计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210400296

TopoObfu:一种对抗网络侦察的网络拓扑混淆机制

刘亚群　邢长友　高雅卓　张国敏
陆军工程大学指挥控制工程学院　南京 210007
(1049178231@qq.com)

摘　要　链路洪泛等典型网络攻击需要在拓扑侦察的基础上针对网络中的关键链路开展攻击行为,具有较强的破坏性和隐蔽性。为了有效抵御这类攻击,提出了一种对抗网络侦察的拓扑混淆机制 TopoObfu。TopoObfu 能够根据网络拓扑混淆的需求,在真实网络中添加虚拟链路,并通过修改探测分组的转发规则使攻击者获得虚假的拓扑探测结果,隐藏网络中的关键链路。为了便于实现,TopoObfu 将虚假拓扑映射为 SDN 交换机的分组处理流表项,并支持在仅部分节点为 SDN 交换机的混合网络中部署。基于几种典型真实网络拓扑的仿真分析结果表明,TopoObfu 能够从链路重要性、网络结构熵、路径相似度等方面有效提升攻击者进行关键链路分析的难度,并在 SDN 交换机流表数量、混淆拓扑生成时间等方面具有较高的实现效率,可以减小关键链路被攻击的概率。

关键词:拓扑混淆;链路洪泛攻击;网络侦察;关键链路
中图法分类号　TP393

TopoObfu:A Network Topology Obfuscation Mechanism to Defense Network Reconnaissance

LIU Ya-qun,XING Chang-you,GAO Ya-zhuo and ZHANG Guo-min
College of Command and Control Engineering,Army Engineering University of PLA,Nanjing 210007,China

Abstract　Some typical network attacks,such as link-flooding attack,need to be carried out on critical links based on topology reconnaissance,which has strong destructiveness and stealthiness. In order to defense these attacks effectively,TopoObfu,a topology obfuscation mechanism against network reconnaissance,is proposed. TopoObfu can add virtual links to the real network according to the requirements of network topology obfuscation,and provide attacker with fake topology by modifying the forwarding rules of probing packets,and hide critical links in the network. To facilitate the implementation,TopoObfu maps the fake topology to the flow table entries used by SDN switches for packet processing,and can be deployed in the hybrid network where only part of the nodes are SDN switches. The simulation analysis based on several typical real network topologies shows that TopoObfu can effectively improve the difficulty of critical links analysis launched by attackers in terms of link importance,network structure entropy,path similarity and so on,and has high implementation efficiency in terms of the number of flow table entries in SDN switches,the generated time of fake topology,and can reduce the probability of critical links being attacked.

Keywords　Topology obfuscation,Link-flooding attack,Network reconnaissance,Critical links

1　引言

随着计算机网络的飞速发展,物联网、网上购物、电子银行等一大批新型产业方兴未艾,人们的生活已经离不开计算机网络。与此同时,网络攻击频发,给个人和国家造成了难以估量的损失。分布式拒绝服务(Distribution Denial of Service,DDoS)[1]攻击通过控制大量僵尸主机向目标节点发送大量数据包,消耗目标节点的资源,影响目标节点正常工作,这是当前 Internet 面临的主要威胁之一。国家互联网应急中心发现,2020 年上半年我国每日峰值流量超过 10 Gbps 的 DDoS 攻击约有 220 起[2]。

与传统的 DDoS 攻击不同,链路洪泛攻击(Link-flooding Attack,LFA)[3-4]通过控制大量僵尸主机向 Internet 中的关键链路注入大量低速率流量,降低甚至切断合法用户与服务器之间链路的连通性,破坏合法用户与服务器之间的正常通信。由于 LFA 发送的是具有真实 IP 地址的低速率流量,这种流量与合法流量的特征基本一致,因此检测 LFA 难度较大。与传统的 DDoS 攻击相比,LFA 的隐蔽性更强,危害性更大。2013 年,国际反垃圾邮件组织曾遭受 LFA,攻击流量的峰值达到了 300 Gbps[5]。

相关研究表明,Internet 是一种无标度网络,其中大多数节点或链路的重要性较低,而少数节点或链路的重要性较高[6]。Internet 的这一特性导致其对攻击者的蓄意攻击极为脆弱,攻击者只需破坏少数关键节点或关键链路就可以对整个网络的连通性造成严重影响。为了降低 LFA 的成本并提高攻击效果,高级攻击者会在攻击前实施网络侦察来推断 In-

到稿日期:2021-04-28　返修日期:2021-05-29
通信作者:邢长友(changyouxing@126.com)

ternet 中的关键链路，并将其作为 LFA 的目标链路。因此，获取目标网络的拓扑信息并以此推断网络中的关键链路是 LFA 的关键步骤。基于以上事实，我们认为利用拓扑混淆技术在 LFA 的侦察阶段隐藏 Internet 的关键链路可以明显降低 LFA 的攻击效果，从而实现对网络的有效保护。

目前隐藏关键链路的拓扑混淆技术主要面临以下两个挑战：1）如何确定网络中的关键链路以及如何定量地描述网络中关键链路的安全性；2）如何利用拓扑混淆技术以较低的成本有效地隐藏网络中的关键链路。

为解决以上问题，本文基于软件定义网络（Software Defined Network，SDN）[7]提出了一种能够有效防御链路洪泛等攻击侦察阶段的拓扑混淆机制 TopoObfu。首先，定义链路在各路径中出现的频率为链路的重要性，并利用基于链路重要性的网络结构熵来衡量网络中关键链路的安全性。其次，考虑到从传统网络到 SDN 网络的更新改造问题，TopoObfu 根据节点更新算法将传统网络中的少量路由器更新为 SDN 交换机，利用 SDN 的灵活性添加虚拟链路并修改数据包转发路径，生成混淆网络拓扑。最后，利用仿真软件验证 TopoObfu 的可行性，并设计一系列实验来评估 TopoObfu 的混淆效果与混淆成本。实验结果表明，TopoObfu 可以用较低的成本隐藏网络中的关键链路，从而有效提升攻击者进行关键链路分析的难度。

2 相关工作

现有的 LFA 防御方法主要分为两种：被动防御和主动防御。LFA 的被动防御方法通过检测网络内部的拥塞链路和动态修改流量的转发路径来缓解 LFA 对网络连通性的影响。LFAdefender[8]首先根据 LFA 目标链路选择算法来确定网络中的关键链路，然后根据 LFA 拥塞检测机制检测 LFA，最后根据多路路由策略和恶意流量阻塞策略缓解 LFA 造成的链路拥塞。与 LFAdefender 类似，Woodpecker[9]首先使用启发式算法将网络中的少量传统路由器更新为 SDN 交换机，并利用 SDN 灵活的数据包转发能力增加入口节点与出口节点之间的可选路径数量。然后，利用 SDN 交换机的统计功能，Woodpecker 提出了一种基于逐跳探测的拥塞链路定位机制，用于检测 LFA。最后，Woodpecker 的中心流量工程模块负责对所有链路上的流量进行负载均衡，消除瓶颈链路和其他链路之间的链路利用率差异。LFA 的被动防御方法必须在 LFA 发生后才能检测 LFA，并且需要修改流量转发路径。

与 LFA 的被动防御方法相比，LFA 的主动防御方法利用拓扑混淆技术来干扰 LFA 的侦察阶段，不影响流量的正常转发，并且在 LFA 发生前即可产生抵御效果。Trassare 等[10]首先通过添加虚拟链接使关键节点的中心度最小化，然后使用智能路由器伪装虚拟节点在边界的响应 traceroute 包。这种单节点响应请求的方式容易造成网络的性能瓶颈。HoneyNet[11]首先利用 SDN 交换机的统计功能找出可能包含瓶颈链路的一组网络关键节点，然后利用 Barabsi-Albert 模型[12]在关键节点的附近节点部署符合幂律分布的无标度诱饵网络，以混淆真实的网络拓扑。Linkbait[13]首先根据各链路的流量密度确定网络中潜在的 LFA 目标链路，然后将攻击者发送的探测流量重定向到关键链路附近的诱饵链路，使攻击者误认为诱饵链路是网络中的关键链路。文献[14]向网络中添加虚拟阴影网络，以虚假拓扑信息欺骗攻击者，解决了可用备选路由受网络拓扑限制的问题。NetHide[15]根据安全性和可用性等指标，将拓扑混淆问题建模为多目标优化问题，在保证 traceroute 等链路追踪工具可用性的条件下给攻击者提供虚假的网络拓扑信息，确保每条链路的安全性。在实现中，NetHide 利用 P4 可编程交换机修改数据包的 IP 地址、TTL 等字段。以上研究均以传统网络或 SDN 网络为实现环境，但均没有考虑传统网络到 SDN 网络的更新改造问题。

3 网络拓扑混淆机制 TopoObfu

本节首先介绍了网络拓扑混淆的应用背景，给出了相应的符号描述，然后讨论分析了拓扑混淆机制的评价指标，并在此基础上设计了 TopoObfu 网络拓扑混淆机制，给出了相应的实现算法。

3.1 威胁模型概述

本文考虑传统 IP 网络，如图 1 所示。网络中的服务器连接出口路由器，对外提供服务。外部用户通过入口路由器进入网络，访问服务器提供的服务。网络使用基于最短路径优先原则的路由策略，因此外部用户从入口路由器访问连接在出口路由器上的各服务器的路径是固定的。网络中的少数链路同时出现在多条路径中，具有较高的重要性。为了实现拓扑混淆功能，网络中的路由器节点可以替换为 SDN 交换机。

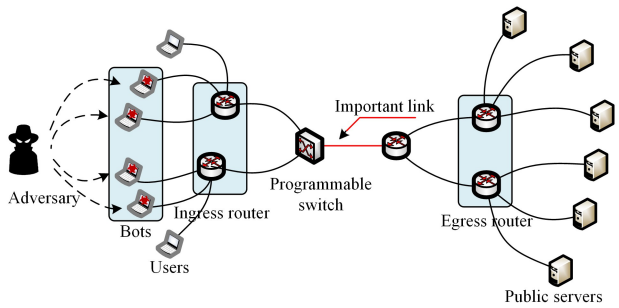


图 1 网络拓扑混淆系统模型

Fig. 1 System model of network topology obfuscation

LFA（如 Crossfire[3]和 Coremelt[4]）通过发送大量低速率流量到网络中的特定链路，破坏了合法用户与服务器之间的连通性。LFA 发送的低速率流量与正常流量的特征基本一致，因此 LFA 具有较高的隐蔽性。此外，考虑到攻击预算有限，高级链路洪泛攻击者会在实施 LFA 前探测目标网络中的关键链路，并选择少量关键链路作为 LFA 的目标，提高攻击效率。

Traceroute[16]等路径追踪工具是攻击者常用的网络侦察工具。Traceroute 通过向目标节点发送 TTL 值递增的数据包来获得中间路由器回复的“超时”信息，进而实现路由探测功能。攻击者只能通过入口路由器进入网络，访问连接在出口路由器上的服务节点，因此攻击者使用 traceroute 工具可获得从入口路由器到出口路由器之间的路由信息，并根据获得的路由信息推断网络中的关键链路。此外，traceroute 是合法用户常用的主机在线状态和网络故障的检测工具，因此攻

击者可将其发送的 traceroute 数据包伪装成合法用户发送的 traceroute 数据包，来隐藏其路由探测行为，但这会造成防御方对攻击探测行为检测的准确率较低，误报率较高。

本文的符号及其含义如表 1 所列。无向图 $G(N,L,P)$ 表示真实的网络拓扑，其中 N 表示网络中的路由节点集合，L 表示路由节点之间的链路集合，P 表示所有节点之间的路径集合。向量 $\boldsymbol{path}_{s,d}=(l_1,l_2,\cdots,l_n)(s,d\in N,l_i\in L)$ 表示节点 s 和节点 d 之间的路径，$\boldsymbol{path}_{s,d}=\infty$ 代表节点 s 和节点 d 之间不连通。特别指出，本文所说的路径指一对节点之间的路由路径，而不是一对节点之间的所有路径。

表 1 符号及含义

Table 1 Symbol and meaning

Symbol	Meaning
N	The set of network nodes
$S\subseteq N$	The set of ingress nodes
$D\subseteq N$	The set of egress nodes
L	The set of links between nodes
$G(N,L,P)$	Real network topology
$G(N',L',P')$	Fake network topology
$\boldsymbol{path}_{s,d}$	The path vector between node s and d
$P_{S,D}$	The set of paths between set S and set D
$L_{S,D}$	The set of links in $P_{S,D}$
F_l	The importance of link l
δ_s	The cost of updating a router
δ_f	The cost of maintaining a flow table
$H(G)$	Network structure entropy of network G

3.2 网络拓扑混淆效果度量指标

本节依据系统模型和攻击模型，用链路在各路径中出现的频率衡量链路的重要性，并进一步引入基于链路重要性的网络结构熵来衡量网络中关键链路的安全性；随后使用路由节点的更新数量和可编程交换机的流表数量来衡量混淆拓扑的混淆成本。

3.2.1 链路的重要性评估

攻击者实施 LFA 的目的是尽可能多地切断正常用户与服务器之间的连通性，因此通过一条链路的路径数量越多，这条链路的重要性就越高。鉴于攻击者可以探测到所有入口节点与出口节点之间的路径，因此我们用链路在从入口节点到出口节点的路径上出现的频率来衡量链路的重要性。用 $P_{S,D}=\{\boldsymbol{path}_{s,d}\mid s\in S,d\in D\}$ 表示入口节点与出口节点之间的路径集合，$L_{S,D}$ 为 $P_{S,D}$ 包含的链路集合。对于 $L_{S,D}$ 中的任意一条链路 l，F_l 为链路 l 在所有从入口节点到出口节点的路径上出现的频率。

$$L_{S,D}=\bigcup_{s\in S,d\in D}\{(u,v)\mid(u,v)\subseteq\boldsymbol{path}_{s,d}\} \tag{1}$$

$$F_l=\sum_{s\in S,d\in D}B^l_{\boldsymbol{path}_{s,d}} \tag{2}$$

如果链路 $l\in\boldsymbol{path}_{s,d}$，则 $B^l_{\boldsymbol{path}_{s,d}}=1$，$F_l$ 代表链路 l 的重要性。

3.2.2 网络结构熵

为定量描述网络中各链路的差异性，我们引入基于网络链路重要性的网络结构熵来衡量网络中关键链路被识别的难易程度。用 $H(G)$ 表示网络 G 的结构熵，则：

$$H(G)=\sum_{l\in L_{S,D}}I_l\cdot\ln I_l \tag{3}$$

$$I_l=\frac{F_l}{\sum_{l\in L_{S,D}}F_l} \tag{4}$$

I_l 为链路 l 的重要性与所有链路重要性之和的比值。网络 G 的链路重要性结构熵可以代表网络 G 中关键链路的安全性，基于链路重要性的网络结构熵越小，网络 G 中各链路的重要性差异越大，网络 G 中关键链路被识别的可能性就越大；基于链路重要性的网络结构熵越大，网络 G 中各链路的重要性差异越小，网络 G 中关键链路被识别的可能性就越小。

3.2.3 成本

TopoObfu 通过链路的增添操作来生成混淆拓扑，而链路的增添操作由 SDN 交换机的分组处理流表具体实现。TopoObfu 生成混淆拓扑的成本主要来源于路由节点的更新和 SDN 交换机流表项的维护，令 δ_s 为将一台普通路由器更新为 SDN 交换机的成本，δ_f 为 SDN 交换机每管理一条流表项的成本，如果 TopoObfu 更新 m 个普通路由器，那么 TopoObfu 生成混淆拓扑的成本 C 为：

$$C=\delta_s\cdot m+\delta_f\cdot\sum_{i=1}^{m}S_i \tag{5}$$

其中，S_i 为第 i 个可编程交换机维护的流表项数量。

3.3 TopoObfu 混淆算法设计

3.3.1 拓扑混淆算法设计

网络中最短路径优先的路由策略导致节点间的路径相对固定，少量关键链路出现在节点间的多条路径上，对网络连通性具有较大影响。受限于攻击资源，理性的攻击者在实施 LFA 前会探测从入口节点到出口节点之间的拓扑信息，并选择其中的少量关键链路作为攻击的目标。拓扑混淆算法 TopoObfu 利用 SDN 交换机灵活的数据包处理能力来添加虚拟链路并修改数据包转发路径，干扰攻击者的路由探测行为，使攻击者获得虚假的网络拓扑信息，降低网络中链路之间的重要性差异，从而有效防御 LFA 对关键链路的攻击。

例如，在图 2 中，入口节点集 $\{R_1,R_2\}$ 与出口节点集 $\{R_8,R_9\}$ 之间的通信都经过链路 (S_1,R_5)，链路 (S_1,R_5) 为网络中的关键链路。拓扑混淆时，TopoObfu 首先将节点 S_1 更新为 SDN 交换机，然后在节点 S_1 和节点 R_6 之间添加虚拟链路，并将入口节点 R_1 与出口节点 R_8 之间的通信路径由 $(R_1,R_3,S_1,R_5,R_6,R_8)$ 更新为 (R_1,R_3,S_1,R_6,R_8)，降低关键链路 (S_1,R_5) 的重要性。

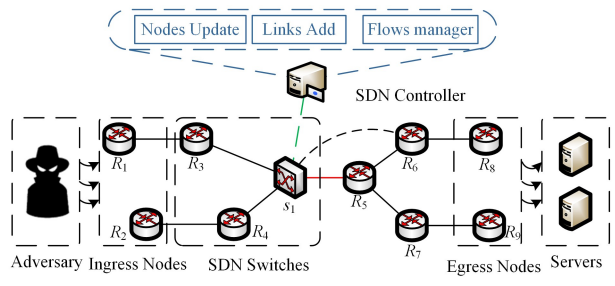


图 2 TopoObfu 系统架构

Fig. 2 System architecture of TopoObfu

拓扑混淆算法 TopoObfu 的目的是在不影响合法用户正常业务访问的前提下，使从入口节点到出口节点的所有路径上的链路的重要性都尽可能相似，以减小关键链路被攻击者识别的概率。拓扑混淆算法 TopoObfu 的输入为网络拓扑 $G(N,L,P)$、入口节点集合 S、出口节点集合 D、SDN 交换机节点的数量 M 和关键链路的数量 K，输出为混淆网络拓扑 G'

$=(N',L',P')$。TopoObfu 的具体执行过程包括两个步骤：节点更新和路由更新。在节点更新步骤中，TopoObfu 计算各链路的重要性，并将 $M(M\leqslant|N|)$ 个传统路由器更新为 SDN 交换机。在路由更新步骤中，TopoObfu 通过在 SDN 交换机上执行链路的添加操作并且修改从入口节点到出口节点的数据包转发路径，来降低真实网络中关键链路的重要性。TopoObfu 的工作流程如算法 1 所示。

算法 1 TopoObfu

Input：Physical topology G=(N,L,P)；Ingress nodes S；Egress nodes D；Number of nodes to be updated M；Number of critical links K

Output：Obfuscated topology G′=(N′,L′,P′)

1. if M≤|N|
2. V,E←NodeUpdate(G,S,D,M,K)
3. end if
4. G′(N′,L′)←RouteUpdate(G,S,D,V,E)
5. return G′

3.3.2 关键路由节点更新算法

在传统 IP 网络中，传统路由器节点只能根据数据包的目的地址进行分组转发。为了修改入口节点与出口节点之间的数据包转发路径，TopoObfu 将传统的路由器更新为 SDN 交换机，并利用 SDN 技术的数控分离特性以及灵活可编程的特点修改真实网络的数据包转发路径。由于实际网络中包含大量的传统路由器，路由节点的全部更新必然会消耗大量的经济和时间，进而影响合法用户对业务的正常访问。为了节约成本，TopoObfu 采取了一种增量式的路由节点更新方案：在一定成本范围内将少量关键的传统路由器更新为 SDN 交换机，并且随着成本预算的提高动态增加更新为 SDN 交换机的节点的数量。考虑入口节点与出口节点之间的所有有向路径 $P_{S,D}=\{path_{s,d}\mid s\in S,d\in D\}$，路径 $path_{s,d}=(s,n_0,u,n_1,n_2,d)$ 上的 SDN 交换机节点 u 可以在 u 与 u 之后的所有节点之间添加虚拟链路 (u,n_2) 和 (u,d)。例如路径 $path_{s,d}=(s,n_1,u,n_1,n_2,d)$，SDN 交换机节点 u 可以添加链路 (u,n_1)，(n_1,n_2) 和 (n_2,d)。为了减少 SDN 交换机的数量，节点更新问题可以转化为在网络拓扑 $G(N,L,P)$ 中从入口节点集 S 到出口节点集 D 的所有路径 $P_{S,D}$ 上选择 M 个适用性最高的节点。节点的适用性指包含该节点的所有路径上位于该节点之后的链路的重要性之和，节点的适用性反映了一个节点适合执行链路增添操作的程度。用 $N_{S,D}$ 表示 $P_{S,D}$ 上包含的节点的集合。

$$N_{S,D}=\bigcup_{s\in S,d\in D}\{n\mid n\in path_{s,d}\} \tag{6}$$

用 L_n 表示包含节点 n 的所有路径的集合。

$$L_n=\bigcup_{s\in S,d\in D}\{path_{s,d}\mid n\in path_{s,d}\} \tag{7}$$

用 A_n 表示节点 n 的适用性，$l\ll n$ 表示链路 l 在节点 n 之后。对于 $N_{S,D}$ 中的任一节点 n：

$$A_n=\sum_{path\in L_n}\Big(\sum_{l\in path,l\ll n}I_l\Big) \tag{8}$$

节点更新算法 NodeUpdate 的输入为网络拓扑 $G(N,L,P)$、入口节点集合 S、出口节点集合 D、SDN 交换机节点的数量 M 和关键链路的数量 K，输出为 M 个 SDN 交换机节点集合以及 K 条关键链路集合。NodeUpdate 首先计算从入口节点集 S 到出口节点集 D 的所有路径 $P_{S,D}$ 上的所有链路的重要性，然后计算 $P_{S,D}$ 上的所有节点的适用性，最后 NodeUpdate 输出适用性最高的 M 个 SDN 交换机节点以及重要性最高的 K 条链路。NodeUpdate 的工作流程如算法 2 所示。

算法 2 NodeUpdate

Input：Physical topology G=(N,L,P)；Ingress nodes S；Egress nodes D；Number of nodes to be updated M；Number of critical links K

Output：SDN Switches V；Critical links E

1. $L_{S,D}\leftarrow\bigcup_{s\in S,d\in D}\{(u,v)\mid(u,v)\subseteq path_{s,d}\}$
2. for l in $L_{(S,D)}$
3. $F_l\leftarrow\sum_{s\in S,d\in D}B^l_{path_{s,d}}$
4. end for
5. for l in $L_{S,D}$
6. $I_l\leftarrow\dfrac{F_l}{\sum_{l\in L_{S,D}}F_l}$
7. Importance[l]←I_l
8. end for
9. $N_{S,D}\leftarrow\bigcup_{s\in S,d\in D}\{n\mid n\in path_{s,d}\}$
10. for n in $N_{S,D}$
11. for path in L_n
12. $A_n\leftarrow\sum_{l\in path,l\ll n}I_l$
13. Adaptation[n]←A_n
14. end for
15. end for
16. V←The largest M values in Adaptation
17. E←The largest K values in Importance
18. return V,E

本文将 NodeUpdate 输出的 M 个 SDN 交换机节点替换成 SDN 交换机，并利用 SDN 控制器控制 SDN 交换机的数据包修改和数据包转发行为。

3.3.3 网络路由更新算法

真实网络中各链路的重要性具有较大差异，因此真实网络中的关键链路容易被攻击者发现。TopoObfu 通过在更新的 SDN 交换机上添加虚拟链路并修改数据包的转发路径，来降低关键链路在入口节点与出口节点之间的路径上出现的频率。

例如，在图 2 中，TopoObfu 添加链路 (S_1,R_6) 并将 R_1 与 R_8 之间的转发路径由 $(R_1,R_3,S_1,R_5,R_6,R_8)$ 更新为 (R_1,R_3,S_1,R_6,R_8)，其余节点之间的路径保持不变，这样可以降低链路 (S_1,R_5) 在路径上出现的频率。转发路径修改前后网络中各链路的重要性如表 2、表 3 所列。转发路径修改前网络的结构熵为 3.122，转发路径修改后网络的结构熵为 3.261。转发路径的修改会增加虚拟链路的重要性，降低真实网络中关键链路的重要性，这有利于减小链路之间的重要性差异，从而增大网络的结构熵。

表 2 路由更新前的网络结构熵

Table 2 Network structure entropy after routing update

Link	Importance	Link	Importance
$(R1,R3)$	2	$(R5,R6)$	2
$(R2,R4)$	2	$(R5,R7)$	2
$(R3,S1)$	2	$(R6,R8)$	2
$(R4,S1)$	2	$(R7,R9)$	2
$(S1,R5)$	4	—	—

表 3 路由更新后的网络结构熵

Table 3 Network structure entropy after routing update

Link	Importance	Link	Importance
(*R*1,*R*3)	2	(*R*5,*R*6)	1
(*R*2,*R*4)	2	(*R*5,*R*7)	2
(*R*3,*S*1)	2	(*R*6,*R*8)	2
(*R*4,*S*1)	2	(*R*7,*R*9)	2
(*S*1,*R*5)	3	(*S*1,*R*6)	1

路由更新算法 RouteUpdate 以真实的网络拓扑 $G(N,L,P)$、入口节点集合 S、出口节点集合 D、SDN 交换机节点集合 V、关键链路集合 E 和路由更新概率 p 为输入，输出混淆的网络拓扑 $G'(N',L',P')$。对于入口节点集 S 和出口节点集 D 之间的每一条路径，RouteUpdate 先在该路径上的 SDN 交换机节点中随机选择一个作为前置节点，并在该路径上关键链路后的节点中随机选择一个作为后置节点，随后 RouteUpdate 在前置节点与后置节点之间添加虚拟链路。最后，RouteUpdate 以概率 p 使该路径通过添加的虚拟链路形成混淆网络拓扑 $G'(N',L',P')$。RouteUpdate 流程如算法 3 所示。

算法 3 RouteUpdate

Input: Physical topology G=(N,L,P); Ingress nodes S; Egress nodes D; SDN Switches V; Critical links E; Route update probability p;

Output: Obfuscated topology G′ (N′,L′,P′)

1. G′←G
2. $P_{S,D}=\{path_{s,d}\mid s\in S,d\in D\}$
3. for path in $P_{S,D}$
4. if path∩V!=∅ and path∩E!=∅
5. Front_node←Random(path∩V)
6. $Back_{node}$←Random({node|node≪(path∩E)})
7. G′.addlink($Front_{node}$,$Back_{node}$)
8. Update routing information with probability p
9. return G′ (N′,L′,P′)

4 TopoObfu 系统的设计

本节详细介绍 TopoObfu 系统的具体实现流程。首先利用 Python 的 NetworkX[17] 模块拓扑混淆算法计算混淆网络拓扑，然后利用 Ryu[18] 和 Mininet[19] 搭建验证系统，证明 TopoObfu 的可行性。

4.1 混淆网络生成

首先，混淆网络拓扑生成程序读取 json 格式的真实网络拓扑，并利用 NetworkX 模块创建网络拓扑。然后，混淆网络拓扑生成程序根据拓扑混淆算法，利用 NetworkX 模块灵活地修改网络拓扑，计算得到混淆的网络拓扑信息。最后，混淆网络拓扑生成程序将混淆拓扑信息以 json 格式保存在文件中。

4.2 控制器与交换机

本文使用 Mininet 构建网络拓扑，Mininet 默认使用 OpenvSwitch[20] 软件交换机。此外，本文使用基于 OpenFlow1.3 协议的 Ryu 作为 SDN 控制器，Ryu 的 Nicira 扩展结构[21] 支持流表项匹配并修改 TTL 字段。

在 Mininet 中，混淆网络中的传统路由器和 SDN 交换机由 OpenvSwitch 实现，受 Ryu 控制器控制。传统路由器根据数据包的目的地址，按照最短路径进行分组转发，SDN 交换机通过匹配并修改 IP 数据包的 TTL 字段来实现虚拟链路添加和路由更新功能。SDN 交换机收到外部用户发送的访问服务器的 IP 数据包或者是服务器响应外部用户的 IP 数据包后，Ryu 控制器计算源节点与目的节点之间真实路径与虚拟路径的差值，并给 SDN 交换机下发流表项，为数据包的 TTL 值增加真实路径与虚拟路径之间的差值。

SDN 交换机的数据包处理流程如图 3 所示。

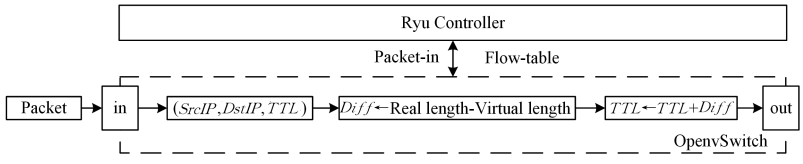


图 3 数据包处理流程

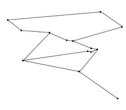
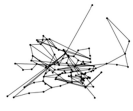
Fig. 3 Processing steps of packet

5 实验评估

本文利用 Mininet 网络仿真器搭建了 Topology Zoo[22] 上的 3 种不同规模的网络，3 种网络的网络拓扑、节点数量与链路数量如表 4 所列。在每种网络拓扑中，本文随机选择 10% 的节点作为入口节点，选择 10% 的节点作为出口节点，并且只考虑入口节点与出口节点之间的路径。传统路由器和 SDN 交换机均为 OpenvSwitch，Ryu 使用 OpenFlow1.3 协议与 SDN 交换机通信。我们对下面给出的每一项评估实验都重复进行了 500 次，并展示出了最终的平均值。实验结果显示，关键节点的数量对实验结果的影响不大，因此我们在入口节点到出口节点路径上选择适用性最高的前 10% 个节点作为关键节点，并将关键节点由传统路由器更新为 SDN 交换机。

表 4 网络拓扑信息

Table 4 Information of network topology

Type of topology	Eunetworks	Iris	Interoute
Topography			
The number of nodes	14	51	110
The number of links	16	64	148

5.1 链路的重要性

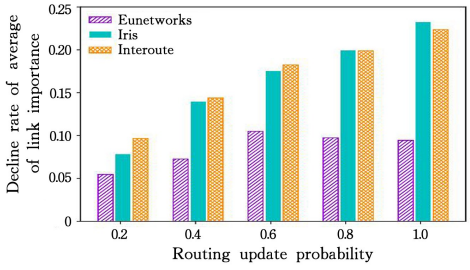
链路承载的流量数量可以反应链路在网络中的重要程度。我们在实验中测量了 3 种网络拓扑中路由更新概率对网络拓扑入口节点与出口节点之间的路径上包含的链路的重要性平均值 F_avg 和重要性方差 F_var 的影响，并使用链路重

要性平均值的下降率 FR_avg 和链路重要性方差的下降率 FR_var 来展示与分析实验结果。

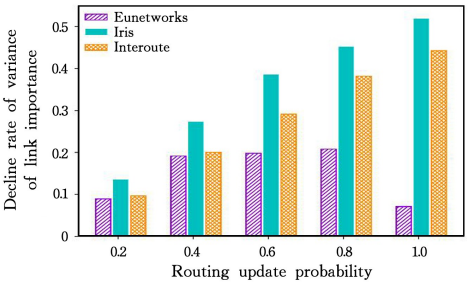
$$FR\_avg=1-\frac{F\_avg}{F_0\_avg} \tag{9}$$

$$FR\_var=1-\frac{F\_var}{F_0\_var} \tag{10}$$

图 4(a)和图 4(b)分别给出了 3 种网络拓扑中链路重要性平均值的下降率随路由更新概率的变化趋势，以及链路重要性方差的下降率随路由更新概率的变化趋势。在 Iris 和 Interoute 中，链路重要性平均值的下降率和链路重要性方差的下降率都随路由更新概率的增长而增大，这与预期相符。但在 Eunetworks 中，链路重要性平均值的下降率和链路重要性方差的下降率都随路由更新概率的增长而先增大后减小，这可能是因为 Eunetworks 的网络规模较小，其真实网络中各链路的重要性较低，较大的路由更新概率导致虚拟链路的重要性超过真实链路的重要性，进而导致链路重要性平均值的下降率和链路重要性方差的下降率变小。



(a)链路重要性平均值的下降率



(b)链路重要性方差的下降率

图 4 链路重要性评估

Fig. 4 Evaluation of link importance

5.2 网络结构熵

网络结构熵代表网络中关键链路的安全性，网络的结构熵越大，网络中各链路的重要性差异越小，网络中的关键链路就越安全；网络结构熵越小，网络中各链路的重要性差异越大，网络中的关键链路就越不安全。根据 3.2.2 节中的定义，我们分析了路由更新概率对 3 种网络的网络结构熵的影响，如图 5 所示。3 种网络的结构熵变化范围较小，其中 Iris 和 Interoute 的网络结构熵随路由更新概率的增长而增大，Eunetworks 的网络结构熵随路由更新概率的增长而先增大后减小。这同样是因为 Eunetworks 的网络规模较小，较大的路由更新概率导致虚拟链路的重要性超过真实链路的重要性，各链路重要性的差异变大。在路由更新概率为 100%时，混淆网络的网络结构熵甚至低于真实网络的网络结构熵。

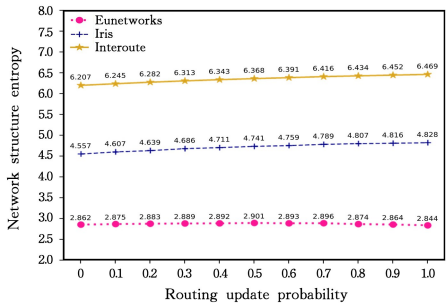


图 5 网络结构熵评估

Fig. 5 Evaluation of network structure entropy

5.3 流表数量

SDN 交换机中的流表数量是影响 SDN 交换机性能的重要因素，流表数量的增多会导致 SDN 交换机的转发效率下降，因此本文使用流表数量来衡量 TopoObfu 的成本。3 种网络在满足所有入口节点与出口节点之间的通信需求时，单个 SDN 交换机的平均流表数量和最大流表数量与路由更新概率的关系如图 6 所示。网络规模越大，流表数量随路由更新概率的增长趋势越明显。由于 Ryu 控制器需要针对每一组 $(SrcIP, DstIP, TTL)$ 下发一条流表，在 Interoute 网络中，SDN 交换机的最大流表数量已经超过 320 条。

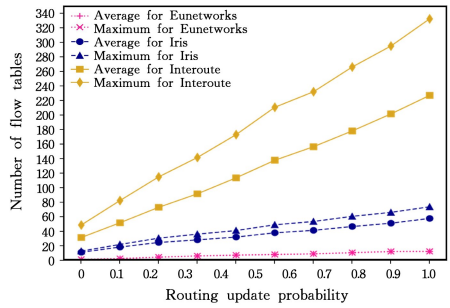


图 6 流表数量评估

Fig. 6 Evaluation of the number of flow tables

5.4 路径长度

Traceroute 等路径探测工具除了能够获得中间节点的信息外，还能获得源节点与目的节点之间的往返时延。利用网络层析技术[23]，攻击者可以根据源节点与目的节点之间的往返时延推测出网络拓扑信息。虽然 TopoObfu 给外部用户展示的是混淆网络，但实际上 TopoObfu 仍按照真实网络拓扑进行数据包转发，这可能会造成由网络层析技术推测得到的网络拓扑信息与 TopoObfu 展示的混淆网络拓扑信息不一致，从而引起攻击者的怀疑。虽然增加路由节点的回复时延能够解决这一问题，但是这样同时会增加合法用户的访问时延，因此 TopoObfu 的一致性是影响 TopoObfu 防御效果的重要因素。在 3 种不同规模的网络中，我们通过比较真实网络与虚拟网络的平均路径长度来分析 TopoObfu 的一致性。

图 7 给出了网络入口节点与出口节点之间路径的平均长度随路由更新概率的变化趋势。虽然 3 种网络的平均路径长度均随路由更新概率的增长而减小，但是 Eunetworks，Iris 和 Interoute 的最大变化幅度分别为 0.292，0.45 和 0.56，3 种网

络的平均路径长度变化幅度非常小。在这种情况下，网络层析技术推测得到的网络信息与 TopoObfu 展示的混淆网络信息基本一致，因此 TopoObfu 具有较好的一致性。

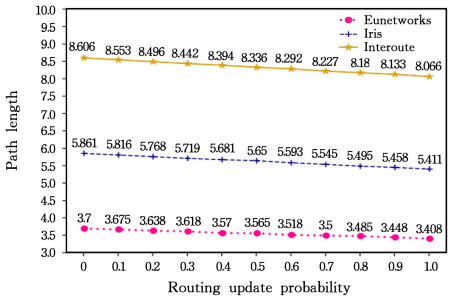


图 7 路径长度评估

Fig. 7 Evaluation of path length

5.5 拓扑生成时间

在实际环境中，网络设备的更新换代和网络的局部故障等会造成网络的节点、链路以及路由信息不断变化。针对真实网络的更新，TopoObfu 需要重新计算混淆网络。为了评估 TopoObfu 对网络更新的灵敏性，我们测量了混淆网络生成程序对 3 种不同规模网络的计算时间。实验结果显示，Eunetworks 的混淆网络计算时间约为 0.0033 s，Iris 的混淆网络计算时间约为 0.0371 s，Interoute 的混淆网络计算时间约为 0.9524 s。3 种不同规模网络的平均混淆网络计算时间均在 1s 以内，这表明 TopoObfu 可以在较短的时间内建立其对抗 LFA 攻击网络侦察的防御机制，有效应对真实网络的更新。

5.6 安全性

网络未进行拓扑混淆时，攻击者可以轻易识别网络中的关键链路并对关键链路实施链路洪泛攻击，进而严重破坏网络的连通性。本节讨论在网络实施拓扑混淆之后，攻击者在不同攻击预算之下攻击到关键链路的概率。我们设定真实网络中重要性由高到低的前 10%的链路为关键链路，路由更新的概率为 100%。

图 8 给出了 3 种网络拓扑中关键链路被攻击的概率随攻击预算的变化趋势，横坐标攻击预算为攻击者攻击的链路数量占网络链路总数量的比例，纵坐标攻击关键链路的概率为关键链路的数量占攻击者攻击链路数量的比例。3 种网络关键链路被攻击的概率均随着攻击预算的增加而减小，且关键链路被攻击的概率均小于 25%。这一结果说明，攻击者攻击的关键链路数量的增长率小于攻击预算的增长率，且 TopoObfu 能极大地增加攻击者的攻击成本。

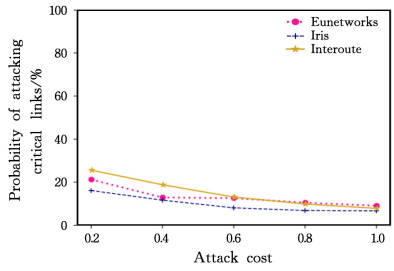


图 8 安全性评估

Fig. 8 Evaluation of security

结束语 针对链路洪泛攻击者在攻击前期实施的网络侦察，本文提出了一种面向欺骗防御的拓扑混淆机制 TopoObfu。TopoObfu 的主要思想是将网络中的少量传统路由器更新为 SDN 交换机，通过添加虚拟链路和修改数据包转发路径来给攻击者提供混淆的网络拓扑信息，从而隐藏网络中的关键链路。实验结果表明，TopoObfu 能快速形成对真实网络的有效保护，提高攻击者的攻击成本，并且 TopoObfu 的防御机制不容易被攻击者发现。

TopoObfu 仅考虑了在真实网络中添加虚拟链路，下一步我们将继续探索添加虚拟节点对网络混淆效果的影响。此外，在大规模网络中，SDN 交换机的流表数量庞大，会影响数据包的转发效率，下一步将利用 P4[24] 可编程交换机实现 TopoObfu，以减少流表数量，提升数据包的转发效率。

参 考 文 献

[1] DOULIGERIS C, MITROKOTSA A. DDoS attacks and defense mechanisms: classification and state-of-the-art [J]. Computer Networks, 2004, 44(5): 643-666.

[2] 国家互联网应急中心. 2020 年上半年我国互联网网络安全监测数据分析报告[EB/OL]. (2020-09-26)[2021-05-18]. https://www.cert.org.cn/publish/main/46/2020/2020092608504265 2505447/20200926085042652505447_html.

[3] KANG M S, LEE S B, GLIGOR V D. The crossfire attack[C]// 2013 IEEE symposium on security and privacy. IEEE, 2013: 127-141.

[4] STUDER A, PERRIG A. The coremelt attack[C]// European Symposium on Research in Computer Security. Berlin: Springer, 2009: 37-52.

[5] BRIGHT P. Can a ddos break the internet? Sure…Just not all of it[EB/OL]. (2013-04-02)[2021-05-18]. http://arstechnica.com/security/2013/04/can-a-ddos-break-the-internet-sure-just-not-all-of-it/.

[6] BARABÁSI A L. Scale-free networks: a decade and beyond[J]. Science, 2009, 325(5939): 412-413.

[7] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.

[8] WANG J, WEN R, LI J, et al. Detecting and mitigating target link-flooding attacks using sdn[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 16(6): 944-956.

[9] WANG L, LI Q, JIANG Y, et al. Woodpecker: Detecting and mitigating link-flooding attacks via SDN[J]. Computer Networks, 2018, 147: 1-13.

[10] TRASSARE S T, BEVERLY R, ALDERSON D. A technique for network topology deception[C]// MILCOM 2013-2013 IEEE Military Communications Conference. IEEE, 2013: 1795-1800.

[11] KIM J, SHIN S. Software-defined HoneyNet: Towards mitigating link flooding attacks[C]// 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2017: 99-100.

[12] BARABÁSI A L, ALBERT R. Emergence of scaling in random networks[J]. Science, 1999, 286(5439): 509-512.

[13] WANG Q, XIAO F, ZHOU M, et al. Linkbait: active link obfuscation to thwart link-flooding attacks[J]. arXiv: 1703. 09521, 2017.

[14] AYDEGER A, SAPUTRO N, AKKAYA K. Utilizing NFV for Effective Moving Target Defense against Link Flooding Reconnaissance Attacks[C]// 2018 IEEE Military Communications Conference(MILCOM), New York. IEEE, 2018: 946-951.

[15] MEIER R, TSANKOV P, LENDERS V, et al. NetHide: Secure and practical network topology obfuscation[C]// 27th USENIX Security Symposium (USENIX Security 18). 2018: 693-709.

[16] KERNEN T. Traceroute[EB/OL]. [2021-05-18]. http://www.traceroute.org/.

[17] NETWORKX D. Networkx[EB/OL]. [2021-05-18]. https://networkx.org/.

[18] Welcome to RYU the Network Operating System(NOS)[EB/OL]. [2021-05-18]. https://ryu.readthedocs.io/en/latest/index.html.

[19] 2021 Mininet Project Contributors. Mininet[EB/OL]. [2021-05-18]. http://mininet.org/.

[20] A LINUX FOUNDATION COLLABORATIVE PROJECT. OpenvSwitch[EB/OL]. [2021-05-18]. http://www.openvswitch.org/.

[21] Nicira Extension Structures[EB/OL]. [2021-05-18]. https://ryu.readthedocs.io/en/latest/nicira_ext_ref.html.

[22] THE UNIVERSITY OF ADELAIDE. The internet topology zoo[EB/OL]. (2013-04-16) [2021-05-18]. http://topology-zoo.org/.

[23] COATES M, CASTRO R, NOWAK R, et al. Maximum likelihood network topology identification from edge-based unicast measurements[J]. ACM SIGMETRICS Performance Evaluation Review, 2002, 30(1): 11-20.

[24] BOSSHART P, DALY D, GIBB G, et al. P4: Programming protocol-independent packet processors[J]. ACM SIGCOMM Computer Communication Review, 2014, 44(3): 87-95.

LIU Ya-qun, born in 1996, postgraduate. His main research interests include software defined network and cyberspace security.

XING Chang-you, born in 1982, Ph.D, associate professor. His main research interests include software defined network and network measurement.