矢量数据包处理加速的动态防护系统设计与实现

苗力仁,扈红超,霍树民,程国振
(1. 国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 针对 IP 地址动态化防护技术引入额外开销而导致正常网络传输性能下降的问题,首次设计并实现了一种基于矢量数据包处理(Vector Packet Processing,VPP)加速的 IP 地址动态防护系统,在隐藏真实 IP 地址的同时增强了系统数据处理能力. 首先,针对控制平面和数据平面处理逻辑不同,分别设计了快转发逻辑和慢转发逻辑,降低数据报文处理过程中的拷贝次数;其次,面向真实 IP-虚假 IP 频繁映射,提出一种共享内存的高效的 IP 地址动态变换机制;再次,采用最优化和哈希链算法制定了 IP 跳变策略与虚假 IP 地址预分配机制,最小化系统性能损耗;最后,实验结果表明,系统能够有效抵御 DoS 攻击并将潜在的侦查攻击命中率控制在 16% 以下,在数据处理性能上也有明显的速度提升.

关键词: 网络主动防御; IP 跳变; 矢量数据包处理; 最优化
中图分类号: TN915.08 **文献标识码:** A **文章编号:** 0372-2112 (2019)08-1724-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.08.016

The Design and Implementation of a Vector Packet Processing Accelerating Dynamic Protection System

MIAO Li-ren,HU Hong-chao,HUO Shu-min,CHENG Guo-zhen
(1. *National Digital Switching System Engineering R&D Center,Zhengzhou,Henan* 450002,*China*)

Abstract: IP address dynamic protection techniques will introduce additional overhead. Therefore,the performance of normal network transmission decreases. A dynamic protection system of IP address accelerated by Vector Packet Processing (VPP) is designed and implemented for the first time,which can hide the real IP address and enhance the system's data Processing ability. Firstly,fast forwarding logic and slow forwarding logic are designed respectively for different logic of control plane and data plane processing,so as to minimize the number of copies in data message processing. Secondly,facing the frequent mapping between real IP and virtual IP,an efficient dynamic IP address transformation mechanism of Shared memory is proposed. Thirdly,the optimization algorithm is used to formulate the IP hopping strategy,and the hashing chain algorithm is used to formulate the efficient virtual IP address pre-allocation mechanism. Minimize system performance losses. Finally,the experimental results show that the system can effectively resist DoS attacks and control the potential detection attack hit rate below 16%,which is significantly improved in data processing performance.

Key words: network active defense;IP mutation;vector packet processing;optimization

1 引言

近些年来,信息技术的高速发展使得计算机网络得以迅速普及. 由于网络本身具有很高的互联性和开放性,导致其很容易遭到恶意攻击,这给互联网资源带来了严重的威胁. 在网络攻击发起之前,攻击者往往会先对目标网络进行侦查,试图收集潜在受害者的信息. 网络配置的静态属性给了攻击者很大的优势,因为他们有相对大量的时间和方法来探索目标,直到收集到足够多的信息来找到目标系统的漏洞并发起攻击. 文献[1]中调查显示攻击者平均花费 45% 的时间用于对目标的侦查.

收稿日期:2019-03-20;修回日期:2019-06-13;责任编辑:马兰英
基金项目:信息工程大学新兴方向研究项目(No. 2016610708);国家自然科学基金(No. 61602509)

为了扳回防御方的天然劣势,研究者们在自适应网络防御领域做了很多工作,其中移动目标防御(Moving Target Defense, MTD)是一种被誉为“改变游戏规则”的技术[2],MTD 试图使用异构、动态、引入不确定性的方式来进行系统防御和增加网络攻击复杂度. IP 地址跳变是 MTD 技术的一种实现方式,通过改变 IP 地址的静态特性,主动更改主机 IP 地址从而使整个网络都变得不可探测.

本文提出了一种基于矢量数据包处理(VPP)加速的 IP 地址动态跳变主动防御技术. VPP 是 cisco 公司开发的一项基于 DPDK(Data Plane Development Kit)的高速软件数据平面技术,是一个可扩展的平台框架. 本文提出的方法就是建立在 VPP 可扩展的模块化设计之上,通过搭建 VPP 网关、添加功能节点向用户主机分配虚假 IP 的方式实现 IP 地址动态跳变.

本文主要贡献如下:

(1) 设计并实现了一种基于 VPP 加速的 IP 动态防护系统.

(2) 提出快、慢转发逻辑分离控制与数据平面;提出共享内存方式解决真实 IP 与虚假 IP 频繁映射问题;提出最优化 IP 跳变间隔算法和哈希链虚假 IP 预分配算法.

(3) 实验验证系统的安全性能和数据处理性能.

2　相关工作

网络 IP 地址跳变技术是移动目标防御的一种手段,其目标是通过不断变化用户主机的 IP 地址来迷惑和欺骗攻击者,使其无法定位攻击目标或发起有效的攻击[3]. 在传统网络安全防御领域,Dunlop[4] 等人提出了 RPAH(Random Port and Address Hopping)技术,通过搭建地址跳变网关、端口跳变引擎、端口和地址跳变网关实现 IP 地址的跳变,但是这种方法需要在网络中搭建多台新的硬件设备,增加了部署成本.

近年来,新型网络架构技术 SDN 备受关注. SDN shuffle[5] 利用 SDN 控制器命令服务器安装网络地址转换规则来动态转换 IP 地址. 这种方法需要修改所有服务器主机和 DNS 服务器来配合 SDN 控制器. OF-RHM[6](OpenFlow Random Host Mutation)使用 SDN 控制器动态地为每个主机分配一个随机的虚假 IP,该虚假 IP 从主机的真实 IP 转换而来. 这些基于 SDN 架构的动态 IP 技术提供了很好的抗识别防御性能,但在系统数据处理性能上存在瓶颈.

随着网络架构的演进,高性能网络编程技术不断突破,出现了很多优秀的高性能网络数据处理框架,比如 6wind、Windriver、Netmap 与 VPP 等,其中 VPP 作为 Linux 基金会开源项目 FD. io 的核心[7],正在被越来越多的软件开发者们接受. 本文设计的 IP 跳变系统就依托于 VPP 数据处理平台,通过在 VPP 数据处理流程中加入 IP 跳变功能节点实现动态防护. 在 IP 跳变节点中,我们结合实际网络状态,将 IP 跳变问题建模为最优停止问题以确定合理的跳变间隔,并使用哈希链算法为每台主机分配对应的虚假 IP 池,并在保证不破坏通信的基础完成了真实 IP 与虚假 IP 的动态映射. 同时,为了进一步提速,我们还在 VPP 框架内设计了快慢转发逻辑,并使用了共享内存机制. 实验结果表明,系统能够有效抵御 DoS 攻击并将潜在的侦查攻击命中率控制在 16% 以下,而且相比现有技术在数据处理性能方面有明显提升.

3　系统架构

本节将详细介绍基于 VPP 加速的 IP 动态防护系统,系统的整体架构如图 1 所示. 我们在网关服务器内搭建了一个平台,主要组成部分为管理服务模块和 VPP 数据处理模块,其中 VPP 工作在用户态.

VPP 的加速对象是数据平面通信过程中产生的大批流量,通过将大量数据报文打包成矢量的形式优化数据处理过程,相比之下,数量较少、内容较短的控制报文并没有加速的必要. 针对这一问题,本文分别设计了快转发逻辑和慢转发逻辑:数据平面流量经由快转发路径实现 VPP 加速,同时完成 IP 跳变. 控制平面的报文比如 dhcp、arp 等经由慢转发路径直接发往 IP 跳变节点,不经 VPP 加速,最大化降低数据报文处理过程中的拷贝次数. 本系统中 VPP 只针对内网流量进行加速,因此带有 vlan tag 的报文也一并经由慢转发路径处理.

VPP 网关平台实现了数据转发、链路控制和 IP 地址跳变等功能. 其中 IP 跳变功能由向 VPP 内添加动态跳变节点实现.

本文通过使用节点的方式向 VPP 内部添加了 IP 地址跳变功能. 具体设计结构如图 2 所示. 本文将通过以下三个方面对 IP 跳变节点进行详细的描述.

3.1　主机、数据包标识

虚假 IP/域名修改单元为每一个接入内网的主机分配了一个与其真实 IP(rIP)对应的虚假 IP(vIP),并为每一个 vIP 生成一个对应的虚假域名(vDomain). 所有修改之后的信息会同步存储在终端信息单元内. 系统采用五元组 < 源 IP、目的 IP、源端口、目的端口、通信协议 > 唯一标识一个数据包,并将接收到的 DNS 数据包发往 DNS 解析模块,DNS 解析模块通过查询终端信息表完成数据包的解析. 除 DHCP 和 DNS 数据包外的其他数据包都会被发往虚假 IP/域名修改单元进行 IP 地址和域名的修改.

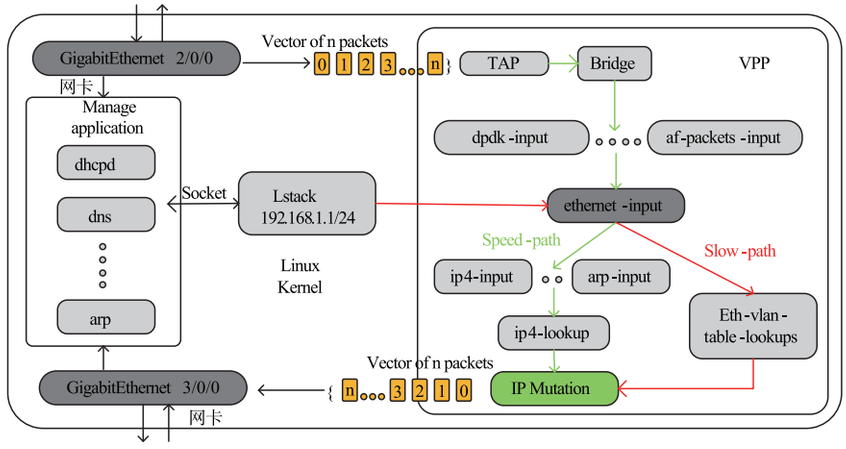


图1 基于VPP加速的动态IP架构

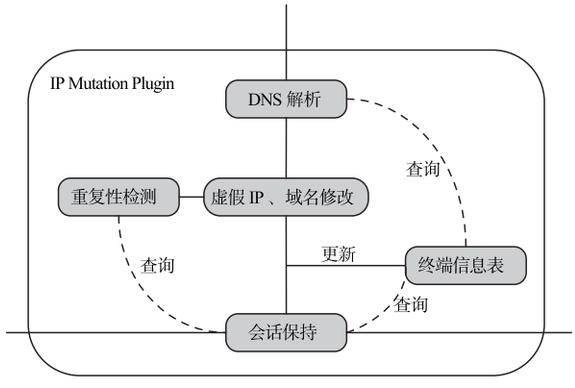


图2 IP跳变节点结构示意

3.2 通信流程

对于内网中通信的两台主机,会话保持单元会首先建立本次通信的会话信息,内容包括 < 源主机信息、目的主机信息、通信协议信息 >. 基于本次会话信息,系统将通信数据包中的源 IP 替换为源主机对应的 vIP,将目的 IP 替换为目的主机对应的 rIP. 基于 VPP 的 IP 地址动态跳变系统的运行流程描述如图 3 所示:

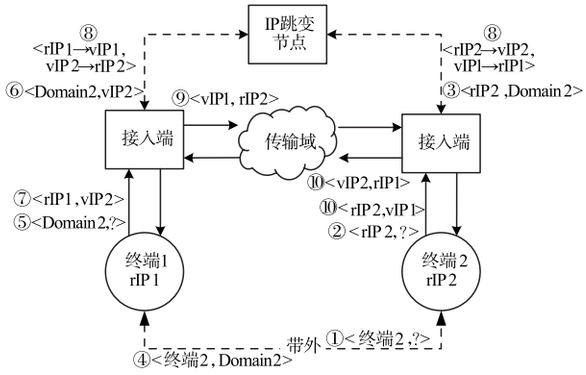


图3 系统运行流程示意

首先,系统为各个终端分配虚假域名. 通信时,终端 1 预先通过带外的方式获取终端 2 的域名,继而通过域名向控制端查询终端 2 的 IP(即 vIP2),之后使用自己真实地址 rIP1 与终端 2 通信,过程中系统将其真实 IP 修改为 vIP1 发往终端 2. 终端 2 与终端 1 的通信与上述步骤相同. 这样通信双方都只能知道对方的 vIP 并且不会察觉,能够达到隐藏主机 IP 地址的目的.

3.3 通信保护机制

为了防止由 IP 改变引起的连接中断,我们设计了会话保持单元,数据包进入网关之后系统会首先查询已建立的会话信息,如果数据包属于当前存在的会话,系统将根据会话信息中的内容进行 IP 地址的替换. 在通信结束后会话信息将从会话列表中删除,并在下次通信开始时从终端信息表内查询新的 vIP 建立会话. 重复性检测单元会遍历生成的 vIP,防止出现一个 vIP 对应多台真实主机的情况发生.

针对真实 IP 与虚假 IP 频繁映射的性能损耗问题,本系统利用 DPDK 在 IP 跳变节点内分别创建了真实与虚假 IP 地址的共享内存. 当第一个相关进程启动时,DPDK 记录内存映射文件的详细内存配置,包括其使用的大页以及映射的虚假内存地址等. 当另一个相关进程启动时,DPDK 读取记录的内存信息并为其重新创建相同的内存配置. 如图 4 所示,本文通过这种方式使所有进程共享内存区域和指针. 当节点内一个单元对 IP 地址进行修改时,其他单元都会察觉到这个更改,无须在各个单元间复制,减少了数据拷贝次数,实现了高效的 IP 地址动态变换.

4 IP 跳变动态策略

本文提出的 IP 地址动态防护系统通过以一定的频率生成并分配虚假 IP 地址的方式实现动态属性. 为了得到最优的动态策略,本文使用了以下两种办法:

(1) 将 IP 地址跳变问题建模为最优停止问题[8],

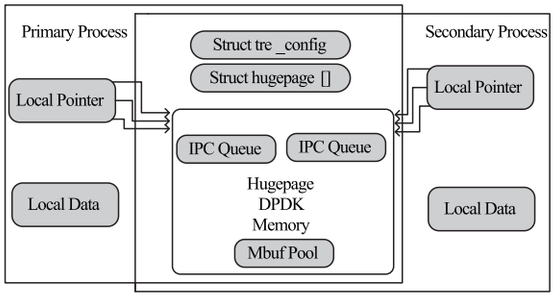


图4　DPDK共享内存示意

推导出最优的 IP 跳变间隔.

(2) 利用单向加密的哈希链算法[9]生成虚假 IP 地址.

4.1　最优化算法

虚假 IP 的有效性取决于在其存在期间内攻击者攻破主机的概率. 如果虚假 IP 存在时间过长,攻击者可能已经得到了想要的信息(网络拓扑、主机信息、系统漏洞等),但是太过频繁的跳变又会给系统带来极大的性能负担. 理想的情况是在攻击者攻击成功的概率达到最大之前进行 IP 跳变,同时还需要考虑跳变带来的性能损失. 因此本文将此问题定义为一个最优化求解问题. 为了方便分析,求解基于以下假设:

(1) 攻击者对网络的探测攻击是一个强度为 μ 的泊松过程.

(2) 目标主机对探测攻击的响应是指数级的,平均响应时间为 $1/\lambda$.

(3) 主机通信连接数符合排队论中的生灭过程,通信产生速率为 α,结束速率为 β.

我们将攻击者在时间 t 内攻破目标网络内主机的概率记为 $G(t)$,将攻击者攻破目标主机的期望时间记为 Et,将主机内正在通信的连接数记为 $M(t)$. 因此,选择下一个跳变时间可以表示为最优停止问题:

$$\min_T\{E(G(T)+\varepsilon M(T))\} \tag{1}$$

其中 ε 是用来衡量性能损失的参数.

简单起见,设初始时间为 0,跳变时间为 T,在时间 T 内,攻击者对网络的探测次数概率分布为:

$$P(N(T)=k)=e^{-\mu(T-\frac{1}{\lambda})}\cdot\frac{(\mu(T-\frac{1}{\lambda}))^k}{k!} \tag{2}$$

由此得到 $G(t)$ 的期望为:

$$\begin{aligned}E(G(T)) &= \omega\cdot\sum_{k=1}^{\infty}(1-\omega)^{k-1}\cdot k\cdot P(N(T)=k)\\ &=\omega\cdot\sum_{k=1}^{\infty}(1-\omega)^{k-1}\cdot k\cdot e^{-\mu(T-\frac{1}{\lambda})}\\ &\quad\cdot\frac{(\mu(T-\frac{1}{\lambda}))^k}{k!}\end{aligned} \tag{3}$$

其中 ω 表示单次探测攻击成功的概率. 上式满足以下约束条件:

$$0<\omega<1,\mu>0,T>\frac{1}{\lambda} \tag{4}$$

时间 T 内攻击者每次探测攻击之后需要等待目标主机的响应,因此攻击者攻击成功的期望时间为:

$$Et=\frac{1}{\omega}(\frac{1}{\mu}+\frac{1}{\lambda})=\frac{\omega(\mu+\lambda)}{\mu\lambda} \tag{5}$$

设主机内通信连接的产生和结束是相互独立的,并且在时刻 0 已经达到稳态,根据生灭过程的数学模型,稳态时下式成立.

$$p_j=\sum_{i=0}^{\infty}p_i\cdot P_{ij},\quad j=0,1,2\cdots \tag{6}$$

i,j 表示生灭系统状态(通信连接数),Pi 和 Pj 分别表示系统处于状态 i,j 的概率. 根据通信产生速率 α 和结束速率 β,有:

$$p_n\cdot\alpha=p_{n+1}\cdot\beta \tag{7}$$

从而有:

$$p_{n+1}=p_n\cdot\frac{\alpha}{\beta}=p_n\cdot\rho \tag{8}$$

其中 $\rho=\frac{\alpha}{\beta}$. 由上式进行递推可得:

$$p_{n+1}=\rho^{n+1}p_0 \tag{9}$$

显然只有在 $\rho=\frac{\alpha}{\beta}<1$ 的条件下,下式才可能成立.

$$\sum_{n=0}^{\infty}\rho^n p_0=\frac{p_0}{1-\rho}=1 \tag{10}$$

从上式可得 $p_0=1-\rho$,进而由式(9)得到:

$$p_n=\rho^n(1-\rho) \tag{11}$$

综上可知,在稳态时,系统内通信连接数 $M(t)$ 的稳态分布为 $\{p_n=\rho^n(1-\rho),n\geq 0\}$. 由此得到 $M(T)$ 的期望如下:

$$E(M(t))=\sum_{n=1}^{\infty}n\cdot p_n=\sum_{n=1}^{\infty}n\cdot\rho^n(1-\rho) \tag{12}$$

因此本文的最优化 IP 地址跳变间隔策略是选择满足下列条件的时间间隔 T 进行 IP 跳变:

$$\min_T\{E(G(t)+\varepsilon M(t))\} \tag{13}$$

$$T\leq Et=\frac{\omega(\mu+\lambda)}{\mu\lambda} \tag{14}$$

$$T\leq\frac{N}{\mu}\ (N\text{ 为内网主机数}) \tag{15}$$

其中条件 3 是为了保证系统在攻击者探测全部内网主机之前进行 IP 跳变,避免泄露系统拓扑等信息. 确定了 IP 地址的跳变间隔之后,接下来本文将对虚假 IP 的选择算法进行讨论.

4.2　虚假 IP 选择算法

1981 年,Lamport 提出哈希链算法用来作为一种防

窃听的密码保护方案,其基本原理是将哈希函数循环地用于一个字符串,凭借其不可逆的属性实现单向加密. 在本文中我们使用哈希链来生成 IP 地址跳变所需要的虚假 IP 地址,具体步骤在算法 1 中描述:

算法 1 vIP 选择算法

输入:The shared key s,the hash function $F(x)$,
the number of hosts N
输出:The Virtual IP pool $vIP[N]$
1:initialize: Set $0 \leqslant i \leqslant N-1$, $0 \leqslant k \leqslant n$
2:for $i = 1,2\cdots\cdots,N-1$ do
3: Choose random initial secret value ri for each host
4: for $k = 1,2\cdots\cdots n$ do
5: To generate the hash chain $F^k(ri)$
6: Set vIP$[i] = F^k(ri)$
7: While jump time is up do
8: Set vIP$[i] = F^{k-1}(ri)$
9: end while
10: end for
11:end for

首先,对内网内所有主机使用整数 i 进行编号,然后为所有主机生成一个共享的密钥 s 以及各自的随机初始密值 ri. 每一个主机 i 都会通过下式构造一个属于自己的哈希链,设其长度为 $n+1$.

$$(\forall k \in [1,n])rIP_i(k) = F^k(r_i) = F(F^{k-1}(r_i),s) \tag{16}$$

F 为单向哈希函数,它通过递归地调用密值 ri 生成哈希链,同时在递归的每一步中引入密钥 s,作用是防止攻击者计算出函数 F. 这样我们就通过哈希链为每一个主机生成了一个虚假 IP 地址池,虚假 IP 地址的选择与哈希链的生成顺序相反,这样凭借哈希链的单向属性,即使攻击者知道了当前的虚假 IP 地址、密值 ri 和密钥 s 也很难计算出下一跳的虚假 IP 地址.

5 实验结果与分析

本节将对实验系统部署进行简单介绍,并对系统的防御效果以及数据处理性能进行评估.

5.1 系统部署

图 5 所示为本文 IP 动态防护系统的实际部署,系统内主要设备包括一台服务器,三台交换机以及十台主机. 为了使主机间的通信全部经由 VPP 网关进行 IP 跳变,选择端口隔离的方式对不同主机进行隔离. 表 1 为实验部署描述,由 Kali 主机充当攻击者的角色.

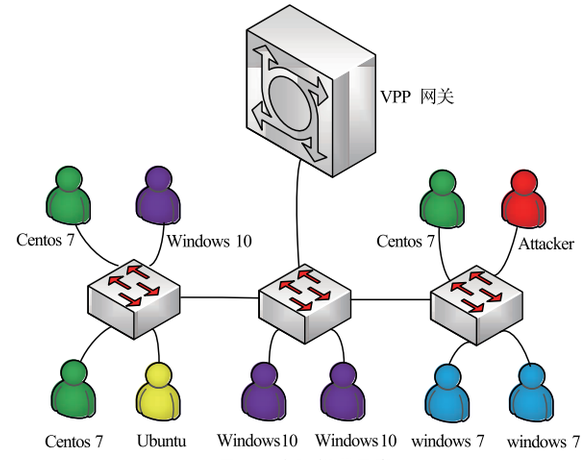


图5 实验部署示意

表 1 实验部署描述

设备	数量	描述
华为 H22H-03 服务器	1	VPP 架构网关
Pica8(P3297)交换机	3	用于二层交换
普通计算机	2	Window7 系统
	3	Window10 系统
	3	Centos7 系统
	1	Ubuntu16.04 系统
	1	Kali 系统

5.2 防御性能分析

5.2.1 抗 DoS 攻击能力分析

DoS 攻击通过目标向发送大量攻击请求来耗尽其服务资源,进而达到使目标系统崩溃的目的[10]. IP 地址跳变技术可以有效的防御针对网络主机的 DoS 攻击. 对于本系统,首先从理论上进行分析. 假设内网中主机数目为 N,可用虚假 IP 数目,即哈希链长度为 L,则网络 IP 地址空间大小为 NL. 相比于静态网络,攻击者成功命中目标主机的难度 H 将大幅增加:

$$H_{跳} = H_{静}\left(1 + \sum_{k=1}^{NL-1} k \frac{C_{NL-1}^{k}}{C_{NL}^{k} C_{NL-1}^{1}}\right) \tag{17}$$

即使攻击者能够成功命中目标,也只能在有限的 IP 跳变间隔内进行攻击,无法达到耗尽目标系统资源的目的.

为测试系统防御能力,假设网络中 ubuntu 系统主机的防御能力较差,使用 Hping3 工具构建 DoS 攻击,策略为随机攻击目标网络中的主机,若目标不是 ubuntu 系统则在短暂延迟后攻击下一台主机. 因为 DoS 攻击的目的是耗尽受害者的系统资源,所以我们以目标系统主机的实时 CPU 占用率作为测试指标,并对每种 DoS 攻击进行了三次重复测试,结果如图 6 所示. 可以看到,一旦 DoS 攻击命中目标,目标主机的 CPU 占用率会迅速上涨,其中 SYN 洪水攻击涨幅最大,泪滴攻击次之. 攻击开始一段时间后,由于目标主机虚假 IP 改变,DoS 攻击失去目标,目标主机 CPU 占用率在短时间内恢复正常. DoS 攻击可能会在虚假 IP 跳变之前耗尽主机资源,但只能使其暂

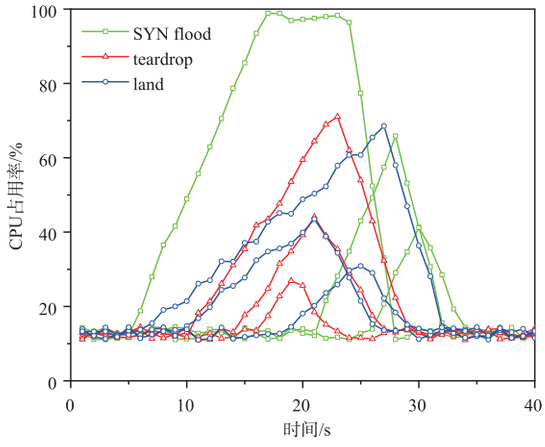


图6　目标主机CPU占用率

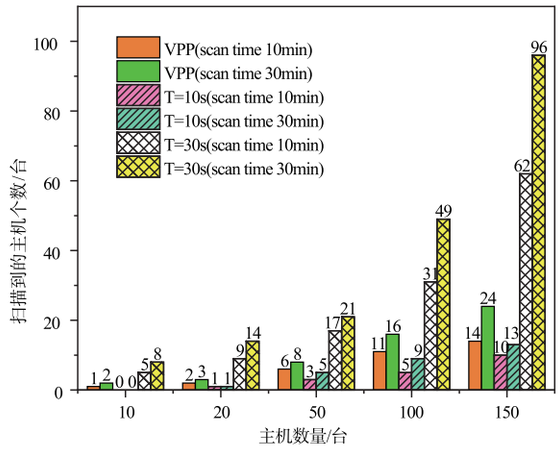


图7　扫描结果

时失去处理能力,不会对系统产生破坏性的损坏,由此可见本系统可以有效的防御 DoS 攻击.

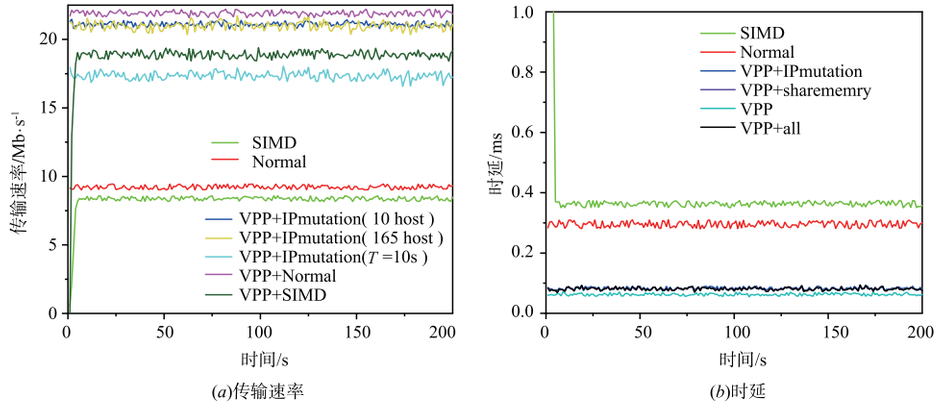
5.2.2　抗识别能力分析

主机地址识别通常是网络攻击链的第一步,攻击者只有在找到目标主机之后才能进行下一步侦查. 在传统静态网络中,只要时间足够,攻击者总是可以扫描出网络中所有存活的主机,引入 IP 跳变之后,只要在跳变间隔 T 时间内攻击者无法扫描出全部网络主机,已经扫描到的 IP 就会失效,迫使其重新开始扫描. 为了验证系统的抗识别能力,分别在不同主机数量时使用 zenmap 扫描工具模拟攻击者对内网进行扫描,在 30 分钟内随机扫描各个网段. 攻击者随着扫描时间嗅探到的主机个数如图 7 所示. 可以看到,IP 跳变的周期越短,攻击者扫描到主机的概率越小. 扫描时间越久,网络主机规模越大,被扫描到的主机数量越多,但攻击者无法扫描出网络中全部的主机,对本系统的扫描成功率只有 16% 左右. 而且这些 zenmap 扫描到的主机中包括使用不同虚假 IP 的同一真实主机,因此被扫描到的主机数量实际上会更少一些.

5.3　数据处理性能分析

VPP 可以通过矢量处理数据包的方式最大程度提高系统性能. 不仅如此,VPP 还集成了 DPDK 技术,采用轮询模式驱动(Poll Mode Driver)代替了传统的 NAPI 方式,完全不使用中断机制,避免不必要的开销. 针对本文提出的基于 VPP 的 IP 跳变系统,我们对其进行了数据处理性能的测试,对内网中两台主机之间的文件传输速率和时延进行了跟踪,并与同样基于华为 H22H-03 服务器的传统网络网关(Normal)以及 SDN 动态 IP 跳变系统(SIMD)[11]进行了对比,结果如图 8 所示.

可以看到,传统静态网络的文件传输速率与时延都相对稳定. SIMD 系统需要将数据包与流表进行匹配,然后再进行转发或修改操作,因此传输速率相对较慢,时延较大. SDN 架构中每次会话建立都需要下发一对流表,因此第一个数据报文会上传到控制器进行处理,会带来系统的最大时延. VPP 为网络提供了极大的速度加成,静态网络部署 VPP 后,速度提升近 240%. 基于 SDN 架构的 SIMD 系统在 VPP 的加持下速度也提升了近225%. 本系统中,由于VPP需要修改报文头部IP



(a)传输速率　　(b)时延

图8　系统性能分析

地址,处理速度略有下降,但是相比于同等条件下的 SIMD 系统也有近 12% 的性能提升. 当将 IP 跳变间隔设置为 10s 时,由于跳变过于频繁,系统的处理性能下降将近 17.5%,这是无法忍受的性能损失. 所以,结合前文实验结果,基于最优化算法得出的最佳跳变间隔综合考虑了防御性能与开销,具有很好的实际应用效果.

本系统不论是数据的传输速率和时延都明显优于传统方式和 SIMD,相比于传统网络其数据处理性能提升将近两倍. 对于动态防御系统来说,极快的数据处理速度会压缩攻击者的攻击准备时间,增加其攻击难度.

6 结束语

本文充分利用 VPP 高速数据处理平面的灵活性和可扩展性,结合 IP 跳变技术设计并实现了一种基于 VPP 加速的 IP 地址动态防护系统. 首先,通过划分快、慢转发逻辑和共享内存的方式减少内核内存读写操作,接着通过最优化算法计算生成系统最优 IP 跳变间隔、通过哈希链算法计算生成预分配虚假 IP 池. 最后在 VPP 中插入动态防护功能节点并对系统性能进行测试. 实验结果表明,本文提出的方法可以有效阻断 Dos 攻击和网络侦查,相比传统静态网络和其他 IP 跳变技术在传输速率和时延方面都有明显的性能提升. 目前,传统网络的数据处理能力已经接近瓶颈,很难满足各种新型网络技术的需求,数据平面加速套件的引入已是必然,本文在这一领域进行了探索,具有很好的参考意义. 未来的工作将进一步丰富系统的功能,加入端口跳变、路径跳变以及操作系统指纹欺骗等功能节点,完成基于 VPP 的多维动态防护系统方案.

参考文献

[1] Kewley D, Fink R, Lowry J, et al. Dynamic approaches to thwart adversary intelligence gathering[A]. Darpa Information Survivability Conference & Exposition II[C]. Anaheim, CA, USA. 2001. 176 – 185.

[2] POOVENDRAN R. Dynamic defense against adaptive and persistent adversaries[A]. Proceedings of the 5th ACM Workshop on Moving Target Defense[C]. Toronto, Canada. 2018. 57 – 58.

[3] Antonatos S, Akritidis P, Markatos E P, et al. Defending against hitlist worms using network address space randomization[J]. Computer Networks, 2007, 51(12): 3471 – 3490.

[4] Luo Y B, Wang B S, Wang X F, et al. RPAH: Random port and address hopping for thwarting internal and external adversaries[A]. 2015 IEEE Trustcom/BigDataSE/ISPA[C]. Helsinki, Finland. 2015. 1: 263 – 270.

[5] Macfarland D C, Shue C A. The SDN Shuffle: Creating a moving-target defense using host-based software-defined networking[A]. Acm Workshop on Moving Target Defense[C]. Denver, Colorado, USA. 2015. 37 – 41.

[6] Jafarian J H, Al-Shaer E, Duan Q. Openflow random host mutation: transparent moving target defense using software defined networking[A]. Workshop on Hot Topics in Software Defined Networks[C]. Chicago, Illinois, USA. 2012. 127 – 132.

[7] Inocybe Technologies: Innocybe_VPP_Whitepaper[OL]. http://www. sdxcentral. com/wp-content/uploads/Innocybe_VPP_Whitepaper. pdf. 2017. 12.

[8] Clark A, Sun K, Poovendran R. Effectiveness of IP address randomization in decoy-based moving target defense[A]. 2013 IEEE 52nd Annual Conference on Decision and Control (CDC)[C]. Firenze, Italy. 2013. 187 – 195.

[9] Wang K, Chen X, Zhu Y. Random domain name and address mutation (RDAM) for thwarting reconnaissance attacks[J]. PLoS ONE, 2017, 12(5): e01777111.

[10] 胡毅勋,等. 基于 OpenFlow 的网络层移动目标防御方案[J]. 通信学报,2017,38(10):102 – 112.
HU Yi-xun, ZHENG Kang-feng, YANG Yi-xian, NIU Xin-xin. Moving target defense solution onnetwork layer based on OpenFlow[J]. Journal on Communications, 2017, 38(10): 102 – 112. (in Chinese)

[11] 陈扬,扈红超,程国振. 软件定义的内网动态防御系统设计与实现[J]. 电子学报,2018,46(11):2604 – 2611.
CHEN Yang, HU Hong-chao, CHENG Guo-zhe. The Design and Implementation of a Software-Defined ntranet Dynamic Defense System[J]. Acta Electronica Sinica, 2018, 46(11): 2604 – 2611. (in Chinese)

作者简介

苗力仁(通信作者) 男,1995 年出生,辽宁丹东人,国家数字交换系统工程技术研究中心硕士研究生,主要研究方向为网络安全.
E-mail: 710266505@qq.com

扈红超 男,1982 年出生,河南商丘人,国家数字交换系统工程技术研究中心研究员,博士生导师,主要研究方向为云计算和网络安全.
E-mail: 13633833568@139.com