软件定义的内网动态防御系统设计与实现

陈 扬,扈红超,程国振
(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 当前,自带设备(BYOD)的兴起对传统基于边界的内网防护观念提出了新的挑战——内部不设防导致堡垒易从内部攻破. 从扰乱攻击链的角度,本文提出了"隔离 + 动态"的防护方法,设计并实现了一种基于软件定义的内网动态防御系统. 通过为内网终端分配虚拟 IP 地址空间,以隐藏各自的真实信息;并且将 IP 跳变和路径跳变结合起来,实现了更全方面的防护. 结果表明,在正常网络应用不受影响的情况下,该系统能大幅降低网络侦察扫描的可用性,阻断网络窃听,提高攻击者实时攻击难度.

关键词: 内网防御; 软件定义; IP 跳变; 路径跳变
中图分类号: TP302.1 **文献标识码:** A **文章编号:** 0372-2112 (2018)11-2604-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.11.006

The Design and Implementation of a Software-Defined Intranet Dynamic Defense System

CHEN Yang, HU Hong-chao, CHENG Guo-zhen
(*National Digital Switching System Engineering R&D Center, Zhengzhou, Henan* 450002, *China*)

Abstract: The rise of Bring Your Own Device (BYOD) now poses new challenges (the internal undefended causes the citadel to break through from within) to the concept of traditional boundary-based intranet protection. Based on the idea of isolation and dynamic, this paper designs and implements a Software-defined Intranet Dynamic Defense system (SIDD) to harass cyber kill chain. We allocate virtual IP address space for intranet terminals to hide the real IP address, meanwhile, combine the maneuvering of IP and path to achieve more comprehensive protection. Our experiments indicate that this method can significantly reduce the availability of network reconnaissance, block the network eavesdropping, and increase the difficulties of attacker's real-time attack without affecting network applications.

Key words: intranet defense; software-defined network; IP maneuvering; path maneuvering

1 引言

企业、机构、政府等部署的内部网络包含有大量高价值的信息资产,是攻击者渗透和控制的主要对象之一. 当前对内网的防御主要有两类:(1)基于边界的传统防御方案;(2)近年来兴起的零信任机制[1].

传统内网防御方案主要通过在边界部署大量防火墙、IDS 等设备,在内外网之间构建 DMZ 区域,一定程度上增加了内网的安全性. 但是,这种防护方式面临着新的挑战. 一方面,攻击手段复杂化;另一方面,内网边界趋于模糊,尤其是近年来,随着移动互联网的发展,自带设备(Bring Your Own Device, BYOD)[2,3] 成为企业、政府等工作场所的新模式. 该模式虽然降低了企业成本,但使企业内网面临极大的安全隐患.

为应对上述挑战,本文以扰乱攻击者攻击链[4],提高攻击难度为目标,基于零信任的理念,通过"隔离 + 动态"的技术思路,设计并实现了一种基于软件定义的内网动态防御系统(Software-defined Intranet Dynamic Defense system, SIDD). 使系统能够保持原有网络配置的完整性,并最小化操作管理. 在内网中隐藏终端的真实 IP 地址(rIP),为终端分配虚拟 IP 地址(vIP),以实现用户之间的隔离. 在正常网络应用不受影响的情况下,实现网络拓扑和 IP 地址协同跳变.

收稿日期:2018-05-23;修回日期:2018-07-30 ;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);国家自然科学基金创新群体项目(No. 61521003);国家重点研发计划项目(No. 2016YFB0800100, No. 2016YFB0800101)

本文主要贡献如下:

(1)设计提出一种基于软件定义的内网动态防御方案.(2)基于 Opendaylight 实现了网络动态 IP、路径跳变系统.(3)搭建了实验平台,验证系统的防扫描,防窃听以及防 DoS 攻击的能力.

2　相关工作

在 BYOD 安全方面,谷歌的 BeyondCorp[5] 和 Duo Security 的 Duo Beyond[6] 都是基于零信任网络模型,通过注册认证的方式管理接入设备. 但零信任网络部署过于复杂和耗费精力财力.

在地址跳变技术研究方面,Ehab Al-Shaer 等人提出了一种具有高速变化和难以预测特性的 IP 地址变化技术(Random Host Mutation,RHM)[7]. 之后结合 OpenFlow 技术进行改进,得到 OF-RHM (OpenFlow Random Host Mutation)[8] 模型.

在路径跳变技术方面,Talipov 等人提出基于 R-ADOV(reverse AODV)[9] 的路径跳变方法. Jafarian 等提出了一种随机路径跳变方法(RRM,Random Route Mutation)[10,11],通过可满足性模理论来计算可选取的转发路径. 随后论文[12] 在 RRM 的基础上提出基于安全容量矩阵的最优跳变路径生成方法.

综上,现有研究成果在通过改变网络静态属性提高攻击者渗透成本方面进行了卓有成效的理论研究,但较少讨论工程实践的可行方案. 本文从当前企业网真实内网部署需求出发,实现了 IP、路由协同跳变,解决了网络动态化工程实践中的关键问题.

3　系统架构设计

该架构主要包括数据层,管理层,控制层,如图 1.

数据层:负责数据转发,减少 IP 动态变换带来的传输速率损失.

管理层:对系统参数进行配置,获取系统运行状态.

控制层:主要实现:(1)终端资源配置,虚拟配置动态变化.(2)为通信建立会话,动态修改网络属性.

本文为一个内网防御系统,所有接入该系统的终端只能使用 dhcp 来获取真实 IP,在获取真实 IP 的同时,控制器为每个终端分配一个动态 vIP 地址和一个内网域名. 内网之间的相互访问只能通过域名解析(DNS)获取当前时刻目的终端的 vIP.

4　SIDD 系统实现

软件定义网络(Software Defined Network,SDN)为 SDN 架构可对网络设备集中管理,精确定义底层设备对数据分组的转发,实现了对底层设备网络行为变化的灵活控制. 为系统的实现提供了可行方案.

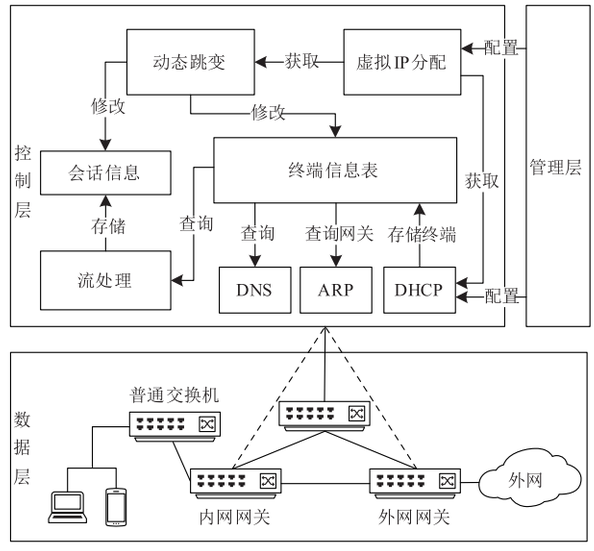


图1　系统框架

4.1　系统核心模块

本文基于 Opendaylight 实现了系统的四个核心模块:DNS 模块,DHCP 模块,ARP 模块和流处理模块,其系统状态图如图 2,途中任何状态失败或结束都会回到监听状态.

DNS 模块:主要负责(正)反向域名查询. 规定终端通过域名只能查询目标终端的 vIP. 通过反向域名查询只能得到终端自己的域名.

DHCP 模块:主要负责为终端分配 rIP 和 vIP. rIP 以 DHCP 报文的形式返回终端,vIP 存储于控制器 DataStore 中,并进行动态跳变.

ARP 模块:主要有两个作用:(1)禁止内网中的 ARP 报文广播(ARP 广播报文是内网中网络嗅探的最佳帮手);(2)回应终端的网关 MAC 请求.

流处理模块:为终端双方通信建立动态路径. 只有符合规则的报文才能到达通信对端. 能够屏蔽掉其他扫描、攻击报文.

4.2　通信流程

本文的通信采用源地址虚拟和目标地址真实的方式,不同于 Al-Shaer 等人提出的源、目的 IP 皆隐藏的策略[7,8]. 这样有如下好处:

(1)SIDD 的机制要求用户只能使用 vIP 对目标终端进行访问. 所以只隐藏源 rIP 与文献[7,8]得到反嗅探效果一样.

(2)报文只需被接入端修改一次. 这样可以提高报文处理效率.

如图 3 所示,具体通信流程如下:

步骤 1:终端 1 通过带外的方式获得终端 2 的内网域名 n2.

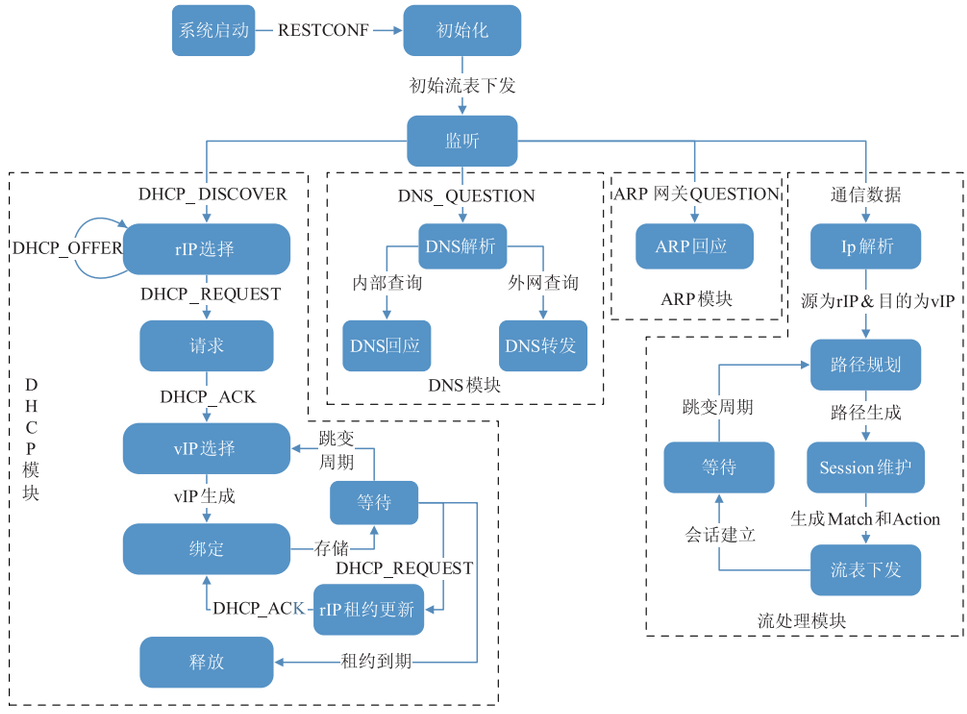


图2 系统状态图

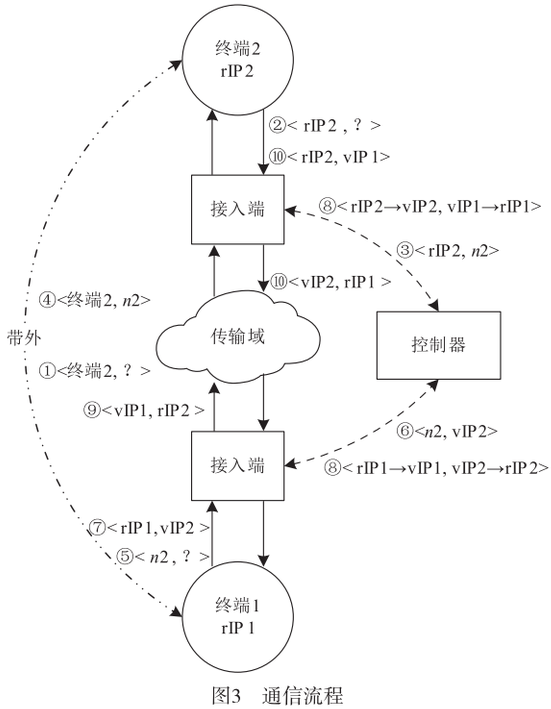


图3 通信流程

步骤2:终端1通过控制端的DNS服务器获得终端2的vIP2.

步骤3:终端1使用自己的真实地址和终端2的当前虚拟地址vIP2向终端2发送报文〈rIP1,vIP2〉.

步骤4:该报文到达终端1端的接入端后,报文被修改为〈vIP1,rIP2〉.然后送达终端2.

步骤5:终端2的报文返回同3、4步骤.

4.3 业务分组

对于IP报文上报到交换机,控制器主要需要判断报文属于:

(*a*)发出端到接入端,下一跳为接收端.

(*b*)发出端到接入端,下一跳不是接收端.

(*c*)非接入端报文,下一跳为接收端.

(*d*)非接入端报文,下一跳不是接收端.

ODL控制器只处理接入端上传的报文.如果该报文源地址为rIP,其目的地址为vIP,则生成路径.对于中的每个交换机依次下发处理转发流表 以及返回处理流表.对应于以上四种类型报文,有四种处理流表(根据 中的不同位置进行区分).具体处理流程如算法1所示.

例如,对于源终端的接入端交换机,如果属于类型a的报文,下发转发处理流表f_{in},添加action修改源rIP为vIP,目的vIP为rIP,添加action将mac修改为目的终端mac.以及添加出端口s.out的action.流表f_{in}对应于算法一中:

$$f_{\text{in}}\left(\begin{matrix} h_i.\ \text{rIP}\rightarrow h_i.\ \text{vIP}, h_j.\ \text{vIP}\rightarrow h_j.\ \text{rIP}, \\ p.\ \text{dst\_mac}\rightarrow h_j.\ \text{mac}, \text{output}:\text{s.\ out}\end{matrix}\right)$$

其中,→意为修改,前面内容为匹配,output为报文出口.

算法 1　通信处理

```
1:for all 报文 p 来自 OF 交换机 do
2:  if p. src 是 rIP and p 来自接入端 then
3:    if p. dst 是 vIP then
4:      R1← hi到 hj的路径
5:      for all 交换机 s in R1 do
6:        if s∈Type. a then
7:          f_in(hi. rIP→hi. vIP, hj. vIP→hj. rIP, p. dst_mac→hj. mac, output; s. out)
8:          f_back(hj. rIP→hj. vIP, hi. vIP→hi. rIP, p. dst_mac→hi. mac, output; s. in)
9:        else if s∈Type. b then
10:         f_in(hi. rIP→hi. vIP, hj. vIP→hj. rIP, p. dst_mac→hj. mac, output; s. out)
11:         f_back(hj. vIP, hi. rIP, output; s. in)
12:       else if s∈Type. c then
13:         f_in(hi. vIP, hj. rIP, output; s. out)
14:         f_back(hj. vIP, hi. rIP, output; s. in)
15:       else if s∈Type. d then
16:         f_in(hi. vIP, hj. rIP, output; s. out)
17:         f_back(hj. rIP→hj. vIP, hi. vIP→hi. rIP, p. dst_mac→hi. mac, output; s. in)
18:       end if
19:       flow_mod(f_in, f_back)
20:     end for
21:   end if
22: end if
23:end for
```

4.4　IP 分配

在后面5.1 节分析中可以得到跳变地址空间越大,系统防嗅探能力越强. 文献[7,8]因为分布式路由限制,折中的选择了两级跳变原则. 但这样分配使得每个子网仅在分配 vIP 段的有限空间内跳变. 而本文在系统实现中采用共享池分配机制,使所有终端共享 IP 跳变池,以最大化跳变的不可预测性.

具体分配算法如算法 2,因为 IPv4 有限的和分散的未使用地址空间,本文将可用 vIP 分段存放在 $vIp[l][2]$ 这个二维数组当中,先随机选段,再在段中随机选 vIP.

算法 2　IP 分配

```
1:for each j < J do
2:  vIp[j][2]←{vIP 起始地址, vIP 终止地址}
3:end for
4:vIpPool ← 0
5:function VIP ALLOCATION
6:  while true do
7:    i←Random( ) mod J
8:    size←vIp[i][1] - vIp[i][0]
9:    ip ←vIp[i][0] + (Random( ) mod size)
10:   if vIpPool 没有 ip then
11:     vIpPool. add(ip)
12:     return ip
13:   end if
14: end while
15:end function
```

4.5　主动式随机路径跳变

当前域内路由转发策略通常采用 oSPF、iS-IS 等动态路由协议. 虽然可以通过增加“主动变换”策略,以提高流量传输的动态性、可靠性. 但作为分布式路由协议,每个节点状态的频繁改变会导致“路由收敛”问题. 因此,本文提出了主动式随机路径跳变技术,基于软件定义网络的全局视图,主动地、并发地随机变换多条流的路径.

SIDD 的主要挑战是随机地改变给定的源地址和目的地址之间的路径,同时考虑以下限制:(1)增加不可预测性;(2)避免网络中的任何链路超载(容量限制);(3)满足 QoS 约束.

物理网络可以表示为一个加权无向图 $G=(N,L)$,其中,N,L 分别表示物理交换机和物理链路的集合. 对于每个物理链路 $l_i \in L$,对应有一个可用的带宽资源 $B(l_i)$,时延 $D(l_i)$,抖动 $J(l_i)$,丢包率 $S(l_i)$. 对于生成的任意路径 $R_j(1 \leq j \leq K)$应满足如下约束:

$$\min_{l_i \in R_j} B(l_i) \geq B$$

$$\sum_{l_i \in R_j} D(l_i) \leq D$$

$$\sum_{l_i \in R_j} J(l_i) \leq J$$

$$\prod_{l_i \in R_j} S(l_i) \leq S$$

根据业务要求,B,D,J,S 取不同值,本文是用 ATM、Diffserv、ITU-T 等技术体系或标准给出的各类业务 QoS 参数推荐值作为依据进行取值.

在本文路径跳变机制中,首先通过 Dijkstra 算法得到初始路径 R_1,然后下发流表建立通信. 接着基于 R_1,使用 K 最短路径算法生成 K 条路径. 最后,跳变周期内随机选择一条路径,生成相应流表进行下发更新. 具体路径生成如算法 3 所示.

流表更新时,交换机会有三种情况:(1)只属于新路径;(2)只属于旧路径;(3)属于新、旧路径. 对于情况(1),报文按新的流表进行转发;对于情况(2),新的流表下发以后,如果报文到来,按照旧的流表进行处理,如果报文不再到来,经过 Idle_Timeout 后,旧流表被删除;对于情况(3),新流表会覆盖旧流表,所以报文按照新流表进行处理. 所以在流表更新过程中,报文的传输不会有丢包情况. 保证了报文的正常传输.

算法 3 动态路径跳变

输入:物理拓扑 $G(N,L)$
输出:流表 *flow*
1: **function** PATH GENERATE
2: **for all** 报文 p 来自 OF 交换机 **do**
3: $w[L]$←根据链路信息计算权重
4: R_1←Dijkstra($G(N,L)$,$w[L]$)
5: *flow*← 根据 R_1 生成流表
6: 下发流表 *flow*
7: **end for**
8: **end function**
9: **function** PATH MUTATION
10: **if** 跳变周期到 **then**
11: R←基于 R_1,利用 K 最短路径算法得到路径组
12: R_j←随机选择一条路径
13: *flow*←根据 R_j 生成流表
14: 下发流表 *flow*
15: **end if**
16: **end function**

5 分析

5.1 IP 跳变效果

为了评估 IP 跳变效果,文本采用 Carroll[13] 使用的瓮模型. 对于网络中 r 台终端,其分配地址空间大小为 v,设攻击者扫描次数为 k,因为地址空间足够大,所以假定 $v>k$.

(a)静态

发现一个内网终端所需要的探查次数 k 可建模为负超几何分布,令 Y 表示探查次数,$Y\sim H^-(1,r,v)$,则有:

$$E[Y]=v+1/r+1 \tag{1}$$

(b)动态

最优跳变旨在每次探查后进行跳变,攻击者从当前的探查中获得的信息将不会对下次探查提供任何有用的信息. 此时 Y 服从几何分布:

$$E[Y]=1/p=v/r \tag{2}$$

其中 $p=r/v$. 可以看出,要想增大探测扫描的难度,就得提高 IP 地址的跳变空间. SIDD 使所有终端共享一个大的 IP 池,跳变空间足够大,探查难度也随之上升. 从式(1)、式(2)的对比中可以发现,在相同的跳变空间内,动态性并不能提升探测难度. 但跳变能阻断攻击者对扫描结果的利用.

5.2 路径跳变效果

对于网络中的流量窃听,假设在一段时间 T 内传输的数据流为 f,则时间 T 内链路跳变次数为 r_d,每次传输路径节点数为 h_i,$i\in[o,r_d]$,网络总节点数为 n.

攻击者一次成功的攻击定义为获得完整的数据流 f. 假设攻击者有能力知道链路是否为目标流传输链路并且是否已经跳变,根据链路情况攻击者能根据策略重新选择监听节点. 攻击者在 T 时间内能连续扫描 k 次.

(a)静态

对于静态链路,攻击者只要探测到传输路径中的一条,就能获取整个数据流 f. 令 Z_k 表示 k 次扫描中扫出目标链路个数,同样可以根据 5.1 中模型得到:

$$\begin{aligned}\Pr(win)&=\Pr(Z_k>0)\\&=1-\Pr(Z_k=0)\\&=1-\binom{n-h_0}{k}\Big/\binom{n}{k}\end{aligned} \tag{3}$$

(b)动态

对于动态链路,攻击者必须在每次跳变后,在其跳变周期内重新探查到 f 的新链路,否则窃听失败. 因此攻击者成功的概率为:

$$\Pr(win)=\prod_{i=0}^{r_d}\left\{1-\binom{n-h_i-\mathrm{sgn}(i)}{q}\Big/\binom{n-\mathrm{sgn}(i)}{q}\right\}, \tag{4}$$

$$q=|k/r_d|$$

其中,sgn(·)为符号函数,q 为路径跳变周期内攻击者最大的探查次数. 考虑到用户流量的正常传输,路径跳变不能过快,所以默认 $q\geq 1$.

由式(3),式(4)可以得到图 4,其中 $n=20$,$h_i=4$. 可以发现随着攻击者探测能力增加,即 k 值加大,数据被窃听成功的概率越大. 当增加系统路径跳变周期,攻击者攻击成功的概率逐渐降低.

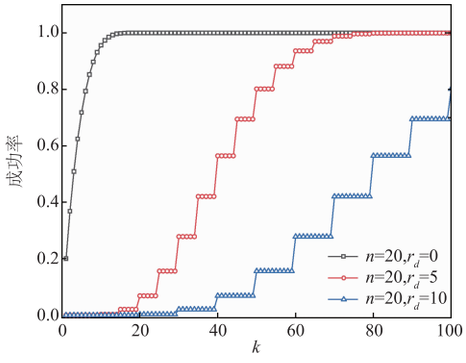


图4 路径跳变效果

6 实验结果及分析

6.1 功能性验证

在功能性验证中,本文搭建了一个小型的物理拓扑,如图 5. 图 6 展示了用户终端 H1 与 H2 的通信过程,此时交换机的流表信息如表 1. 可以看出,所有的 LLDP,DHCP,ARP,DNS 都交由控制器进行处理. 控制器会在接入端将报文的源 rIP 修改为 vIP,目的 vIP 修改为 rIP,修改目的 mac 地址.

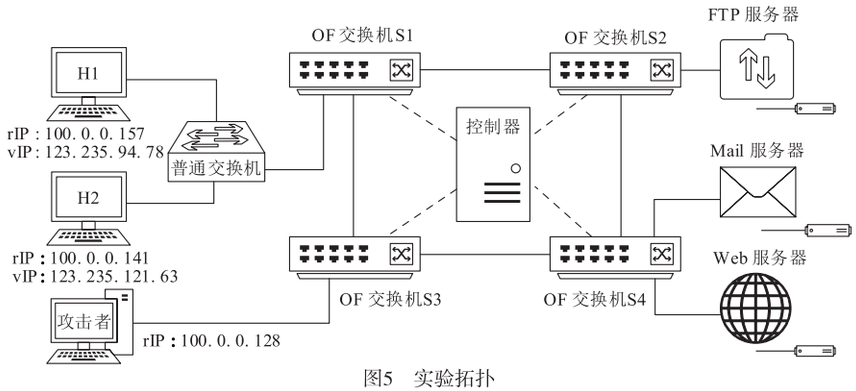
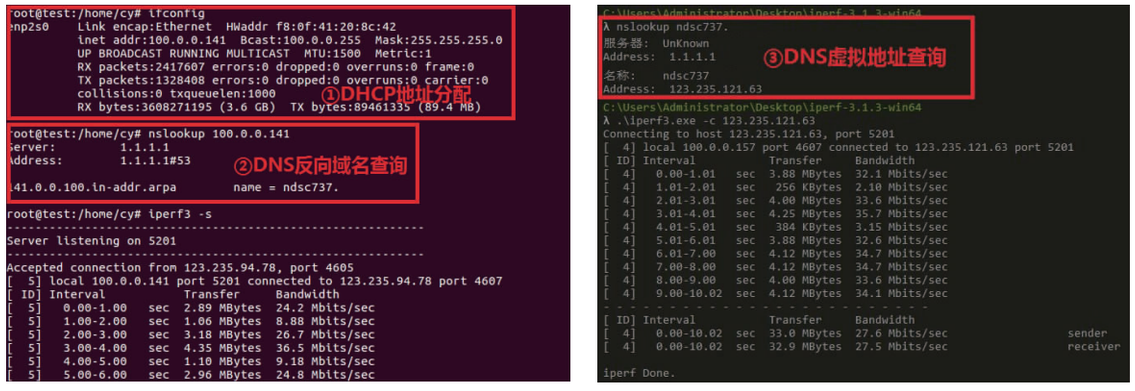


图5　实验拓扑



(*a*) H2　　(*b*) H1

图6　通信过程

表 1　下发流表

Match	Action
LLDP,DHCP,ARP,DNS	controller
tcp,nw_src:100. 0. 0. 157, nw_dst:123. 235. 121. 63	mod_dl_dst:f8:0f:41:20:8c:42, mod_nw_src:123. 235. 94. 78, mod_nw_dst:100. 0. 0. 141, output:1
tcp,nw_src:100. 0. 0. 141, nw_dst:123. 235. 94. 78	mod_dl_dst:6c:0b:84:42:84:51, mod_nw_src:123. 235. 121. 63, mod_nw_dst:100. 0. 0. 157, output:1

6.2　防御效果评估

(*a*)防扫描实验测试

为了显示 SIDD 对攻击列表攻击的有效性,本文用 Mininet 生成了 150 个在线终端,其中 50 台运行 Nmap 作攻击者对 100 台靶机进行扫描的. 攻击者通过 PING、TCP 报文(TCP_SYN)和反向域名(DNS_PTR)三种方式扫描 120 分钟. 结果如图 8 所示,在任何扫描中,发现的终端 vIP 不超过 4%,并且后续通过对扫描到的靶机 vIP 进行攻击,发现所有 IP 地址已经失效,由此可得 SIDD 能够使得基于 IP 扫描的网络攻击全部失效. 其中 PING 和 DNS_PTR 攻击无效.

(*b*)*K* 值选取

跳变路径数目 K 的选取跟网络规模有关,为此,定义网络节点 i 的重叠率为 δ:

$$\delta = \frac{n_i}{N}$$

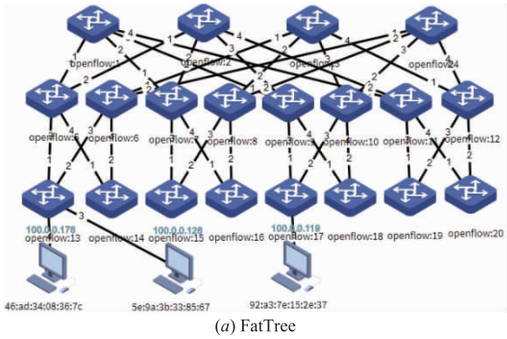
其中,N 为时间 T 内跳变的次数,n_i 为时间 T 内节点 i 的出现次数.

考虑到实际的网络拓扑的往往是如 FatTree[14] 和 VL2[15] 等一些经典拓扑,如图 7 所示. 所以图 9 统计了在这两种标准拓扑下,不同 K 值,随着时间变化,网络最大重叠率 δ_{max} 的变化. 可以看出,动态路径的跳变可以有效降低网路的重叠率. 但 K 取值越大,反而生成路径重叠率较高. 这是因为随着 K 值增大,生成的路径节点数开始增大,使得一些关键节点出现频率增大. 所以通过实验结果验证建议 K 值不要大于生成路径跳数的一半.

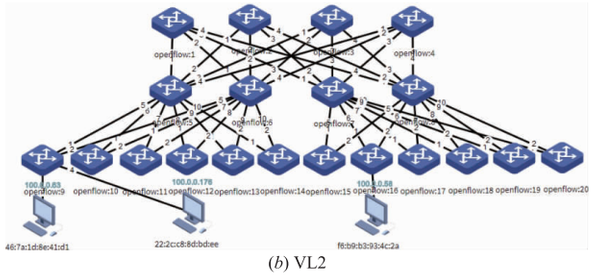
6.3　性能开销评估

(*a*)网络性能开销

本系统旨在控制层面实现了网络的动态 IP 跳变和路径跳变. 对用户端而言无感,所以用户方面没有额外的网络开销. 而对于控制器,需要增加整个网络拓扑和



(*a*) FatTree



(*b*) VL2

图7 数据中心网络架构

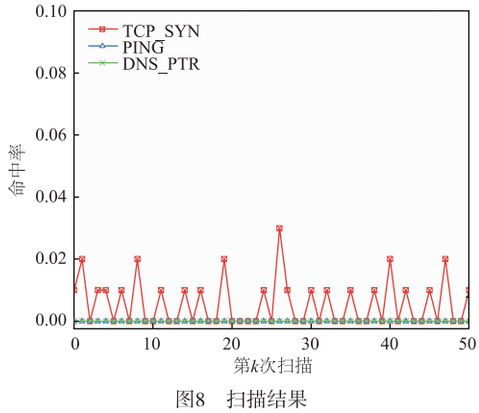


图8 扫描结果

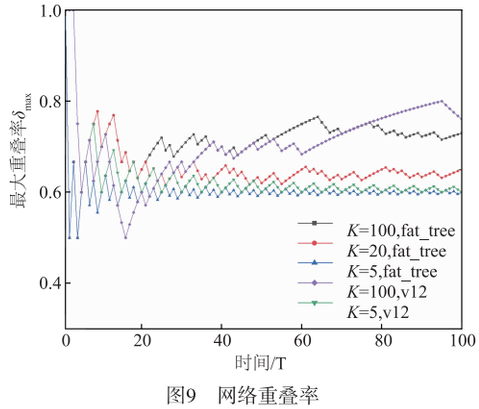


图9 网络重叠率

接入终端跳变信息的动态维护方面的开销. 对于一个经过商用测试的 Opendaylight 控制器来说,这些开销都在可控范围内.

对于 SDN 交换机方面的开销,图 10 对比了 RHM, SIDD 和普通模式下的交换机转发带来的时延开销. 从图中可以看出,SIDD 带来的时延开销小于 RHM 但高于普通模式. 因为 SIDD 会在报文入口进行一次修改,所以平均时延相对于普通模式稍高. 但 RHM 的 MG 网关会在报文入口和出口修改两次,交换机的性能开销和报文时延开销相对于 SIDD 将会加倍.

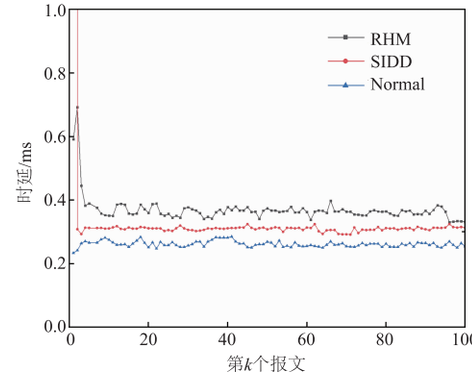


图10 网络时延

因为在 SIDD 的一次会话中,第一个报文会经过交换机上传到控制器进行处理,所以图中出现峰值.

(*b*)流表开销

因为 SIDD 为每两个终端会话建立下发一对流表,所以交换机存储的流表非常大. 对于 n 台终端,下发的流表最大为 C_n^2+4,与 n^2 呈正比. 当交换机同时处理 100 台终端的会话时,下发流表最高能到达 4954 条,这无疑是对交换机性能的一个巨大挑战. 相较而言,普通情况下,交换机只需要 100 条流表就够了.

7 结论

本文针对现有内网安全问题,以扰乱攻击者攻击链从而提高攻击者攻击难度为目标,通过“隔离 + 动态”的技术思路,设计与实现了一种基于软件定义的内网动态防御系统(SIDD),通过随机且不可预知的终端 IP 地址跳变和路径跳变来阻止侦查扫描的可用性,实验结果表明,SIDD 可以使得基于 IP 扫描的网络攻击全部失效,同时可以有效阻断零日漏洞,蠕虫,DoS 等网络攻击. 但系统带来的流表开销需要做进一步的优化处理.

参考文献

[1] Kindervag J. Build Security Into Your Network's DNA:The zero trust network architecture[R]. Cambridge:Forrester Research,2010. 1 – 26.

[2] Flores D A,Qazi F,Jhumka A. Bring your own disclosure: analysing BYOD threats to corporate information[A]. 2016

IEEE Trustcom/BigDataSE/ISPA [C]. Tianjin: IEEE, 2017. 23 – 26.

[3] Escobedo V, Beyer B, Saltonstall M, et al. BeyondCorp: The user experience[J]. Login, 2017, 42(3): 38 – 43.

[4] Hutchins E M, Cloppert M J, Amin R M. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains[R]. Montgomery: Lockheed Martin Corporation, 2011. 113 – 125.

[5] Beske C M C, Peck J, Saltonstall M. Migrating to beyondcorp: maintaining productivity while improving security [J]. Login, 2017. 42(3): 49 – 55.

[6] Liftoff: Guide to duo deployment best practices[Z/OL]. https://duo. com/assets/pdf/Duo-Liftoff-Guide. pdf. Duo, 2017-03-28.

[7] Jafarian J H, Al-Shaer E, Duan Q. An effective address mutation approach for disrupting reconnaissance attacks[J]. IEEE Transactions on Information Forensics & Security, 2015, 10(12): 2562 – 2577.

[8] Jafarian J H, Al-Shaer E, Duan Q. Openflow random host mutation: transparent moving target defense using software defined networking[A]. HotSDN '12 Proceedings of the First Workshop on Hot Topics in Software Defined Networks[C]. Helsinki: ACM, 2012. 127 – 132.

[9] Talipov E, Jin D, Jung J, et al. Path hopping based on reverse AODV for security[A]. APNOMS'06 Proceedings of the 9th Asia-Pacific International Conference on Network Operations and Management: Management of Convergence Networks and Services[C]. Busan: Springer-Verlag, 2006. 574 – 577.

[10] Duan Q, Al-Shaer E, Jafarian H. Efficient random route mutation considering flow and network constraints[A]. 2013 IEEE Conference on Communications and Network Security (CNS)[C]. Washington: IEEE, 2013. 260 – 268.

[11] Jafarian J H, Al-Shaer E, Duan Q. Formal approach for route agility against persistent attackers[A]. European Symposium on Research in Computer Security[C]. Egham: Springer, 2013. 237 – 254.

[12] 雷程,马多贺,张红旗,等. 基于最优路径跳变的网络移动目标防御技术[J]. 通信学报, 2017, 38(3): 133 – 143.
Lei Cheng, Ma Duo-he, Zhang Hong-qi, et al. Network moving target defense technique based on optimal forwarding path migration[J]. Journal on Communications, 2017, 38(3): 133 – 143. (in Chinese)

[13] Carroll T E, Crouse M, Fulp E W, et al. Analysis of network address shuffling as a moving target defense[A]. IEEE International Conference on Communications[C]. NSW: IEEE, 2014. 701 – 706.

[14] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture[A]. Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication[C]. Washington: ACM, 2008. 63 – 74.

[15] Greenberg A, Hamilton J R, Jain N, et al. VL2: a scalable and flexible data center network[J]. Communications of the ACM, 2009, 54(4): 95 – 104.

作者简介

陈　扬　男,1994 年出生,四川南充人,国家数字交换系统工程技术研究中心硕士生,主要研究方向为云计算、SDN、网络安全.
E-mail:2547756390@qq. com

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