

软件定义网络中控制器调度时间机制设计与实现

卢振平, 陈福才, 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: 在软件定义网络的控制平面引入多个控制器, 从而动态地改变控制器与底层网络的映射关系是一种有效缓解控制器安全威胁的方法。然而, 现有的工作缺乏研究控制器动态调度的时间问题, 由此, 在分析软件定义网络的安全现状以及动态控制器研究的基础之上, 首先说明了动态平面控制器调度时间的重要性, 并在先前的研究基础之上刻画了动态控制平面的安全工作流程; 进一步地, 将动态调度的时间问题建模为更新收益过程, 并提出一种对防御者最优的调度算法。实验仿真基于真实的网络攻击信息集, 对比分析表明, 提出的算法明显优于固定周期的算法和随机算法。

关键词: 软件定义网络; 网络安全; 控制器

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2018003

Design and implementation of the controller scheduling-time in SDN

LU Zhenping, CHEN Fucui, CHENG Guozhen

National Digital Switching System Engineering & Technological R&D Center, Zhengzhou, 450002, China

Abstract: Introducing multiple controllers to software defined network so that dynamically altering the mapping relationship between controller and the underlying network is an effective method to ease the security threats in control plane. However, little previous work has been done to investigate the economical time in dynamic-scheduling controllers. Firstly, the importance of scheduling-time in dynamic control plane based on security in SDN and dynamic control plane was introduced. Further, this problem was modeled as a renewal reward process and an optimal algorithm in deciding the right time to schedule was proposed. In our experiments, Simulations based on real network attack dataset are conducted and it demonstrate that proposed algorithm outperforms fixed-cycle algorithms and random algorithm.

Key words: software defined network, network security, controller

1 引言

软件定义网络 (SDN, software defined net-

work) 将控制与转发分离, 形成应用层、控制层、数据层的 3 层结构, 通过南向可编程接口开放了网络, 促进了网络创新, 简化了网络管理^[1]。控

收稿日期: 2017-11-25; 修回日期: 2017-12-20

通信作者: 卢振平, 13203728376@163.com

基金项目: 国家自然科学基金创新群体基金资助项目 (No.61521003); 国家重点研发计划基金资助项目 (No.2016YFB0800100, No.2016YFB0800101); 国家自然科学基金青年基金资助项目 (No.61309020, No.61602509)

Foundation Items: The Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No.61521003), The National Key R&D Program of China (No.2016YFB0800100, No.2016YFB0800101), The National Natural Science Foundation of China (No.61309020, No.61602509)

制器作为其核心组件，掌握着整个网络的视图，将上层决策翻译为数据面的转发规则，控制器这种天生的特性意味着一旦被攻击，可导致信息泄露，甚至瘫痪整个 SDN 网络^[2]。

大量的研究和实践表明，传统静态单一的控制器在面对网络攻击时显得十分被动。首先，攻击可能来自上层应用或者管理员。例如，一个简单的 exit 命令可致使 Floodlight 退出，造成整个网络瘫痪^[3]。其次，攻击可能来自底层数据面。当交换机收到一个没有匹配项的数据分组时，交换机就会把这个数据分组发送给控制器征询意见。攻击方利用这个缺陷向 SDN 交换机发送数据，从而获取控制器的关键信息。例如，攻击方通过南向接口发送探测报文、探测控制器的状态和身份信息，然后利用其掌握的控制器漏洞，进一步发动劫持、信息篡改、甚至 DoS 等攻击^[4,5]。总之，由于目前控制器单一静态的特性，攻击方可持续对控制层进行探测和尝试，总能找到弱点实施攻击。而且一旦攻击成功，由于集中控制的特性，其所造成的危害将比传统网络更为严重。

为了提高控制层的防御能力，动态地映射底层网络要素到多个控制器是一种有效的防御手段。在主流的分布式控制平面，如 ONOS^[6]和 ODL^[7]中，当一个组件或实例失败，且有其他剩余实例，允许重新分配，保障防御者仍能继续工作。然而，这些分布式控制器实例的同构性可能存在共有的安全漏洞，一旦被攻击方探测利用便可以发动持续攻击，劫持控制平面，进而破坏整个网络。除了基于部署同构多个控制器的方法外，一些学者提出基于拜占庭容错机制的控制器安全解决方案^[8,9]。每个底层交换机由多个控制器控

制，控制器数量根据不同的容错需求变化底层交换机与控制器的映射关系。这种安全方案有效地防御了控制器的拜占庭攻击以及控制器和交换机通信之间的攻击，避免了单点失效，保证了网络的正常运行。拜占庭式的容错机制的控制平面强调可用性，其并没有根本地解决控制器劫持等攻击。文献[10]通过在数据中心之中动态地分配控制器以达到最佳网络性能的目标，虽然动态分配提高了控制层的安全性，但是并没有对安全进行评估。随后的一些研究基于拟态防御的思想防御流表篡改以及控制器中的漏洞和后门威胁为基础^[11~13]。文献[13]在控制层选调不同的控制器执行体、输出结果，并通过判决器进行大数判决，从而确保即便一个控制器流表篡改之后底层网络依然可以保证正确的应用服务需求。文献[14]首次提出一种部署了动态多样冗余的安全控制层，通过动态地选调其中一个作为主控制器管理网络，从而缓和目前商业控制器以及开源控制器不断曝出的安全漏洞以及未知的漏洞和植入的后门等。

攻击方成功攻击一般经过 3 个阶段：探测阶段、组织阶段、实施阶段^[15]。控制器与底层交换机的动态映射阻碍了攻击链实施，增加了远程攻击控制器的难度。在 SDN 中，何时选择调度控制器对网络的性能和安全至关重要。图 1 说明了 3 种不同调度情景的攻防过程。

1) 不调度（静态）。由于静态特性，攻击方在无限时间的条件下，不断对控制平面进行分析探测，最终成功发起攻击；2) 傻瓜式调度。能够阻碍攻击阶段的实施，可是并没有考虑调度导致的开销以及攻击方的能力和攻击频率等实际因



图1 动态控制层防御中3种调度时间确定

素, 如果调度开销过大, 防御系统得不偿失;

3) 智能调度。在攻击组织阶段或者在攻击阶段之前调度控制器, 既降低防御开销, 又阻碍了攻击。总之, 在动态调度控制器中, 频繁地重新映射控制器与交换机的连接会导致 SDN 网络性能的下降, 而周期过长的调度控制器则无法有效地应对潜在的攻击。因此, 本文所研究的问题即是如何确定最佳的调度时间, 在保证控制层安全的前提下, 同时又降低调度引起的服务质量缺失等代价。

已有的动态性的研究大多是侧重于控制器与底层网络的策略问题, 没有研究者研究控制器动态调度的时间问题。为了弥补控制平面控制器的调度时间问题研究, 本文基于先前的研究工作^[15], 首先对控制层的威胁模型进行分析, 刻画一种闭环的控制器调度安全工作流程, 进一步将动态调度的时间问题建模为随机理论中的更新过程, 提出了一种最优调度算法确定控制器调度时间。实验分析以实际攻击数据作为仿真样本, 分别计算比较了 5 种固定周期以及随机算法的结果, 并与本文提出的算法进行了比较分析。结果表明, 本文提出的最优的调度算法明显优于上述 6 种算法。

2 威胁模型

1) 防御者

防御者旨在保护控制器配置参数的机密性、完整性和可用性^[16]。

定义 1 关键信息参数 (CIP, critical information)。为了增加攻击成功率, 攻击方在发动攻击之前一般需要探测阶段。攻击方发动攻击探测所需要的网络配置称为关键信息参数, 包括学习网络中控制器的 IP 地址、负载情况、控制器种类等配置信息。

攻击方通过探测阶段获取关键信息参数之后, 针对性地组织攻击。例如, Tming 攻击发送探测和测试两种包, 统计 RTT 分布, 推断控制器的负载状况, 进而有针对性地进行 DDos 攻击^[4]; 而如果攻击方侦测到控制器的 IP 地址, 则为建立控制器和底层网络会话劫持攻击提供条件^[17]。与攻击方相反, 防御者的目标是尽可能减少攻击方探测周期内所获的信息参数量, 以降低攻击成功

的概率。动态调度多样的控制器可以有效地阻碍 SDN 中典型的一些安全威胁。例如, 改变控制器的 IP 地址迫使攻击方已建立的会话劫持失效, 攻击方再次发动攻击时需要重新学习新控制器的 IP 地址。

定义 2 调度代价。控制层调度多样的控制器造成网络系统性能和服务质量暂时性下降, 将这种负面影响称为调度代价。

控制层的调度机制将会改变控制器的 IP, 新的控制器需要重新与底层交换机建立映射关系, 包括学习端口信息、主机位置、维持网络视图的一致性。而且根据应用层的服务需求, 可能需要对路由进行二次更新或者某些中间设施 (防火墙) 的重新布置。总之, 调度会对网络服务造成影响, 影响服务质量。

2) 攻击方

攻击方与防御者针锋相对, 通过探测目标的关键信息参数, 进而破坏或者控制目标。依据攻击链理论, 一般地, 攻击方为了成功发起一次攻击, 第一步通常学习防御者的关键信息参数, 即在探测阶段, 攻击方付出时间资源, 计算资源和财力资源等相关代价。同时, 一旦攻击方成功攻击, 便会获得相应的攻击收益, 相应地, 防御者蒙受对应的攻击损耗。攻击损耗与获取关键信息参数的信息量以及攻击类型密切相关^[18]。控制器所控制的交换机和主机数量越多, 控制器被攻击之后, 目标的损失也就越大。

3) 目标

服务提供商需要为用户保证一定的安全性, 但是也必须考虑在动态调度提供安全的基础上如何降到本身的代价。本文探讨的动态调度控制器的攻防过程中, 何时调度使防御者既能达到防御目标又可以降低调度代价是本文研究的目标。

3 安全防御机制和算法

本节首先描述一种动态调度控制器的防御工作流程, 然后提出一种调度算法决定最佳的时间调度控制器。

3.1 动态控制层防御流程

动态控制层的防御机制如图 2 所示。

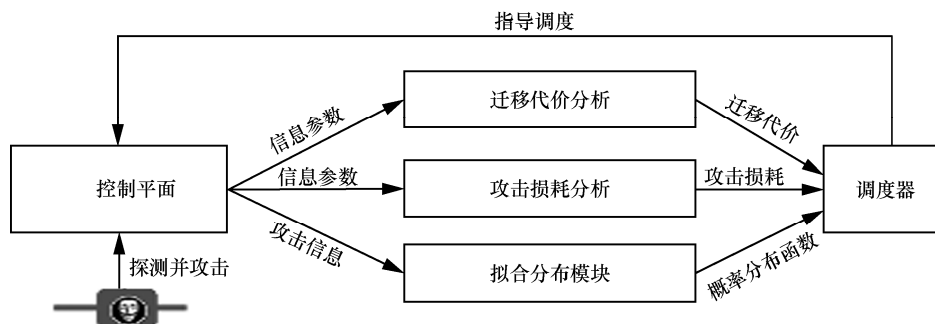


图 2 动态控制层的防御机制

控制平面：控制平面是防御者，攻击方的目标是探测并对其实施攻击。为了降低攻击成功概率增加攻击概率，控制平面根据调度器的指导在异构控制器执行体之间实施切换。

调度代价分析：基于控制层的关键信息衡量计算切换控制器的开销值。

攻击损耗分析：基于防御者的关键信息参数以及攻击类型，评估攻击成功对网络造成的危害。

拟合分布模块：统计历史的攻击时刻，拟合攻击间隔满足的分布函数，为调度提供依据。攻击间隔指的是 2 次成功攻击之间的时差。

调度模块：根据输入的调度代价、攻击损耗，以及攻击分布函数决定下次切换时间。调度过程是一种典型的随机过程，可以刻画为一个更新收益过程。3.2 节基于更新收益理论求解了最佳的调度时间。

动态控制层作为缓和网络威胁的一种防御手段，主要思想就是动态地改变控制层的异构控制器，增加了攻击方探测目标控制器的难度。即使攻击方已经掌握此时控制网络控制器的关键敏感信息，动态调度之后，攻击方需要重新搜集切换之后控制器的信息，缓和了潜在攻击造成的威胁。

3.2 调度时间算法

本文所用到的符号描述如表 1 所示。

本节基于更新收益理论设计了一种调度算法，防御者依据该算法确定最佳的调度时间。算法 1 步骤如下。

算法 1 最优调度算法

输入 攻击间隔时间序列，调度开销，攻击损耗

输出 最佳的调度时间

表 1 符号定义

符号	描述
Ψ	敏感信息参数集合
C_s	调度开销
A_{loss}	攻击损耗
S_{list}	调度时刻序列
t_{past}	距上次调度已过去的时间
T	计算的下次调度的时间
X_i	代表第 i 次调度和第 $i-1$ 次调度之间的的时间间隔
Y_i	第 i 次调度之后，攻击方攻击成功需要的探测时间
$f(x)$	攻击历史的概率密度函数
R_i	X_i 期间，防御者代价

- 1) While true do;
- 2) 依据控制层的敏感信息参数评估 C_s , A_{loss}
- 3) 依据攻击历史等计算决定调度的时间 T ;
- 4) $t_{past} = 0$;
- 5) While $t_{past} < T$ do;
- 6) If 感知被攻击 then
- 7) Break;
- 8) Else
- 9) 更新 t_{past}
- 10) End if
- 11) End while
- 12) 启动调度机制，调度控制器
- 13) 更新调度时间序列: $S_{hist} = S_{list} \cup \min\{T, t_{past}\}$
- 14) End while

在攻防过程中，防御者对控制器异构执行体以及网络具有完全信息，进而评估出调度代价和被攻击之后的攻击损耗，并依据调度时间算法（如算法 2 所示）计算最佳的调度时间。并且，如果

在计算调度时间之前，防御者在感知被攻击或者曝出新的安全漏洞时，则必须立即调度控制器，并将此时距上次调度的时间写入调度时间序列。算法 1 的步骤 6) 中防御者可以通过入侵检测机制^[19]判断是否受攻击，以及漏洞发现机制检测 SDN 中新型漏洞^[20]。

算法 2 调度时间确定算法

输入 S_{hist}

输出 T

- 1) 统计信息 S_{hist} 序列，拟合间隔分布 $F(t)$;
- 2) If 拟合不出任何封闭形式的分布 then
- 3) T 为序列的平均值;
- 4) Else
- 5) 通过公式计算 T
- 6) End if
- 7) 返回 T

算法 2 描述了如何确定最佳的调度时间。步骤 5 基于更新收益理论计算最佳的调度时间，最小化每次调度的单位代价，详述如下。

定理 1 对于探测扫描的攻击，该类的攻击时间间隔满足独立同分布，并且存在一个最优算法决定最优的调度时间使防御代价最小。

SDN 中的侧信道攻击、IP 地址扫描以及隐秘端口扫描等安全威胁可以归为一类攻击^[21]。

证明 动态调度控制器在时间轴上顺序产生，一次调度之后也是等待下一次调度的重新开始，是一种典型的随机过程。基于以上分析，控制器调度时间问题可以建模为随机过程中的更新过程，目标为求解对防御者最佳的重新开始（调度）时间，具体分析如下。

对于防御者 d ， X_i 代表第 $i-1$ 次调度和第 i 次调度之间的时间间隔， Y_i 代表第 i 次调度之后攻击方为了攻击成功需要的探测时间。假定对防御者 d 的 Y_i 存在一个分布 F_d ，其概率分布函数为 $f(x)$ ：表示攻击方花费 x 时长之后，成功探测到脆弱点的可能性。 $f(x)$ 一般根据历史攻击获得，累积分布函数为

$$F(x) = \int_0^x f(x) dx \quad (1)$$

T_i^d 为计算的防御者第 i 次调度的时间间隔。若防御者未被攻击的话，则按照此时间调度。然

而，若在 T_i^d 之前受到攻击，则立即调度。故在第 i 次调度的时间应为

$$X_i = \begin{cases} T_i^d, & Y_i > T_i^d \\ Y_i, & Y_i < T_i^d \end{cases} \quad (2)$$

若第二个不等式成立表示防御者在未调度之前遭受攻击，防御者蒙受一个攻击损耗 A_{loss} 。一旦防御者被攻击，防御者会立即调度，支付调度代价 C_s ，所以，防御者在第 i 次调度时的代价为

$$R_i = \begin{cases} C_s^d, & Y_i > T_i^d \\ A_{loss}^d + C_s^d, & Y_i < T_i^d \end{cases} \quad (3)$$

$E(X_i)$ 和 $E(R_i)$ 分别定义为相应期望值，表示第 i 次调度的总代价以及所用时间。

$$\begin{aligned} E(X_i) &= \int_0^{T_i^d} xf(x) dx + \int_{T_i^d}^{\infty} T_i^d f(x) dx \\ &= \int_0^{T_i^d} xf(x) dx + T_i^d (1 - F(T_i^d)) \end{aligned} \quad (4)$$

$$\begin{aligned} E(R_i) &= C_s^d \Pr(Y_i > T_i^d) + (A_{loss}^d + C_s^d) \cdot \\ &\Pr(Y_i \leq T_i^d) = C_s^d + A_{loss}^d F(T_i^d) \end{aligned} \quad (5)$$

定义 3 单位代价。第 $i-1$ 次和第 i 次的调度间隔中，代价与时间长度的比值。这个值衡量了单位时间内调度的性价比。

$$\gamma(T_i^d) = \frac{E(R_i)}{E(X_i)} = \frac{C_s^d + A_{loss}^d F(T_i^d)}{\int_0^{T_i^d} xf(x) dx + T_i^d (1 - F(T_i^d))} \quad (6)$$

$\gamma(T_i^d)$ 定义为时间阈值为 T_i^d 下的单位时间代价，本次调度的目标是寻找 T_i^d 最小化 $\gamma(T_i^d)$ ，即

$$\min \gamma(T_i^d) \Rightarrow \frac{\partial \gamma(T_i^d)}{\partial T_i^d} = 0 \quad (7)$$

求得的 T_i^d 为第 i 次调度的最佳时间。

4 仿真实验

本节分析评估第 3 节确定最佳时刻的调度算法，实验在配 Intel(R) Core i7-4790 CPU 3.6 GHz，16 GB RAM 的主机上进行，编程基于 Mathematica 平台实现。

4.1 实验设置

仿真以第三方提供的真实网络攻击数据作为输入^[22]，包括攻击方的标识，防御者 IP 地址和攻击时间戳。每组数据认为是一次攻防对抗的数据，由于这是一个长期的过程，为了得到

合理可信的结果, 本节选取攻击次数超过 30 次的攻击数据集作为仿真样本数据输入, 共 151 组攻击样本。

1) 攻击数据样本

利用 SPSS 软件通过柯尔莫可洛夫-斯米洛夫检验 151 组攻击样本的时间间隔拟合为正态分布、指数分布、均匀分布和泊松分布的拟合优度, 结果如表 2 所示。

表 2 不同显著性水平下的拟合结果	
显著性水平	结果
0.05	108 组为正态分布, 其余 43 组不属于任何分布
0.1	几乎全部为正态部分

表 2 的结果也证实了定理 1 中的攻击时间间隔分布符合独立同分布规律的合理性。

2) 攻击损耗

现有的中大型商业网络, 一般设有特定的安全机构进行攻击分析和代价评估。而一些小型组织也可以应用目前开发的一些工具计算攻击损失。衡量攻击代价是一个比较复杂的研究过程。本节在仿真中, 认为防御者遵循文献[23]设置恢复时间为 5 min 的攻击损耗分布。

3) 调度开销

在调度前后, 控制器需要维持网络状态的一致性, 在学习的过程中会产生一定的时延。不同的商业组织产生的时延以及对服务质量的影响是不同的。时延通常随着处理包的数量线性增长^[24]。为了方便研究, 设定了 3 种 SDN 情景: 小型商业网络时延 0.1 s, 中型商业的时延 1 s, 大型商业网络的时延 10 s, 调度开销分别为 2、20、200。

4) 攻击数据样本局限性

从攻击数据样本并不能判断样本数据集中的防御者是否部署动态机制, 而调度算法是在动态的条件下, 为了确定最佳的调度时刻而提出的。考虑这种局限性, 对数据样本做两点考虑。

情形 1 在样本攻击数据集中, 防御者采取了动态调度。

这种情况下, 直接输入原始攻击数据样本模拟攻击。

情形 2 在样本攻击数据集中, 防御者均未做动态调度。

该情形下, 需要考虑动态调度控制器的影响。

定义 4 效能系数。由于动态的效果, 在调度前后, 攻击方不得不重启一个攻击链, 重新搜集调度之后的关键敏感信息进而发动对控制器的攻击。故而, 动态调度会产生一种攻击到达的延迟效果。

$$\text{效能系数} = \frac{\text{动态调度控制器的攻击时间间隔}}{\text{静态时的攻击时间间隔}}$$

延迟效果可通过动态调度控制器的攻击时间间隔与静态时的攻击时间间隔之比刻画, 表明了动态之后的有效性。在情形 2 下可以通过设置不同的效能系数增加采集的攻击数据中 2 个成功攻击之间的间隔, 进而反映假定采取动态行为的效果。

4.2 评估方法

仿真以 151 个数据样本模拟 151 个攻防情景, 每个样本的攻击时间戳为攻击发生的时刻。仿真根据算法确定每个数据样本下的调度时间, 直至攻击数据样本在时间轴上结束。

为了比较不同方法的差异, 验证该算法的有效性, 将之与 6 种不同的调度方法比较, 包括 5 种固定周期的调度和 1 种随机调度, 具体如表 3 所示。

表 3 6 种对比的调度方法	
算法	描述
SCH_{random}	随机地从历史攻击间隔中选取一个时间
SCH_{average}	历史攻击间隔的平均值
SCH_{median}	历史攻击间隔的中值
SCH_{short}	选取历史攻击间隔最小值和中间值之间的中值
SCH_{long}	选取历史攻击间隔中值和最大值之间的中值
SCH_{longest}	选取历史攻击间隔最大的时间

通过与这 6 种具有代表性的调度方法的比较, 进而说明本文的调度算法相比于固定周期以及随机算法占优的普适性。

定义 5 长期单位代价。攻击开始和结束之间, 总代价与总运行时间之比称为长期单位代价, 刻画了各个算法的有效性。

防御者记录在不同调度方法下的调度代价和攻击损耗, 也记录了其运行时间。定义 2 的单位时间代价衡量了一次调度的效益, 长期的单位代

价则是衡量了长期的效益率。通过比较长长期单位代价衡量各个算法的性能优劣,单位代价越低,说明算法越优。

4.3 仿真结果

4.3.1 单位代价评估

在3种不同调度代价设置下,计算每个防御者在每种调度算法的单位代价,对所有代价取平均,统计结果如图3所示。

1) 从图3(a)可以看,Short算法的单位代价最高。原因是在这种情况下,控制器切换过于频繁,并没有考虑攻方的攻击能力与自身的调度代价。而且可以看出,当调度代价从2上升到200时,其单位代价比其他的算法更为敏感,因为为了避免攻击,调度的频率特别快,所以在图3(a)中平均代价增加得更多。

2) 相反地,Longest调度方法对调度代价的敏感性比较低,但是与OSA算法相比其单位代价比较高,尤其当调度代价比较低的时候,这个特点更为明显,这是因为调度间隔总是为历史攻击

间隔的最大间隔,被攻击成功的概率较大。

3) 随机算法的单位代价高于OSA以及Longest的单位代价。如果调度比较频繁,随机地调度也会引入较高的调度代价;如果调度周期较长,则被攻击的概率变大,防御者的攻击损耗相对变高。如果攻击历史并不是随机特性,随机地调度并不能有效地均衡两方面的代价。

结果分析表明,OSA算法考虑了调度代价以及攻击损耗2个方面的因素,相比于固定周期调度以及随机调度这些方法对网络服务提供商更节约成本。

4.3.2 效能系数

通过比较图3(a)和图3(b)可以发现,将效能系数设为2之后,各种算法下的单位代价大约减半。这是由于将效能系数设为2之后,攻击时间间隔增加一倍,而每一次的调度代价和攻击代价却是不变的。然而,改变攻击数据集中的时间间隔将会影响拟合之后的分布函数,防御者遭受的代价可能会受到影响,这是否影响所提出算法确

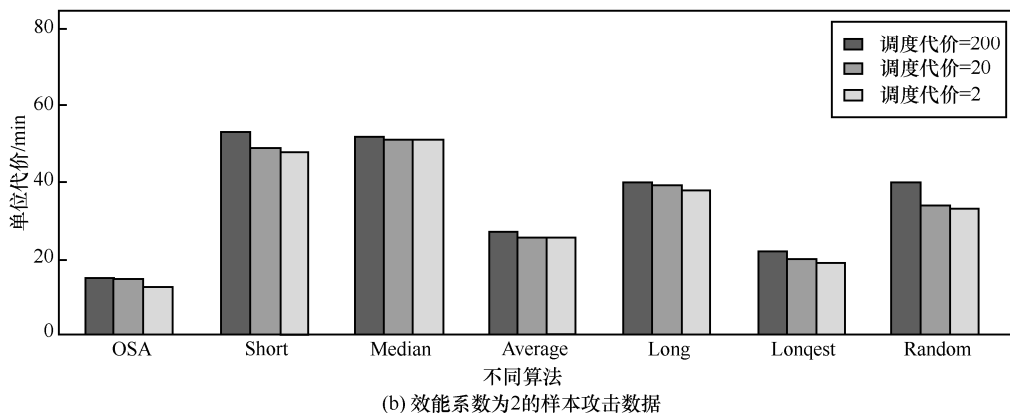
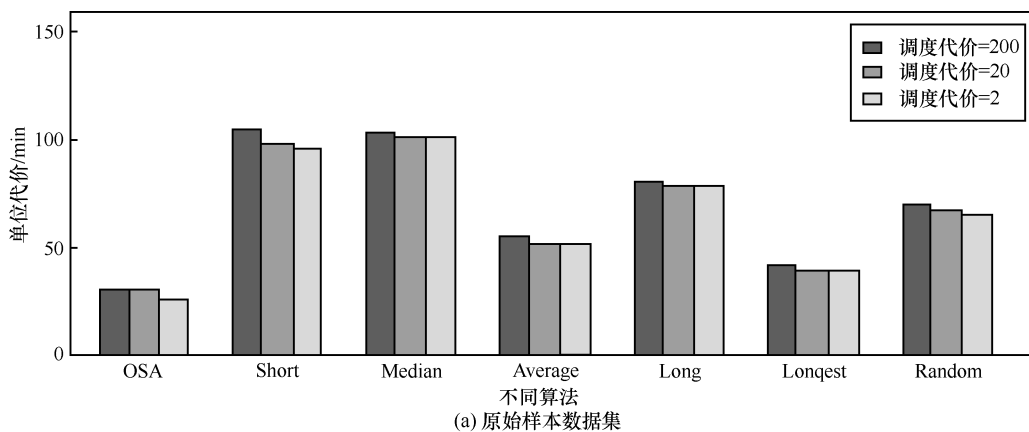


图3 各种调度方法下的所有防御者的平均单位代价

定的调度时间？基于此考虑，本节分析了单位代价和效能系数之间的关系。如图4所示。

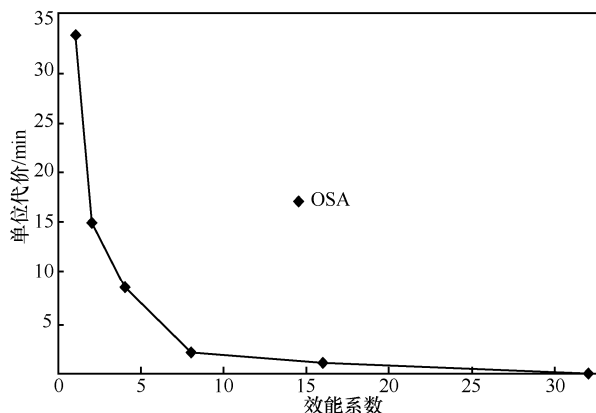


图4 调度代价设为200的单位代价与效能系数之间的关系

通过观察图4，效能系数与单位代价两者为反比例关系，也就意味着如果动态控制层的调度机制能尽可能地延迟攻击的话，则会降低防御者的单位代价。

4.3.3 调度间隔

通过记录在每个攻击数据、各个算以及不同调度开销下的调度间隔，加权平均之后，如图5所示。显而易见，当调度代价比较高时，频繁地做调度对防御者来说并不划算。图5表明，随着调度代价的增加，OSA算法的平均调度间隔趋于更大。其他6种算法没有考虑调度代价这个因素，所以并不受调度开销的影响，在整个过程中，调度的间隔几乎不变。

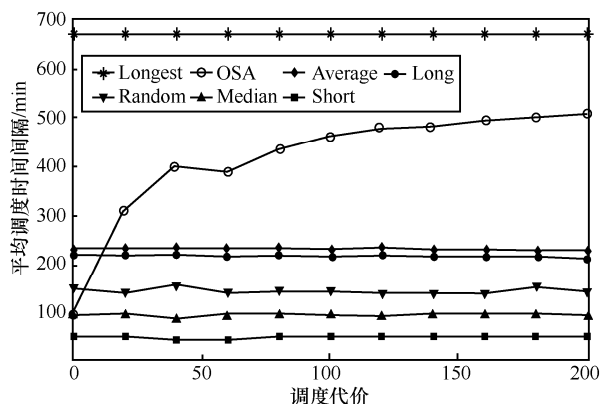


图5 原始数据集下(效能系数为1)各种调度方法的平均调度时间间隔

此外，本文提出的调度算法的平均运行时间是1.91 s，与图5中的调度时间相比几乎可以忽略不计。

5 结束语

在软件定义网络中部署动态控制器是一种有效缓和攻击层安全威胁的有效方式。本文探讨了控制器动态过程中的调度时间问题，在已有的研究基础之上，基于更新收益理论提出了一种最佳效益的调度时间算法。该算法综合权衡动态带来的开销以及攻击带来的损耗2个重要因素决定调度时刻。通过仿真评估，与传统的5种固定周期算法和一种随机的算法进行了比较，表明了其占优的一般性。

参考文献：

- [1] JAIN S, KUMAR A, MANDAL S, et al. B4: experience with a globally-deployed software defined wan[J]. Computer Communication Review, 2013, 43(4):3-14.
- [2] KREUTZ D, RAMOS F M V, VERISSIMO P. Towards secure and dependable software-defined networks[C]// ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. 2013:55-60.
- [3] LENG J, ZHOU Y, ZHANG J, et al. An inference attack model for flow table capacity and usage: exploiting the vulnerability of flow table overflow in software-defined network[J]. Water Air & Soil Pollution, 2015, 85(3):1413-1418.
- [4] SONCHACK J, DUBEY A, AVIV A J, et al. Timing-based reconnaissance and defense in software-defined networks[C]// Conference on Computer Security Applications. 2016:89-100.
- [5] CUI H, KARAME G O, KLAEDTKE F, et al. On the fingerprinting of software-defined networks[J]. IEEE Transactions on Information Forensics & Security, 2016, 11(10):2160-2173.
- [6] BERDE P, GEROLA M, HART J, et al. ONOS: towards an open, distributed SDN OS[C]// The Workshop on Hot Topics in Software Defined Networking. 2014:1-6.
- [7] MEDVED J, VARGA R, TKACIK A, et al. OpenDaylight: towards a model-driven SDN controller architecture[C]// International Symposium on a World of Wireless, Mobile and Multimedia Networks. 2014:1-6.
- [8] LI H, LI P, GUO S, et al. Byzantine-resilient secure software-defined networks with multiple controllers in cloud[C]// International Conference on Communications. 2014:695-700.
- [9] ELDEFRAWY K, KACZMAREK T. Byzantine fault tolerant software-defined networking (SDN) controllers[C]// Computer Society International Conference on Computers, Software & Applications. 2016:208-213.
- [10] WANG T, LIU F, GUO J, et al. Dynamic SDN controller assignment in data center networks: Stable matching with transfers[C]// The IEEE International Conference on Computer Communications. 2016:1-9.
- [11] QI C, WU J, HU H, et al. An intensive security architecture with multi-controller for SDN[C]// Computer Communications Workshops. 2016:401-402.

- [12] 扈红超, 陈福才, 王祺鹏. 拟态防御 DHR 模型若干问题探讨和性能评估[J]. 信息安全学报, 2016, 1(4):40-51.
HU H C, CHEN F C, WANG Z P. Discussion and performance evaluation of the proposed DHR model[J]. Journal of Cyber Security, 2016, 1(4):40-51.
- [13] QI C, WU J, HU H, et al. Dynamic-scheduling mechanism of controllers based on security policy in software-defined network[J]. Electronics Letters, 2016, 52(23):1918-1920.
- [14] LU Z P, CHEN F C. Poster: A Secure control plane with dynamic Multi-NOS for SDN [C]//Network and Distributed System Security Symposium (NDSS). 2017.
- [15] ZHUANG R, DELOACH S A, OU X. Towards a theory of moving target defense[C]// ACM Workshop on Moving Target Defense. ACM, 2014:31-40.
- [16] RUI ZHUANG, ALEXANDRU G, et al. A theory of cyber attacks a step towards analyzing mtd systems. In Proceedings of the 2nd ACM Workshop on Moving Target Defense (MTD), 2015.
- [17] BROOKS M, YANG B. A Man-in-the-Middle attack against OpenDayLight SDN control-ler[C]// ACM Conference on Research in Information Technology. ACM, 2015:45-49.
- [18] ECESSA. What does network downtime & latency cost your business [EB/OL]. <http://www.eccessa.com/blog/downtime-costs>.
- [19] ILGUN K, KEMMERER R A, PORRAS P A. State transition analysis: a rule-based intrusion detection approach[J]. IEEE Transactions on Software Engineering, 1995, 21(3):181-199.
- [20] LEE S, YOON C. DELTA: a security assessment framework for software-defined networks [C]// Network and Distributed System Security Symposium. 2017.
- [21] WANG H, LI F, CHEN S. Towards cost-effective moving target defense against DDoS and covert channel attacks[C]//ACM Workshop on Moving Target Defense. 2016:15-25.
- [22] WANG A, CHANG W, MOHAISEN A, et al. Delving into internet DDoS attacks by botnets: characterization and analysis[C]// The 45th IEEE International Conference on Dependable Systems and Networks (DSN). 2015.
- [23] MATTHEWS T. INCAPSULA survey : what DDoS attacks really cost businesses[EB/OL]. <http://lp.incapsula.com/rs/incapsulainc/images/eBook%20%20DDoS%20Impact%20Survey.pdf>.
- [24] JAFARIAN J H, AL-SHAER E, QI D. Adversary-aware IP address randomization for proactive agility against sophisticated attackers[C]//The 34th IEEE Conference on Computer Communications (INFOCOM). 2015:238-276.

[作者简介]



卢振平 (1992-), 男, 河南商丘人, 国家数字程控交换系统工程技术研究中心硕士生, 主要研究方向为软件定义网络、网络先进防御。



陈福才 (1974-), 男, 江西南昌人, 硕士, 国家数字程控交换系统工程技术研究中心研究员, 主要研究方向为电信网关防、网络安全。



程国振 (1986-), 男, 山东菏泽人, 博士, 国家数字程控交换系统工程技术研究中心助理研究员, 主要研究方向为云数据中心、软件定义网络、网络安全。