端的跳变技术与服务器的跳变技术是相互独立的，例如，客户端可以为用户层跳变，而服务器可以为网络层跳变。具体采用哪一层空间的跳变，根据具体的情况而定。为此，将跳变栈模型中各层的性能指标进行了对比，如表 2 所示。

表 2 跳变栈模型中各层的指标对比

层次	安全	性能	透明度	兼容性	费用
用户层	高	中	中	高	低
内核层	高	高	高	低	低
网络层	高	高	高	高	高

跳变栈模型能够较好地适用于跳变技术，并满足跳变技术的各项基本特性。

① 安全性，将服务器的服务端信息隐藏起来，使用不断变化的跳变端信息作为通信端信息，使得攻击者无法确定攻击目标。

② 高效性，消息篡改所消耗的性能是相当微小的，只需要修改源端信息或目的端信息。

③ 透明性，原网络服务的程序不需要做任何的代码修改或配置更新。

④ 兼容性，跳变技术独立于网络服务，能够兼容于所有的基于 TCP/IPv4 的网络服务。

5 实验验证

5.1 实验环境

实验从两方面进行验证，一个方面是跳变技术的安全性能，如抵御网络攻击；另一个方面是跳变技术的服务性能，如传输延迟、延迟抖动、吞吐量、服务率等。实验设备环境如表 3 所示，所有主机系统都有统一的硬件配置。

表 3 实验硬件环境

系统	主机系统	内存/MB	处理器/GHz	带宽/(Mbit·s^{-1})
服务器	Windows XP	512	2.66	100
攻击者	Windows XP	512	2.66	100
客户端	Windows XP	512	2.66	100
路由器	Windows 2003	512	2.66	100

实验网络环境如图 6 所示。服务器、攻击者和客户端的 IP 地址分别处于不同的网段内，分别为 1.1.1.2-1.1.1.5/24、1.1.2.2/24 和 1.1.3.2/24 网段。路由器配置了 3 个网卡设备（1.1.1.1/24、1.1.2.1/24 和 1.1.3.1/24），分别用于连接服务器、攻击者和客户端。当服务器采用防火墙技术时，IP 地址为 1.1.12 和端口为 80；当服务器采用跳变技术时，IP 地址的变化范围从 1.1.1.2 到 1.1.1.5，端口变化范围从 1000 到 65535，平均跳变时隙（周期）为 2 s。

实验软件环境如表 4 所示，服务器安装了 IIS 网页服务器和基本防火墙，攻击者可以发起 CC 攻击和 SYN Flooding 攻击等攻击，客户端能够测试服务器的各项性能指标，服务器采用网络层跳变，客户端采用用户层跳变。

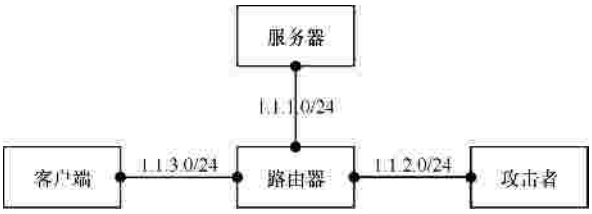


图 6 实验网络拓扑环境

表 4 实验软件环境

软件	服务器	攻击者	客户端
防护软件	防火墙、网络层跳变	无	用户层跳变
服务软件	IIS 5.1	无	无
其他软件	无	CC	ab.exe

5.2 安全性能实验

1) CC 攻击实验

Challenge Collapsar 攻击，简称 CC 攻击，是经典的拒绝服务攻击。CC 攻击连续不断地向服务器发起大量的服务请求来占据服务器的连接资源，从而拒绝其他合法用户的服务请求。在 CC 攻击中，由于用于攻击的数据消息都是合法的，其可以轻易地穿透防火墙或入侵检测等防护技术。

攻击者分别向安装了防火墙和跳变技术的服务器发起攻击，实验结果如表 5 所示。对于每个不同的攻击速率，分别实验了 6 到 8 组的攻击实验。其中平均耗时是指从开始实施攻击到服务器出现拒绝服务现象的平均时间，平均连接数是指出现拒绝服务现象时攻击者所保持的平均连接数。

表 5 CC 攻击实验结果

攻击速率/(次·s^{-1})	防火墙		跳变技术	
	平均耗时/s	平均连接数/个	平均耗时/s	平均连接数/个
5	390.6	912.8	-	0
10	239.6	1070	-	0
15	176.2	1248.6	-	0
20	53.8	466.8	-	0
25	25.6	188.2	-	0
30	48.2	206.4	-	0

从实验结果中可以得出如下结论。

① CC 攻击能够轻易穿透防火墙。在较短的时间内，CC 攻击可以使得服务器拒绝服务。如表 5

所示，随着攻击速率的增大，平均攻击耗时变短，平均连接数减少。当攻击速率超过 20 次/秒后，平均耗时和平均连接数趋于稳定，这是因为单位时间内服务器能够处理的 TCP 连接请求数是有限的，即使攻击速率超过了处理能力，有效的 TCP 连接数也为 TCP 连接请求上限数。

② 跳变技术拥有较好的防御能力。当采用跳变技术时，由于服务器的通信端信息不断变换，攻击者无法获得通信端信息，只能采取盲目方式的 CC 攻击，即攻击的目标是随机的端信息。在盲目方式的 CC 攻击中，如表 5 所示，平均耗时全为“-”，这意味着无论攻击者使用多长时间也无法使服务器到达拒绝服务的状态，因为即使攻击者以 30 次/秒的攻击频率，一个跳变时隙内攻击次数为 60 次，所有可能的端信息为 4×64 536=258 144，则能够命中通信端信息的概率为 60/258 144=0.000 2 趋向于零。

因此，相比于防火墙，跳变技术能够更好地保护网络服务器免受 CC 攻击。

2) SYN Flooding 攻击实验

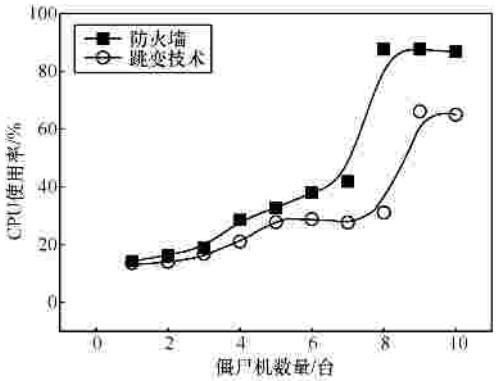
SYN Flooding 攻击也称为半连接攻击，基本思想是利用大量的 SYN 数据分组耗尽服务器的内存或处理器资源。由于 SYN Flooding 攻击的实现过程简单、攻击效果明显、隐蔽性强，被攻击者广泛用于网络攻击。据调查显示，在 2012 年一季度，SYN Flooding 攻击以 23%的比例上升为所有网络攻击类型的第一位[1]。

本文利用 SYN Flooding 攻击验证跳变技术的安全性能，对比了防火墙的服务器性能。在 SYN Flooding 攻击实验中，利用多台计算机作为僵尸机，每台僵尸机以平均 20 000 个/秒的攻击速率向服务器发送 SYN 数据分组，分别测试了防火墙和跳变技术的安全性能，实验结果如图 7 所示。

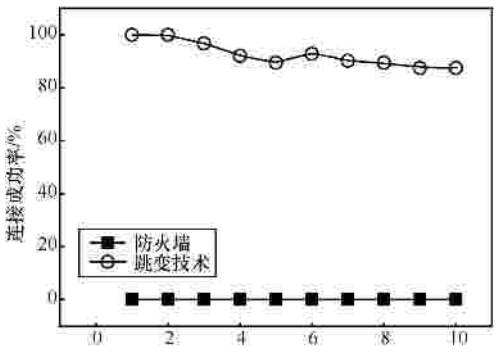
从实验结果可以得出如下结论。

① CPU 使用率。当采用防火墙时，通信端信息是固定的，攻击数据分组全部发向通信端信息，因此服务器需要消耗大量的资源用于处理数据分组的校验和应答。当采用跳变技术时，通信端信息是变化的，攻击数据分组发向随机的通信端信息，但是攻击者为了提高攻击效果可以缩小随机的范围为通信端信息的变化范围，于是攻击数据分组虽然不能够发向准确的通信端信息，但也能够全部到达服务器，服务器需要消耗大量的资源用于发送 RST 数据分组。因此，如图 7(a)所示，防火墙和跳变技术的 CPU 使用率都随着僵尸机数量的增大而增大；尤其是当采用防火墙时，僵尸机数量超过 5 台就使得 CPU 使用率高达 80%以上。

② 连接成功率。当采用防火墙时，单台僵尸的攻击就足以将服务器的内存资源耗尽，如图 7(b)所示，防火墙连接成功率全部为零；而跳变技术通过动态变化的通信端信息，使得连接成功率一直维持在 83%以上。



(a) 服务器的CPU使用率



(b) 服务器的连接成功率

图 7　SYN Flooding 攻击实验结果

因此，相比于防火墙，跳变技术能够更好地保护网络服务器免受 SYN Flooding 攻击。

5.3　服务性能实验

ApacheBench.exe 工具，简称 ab.exe，是 Apache 为测试其服务器性能的测试工具（在 Apache 安装目录下的 bin 目录中），能够在一定程度上测试服务器的服务性能。

在进行实验之前，首先利用 ab.exe 对没有采用任何防护措施的服务器进行了测试，以确定服务器的最大并发请求数，然后用这个最大请求并发数进行测试，因为本文认为最大请求并发数也是服务器的重要性能。测试结果表明：服务器的最大支持并发请求数为 9，当并发数超过 9 后正常的网络请求

也会出现一定程度的失败情况。

然后，分别对无防护技术、采用防火墙、用户层跳变、内核层跳变和网络层跳变的服务器进行了性能测试，测试请求次数为 1 000 次、并发数为 9（保证服务器的最大请求并发数），实验结果如表 6 所示。

表 6 服务性能实验结果

服务器	吞吐量	单次耗时/ms	传输速率/(kbit·s^{-1})	成功率/%
无防护	22.56	44.31	538.83	100
防火墙	22.47	44.50	536.63	100
用户层	20.76	48.35	495.82	99.89
内核层	20.73	48.23	495.04	99.97
网络层	21.63	46.35	516.58	99.99

从实验结果可以得出如下结论。

① 跳变技术能够维持较高的服务性能。如表 6 所示，平均吞吐量下降不到 5%，单次请求的平均耗时增加不超过 5%，平均传输速率下降不到 8%，并且请求成功率都维持在 99%以上。

② 网络层跳变的综合性能最好，内核层跳变次之，用户层跳变最差。首先，就网络带宽的消耗方面而言，利用 Wireshark 抓分组软件发现，相比内核层和网络层跳变，用户层跳变在 1 000 个网页请求过程中，客户端需要额外发送 13 101 个数据分组，其中占总通信数据分组数量的 28%。其次，就计算机的资源消耗而言，用户层跳变需要将内核空间中的数据消息复制到用户空间，篡改后再传回内核空间，消耗了一定的资源。

6 结束语

本文总结如下，给出了端信息跳变技术的基本特性。立足于此，指出了将跳变技术应用于现实网络服务的技术难点。然后提出了基于消息篡改的端信息跳变技术，并建立了跳变栈模型，包括：用户层跳变、内核层跳变和网络层跳变。分析了各层跳变方案的优势和不足。最后，利用实验验证了基于消息篡改的端信息跳变技术具有较高安全性能和服务性能。本文对基于端信息跳变的网络防护技术具有较高的应用价值。

对跳变技术的展望为，如何将跳变技术应用于现实的网络服务是十分值得研究的课题，本文仅对简单网络服务进行了研究分析，适用的领域相对较小。在今后的研究工作中，可以研究更为复杂的网络服务，如 P2P 网络、传感器网络等，在这些网络领域也存在严峻的安全问题，跳变技术在这些领域也有巨大的研究空间。

参考文献：

[1] Q1 2012 prolexic attack report[EB/OL]. http://www.prolexic.com/ddos-attack-reports.html.

[2] COOKE E, JAHANIAN F, MCPHERSON D. The zombie roundup: understanding, detecting, and disrupting botnets[EB/OL]. https://db.usenix.org/events/sruti05/tech/ full_papers/cooke/cooke.pdf, 2005.

[3] FREILING F C, HOLZ T, WICHERSKI G. Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks[EB/OL]. http://aib.informatik.rwth-aachen.de/, 2005.

[4] PRASAD K M, REDDY R M, KARTHIK M G. Flooding attacks to internet threat monitors (ITM): modeling and counter measures using botnet and honeypots[J]. International Journal of Comp Science & Information Technology (IJCSIT), 2011, 3(6): 159-172.

[5] MOSCOLA J, JOHN W. Lockwood, ronald prescott loui, michael pachos[A]. Implementation of a Content-Scanning Module for an Internet Firewall[C]. 2003.

[6] SALAMA S E, MARIE M I, EL-FANGARY L M, *et al*. Web anomaly misuse intrusion detection framework for SQL injection detection[J]. International Journal of Advanced Computer Science and Applications (IJACSA), 2012, 3(3):123-128.

[7] PING W, LEI W, RYAN C. Honeypot detection in advanced botnet attacks[J]. International Journal of Information and Computer Security, 2010, 4(1): 30-51.

[8] LEE K, CAVERLEE J, WEBB S. The social honeypot project protecting online communities from spammers[A]. Proceedin of The 19th International Conference on World Wide Web[C]. New York, USA, 2010. 1139-1140.

[9] **林楷**, **贾春福**, **石乐义**. **分布式时间戳同步技术的改进**[J]. **通信学报**, 2012, 33(10): 110-116.
LIN K, JIA C F, SHI L Y. Improvement of distributed ti tamp synchronization[J]. Journal on Communications, 2012, 33(10): 110-116.

[10] **石乐义**, **贾春福**, **吕述望**. **基于端信息跳变的主动网络防护研究**[J]. **通信学报**, 2008, 29(2): 106-110.
SHI L Y, JIA C F, LV S W. Research on end hopping for active network confrontation[J]. Journal on Communications, 2008, 29(2): 106-110.

[11] LIN K, JIA CF, WENG C. Distributed timestamp synchronization for end hopping[J]. China Communications, 2011, 8(4): 164-169.

作者简介：

林楷（1985-），男，江西上饶人，博士，主要研究方向为网络和信息安全。

贾春福（1967-），男，河北文安人，博士，南开大学教授、博士生导师，主要研究方向为网络与系统安全、密码学应用和恶意代码分析等。