计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200900126

一种对抗网络侦察的自适应欺骗防御机制

赵金龙[1] 张国敏[1] 邢长友[1] 宋丽华[1] 宗祎本[2]
1 中国人民解放军陆军工程大学指挥控制工程学院 南京 210007
2 中国人民解放军 61789 部队 上海 200000
(zhaojl_a@163.com)

摘 要 静态配置的网络主机信息在面对攻击者侦察时易于暴露,进而带来了严重的安全隐患。主机地址跳变及部署虚假节点等欺骗方法能够扰乱攻击者对网络的认知,增加其网络侦察的难度。但如何高效地利用这些手段来对抗攻击者的侦察行为仍存在诸多困难。为此,在对攻防双方行为进行建模描述的基础上,提出了一种高效的自适应欺骗防御机制(Self-adaptive Deception Method,SADM)来应对网络侦察。SADM 结合网络侦察过程中攻防双方多阶段持续对抗的特点,以资源约束下防御方的综合收益最大化为目标进行建模,并在此基础上通过启发式方法进行自适应防御决策,以快速应对攻击者的多样化扫描行为。仿真实验结果表明,SADM 能够有效延缓攻击者的探测速度,在保证防护效果的同时降低部署欺骗场景的代价。

关键词:网络侦察;欺骗防御;扫描攻击;软件定义网络
中图法分类号 TP393

Self-adaptive Deception Defense Mechanism Against Network Reconnaissance

ZHAO Jin-long[1],ZHANG Guo-min[1],XING Chang-you[1],SONG Li-hua[1] and ZONG Yi-ben[2]
1 Command & Control Engineering College,Army Engineering University of PLA,Nanjing 210007,China
2 Unit 61789 of PLA,Shanghai 200000,China

Abstract The statically configured network host information is easy to be exposed in the face of network reconnaissance,which brings serious security risks. Deception methods such as host address mutation and deployment of fake nodes can disrupt attacker's awareness of the network and increase the difficulty of reconnaissance. However,there are still many challenges in using these methods to counter attacker's reconnaissance behavior effectively. For this reason,by modeling the behaviors of both attaker and defender,an efficient self-adaptive deception defense mechanism SADM (Self-adaptive Deception Method) is proposed. SADM considers the characteristics of the multi-stage continuous confrontation between attacker and defender in the network reconnaissance process,modeling with the goal of maximizing the defender's accumulative payoffs under cost constraints,and then makes adaptive defense decisions through heuristic methods,to respond quickly to attacker's diverse scanning behavior. The simulation experiment results show that SADM can effectively delay the attacker's detection speed and reduce the cost of deploying deception scenarios while ensuring the defense effect.

Keywords Network reconnaissance,Deception defense,Scanning attack,Software-defined network

1 引言

网络侦察是网络攻击的一个重要步骤,攻击者在发动攻击前通常需要执行网络侦察来确定可供利用的目标[1]。主机探测是网络侦察的一种常用方式,攻击者通过主机探测确定未知网络中的可访问主机及其 IP 地址,以便执行后续攻击[2],在网络中进行横向渗透。而在线主机和网络的静态特性更是简化了这种侦察,使攻击者可以通过 Nmap 等扫描工具轻易获取到目标网络中的活跃主机信息[3-4]。

为了应对这种攻防的不对称性带来的挑战,研究人员将主动防御的思想引入到对网络侦察的防护中。通过在网络中添加大量虚假节点[5-9]、快速变化节点的 IP 地址[10-14]等方式动态改变系统的侦察面,为攻击者提供一个欺骗性的网络视图,以有效地降低攻击者的侦察效率。然而,如何更高效地设置和调整欺骗视图,现有的方案缺乏有效的决策机制:要么缺乏对攻击者行为的描述,不能自适应地根据攻击者行为调整防御策略,进而导致防御效率低下[7-9,15-17];要么没有考虑防护的成本,需要消耗大量的系统资源,对网络性能造

到稿日期:2020-09-16 返修日期:2020-12-01 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61379149,61772271);国家博士后科学基金项目(2017M610286)
This work was supported by the Natural Science Foundation of China (61379149,61772271) and China Postdoctoral Science Foundation (2017M610286).
通信作者:张国敏(zhang_gmwn@163.com)

成了较大的影响[13-14,18]。

针对上述问题，本文提出了一种高效的自适应欺骗防御机制 SADM 来应对主机探测。首先，为了应对不同类型的攻击者，我们提出了更精确的模型来描述高级攻击者的行为，进而根据攻击者的攻击策略自适应地提供最佳应对方案。为了提高探测效率，攻击者往往遵循一定的策略来探测地址空间[11,19]。因此，SADM 通过分析攻击者的历史探测轨迹建立其行为模型，并预测其下一步探测的区间，然后将高概率地址区间中的真实主机迁移到低概率区间，并在其中设置大量蜜罐等虚假节点，来吸引攻击者的注意力，从而在提高防护效果的同时减少变换欺骗视图的次数，降低整体所需的防御成本。为了方便叙述，后文用节点统一表示虚假节点和真实主机。其次，为了获取一次对抗中防御者的最优策略，最大化整个对抗过程中防御者的收益并降低防御成本，SADM 将欺骗视图设置问题建模为在性能和代价约束下的优化问题，并给出了启发式算法，来指导防御者每一步的动作。最后，我们在 SDN 环境下实现了 SADM 机制，实验结果证明了该方法的高效性和有效性。与相同条件下的随机变化方式相比，SADM 可以在将攻击者的探测速率减缓至不到其原来的 1/4 的同时，获取到更高的防护收益。

2 相关工作

根据使用的方法，现有对抗攻击者进行网络侦察的主动防御方案大致可以分为两类。

(1)基于虚假视图。虚假视图主要利用大量轻量级虚拟节点或蜜罐来混淆真实的网络拓扑，将真实主机隐藏在大量虚假节点中，并通过控制流量使节点只能获取虚假的网络拓扑视图。Achleitner 等[7]提出的分布式侦察欺骗系统(RDS)通过为不同节点分配不同的虚拟视图来欺骗恶意主机，限制攻击者获取到的真实网络的细节。在此基础上，XU 等[5]提出将真实网络拓扑按照物理结构划分为不同的分组，分别实施不同的欺骗策略。Kelly 等[6]利用遗传算法分析了防御方的最优虚假视图配置策略。Robertson 等[8]提出了一种定制信息网络 CINDAM，为网络上的每个主机创建一个临时的虚拟网络视图，从而将网络的恒定拓扑结构转换为欺骗性的、可变的和个性化的拓扑结构。类似的方案还有ACyDS[4]，CyberChaff[9]。然而，较少的虚假节点数量使攻击者有足够的时间去排查信息的真伪，而较多的虚假节点又会带来高昂的维护成本，静态的虚拟视图同样不能应对长期潜伏收集信息的攻击者。

(2)IP 地址跳变。IP 地址跳变技术通过周期性地改变主机的 IP 地址来躲避攻击者的探测。网络地址空间随机化(NASR)[16]通过修改动态主机配置协议(DHCP)服务器，来频繁地改变系统 IP 地址。自屏蔽动态网络结构(SDNA)[17]通过重写进入和离开操作系统的数据包，来防止攻击者从主机观察到网络中的真实地址。OF-RHM[10]，SDN Shuffle[15]将真实地址动态地映射到一个虚假地址。Jafarian 等[13]提出了一种自适应的地址随机化技术，通过预测攻击者的探测模式来提高地址随机化的有效性。Wang 等[20]量化地分析了不同攻击场景下的不同地址变换周期的防护效果。较长的变化周期让攻击者足以完成扫描，而频繁的变换将带来巨大的性能损耗，此外还需要复杂的机制来维持正常连接不被中断。

和本文内容类似的研究有文献[14,18]，它们都提出结合虚假节点和 IP 地址跳变技术的欺骗防御方式。通过预测攻击者行为以及动态变换网络中真假节点的分布，来提高攻击者探测到虚假节点的概率，从而降低探测到真实主机的概率。然而，文献[18]只关注了变换方案的有效性而没有考虑代价，文献[14]则重点考虑的是小规模网络下变换方案对已建立连接的影响。

3 威胁模型

3.1 攻击者行为建模

攻击者在渗透到网络后，通常需要扫描特定的地址空间来探测其他节点，以收集网络中活跃主机的信息。为了探测的隐蔽性和高效性，攻击者会采取一些高效隐蔽的策略，常用的探测策略包含 3 个要素[7]，即扫描空间 Ω、扫描速度 v 和跳转间隔 δ。扫描空间表示要探测的 IP 地址空间，通常根据要探测的 IP 地址前缀来选择。跳转间隔表示每次探测选取的下一个地址和其所在节点地址的间隔。而根据攻击者每次的跳转间隔，可以将整个扫描空间划分为不同的区间，即 $\Omega=\Omega_1\cup\Omega_2\cup\cdots\cup\Omega_k$，攻击者在一定的时间内探测各个区间的概率不同。用 t 表示攻击步骤，则攻击者下一次选择的探测目标可以表示为攻击者在第 $t+1$ 次探测时选择区间 Ω_i 中的地址的概率 $p_{t+1}(\Omega_i)$，其中：

$$\sum_{i=1}^{k} p_{t+1}(\Omega_i)=1,\Omega_i\subseteq\Omega \tag{1}$$

攻击者的每次探测都会耗费一定的成本，并且会增加暴露的风险。因此，为了提升探测效率，高级的攻击者会使用一些高效的智能化方式，根据探测结果动态自适应地更新探测不同区间的概率[18]，如在发现活跃主机后，逐渐增加活跃主机所在区间的探测概率。同时，为了逃避检测，攻击者可能会在感染其他节点后改变发起探测的位置等。

3.2 防御者行为建模

为了防止攻击者获取网络中活跃主机的信息，防御者通过设置蜜罐等虚假节点并动态改变网络中主机分布的方式来为攻击者提供一个欺骗视图，从而躲避探测。为了避免不必要的变化，防御者初始在网络中随机设置大量虚假节点来监控网络，如文献[5-9]所述。当有节点连接到这些虚假节点后，认为其为潜在的恶意节点，然后根据攻击者行为动态调整欺骗视图，当观察到同一节点一定次数的探测后，确认该节点为恶意节点，并将其隔离出网络。观察到潜在的恶意节点后，防御者希望尽快确认攻击节点，以避免暴露更多的真实节点。同时，考虑到攻击者可能改变发起探测的位置，因此很有必要尽快确认攻击者。

防御者拥有网络的全局视图，在 SDN 网络中可以通过流表统计信息获取攻击者的历史探测动作 A_t 构成的轨迹序列 $\{A_0,A_1\cdots,A_t\}$。为了使每次变化带来的收益最大，防御者根据攻击者的历史轨迹来预测攻击者的探测区间以及区间上的概率分布，计算最佳的方案来变换主机地址分布，以降低真实主机被发现的概率，提高虚假节点的命中率，如图 1 所示。因

为通过虚假节点可以监测攻击者具体的攻击动作，所以可以获取到攻击者的详细攻击行为，提供更全面的威胁安全情报，并降低误报率。具体主机地址变换的实现方法如文献[10]所述。这里我们假设攻击者无法区分网络中节点是真实主机还是虚假节点。

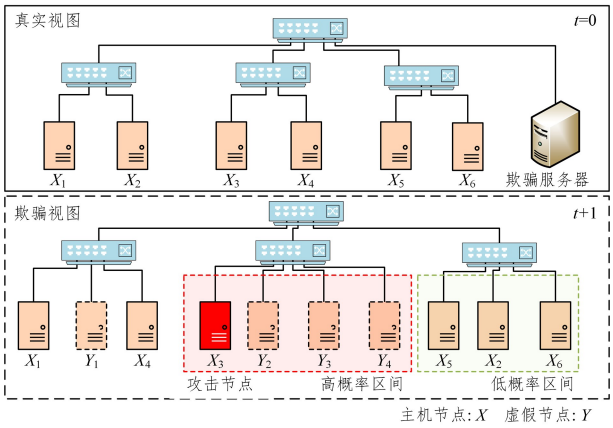


图 1 欺骗视图设置和调整方式

Fig. 1 Way to set up and adjust deception views

攻击者探测一个区间的概率可以通过其探测这个区间的历史概率 $\hat{p}_{t+1}(\Omega_i)=r_{it}/r_t$ 来预测[13]，r_t 表示总的历史探测次数，r_{it} 表示探测区间 Ω_i 的次数，$\hat{p}_{t+1}(\Omega_i)$ 表示预测值。同时，假设攻击者会通过当前区间的探测情况自适应地更新探测概率，因此，我们用探测情况的加权值来修正历史概率 $\hat{p}_{t+1}(\Omega_i)$：

$$\hat{p}_{t+1}(\Omega_i)=\begin{cases} w_t S_t(\Omega_i)+(1-w_t)\dfrac{r_{it}}{r_t}, & S_t=1 \\ \dfrac{r_{it}}{r_t}, & S_t=0 \end{cases} \tag{2}$$

其中，$S_t=1$ 表示攻击者在 t 时刻探测到在线主机(包含真实主机和虚假节点)，$S_t=0$ 则表示未探测到。权重 $w_t\in[0,1)$ 表示攻击者在当前区间探测到在线主机时继续探测当前区间的可能性。我们取其指数移动加权平均来表示这个值：

$$w_{t+1}=\beta w_t+(1-\beta)\,p_t(\Omega_{i\to j}\,|\,S_t(\Omega_i)=1) \tag{3}$$

其中，$p_t(\Omega_{i\to j}\,|\,S_t(\Omega_i)=1)=\{0,1\}$ 表示在第 t 步探测中在区间 Ω_i 探测到活跃主机时是否跳转到另一区间 Ω_j，跳转时为 0，否则为 1。这样，如果攻击者采取自适应的方式，在探测到活跃节点后加大当前区间的概率而在连续未探测到节点时降低探测概率，或者按照固定的策略进行探测都可以通过式(2)描述其行为。

在预测到攻击者的探测概率分布后，防御者将真实主机从高概率区间转移，然后利用生成的虚拟节点来监控这些区间，引诱攻击者继续探测这些区间，以便尽快地确认恶意节点。防御者在转移节点时，需要考虑迁移的有效性 R_t 和迁移代价 C_t。有效性可以用一次变化后是否探测到真实主机或虚假节点来衡量，而代价表示每次迁移本身的代价和维护虚拟节点的代价。用 P_t 表示 t 时刻的收益函数：

$$P_t=w_1R_t-w_2C_t \tag{4}$$

其中，w_1 和 w_2 分别为防御者为有效性和代价分配的权重因子。防御者的目标是最大化累计收益：

$$f=\text{maximize}\sum_{t+1}^{T}\gamma^t P_{t+1} \tag{5}$$

其中，折扣因子 $\gamma\in[0,1]$ 表示当前步骤操作的长期影响，T 表示防护停止的阈值，即确认攻击者身份所需要的探测次数[21]。

3.3 防护场景模型

假设防御者可利用的地址空间为 Ω_m，空间大小为 m，包含 n 个子网。在一定的时间内，网络中活跃的真实主机数目为 x，并维持这个数目不变，虚假节点的数目为 y。则网络结构可以描述为两个一维向量 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$，分别表示真实主机和虚拟节点在地址空间中的坐标。$\boldsymbol{X}_t$ 和 $\boldsymbol{Y}_t$ 分别表示 t 时刻下的节点分布，则防御问题可以表示为在给定的时刻 t 和系统状态 $\boldsymbol{X}_t$，$\boldsymbol{Y}_t$ 下，防御者希望通过一个最优的变换得到下一步的系统状态 $\boldsymbol{X}_{t+1}$，$\boldsymbol{Y}_{t+1}$，使变换获得的期望收益最高。

一次变换的有效性 R_t 用变换后获取的奖励表示。一次变换后，如果下一次攻击者命中真实节点，即 $S_{t+1}(\boldsymbol{X}_{t+1})=1$，则奖励为负；命中虚假节点，即 $S_{t+1}(\boldsymbol{Y}_{t+1})=1$，则奖励为正；探测到空地址 $S_{t+1}=0$，则奖励为 0。奖励的具体值和节点本身的价值有关，用 $v(X_i)$ 表示主机 X_i 的价值，$v(Y_i)$ 表示虚假节点 Y_i 的价值，则一次变化后的奖励可以表示为：

$$\begin{aligned} R_{t+1}&=R^X_{t+1}+R^Y_{t+1} \\ &=-\sum_{i=1}^{x}S_{t+1}(X_i^{t+1})v(X_i)+\sum_{i=1}^{y}S_{t+1}(Y_i^{t+1})v(Y_i) \end{aligned} \tag{6}$$

每一次变化的代价包含迁移操作的代价和维护虚拟节点的代价。其中一次变化对真实主机的影响可以表示为：

$$C^X_{t+1}=\frac{1}{x}\sum_{i=1}^{x}(X_i^{t+1}\odot X_i^t)D(X_i^{t+1}-X_i^0) \tag{7}$$

其中，同或运算$\odot$表示在第 t 次变化中第 i 个主机 X_i 的位置是否发生了变化。而变化的具体代价和变化的方式有关，如在同一子网中迁移和跨子网迁移的代价不同。在 SDN 网络中，改变主机的 IP 地址需要修改相关转发节点上的转发规则，变化代价主要与未作变化前主机的真实分布 X^0 有关。因此，我们用变化后主机在网络拓扑中的位置到初始位置的路径上涉及的 SDN 节点数目表示，用函数 $D(X_i^{t+1}-X_i^0)$ 表示这个数目。如在一个树形网络拓扑中，如图 2 中的箭头所示，将主机 X_i^0 变换到同一网段内的 X_i^t 涉及的 SDN 节点的数目为 1，而跨网段变换到 $X_i^{t'}$ 涉及的 SDN 节点的数目为 3。

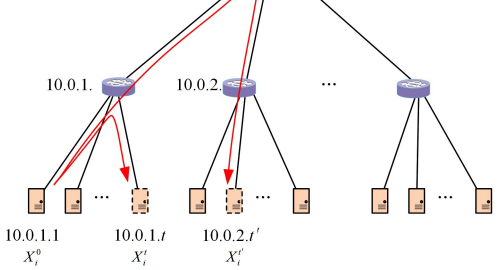


图 2 变换成本的计算示意

Fig. 2 Example of calculation of transformation costs

这里通过虚拟化技术实现虚假节点，因此变换虚假节点的代价主要与维持的虚假节点数量有关，即：

$$C^Y_{t+1}=|\boldsymbol{Y}_{t+1}|/y_0 \tag{8}$$

用 w_x 和 w_y 表示权重因子，则一次变化总的防护代价可以表示为：

$$C_{t+1}=w_xC_{t+1}^X+w_yC_{t+1}^Y \tag{9}$$

4 防护策略计算

对于多目标多阶段随机优化问题，在一次防御过程中，攻击者的可选攻击动作为 $O(m)$，m 为可用地址空间大小，防御者可选动作空间为 $O(A_m^xC_{m-x}^y)$，x 为活跃主机的数目，y 为虚假节点的数目，则最优解的计算复杂度是 $O(m\times A_m^xC_{m-x}^y)$。而实际中防御者需要匹配攻击者的扫描速度来做出应对，需要一种快速的计算方式，因此我们提出一种经济高效的启发式算法(SADM)来解决这个问题。

当攻击者完成第 t 次探测后，防御者可以由式(2)预测得到攻击者 $t+1$ 次的探测概率分布 $\hat{p}_{t+1}$，然后实施变换策略，代价由式(9)决定，奖励值由式(6)的期望值确定：

$$\begin{aligned}R_{t+1}&=E(-\sum_{i=1}^{x}S_{t+1}(X_i^{t+1})v(X_i)+\sum_{i=1}^{x}S_{t+1}(Y_i^{t+1})v(Y_i))\\&=-\sum_{k=1}^{n}\left(\frac{\hat{p}_{t+1}(\Omega_k)}{|\Omega_k|}\sum_{X_i^{t+1}\in\Omega_k}v(X_i)\right)+\\&\quad\sum_{k=1}^{n}\left(\frac{\hat{p}_{t+1}(\Omega_k)}{|\Omega_k|}\sum_{Y_i^{t+1}\in\Omega_k}v(Y_i)\right)\end{aligned} \tag{10}$$

然而，攻击者探测不同区间的长期概率分布是不确定的，因此无法获取一次防御动作的长期奖励的期望。但通常攻击者遵循一定的策略进行探测，其探测不同区间的概率分布在一段时间内近似不变，因此可以通过 $t+1$ 步的预测值来近似 $t+1$ 之后的概率分布。则当 $\boldsymbol{X}_{t+1}^*,\boldsymbol{Y}_{t+1}^*=\underset{\boldsymbol{X},\boldsymbol{Y}}{\operatorname{argmax}}(R_{t+1})\ \vee\ \hat{p}_{t+l}(\Omega_i)=\hat{p}_{t+1}(\Omega_i)$，$l>1$ 时，$\boldsymbol{X}_{t+1}^*,\boldsymbol{Y}_{t+1}^*=\underset{\boldsymbol{X},\boldsymbol{Y}}{\operatorname{argmax}}(R_{t+l})$，$l>1$。即如果当前步骤的期望奖励在系统状态 $\boldsymbol{X}_{t+1}^*,\boldsymbol{Y}_{t+1}^*$ 下取得最大值，则未来($l>2$)的期望奖励的最大值也在系统状态 $\boldsymbol{X}_{t+1}^*,\boldsymbol{Y}_{t+1}^*$ 下取得。

另一方面，为了匹配攻击者的扫描速率，并减小频繁变换对合法用户的影响，一个好的策略是一次动作的有效性能够保持较长时间，即防御者完成当前动作后，在随后的一段时间内保持当前的动作也可以获取最大的收益。因此，我们可以用单步收益的最大值来近似累计收益的最大值，即考虑求解如下目标函数：

$$\underset{\boldsymbol{X},\boldsymbol{Y}}{\operatorname{argmax}}(w_1R_{t+1}-w_2C_{t+1}) \tag{11}$$

算法 1 给出了上述模型的启发式求解算法。防御者的目标函数可以分为两部分，有关真实主机部分 $P_{t+1}^X=-w_1R_{t+1}^X-w_2w_xC_{t+1}^X$ 和虚假节点部分 $P_{t+1}^Y=w_1R_{t+1}^Y-w_2w_yC_{t+1}^Y$。防御者需要寻找一种成本最低的策略来将高概率区间的真实主机转移到低概率区间，同时在不同的区间设置虚假节点，以最大化期望收益。为了提高变换收益，从概率最高的区间开始，对于每个概率大于阈值 τ 的区间(见算法 1 中的第 3 行)，计算将其中的主机 x 迁移到其他概率更小的区间时 P_{t+1}^X 的值，使 P_{t+1}^X 取得最大值，然后更新 $\boldsymbol{X}_{t+1}$(见算法 1 中的第 4,5 行)。然后计算在该区间设置的虚假节点的数量，由 P_{t+1}^Y 可知，当满足 $P_{t+1}^Y>0$ 时，虚假节点数量越多，期望收益就越大，因此我们这里设置一个最高阈值 $|\boldsymbol{Y}|_\tau$(见算法 1 中的第 6,7 行)，在实验中我们将其设置为真实主机数目的两倍。然后，依次对下一个概率次高的区间中的主机进行迁移，直到迁移的预期收益小于等于不迁移时停止变换(见算法 1 中的第 8,9 行)。

算法 1 自适应变换决策算法 SADM

输入：t 时刻的攻击步骤 A_t，概率分布 $p_t(\Omega_i)$，系统参数 $\mathbf{X}_t,\mathbf{Y}_t$

输出：t+1 时刻的系统参数 $\mathbf{X}_{t+1},\mathbf{Y}_{t+1}$

1. 更新不同区间的概率分布 $\hat{p}_{t+1}(\Omega_i)$
2. 按照 $\hat{p}_{t+1}(\Omega_i)$ 对 Ω_i 降序排列
3. for i=1,2,…,$|\Omega|$，$\hat{p}_{t+1}(\Omega_i)>\tau$：
4. for x in Ω_i
5. $\mathbf{X}_{t+1}\leftarrow\underset{x\in\Omega_n,n\geqslant i}{\operatorname{argmax}}(P_{t+1}^X(\Omega_n))$
6. if $|\mathbf{Y}_{t+1}(\Omega_i)|<|\mathbf{Y}|_\tau$：
7. $\mathbf{Y}_{t+1}\leftarrow\underset{Y_{t+1}\in\Omega_i}{\operatorname{argmax}}(P_{t+1}^Y(\Omega_i))$
8. if $P_{t+1}^X(\Omega_i)\leqslant P_t^X(\Omega_i)$：
9. stop

算法的计算次数取决于可用子网数量 n 及子网中的主机数 x，因此其时间复杂度为 $O(n^2x)$。但实际上，将主机迁移到概率最低的区间的预期收益往往是最高的，主机恢复真实地址的收益要比分配新的虚拟地址的收益要高，即在当前时刻下的最优迁移方式实际是固定的。此外，只需要迁移概率最高的几个区间。因此，每次迁移的搜索复杂度最优为 $O(1)$，整体复杂度则最优为 $O(x)$。这保证了本文算法的运行效率，为匹配攻击者探测速率提供了基础。

5 实验评估

我们在一台 8×Intel® Core™ i7-7700HQ@ 2.80 GHz CPU、16 GB 内存的 Ubuntu 16.04 虚拟机上，用 python 在 RYU 控制器中实现了 SADM 原型，并通过 mininet 实验平台模拟了一个 B 类网络的树形拓扑来验证其有效性。该 B 类网络被划分成 255 个 C 类子网，每个 C 类子网设置 250 个可用地址，具体如图 2 所示。实验中我们按照子网划分攻击者的探测区间，主机和虚假节点的价值统一设置为 1，权重因子 w_1,w_2,w_x,w_y 均为 0.5。

由于主机在地址空间中分布的不均匀性，像本地偏好这样的非均匀扫描策略能够大大增加探测成功的概率，且局部扫描有助于降低被安全设备发现的概率，因此实际中攻击者往往采用非均匀扫描方式[6]。鉴于此，为了评估 SADM 在不同的攻击策略下的性能，我们同样假设攻击者采用目前常用的 3 种非均匀扫描策略[12,19,22]。

(1)顺序扫描。按照顺序依次扫描整个地址空间。为了加快扫描速度，这里根据是否探测到活跃主机动态调整扫描的跳跃间隔。探测到活跃主机时降低跳跃间隔为 1，否则依次增大跳跃间隔。

(2)本地偏好扫描。利用网络中主机的本地化分布来提高传播速度。以概率 p 探测被感染主机所属的本地子网，以概率 1—p 探测整个地址空间中随机选择的地址。实验中设置攻击者探测本地子网的概率 $p=0.4$。

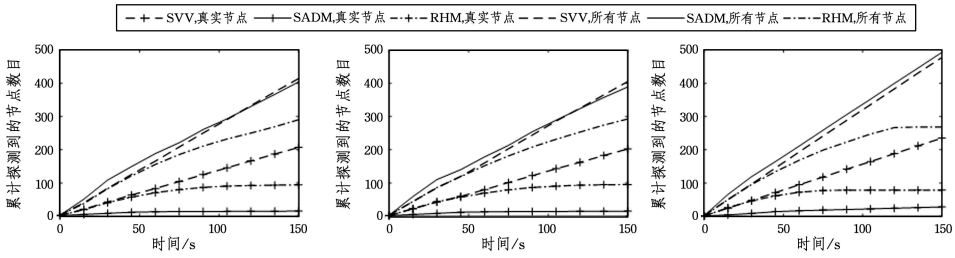
(3)自适应扫描。随机从整个地址空间中选择一个地址进行探测。但为了提高探测速度，假设攻击者会根据是否在

当前区间发现活跃主机基于指数加权移动平均来动态更新选择当前区间的概率。

扫描策略的有效性是通过降低对特定 IP 的多次扫描的概率来确定的，因此我们假设地址不会被重复扫描。攻击者扫描速率设置为每秒 10 个地址，扫描空间为整个 B 类子网。

5.1 延迟扫描

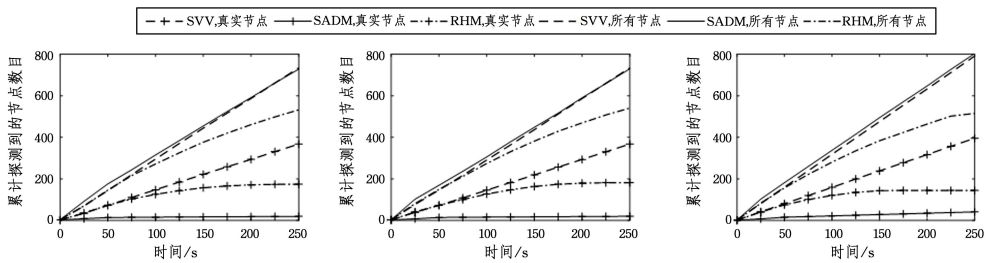
为了对比 SADM 变换算法在延迟扫描攻击方面的性能，我们在不同的扫描方式下，将本文算法分别与设置静态虚假视图[7](SVV)、随机跳变[13](RHM)两种防护方法进行了对比。静态虚假视图随机为网络中的主机分配一个虚拟地址并在子网中分配同样数量的虚假节点。SADM 方法和随机变化 RHM 在与静态虚假视图同样的初始条件下根据检测到的扫描结果动态调整主机和虚假节点的分布。为了便于比较，我们设置随机跳变的频率和 SADM 算法一致。我们针对每种策略分别进行 40 次测试，然后求取平均值，得到的结果如图 3、图 4 所示。



(a)自适应扫描 (b)本地偏好扫描 (c)顺序扫描

图 3 20 个可用子网下不同扫描策略的探测速率

Fig. 3 Detection rate of different scanning strategies within 20 available subnets



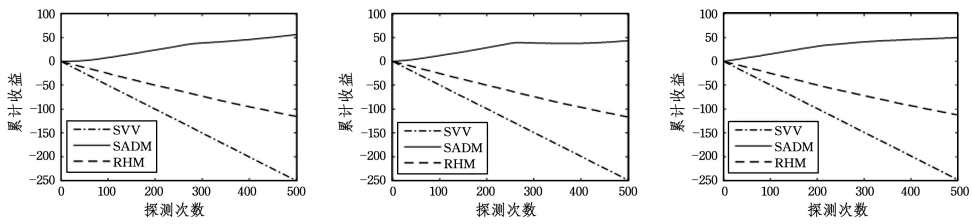
(a)自适应扫描 (b)本地偏好扫描 (c)顺序扫描

图 4 40 个可用子网下不同扫描策略的探测速率

Fig. 4 Detection rate of different scanning strategies within 40 available subnets

图 3 给出了在 20 个可用子网、每个子网 40 台主机的情况下用不同的扫描策略分别探测活跃主机的速率，图 4 给出了在 40 台可用子网、每个子网 40 台主机的情况下用不同的扫描策略分别探测活跃主机的速率。真实节点表示真实脆弱主机，所有节点表示真实主机和虚假节点之和。明显可以看到，由于扫描策略的自适应性，只设置静态虚拟视图的情况下，攻击者探测到活跃主机的数目几乎成线性增长。而随机跳变有效延缓了攻击者探测到活跃主机的速率，但同时也降低了攻击者探测虚假节点的速率。而 SADM 方法通过有效避开高探测概率的区间，将高风险区间的主机迁移到低风险区间，更高效地避免了真实主机被探测到的概率，进一步降低了攻击者的探测效率。同时，通过在高概率区间设置虚假节点，保持了与静态虚假视图同样的活跃节点探测速率，即 SADM 算法大大提高了攻击者探测到虚假节点的概率。可以看到，即使是同样的变换频率，SADM 的效果也比随机跳变方法高 4 倍以上。

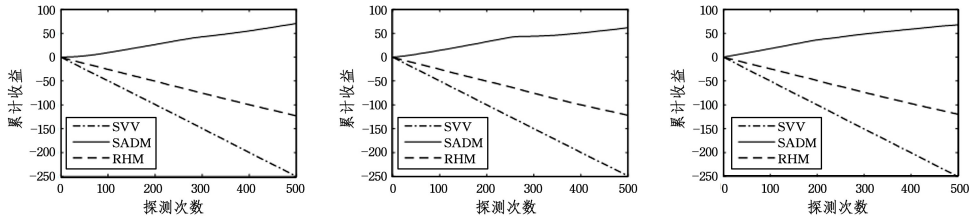
为了证明 SADM 算法的经济性，我们在同样的条件下比较了各种策略前 500 次探测的累积收益。图 5 给出了 20 个可用子网时各种策略情况下的累计收益，图 6 给出了 40 个可用子网的情况。从图中可以明显看到，设置静态虚假视图的收益函数最低，因为维持大量的虚假节点需要持续消耗代价，且其防护效果不如随机跳变和 SADM 方法。而由于每次变换主机地址都寻找代价最小的方式，因此 SADM 的代价低于随机算法，主动调整虚假节点和真实主机的分布也可以使攻击者命中更多的虚假节点，从而收益远高于随机跳变。



(a)自适应扫描 (b)本地偏好扫描 (c)顺序扫描

图 5 20 个可用子网不同扫描策略下的累积收益

Fig. 5 Cumulative payoffs of different scanning strategies within 20 available subnets



(a)自适应扫描 (b)本地偏好扫描 (c)顺序扫描

图 6 40 个可用子网下不同扫描策略下的累积收益

Fig. 6 Cumulative payoffs of different scanning strategies within 40 available subnets

5.2 运行效率

由于在一次变化中，SADM 算法需要寻找最优的变换策略，其计算复杂度和子网中的主机数有关，因此我们分析了算法在不同的主机数量下的运行效率。这里我们假设确认攻击者的阈值为 100，即从触发变化到监控到 100 次来自同一节点的探测我们便确认该节点为恶意节点，终止变化。而通常探测过程越靠前，攻击者的攻击数据越少，预测的准确性越低，需要变换的次数也越频繁，因此越是早期的数据越能代表算法的最差运行情况，越能反应算法的性能。我们计算这 100 次的平均运行时间作为算法运行一次的时间。在可用子网数为 40 的情况下，每种扫描策略进行 100 次实验，然后取平均值作为最后结果，得到的结果如图 7 所示。

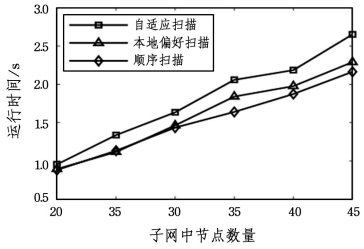


图 7 平均每次运行的时间

Fig. 7 Average running time per run

从图 7 中可以看到，在不同的扫描策略下，平均每次变化的时间随着主机数量的增长呈线性增长。这说明 SADM 算法具有很好的扩展性，计算复杂度并不会在大规模网络中快速增长，可以很好地适应不同规模的网络。此外，探测规律越明显、越能准确预测的探测方式，如顺序扫描，其对应的算法平均运行时间越短。因为攻击者的概率变化越稳定，平均每次需要变化的主机数越少，需要变化的次数也越少，则平均运行时间也越短。在每个子网中有 45 台活跃主机、总共 1 800 台主机的网络中，平均变化一次的时间为 3 ms，这样即使攻击者以极高的扫描速率(每 10 ms 探测一个地址)扫描，SADM 也可以有效地应对。

5.3 性能损耗

频繁地变换主机地址变换会给网络性能带来很大的影响，也会对合法用户建立的连接造成较大影响，因此本节评估 SADM 方法对网络性能的影响。在 SDN 环境中，制约网络性能的一个重要原因是交换机中的流表数和平均每秒更新的流表数量，而这两者都与平均每次变换中地址跳变的主机数量有关。因此，我们用在防护期间平均每次变换中需要跳变的主机数量来衡量算法对系统的损耗。同样地，我们以 100 次探测作为防护的截止条件，然后以这 100 次跳变的主机数目的平均值作为一次跳变的数量，然后测试 100 次攻击来求取平均值。最终得到的结果如图 8 所示。

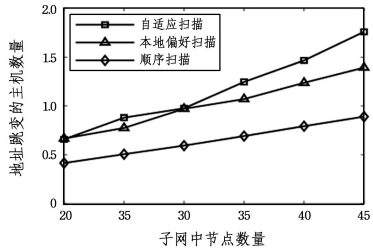


图 8 平均每次运行中地址需要跳变的主机数量

Fig. 8 Average number of mutated hosts per run

图 8 表明了不同的攻击策略下，平均每次需要跳变的主机数目也不一致。自适应扫描下需要变化的次数最多，而顺序扫描则最少。同样地，每个子网中的主机数目越多，每次变换要迁移的主机数目越多，平均每次变换的主机数就越多。可以看到，即使每个子网中包含 45 台活跃主机和相应数量的虚假节点，即网络中节点数目占整个子网地址数目的 1/3 以上时，平均每次变换的主机数目也不超过两个。即攻击者每次扫描时，平均变换两台主机的 IP 地址就可以很好地防止攻击者探测到活跃主机。也就是说，在进行一次变化后，相当长的一段时间内则不需要再做频繁的变化，对系统性能的影响则被控制在了一定的范围内。

结束语 网络侦察的目标是发现网络系统中有价值的主机信息，以便发起进一步的攻击。但现有的防护方案存在缺乏自适应性、防护效率低和防护成本高等问题。为此，我们提出了一种对抗网络侦察的自适应欺骗防御机制 SADM，SADM 对攻防双方的行为进行统一建模描述，将防御者最优策略计算问题建模为一个有效性和性能约束下的优化问题，为如何平衡侦察防护方案的效率和成本问题提供了一种求解思路，并设计实现了一种启发式求解算法，在一个 SDN 模拟环境中评估了其效果。实验结果表明，SADM 可以有效延缓攻击者的探测速度，并降低每次变化带来的代价，实现了变换成本和效率之间的有效平衡。

然而，本文存在一定的不足之处。首先，本文的防御模型只针对采用非均匀策略的攻击者，当攻击者策略的非均匀性不明显时，不能很好地预测攻击者的行为，从而导致防护的效率降低。其次，在算法的求解过程中，采用单步最优来近似累计最优，即只考虑了当前收益的最大化，这样得到的结果有可能与最优解存在较大距离。在后面的工作中，针对这类多阶段随机优化的问题，可以采用神经网络来学习攻击者模式，以提高预测攻击者行为的准确性，并结合强化学习的方法来寻

求不同攻击策略下防御者的最优应对策略，从而提高求解的精度。

参 考 文 献

[1] PANJWANI S, TAN S, JARRIN K M, et al. An experimental evaluation to determine if port scans are precursors to an attack [C]// 2005 International Conference on Dependable Systems and Networks (DSN'05). IEEE, 2005: 602-611.

[2] WANG L, WU D. Moving target defense against network reconnaissance with software defined networking[C]// International Conference on Information Security. Springer, 2016: 203-217.

[3] SOOD A K, ENBODY R J. Targeted cyberattacks: A superset of advanced persistent threats[J]. IEEE Security & Privacy, 2013, 11(1): 54-61.

[4] CHIANG C-Y J, GOTTLIEB Y M, SUGRIM S J, et al. Acyds: An adaptive cyber deception system[C]// 2016 IEEE Military Communications Conference. IEEE, 2016: 800-805.

[5] XU M, GAO Y, FENG C. Dds: A distributed deception defense system based on sdn[C]// 2018 14th International Conference on Computational Intelligence and Security (CIS). IEEE, 2018: 430-433.

[6] KELLY J, DELAUS M, HEMBERG E, et al. Adversarially adapting deceptive views and reconnaissance scans on a software defined network[C]// 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2019: 49-54.

[7] ACHLEITNER S, LA PORTA T F, MCDANIEL P, et al. Deceiving network reconnaissance using sdn-based virtual topologies[J]. Ieee Transactions on Network and Service Management, 2017, 14(4): 1098-1112.

[8] ROBERTSON S, ALEXANDER S, MICALLEF J, et al. Cindam: Customized information networks for deception and attack mitigation[C]// IEEE International Conference on Self-adaptive & Self-organizing Systems Workshops. IEEE, 2015: 114-119.

[9] Cyberchaff[EB/OL]. (2020-8-14)[2020-8-14]. https://formal.tech/cyberchaff/.

[10] JAFARIAN J H, AL-SHAER E, DUAN Q. Openflow random host mutation: Transparent moving target defense using software defined networking[C]// Proceedings of the First Workshop on Hot Topics in Software Defined Networks. ACM, 2012: 127-132.

[11] DU J, GUAN H S, JIANG B C. Defending against hitlist worms using network address space randomization[J]. Microcomputer Information, 2009(6): 85-87.

[12] JAFARIAN J H, AL-SHAER E, DUAN Q. An effective address mutation approach for disrupting reconnaissance attacks[J]. IEEE Trans Information Forensics and Security, 2015, 10(12): 2562-2577.

[13] JAFARIAN J H, AL-SHAER E, DUAN Q. Adversary-aware ip address randomization for proactive agility against sophisticated attackers[C]// 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015: 738-746.

[14] CLARK A, SUN K, POOVENDRAN R. Effectiveness of ip address randomization in decoy-based moving target defense[C]// Decision & Control. IEEE, 2013: 678-685.

[15] MACFARLAND D C, SHUE C A. The sdn shuffle: Creating a moving-target defense using host-based software-defined networking[C]// Proceedings of the Second ACM Workshop on Moving Target Defense. ACM, 2015: 37-41.

[16] 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.

[17] YACKOSKI J, XIE P, BULLEN H, et al. A self-shielding dynamic network architecture[C]// Military Communications Conference. IEEE, 2011: 1381-1386.

[18] XING J, YANG M, ZHOU H, et al. Hiding and trapping: A deceptive approach for defending against network reconnaissance with software-defined network[C]// 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), London, United Kingdom. IEEE, 2019: 1-8.

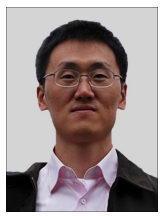
[19] ZOU C C, TOWSLEY D, GONG W B. On the performance of internet worm scanning strategies[J]. Performance Evaluation, 2006, 63(7): 700-723.

[20] WANG S, ZHOU Y, LI Y, et al. Quantitative analysis of network address randomization's security effectiveness[C]// 2018 IEEE 18th International Conference on Communication Technology (ICCT). IEEE, 2018.

[21] STAFFORD S, LI J. Behavior-based worm detectors compared [C]// Recent Advances in Intrusion Detection. International Symposium, Raid, Ottawa, Ontario, Canada. DBLP, 2013.

[22] LI Y, CHEN Z, CHEN C. Understanding divide-conquer-scanning worms[C]// 2008 IEEE International Performance, Computing and Communications Conference. IEEE, 2008: 51-58.

ZHAO Jin-long, born in 1994, postgraduate. His main research interests include network security, deception defense and software defined networking.



ZHANG Guo-min, born in 1979, Ph.D, associate professor. His main research interests include software defined networking, network security, network measurement and distributed system.