计算机科学
COMPUTER SCIENCE

面向网络侦察欺骗的差分隐私指纹混淆机制

何源，邢长友，张国敏，宋丽华，余航

引用本文

何源，邢长友，张国敏，宋丽华，余航. 面向网络侦察欺骗的差分隐私指纹混淆机制[J]. 计算机科学, 2022, 49(11): 351-359.

HE Yuan, XING Chang-you, ZHANG Guo-min, SONG Li-hua, YU Hang. Differential Privacy Based Fingerprinting Obfuscation Mechanism Towards NetworkReconnaissance Deception[J]. Computer Science, 2022, 49(11): 351-359.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220400285

面向网络侦察欺骗的差分隐私指纹混淆机制

何 源 邢长友 张国敏 宋丽华 余 航
陆军工程大学指挥控制工程学院 南京 210007
(784510649@qq.com)

摘 要 网络指纹探测作为一种重要的网络侦察手段,可以被攻击者用于获取目标网络的指纹特征,进而为后续开展有针对性的攻击行动提供支持。指纹混淆技术通过主动修改响应分组中的指纹特征,能够让攻击者形成虚假的指纹视图,但现有的混淆方法在应对攻击者策略性探测分析方面仍存在不足。为此,提出了一种面向网络侦察欺骗的差分隐私指纹混淆机制(Differential Privacy based Obfuscation of Fingerprinting,DPOF)。DPOF 参考数据隐私保护的思想,首先建立了效用驱动的差分隐私指纹混淆模型,通过差分隐私指数机制计算不同效用虚假指纹的混淆概率,在此基础上进一步设计了资源约束下的指纹混淆决策方法,并实现了基于粒子群优化的混淆策略求解算法。仿真实验结果表明,相比现有的典型指纹混淆方法,DPOF 在不同问题规模和预算情况下均具有更优的指纹混淆效果,且能够以更快的速度获得更好的近似最优策略。

关键词:指纹混淆;差分隐私;网络侦察;网络欺骗防御
中图法分类号 TP393

Differential Privacy Based Fingerprinting Obfuscation Mechanism Towards Network Reconnaissance Deception

HE Yuan,XING Chang-you,ZHANG Guo-min,SONG Li-hua and YU Hang
College of Command and Control Engineering,Army Engineering University of PLA,Nanjing 210007,China

Abstract Network fingerprinting detection is an important network reconnaissance method,which can be used by attackers to obtain the fingerprinting characteristics of the target network,and then provide support for subsequent targeted attacks. Fingerprinting obfuscation technology enables attackers to form fake fingerprinting views by actively modifying the fingerprinting features in response packets. However,existing obfuscation methods are still insufficient in dealing with attackers' strategic detection and analysis. To this end,a differential privacy based fingerprinting obfuscation mechanism(DPOF) towards network reconnaissance deception is proposed. Taking the idea of data privacy protection as a reference,DPOF first establishes a utility-driven differential privacy fingerprinting obfuscation model,and calculates the obfuscation probability of fake fingerprints with different utilities through the differential privacy exponential mechanism. On this basis,a fingerprinting obfuscation decision method under resource constraint is further designed,and an obfuscation strategy solving algorithm based on particle swarm optimization is implemented. Simulation results show that compared with the existing typical fingerprinting obfuscation methods,DPOF has better fingerprinting obfuscation effect with different problem scales and budgets,and can obtain a better approximate optimal strategy at a faster speed.

Keywords Fingerprinting obfuscation,Differential privacy,Network reconnaissance,Cyber deception defense

1 引言

网络指纹探测是攻击者利用 Nmap 等探测工具[1-3],向目标系统发送指纹探测分组,分析系统响应分组中的关键字段,匹配指纹库得到目标系统的网络配置信息、操作系统属性、服务信息等特征信息的行为。指纹探测往往是攻击者发起网络攻击的第一步,基于上述特征信息,攻击者可以制定针对性的计划对目标系统发起攻击[4]。

为了主动对抗攻击者的指纹探测,现有研究引入了欺骗防御的思想,通过在网络中策略性地对网络指纹特征进行混淆来误导和迷惑攻击者,达到对抗指纹探测行为的目的[5]。具体而言,网络指纹特征混淆主要是消除网络实体的系统、通信协议、应用等指纹信息,使攻击者在进行指纹探测时获得错误的指纹信息,进而难以基于该信息有效描述探测

到稿日期:2022-04-28 返修日期:2022-07-23
基金项目:国家自然科学基金面上项目(62172432,61772271)
This work was supported by the National Natural Science Foundation of China(62172432,61772271).
通信作者:邢长友(changyouxing@126.com)

对象和发起攻击行为。

然而，尽管当前存在一些通过主动修改协议字段、部署蜜罐等方法进行指纹特征混淆的方法，但它们大多采用简单映射的方式进行指纹特征混淆，将目标系统的指纹转换为另一种指纹以响应攻击者的扫描探测。这样做虽然能够在一定程度上实现指纹特征的隐藏，但并没有消除指纹信息的统计特征，难以抵御基于大数据分析收集信息等方法[6-8]发起的探测行为，攻击者仍可以在多轮次探测的基础上，通过分析网络统计信息找出网络的真实指纹信息。

如图 1 所示，假设网络中有 3 个系统：s_1，s_2 和 s_3，以及两种指纹 f_1 和 f_2。s_1 和 s_2 的指纹类型为 f_1，s_3 的指纹类型为 f_2。防御者进行指纹混淆，将 f_1 混淆为 f_3，将 f_2 混淆为 f_4。攻击者对网络进行扫描探测，得知网络中存在 3 个系统和 2 种指纹，并且 s_1 和 s_2 指纹类型相同，s_3 是另一种指纹的统计信息，网络中指纹的统计信息特征并没有随着指纹混淆而消除。基于这样的统计信息特征，攻击者可以结合目标网络环境对真实的指纹信息进行推断，如办公环境中 Windows 主机占多数，数据中心环境中 Linux 主机占多数等。

进一步，如果网络中新加入 1 个系统 s_4，其指纹类型为 f_2，按照混淆策略其指纹类型变为 f_4。攻击者可以得知新加入的系统 s_4 的指纹类型与 s_3 相同，防御者指纹混淆没有达到预期的目的。

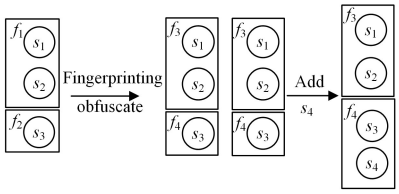


图 1 基于简单映射的指纹混淆方法的局限性

Fig. 1 Limitations of fingerprinting obfuscation methods based on simple mapping

本质上，上述问题与大数据背景下防止隐私泄露有诸多相似之处。尽管防御者采用了去标识化和匿名化等手段，但攻击者仍有可能通过统计分析的方法获取系统的指纹信息。在指纹混淆过程中引入差分隐私是解决这一问题的有效手段。差分隐私机制能够针对目标网络环境，以特定的概率分布形成虚假混淆指纹特征，达到统计意义上的指纹隐私保护的目的，并保证在增加或者去除部分系统后指纹统计特征几乎不变，从而有效对抗攻击者的策略性统计分析行为[9]。如图 2 所示，经差分隐私混淆之后，攻击者扫描探测得到的是错误的统计信息，并且在加入新的系统之后，攻击者也难以推断其指纹类型及相互间关系。

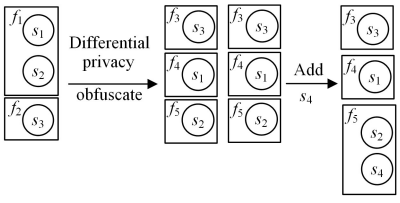


图 2 差分隐私混淆示意图

Fig. 2 Schematic diagram of differential privacy obfuscation

本质上，基于差分隐私的网络指纹混淆就是针对目标网络环境，策略性地将网络中不同的真实指纹变换为多种虚假指纹，进而使攻击者形成错误的网络视图。但与数据隐私保护相比，网络指纹混淆过程中的成本与效用具有较大的差异性[10]，例如将 1 台 Ubuntu 主机混淆为 Debian 所需要的成本可能低于将其混淆为 Windows 的成本，同时出于安全性的考虑，某些关键服务器的指纹信息必须要进行混淆。因此，基于差分隐私的网络指纹混淆问题可以描述为在特定混淆成本和混淆目标约束下，如何构建一种差分隐私混淆方案，使得目标网络的真实指纹特征对外暴露得尽可能少，即攻击者难以根据探测结果推断网络指纹信息。

然而，尽管差分隐私技术为开展指纹混淆提供了一种可行思路，但仍存在成本约束下混淆效果不佳和防御策略求解困难的问题。面对多样化指纹混淆需求以及混淆成本等约束，我们仍需要从混淆模型设计、混淆策略优化等多个角度进行突破。为此，本文建立了一种面向网络侦察欺骗的差分隐私指纹混淆机制对防御策略进行改进，并实现了相应的启发式混淆策略优化算法。具体而言，本文的主要贡献如下：

(1)提出了一种面向网络侦察欺骗的差分隐私指纹混淆机制(DPOF)。DPOF 从差分隐私保护的角度对指纹混淆过程进行了分析，并建立了资源约束下的策略优化模型，支持在成本和指纹变换能力等约束下进行最优混淆决策，达到成本预算受限情况下尽可能减小攻击者收益的目的。

(2)对现有差分隐私指纹混淆方法中的防御策略进行改进。将防御策略优化为选择进行混淆的系统数量，减小差分隐私随机性对防御策略的影响，便于最优防御策略的求解。

(3)实现了一种基于粒子群优化的混淆策略求解算法(Defense Strategy Solve based on Particle swarm optimization，DSSP)，DSSP 针对网络规模增大导致的最优指纹混淆策略解空间爆炸等难题，利用粒子群优化方法提高解空间搜索程度和求解速度，进而高效实现混淆策略的优化求解。

2 相关工作

面向网络侦察的欺骗防御技术主要通过构建虚假的网络指纹等信息来破坏攻击者的探测行为，诱使攻击者形成错误的认知。为此，本文首先介绍本领域当前的典型研究成果和研究动态，在此基础上分析现有欺骗防御手段在抵抗网络指纹探测方面存在的局限性，以及应用差分隐私进行指纹混淆待解决的问题。

为了阻止操作系统指纹识别，Albanese 等设计了一个系统处理传出流量，使其类似于使用不同操作系统的主机生成的流量[11]。Wang 等提出了一个指纹探测对抗系统(Moving OS Fingerprint Adaptively，MOFA)[12]。MOFA 利用 SDN 交换机的数据监控能力，动态地修改 SDN 交换机中的流表项，实现对传输数据包字段的修改，达到指纹信息混淆的目的。Shi 等设计了一种基于 SDN 的 MTD 系统 Chaos，该系统利用混沌塔(CTO)混淆方法，刻画网络中的所有主机层次结构，以此进行 IP 混淆、端口混淆和指纹混淆[13]。

多数研究采用博弈论对网络安全和隐私中的战略性和对抗性互动进行建模[14-15]。Li 等提出一个完全信息动态博弈模型来分析网络攻防双方从侦察到攻击的交互过程[16]。Schlenker 等提出一种博弈模型来刻画欺骗防御，并针对两类攻击者提出了近似最优策略计算算法[10]。Jajodia 等提出一个欺骗概率逻辑，通过防御者在不同的状态下生成假扫描结果的方法，最小化攻击者可能产生的损害[17]。Rahaman 等[18]分别从防御者和攻击者的视角对其交互进行博弈建模，分析了双方的均衡策略。Pawlick 等使用了带证据的信号博弈，将带有欺骗概率证据的检测器引入廉价谈话信号博弈，对攻防双方交互进行建模[19]。Bilinski 等提出一种伪装博弈的模型，在每一轮博弈中，攻击者询问防御者设备的性质（真实或者虚假），防御者可以选择欺骗策略（支付代价）或者如实相告[20]。尽管上述研究提出了基于欺骗防御的解决方案，但没有考虑消除网络统计信息的特征，也没有考虑系统数量动态变化情况下的指纹隐藏的问题，有一定的局限性。为了解决这个问题，Ye 等首次提出利用差分隐私机制实现对系统指纹的混淆[9]，缓解系统动态变化对网络的安全性的影响。但这项工作存在两个问题。一是在成本等约束情况下的防御效果不佳。Ye 等采用的防御者防御策略是从低效用系统开始指纹混淆，这样的防御策略在网络规模较大、防御资源不足以支撑所有的系统进行指纹混淆的情况下存在局限性。二是最优策略求解困难。防御者的防御策略由差分隐私机制对系统指纹添加的噪声扰动决定，噪声的随机性会导致可用防御者策略的不确定，使得最优策略的求解非常困难。其他相关研究在该项工作的基础上进行了扩展和改进[21-22]，侧重点在于博弈不完全信息的优化和博弈模型的创新，但是均未解决上述差分隐私指纹混淆中存在的两个问题。

综上所述，传统的欺骗方法在网络侦察防御中无法消除网络统计特征，也无法应对网络动态变化对网络安全性的影响。基于差分隐私的指纹混淆方法能够解决这一问题，但缺乏在成本等约束条件下的混淆策略求解和优化等方面的研究。

3　指纹混淆机制

针对现有指纹混淆研究中存在的问题，我们提出了一个基于差分隐私的指纹混淆机制。假设网络场景为防御者在网络中部署防御措施，攻击者向目标网络系统发送指纹探测数据包，分析返回响应中的指纹信息。防御者保护在网络中部署的系统，系统集合为 N，其数量为 $|N|$。每个系统具有一定的指纹特征，如操作系统版本、服务类型等，这些指纹特征被统称为指纹配置。整个网络的指纹配置集合为 F，数量为 $|F|$。每个系统拥有一个指纹配置 f_i，$i\leqslant|F|$。系统集合 N 根据指纹的不同分为若干子集 N_{f_i}，$\bigcup_{f_i\in F}N_{f_i}=N$。每个子集为与指纹配置 f_i 相关联的系统集合，每个系统都具有与配置相关联的效用 u_{f_i}，表示该配置的价值和安全级别的综合度量。攻防双方对网络系统的效用具有一致的计算方法。在此场景下，为了对系统指纹进行保护，我们在指纹混淆中加入差分隐私。表 1 列出了本文使用的主要符号及其含义。

表 1　主要符号说明

Table 1　Description of main symbols

符号	含义
N	系统集合
F	配置集合
F'	虚假配置集合
u_{f_i}	配置 f_i 的效用函数
$c_{(i,j)}$	配置 f_i 混淆为配置 f_j 所需成本
η_{f_j}	配置 f_i 混淆为配置 f_j 的概率
B	防御成本预算
$\boldsymbol{\Gamma}$	可行性矩阵
π^D	防御者策略
x_{f_i}	从配置 f_i 选择混淆的系统数量
R^A	攻击者实际收益
ε	隐私预算
ΔS	全局灵敏度

3.1　差分隐私指纹混淆定义

差分隐私由 Dwork 于 2006 年首次提出[23]，其思想来源于密码学中的语义安全，目的是让攻击者无法根据数据集的输出判断一个结果是否存在于该数据集。由于差分隐私不需要依赖于攻击者的先验知识，因此被广泛应用于网络物理系统[24]、人工智能[25]、物联网[26]等领域的隐私保护中。定义 1 给出了差分隐私的正式定义。

定义 1（ε-差分隐私[9]）　对于两个相邻的数据集 D 和 D'，一个机制 M 对所有 M 可能的输出值的集合 Ω 的任意子集 Ω_m，满足：

$$Pr[M(D)\in\Omega_m]\leqslant\exp(\varepsilon)\cdot Pr[M(D')\in\Omega_m] \tag{1}$$

则称机制 M 满足 ε-差分隐私，ε 为隐私预算。如果两个数据集 D 和 D' 仅相差一条记录，则它们是相邻的数据集。查询 Q 是将数据集 D 映射到抽象范围 R 的函数，$Q:D\to R$。差分隐私将查询函数 Q 的结果映射到一个随机值域上，并以一定的概率返回一个查询结果，使用隐私预算 ε 控制相邻数据集查询结果概率分布的接近程度。ε 越小，概率分布越相似，隐私保护程度越高，数据的可用性越低。

定义 2（全局敏感度[9]）　一个查询函数 $Q:D\to R$，在对应的相邻的数据集 D 和 D' 上的全局敏感度定义为：

$$\Delta S=\max_{D,D'}\|Q(D)-Q(D')\|_1 \tag{2}$$

其中，$\|Q(D)-Q(D')\|_1$ 是 $Q(D)$ 和 $Q(D')$ 的曼哈顿距离。全局敏感度是对查询函数在相邻数据集 D 和 D' 上查询的最大变化范围，用于控制在数据集查询输出中的干扰噪声的大小。差分隐私最常用的机制之一是指数机制，其定义如下。

定义 3（指数机制[9]）　对于数据集 D 上的可用性函数 $q(D\times\Omega)\to R$，存在一个机制 M，使得以正比于 $\exp\left(\frac{\varepsilon\times q(D,\Omega)}{2\times\Delta q}\right)$ 的概率输出一个结果，则称 M 为指数机制。

其中可用性函数 $q(D,\Omega)$ 表示数据集中可能输出的结果的价值，该结果的价值越大，输出的概率越高。Δq 表示可用性函数 $q(D,\Omega)$ 的全局敏感度。

本文采用差分隐私中的指数机制对指纹进行保护。为了进行指纹混淆，假设网络中存在一个备选虚假指纹配置集 F'，集合中包含若干虚假配置 f_j，$j\leqslant|F'|$，每个虚假配置 f_j 具有效用 u_{f_j}。在对系统进行指纹混淆时，将虚假配置效用作为

可用性函数，并使用指数机制计算出将网络中的原始指纹混淆为不同虚假配置f_j的概率，如式(3)所示。根据计算出的混淆概率分布从F'中选择一个配置f_j，将系统原配置f_i混淆为f_j。

$$\eta_{f_j}=\frac{\exp\left(\frac{\varepsilon u_{f_j}}{2\Delta u}\right)}{\sum\limits_{f_w\in F'}\exp\left(\frac{\varepsilon u_{f_w}}{2\Delta u}\right)} \tag{3}$$

其中，$\Delta u=\max\limits_{f_w\in F'}(u_{f_w})$。本文将网络中的虚假配置集合$F'$视作数据集，将系统混淆之后的指纹$f_j$视作数据集的输出。因此数据集输出结果的价值即$q(D,\Omega)$等同于指纹$f_j$的效用，相邻的数据集输出的最大距离即全局敏感度是$f_j$的最大效用。根据定义1和定义3，使用差分隐私指数机制混淆指纹，能够确保经过混淆之后的系统指纹符合预先设置的概率分布，保护进行混淆的系统指纹的隐私，使得攻击者难以推测出网络内部存在的真实指纹类型的数量，也无法推断出每个系统与指纹类型之间的对应关系，消除了统计信息上的特征。同时，由于差分隐私可以保证攻击者无法推断给定的数据记录是否在数据集中，因此该机制也可以保证攻击者无法推断给定的系统是否在该种类型的指纹配置集合中，从而在系统数量动态变化时保护指纹信息，而一般的随机算法无法保证这一点。图3给出了一个差分隐私指纹混淆机制示例。

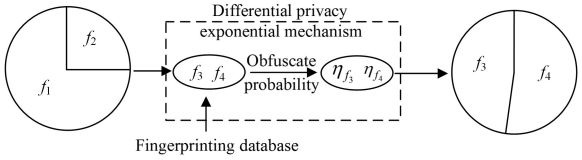


图3 差分隐私混淆机制示例

Fig. 3 Example of differential privacy obfuscation mechanism

假设在网络中存在系统集合，配置集合$F=\{f_1,f_2\}$，其各自的系统数量分别占总数的75%和25%。防御者进行差分隐私混淆，指纹库中存在两种虚假指纹配置$F'=\{f_3,f_4\}$，效用$u_3=8$，$u_4=5$，隐私预算$\varepsilon=0.5$，将u_3和u_4带入式(3)求得其混淆概率为$\{\eta_{f_3}=0.52,\eta_{f_4}=0.48\}$。将所有系统按照$\eta_{f_3}$和$\eta_{f_4}$从指纹库中选择虚假指纹对系统指纹进行混淆，攻击者从混淆后的网络中获得的关于指纹的信息为：网络中存在若干系统，具有两种配置，系统数量分别占52%和48%。可见攻击者通过探测获得了错误的指纹统计信息，无法利用统计分析的方法获取系统的指纹信息。相比使用一般的随机算法提供的概率，通过差分隐私计算混淆概率，能够保证在混淆概率满足ε-差分隐私定义的情况下，使攻击者无法通过对数据集的查询输出判断其中数据记录的隐私信息，即使网络中加入了新的系统，攻击者也难以通过多次探测，从统计信息的变化中推测出该系统指纹。

3.2 混淆策略优化模型

针对特定的系统，我们可以采用前述方法对其指纹进行混淆，但网络中存在若干不同指纹的系统，防御者需要在网络规模较大的情况下利用有限的成本进行混淆决策，达到尽可能降低攻击者收益的目的。因此本文对现有的差分隐私指纹混淆策略进行改进，提出一个在成本约束下的差分隐私指纹混淆优化模型DPOF。

进行指纹混淆时，由于成本有限，防御者必须对每一个系统子集N_{f_i}决定是否混淆其配置，若确定混淆，则选择进行混淆的系统数量。即防御策略可表示为$\pi^D=\{x_{f_i}\mid 0\leqslant x_{f_i}\leqslant|N_{f_i}|\}$，其中$x_{f_i}$表示从系统子集$N_{f_i}$中选择进行混淆的系统数量。这样做的好处是，在防御者成本预算有限的情况下，防御者能够决定每一种指纹配置中被混淆的系统数量，同时减小差分隐私添加的随机噪声对防御者策略的影响。如果防御者认为某一种配置极有可能是攻击者的攻击目标或者网络内某些关键配置的系统必须进行混淆，它可以更多地从这一配置中选择系统进行指纹混淆来尽可能降低攻击者攻击这个配置获得的收益。这样能够让防御者根据自身预算情况和对攻击者的推测灵活地改变混淆策略。

将配置f_i混淆成配置f_j存在一定的成本$c_{(i,j)}$，混淆成本应该和混淆前后的系统配置有关，因此定义为$c_{(i,j)}=\alpha|u_{f_i}-u_{f_j}|$，$\alpha$为混淆成本系数。同时在实际应用中，系统的配置并不能随意改变，或者说改变的成本太高，防御者的混淆策略可能不会使任意的系统配置f_i混淆为任意的虚假配置f_j。这个条件被称为可行性约束，用$|F|\times|F'|$的(0,1)矩阵$\boldsymbol{\Gamma}$表示。如果$\boldsymbol{\Gamma}_{(i,j)}=1$，说明配置$f_i$能被混淆成虚假配置$f_j$。防御者的成本预算定义为$B$。

对攻击者而言，其通过扫描能够探测得知网络中的部分统计信息：系统的数量$|N|$、存在配置的数量以及与每个配置相关联的系统数量。但是，攻击者不知道网络中的真实指纹配置情况，也不知道系统与真实配置的对应关系。防御者将系统的真实配置f_i混淆为f_j后，攻击者观察到的是虚假配置f_j，但系统的真实配置信息和效用等并不会改变。因此，攻击者选择攻击该系统能够获得的收益依然为u_{f_i}，而非其认为的u_{f_j}。这里指纹混淆的作用就是让攻击者做出错误的判断，进而降低其收益。如果攻击者选择攻击具有某一类配置f_t的主机，其实际收益如式(4)所示：

$$R^A=\begin{cases}u_{f_t}(|N_{f_t}|-x_{f_t}), & f_t\in F\\ \eta_{f_t}\sum\limits_{f_i\in F}x_{f_i}u_{f_i}, & f_t\in F'\end{cases} \tag{4}$$

其中，当$f_t\in F$时，攻击者攻击的是真实配置f_t，收益由系统集合N_{f_t}中未进行混淆的系统效用组成；当$f_t\in F'$时，攻击者攻击的是虚假配置f_t，收益由混淆为虚假配置f_t的所有系统效用组成。

针对攻击者的策略性攻击行为，防御者需要考虑在混淆成本、混淆可行性等约束情况下，如何选择每一种配置中的混淆数量，使得攻击者获得的收益最小，即防御者需要在满足约束条件的情况下找到一个最优策略π^{D*}，使得R^A最小：

$$\pi^{D*}=\arg\min_{\pi^D}R^A \tag{5}$$

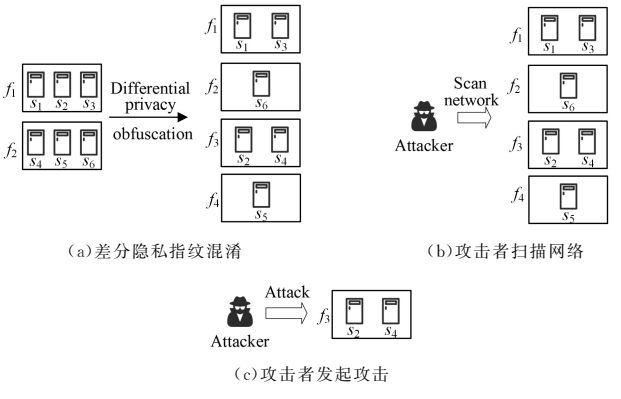
$$\text{s.t.}\sum_{f_j\in F'}\sum_{f_i\in F}\eta_{f_j}x_{f_i}c_{(i,j)}\leqslant B \tag{6}$$

$$\sum_{f_i\in F}x_{f_i}\leqslant|N|,x_{f_i}\leqslant|N_{f_i}| \tag{7}$$

$$\bigcup_{i=1}^{|F'|}\boldsymbol{\Gamma}_{(i,j)}=1,\forall f_j\in F' \tag{8}$$

式(6)表示防御成本预算约束，即防御者用于混淆配置的成本开销不能超过预算；式(7)表示混淆数量约束，即用于混淆的系统数量不能多于系统总配置数量；式(8)表示可行性约束，即系统配置必须能够混淆成某一虚假配置。

整个指纹混淆的过程可以用一个示例简单说明，如图 4 所示。



(a)差分隐私指纹混淆　(b)攻击者扫描网络

(c)攻击者发起攻击

图 4　指纹混淆过程示例

Fig. 4　Example of fingerprinting obfuscation process

假设在网络中存在系统集合 $N=\{s_1,s_2,s_3,s_4,s_5,s_6\}$，配置集合 $F=\{f_1,f_2\}$，配置效用分别为 $u_1=3$ 和 $u_2=9$，防御者预算最多只能混淆 3 个系统。

如图 4(a)所示，防御者决定每一种配置混淆系统的数量。假设防御者认为配置 f_2 的系统效用相对较高，攻击者攻击这种配置的系统获得的收益就较高。因此防御者会选择从配置 f_2 中选择 2 个系统、从配置 f_1 中选择 1 个系统进行指纹混淆。假设指纹库中存在两种虚假指纹配置 $F'=\{f_3,f_4\}$，效用 $u_3=8$，$u_4=5$，隐私预算 $\varepsilon=0.5$，其混淆概率为 $\{\eta_{f_3}=0.52,\eta_{f_4}=0.48\}$。假设所有配置都能任意混淆，经过混淆之后的网络如图 4(a)所示，系统 s_2 和 s_4 被混淆成了配置 f_3，s_5 被混淆成了配置 f_4。

如图 4(b)所示，攻击者对网络进行扫描，可以得知网络中存在 4 种指纹配置 f_1,f_2,f_3,f_4，也知道每一种指纹配置中包含的系统数量。经计算，攻击者发现攻击 4 种配置获得的收益分别为 6，9，16，5。由于攻击者会选择攻击当前收益最高的配置，因此攻击者会选择攻击配置 f_3，如图 4(c)所示，但其实际获得的收益是 3+9=12。

从整个过程中可以看到，如果防御者选择按照从低效用开始的固定顺序进行混淆，由于成本限制，f_2 中的 3 个系统将无法进行混淆。出于收益最大化的目的，攻击者会攻击配置 f_2 得到收益 $9\times3=27$。而使用优化的混淆策略，如果配置 f_2 不进行混淆，最坏的情况下攻击者收益会非常高，防御者在意识到这一情况后，综合约束条件决定在配置 f_2 中选择 2 个系统进行混淆，降低最坏情况下攻击者的收益。相比改进后的灵活混淆策略，按照固定顺序进行指纹混淆的策略在预算有限的情况下防御效果并不理想。通过灵活的混淆策略，防御者能够降低在预算有限的情况下的攻击者收益。

4　基于粒子群优化的混淆策略求解算法

如前文所述，防御者的最佳混淆策略是对每一种配置中的混淆系统数量进行决策，使得攻击者攻击当前收益最高的配置所获得的收益最小，即最坏情况下的攻击者收益最小。但是由于防御者的混淆策略在每一种配置集合 N_{f_i} 上有 $|N_{f_i}|$ 种可能的取值，网络中共有 $|F|$ 个配置集合，因此防御者的混淆策略解空间大小为 $\prod_{f_i\in F}|N_{f_i}|$。在网络规模较大的情况下配置集合数量 $|F|$ 较大，每一种配置集合 N_{f_i} 中系统的数量 $|N_{f_i}|$ 同样较大，防御者混淆策略解空间会呈指数级增长，求解较为困难，因此考虑使用启发式算法进行防御者混淆策略的求解。

本文实现了一种基于粒子群优化的混淆策略求解算法 DSSP 对 DPOF 中防御者的混淆策略进行求解。粒子群算法的主要思想是将防御者的每一种策略理解为空间中的一个位置，每个位置的维度表示在对应的配置集合中选择混淆的系统数量。通过若干个具有不同位置和速度的粒子在解空间中随机运动，用适应度函数评估位置的好坏，粒子间共享彼此间最佳位置，从而在解空间内搜索最优解。DSSP 的伪代码如算法 1 所示。

算法 1　防御者策略优化算法 DSSP

输入：$K,F,F',\varepsilon,N,B,u_{f_i},u_{f_j},\boldsymbol{\Gamma}$

输出 π^{D^*}

1. 初始化所有粒子速度 v_n 和位置 $\boldsymbol{z}_n$
2. 初始化粒子最佳位置 $\boldsymbol{z}_n^*$ 和种群最佳位置 $\boldsymbol{z}_G^*$
3. while $k\leqslant K$ do
4.　for 每个粒子 n
5.　　$\boldsymbol{z}_n^*\leftarrow\underset{\boldsymbol{z}}{\operatorname{argmin}}(\phi(\boldsymbol{z}_n^{best}),\phi(\boldsymbol{z}_n))$
6.　　$\boldsymbol{z}_G^*\leftarrow\underset{\boldsymbol{z}}{\operatorname{argmin}}(\phi(\boldsymbol{z}_G^{best}),\phi(\boldsymbol{z}_n))$
7.　for 每个粒子 n
8.　　$v_n\leftarrow w\,v_n+c_1r_1(\boldsymbol{z}_n^*-\boldsymbol{z}_n)+c_2r_2(\boldsymbol{z}_G^*-\boldsymbol{z}_n)$
9.　　$\boldsymbol{z}_n\leftarrow\boldsymbol{z}_n+v_n$
10.　$k\leftarrow k+1$
11. return z_G^*
12. function $\phi(z_n)$
13.　计算混淆概率 η
14.　for 粒子 z_n 每个维度 m
15.　 if $\boldsymbol{z}_{n,m}\neq N_{f_m}$ then
16.　　$R_m^A\leftarrow(N_{f_m}-z_{n,m})u_{f_m}$
17.　 for j=|F| to |F|+|F'|
18.　　$R_j^A\leftarrow\boldsymbol{z}_{n,m}\eta_j u_{f_j}+R_j^A$
19.　根据 u_{f_j} 计算收益最大的配置 f^*
20.　计算防御者成本 C^D
21.　if $C^D>B$ then
22.　return MaxCount
23.　return R_f^A

算法的输入是迭代次数 K、系统配置集合 F、虚假配置集合 F'、隐私预算 ε、系统集合 N、防御成本预算 B、系统配置的效用函数 u_{f_i}、虚假配置的效用函数 u_{f_j} 以及可行性矩阵 $\boldsymbol{\Gamma}$。输出是最优策略 π^{D^*}。算法第 1－2 行进行初始化，随机赋予所有粒子初始速度 v_n 和位置 $\boldsymbol{z}_n$，并对粒子历史最佳位置 $\boldsymbol{z}_n^*$ 和种群历史最佳位置 $\boldsymbol{z}_G^*$ 进行初始化。第 3－10 行进行迭代，寻找近似最优解直到迭代次数达到上限。第 4－6 行计算每一个粒子位置的适应度 $\varphi(s_i)$，并对 $\boldsymbol{z}_n^*$ 和 $\boldsymbol{z}_G^*$ 进行更新。适应度计算函数的输入为粒子位置 $\boldsymbol{z}_n$。第 13 行根据式(3)和可行性约束，为每一个系统配置计算混淆概率分布 η。第 14－18 行根据位置 $\boldsymbol{z}_n$ 代表的策略计算当前攻击者攻击各个配置的收益。

其中粒子位置在维度 m 上的取值代表防御者从配置 f_m 中选择进行混淆的系统数量，R_m^A 代表攻击者攻击配置 f_m 所能获得的实际收益。第 19 行攻击者计算当前收益最高的配置 f^*。第 20 行计算当前位置 $\boldsymbol{z}_n$ 代表的策略所需的防御成本。第 21—23 行计算函数返回值。若当前策略的防御成本超过预算，则返回常数 $MaxCount$，代表攻击者的收益趋近无穷大，否则返回攻击者实际收益作为当前位置向量 $\boldsymbol{z}_n$ 的适应度值。

第 7—9 行更新每一个粒子的速度和位置。粒子更新的速度包含 3 个部分：第一部分来自粒子更新前的速度，比例由参数 w 控制，可以理解为惯性；第二部分来自 $\boldsymbol{z}_n^*$ 与当前位置之间的距离，比例由参数 c_1 和随机数 r_1 的乘积控制；第三部分来自 $\boldsymbol{z}_G^*$ 和粒子当前位置之间的距离，比例由参数 c_2 和随机数 r_2 的乘积控制。粒子的速度更新综合考虑了三者之间的关联关系，使得粒子在能够搜索到更多解空间的同时，朝着全局最优的方向进行搜索，找到最优解的概率更大。最后第 11 行返回 $\boldsymbol{z}_G^*$ 作为防御者近似最优策略。

5 实验与评估

我们一共进行了 3 个模拟实验来评估本文方法，并与文献[9]和文献[10]提出的模型进行对比。

5.1 实验设置

实验共评价了 3 个模型和 4 个防御策略求解算法，分别为本文的解决方案 DPOF 和 DSSP，CDG[10] 和其算法 Greedy-Minimax，CDG-Fixed 以及文献[9]中的差分隐私方法和其防御部署算法（为方便区分称之为 DPCDG）。CDG 代表传统指纹混淆模型，因此将其作为实验一的基准。该模型在完全信息博弈和不完全信息博弈的情况下分别提出了最优策略求解算法 Greedy-Minimax 和 CDG-Fixed。为了统一实验设置，我们对 Greedy-Minimax 算法进行了修改，使其适用于不完全信息条件的实验。DPCDG 代表现有差分隐私指纹混淆研究，因此将其作为实验二和实验三的基准。由于 DPCDG 的防御策略随机性太大，因此在实验中的度量指标均在 1 000 轮博弈之后取平均值，实验数据均为 10 个博弈实例的平均结果。模拟实验如图 5 所示。模拟实验中以操作系统版本代表配置，一共设置了 4 种真实配置和 4 种虚假配置，每种真实配置关联若干服务器，代表网络系统。虚假配置分别设置为 Windows 7，CentOS 9.0，FreeBSD 10.1 和 Ubuntu 18.04。

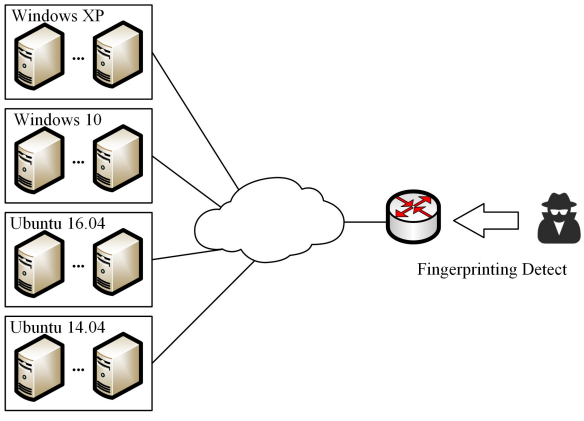


图 5 实验示意图

Fig. 5 Schematic diagram of experiment

实验设置如下：

实验一：评估指纹混淆中引入差分隐私的效果和策略优化算法的性能。将 DPOF 与 CDG 进行实例化，分别采用 DSSP 以及 Greedy-Minimax 和 CDG-Fixed 算法求解防御者最优策略，以攻击者总收益和算法最小迭代次数为度量指标进行对比分析，同时分析混淆之后的指纹分布情况。使用算法最小迭代次数是因为在评估算法性能的实验中，不同算法的实现细节不同，为简化实验，统一采用算法最小迭代次数作为算法运行时间的度量指标（其中 CDG-Fixed 算法迭代次数由配置数量决定，相对固定，因此不进行比较）。

实验二：评估防御策略改进之后的指纹混淆效果。将 DPOF 与 DPCDG 实例化，分别采用 DSSP 和文献[2]中的防御策略算法求解防御策略，以攻击者总收益和防御者成本作为度量指标进行对比分析。

实验三：评估防御策略在不同网络规模下的扩展性。将 3 个模型实例化，逐渐增加系统数量，分别求解各个模型中的防御策略和攻击者总收益，进行对比分析。

为统一实验参数，在实验中，配置效用 u_{f_i} 和虚假配置效用 u_{f_j} 的值在[1,20]之间采样，防御成本和攻击成本的计算方式相同，指纹混淆成本系数 $\alpha=0.2$，$MaxCount=10\,000$。防御者近似最优策略求解算法的参数 $c_1=4$，$c_2=4$，$w=1$，$K=1\,000$，粒子种群大小为 100，可行性约束矩阵随机生成为：

$$\boldsymbol{\Gamma}=\begin{bmatrix}1&0&1&0\\0&1&0&1\\1&0&0&0\\1&1&1&0\end{bmatrix}$$

5.2 实验结果

(1)实验一：本实验的目的是评估引入差分隐私后的指纹混淆效果和指纹分布情况，以及 DSSP 算法的性能。在实验中，我们使用 CDG 中的 Greedy-Minimax 和 CDG-Fixed 算法求解防御者最优策略作为对比。在评估指纹混淆效果的实验中，使用攻击者获得的收益作为度量指标，在评估指纹分布情况的实验中，使用不同配置的系统数量作为度量指标。隐私预算 $\varepsilon=0.5$，防御成本预算 $B=100$，为统一实验参数，将指纹混淆效果评估实验中的 Greedy-Minimax 算法迭代次数设置为 1 000 次。实验结果如图 6—图 8 所示。

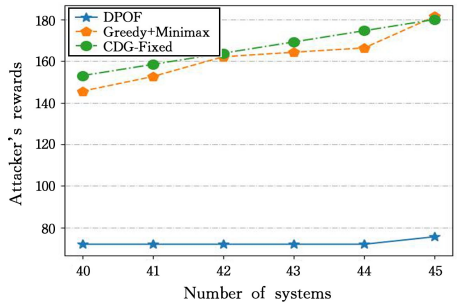


图 6 指纹混淆效果对比

Fig. 6 Comparison of fingerprinting obfuscation effects

图 6 给出了系统总数从 40 逐一增加至 45 的情况下的攻击者的收益变化。从图中可以看出，在引入差分隐私之后的

博弈模型 DPOF 中，攻击者的收益增加非常缓慢甚至停滞。相比之下，CDG 模型中，在系统数量逐一增加的情况下，攻击者的收益随之明显增加。由此可见，引入差分隐私后，新加入系统对攻击者收益的影响非常小，攻击者难以推断出系统指纹信息。其原因是：首先差分隐私机制对数据集中的个体隐私提供保护；其次在改进后的防御策略中，存在新增加系统对防御者近似最优策略造成的影响很小的情况，因此攻击者收益变化不大。除此之外，在 DPOF 模型中的攻击者收益均小于在 CDG 模型中应用不同算法情况下的攻击者收益。这说明应用差分隐私之后防御效果更好。

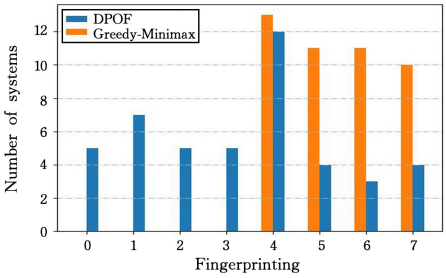


图 7　混淆后指纹分布图

Fig. 7　Fingerprinting distribution after obfuscation

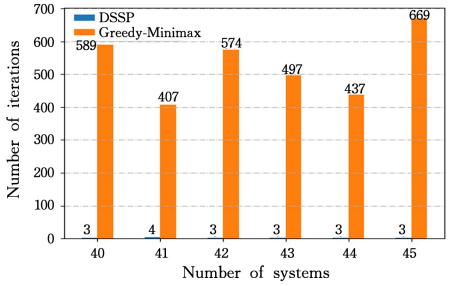


图 8　策略求解算法性能对比

Fig. 8　Performance comparison of policy solving algorithms

图 7 给出了当系统数量为 45 的情况下，使用两种方法混淆指纹之后网络中的指纹分布情况。其中横轴指纹配置 0-3 表示真实配置，4-7 表示虚假配置。从图中可以看到，使用 CDG 中的方法进行指纹混淆后，网络中只存在虚假指纹，且其数量分布接近原真实系统数量分布，即均匀分布，指纹统计特征并没有被消除。而 DPOF 中的方法在混淆指纹后将真实和虚假的指纹信息混合返回给攻击者，以迷惑攻击者的认知。使用 DPOF 方法混淆指纹，混淆后的指纹分布呈现随机性，消除了指纹统计信息中的特征，使攻击者难以从统计信息中推测指纹。

图 8 给出了两个模型在不同系统数量条件下求解最优策略的最小迭代次数对比。从图中可以明显看出，DSSP 算法求解最优策略的速度比 Greedy-Minimax 更快。

综上可以看出，应用差分隐私指纹混淆对攻击者指纹探测的对抗效果优于传统的指纹混淆方法。

(2)实验二：本实验的目的是评估改进差分隐私指纹混淆防御策略之后的防御效果。在实验中，我们使用 DPCDG 作为对比，采用攻击者收益和防御者成本作为度量指标，隐私预算 $\varepsilon = 0.5$。实验结果如图 9 和图 10 所示。由于在实验中，DPCDG 的策略随机性太大，设置过小的预算会导致部分情况下高效用的系统因预算限制而不能进行混淆，但已经有一部分低效用系统混淆为了高效用配置引起了数量的冲突。因此在实验中我们将预算设置为满足 DPCDG 的最大需求，同时用相同条件下的攻击者收益和防御者成本评估不同预算情况下的防御效果。

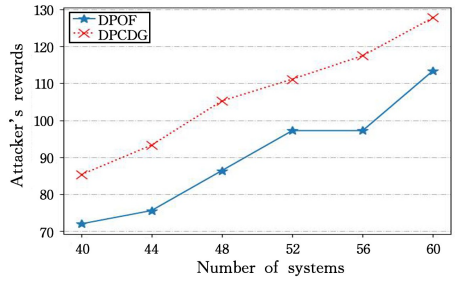


图 9　不同系统数量下的攻击者收益

Fig. 9　Attacker's reward with different number of systems

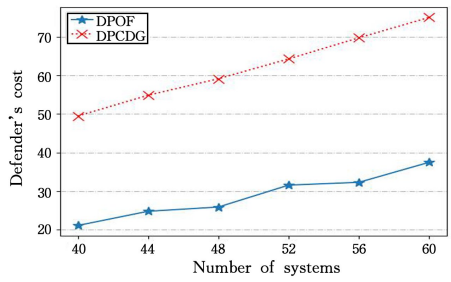


图 10　不同系统数量下的防御成本

Fig. 10　Defense cost with different number of systems

图 9 给出了在不同系统数量的条件下，攻击者在两种方法求解的防御策略部署情况下的收益。从图中可以看出，攻击者在 DPOF 模型求解的防御策略部署情况下的收益增长缓慢，且相比 DPCDG 求解的防御策略降低了约 10%18%。这说明 DPOF 模型在改进了防御部署策略之后，防御效果优于现有的差分隐私指纹探测对抗博弈模型。

图 10 给出了在不同系统数量条件下，防御者部署防御策略所需要的防御成本。从图中可以看出，DPOF 模型的防御策略所需的防御成本随系统数量的增加而缓慢增加，DPCDG 模型的防御策略所需的防御成本增加相对迅速，这是因为 DPCDG 模型中的防御策略是按照固定顺序混淆尽可能多的系统，当系统数量增加时，其成本也随之迅速增加。而在 DPOF 中，防御策略是有决策地从配置集合中选择若干数量的系统进行指纹混淆，存在增加系统数量对防御者的策略影响很小甚至可能不受影响的情况，因此成本增加缓慢。

综合观察图 9 和图 10，在系统数量一定的情况下，DPOF 近似最优策略进行混淆所需的成本比 DPCDG 降低了 50%-57%，同时防御效果相对更好。这说明，改进防御策略之后，本文方法能使用更低的成本，达到比 DPCDG 更好的防御效果。这是因为在本文方法中，高效用的系统不受混淆顺序的制约，能够有效进行指纹混淆，这样的防御策略减小了攻击者扫描这一部分系统而产生的收益。

综上可以得出结论，DPOF 模型改进后的防御策略比现有

的差分隐私指纹探测对抗博弈模型 DPCDG 效果更好，且使用的成本更少，在预算有限的条件下具有更好的防御效果。

(3)实验三：本实验的目的是评估 3 个模型在不同问题规模下的扩展性。隐私预算 $\varepsilon = 0.5$，防御成本预算设置为满足 DPCDG 的最大需求，为统一实验参数，Greedy-Minimax 算法迭代次数设置为 1000 次，度量指标选择攻击者最大收益。实验结果如图 11 所示。

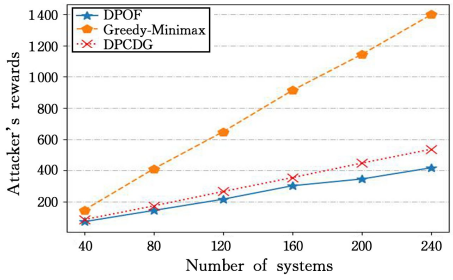


图 11 不同问题规模下的攻击者收益

Fig. 11 Attacker's reward under different problem scales

图 11 给出了在系统数量从 40 增加至 240，以模拟不同网络规模的情况下，攻击者在各个模型中部署的防御策略下的收益。可以看到，在系统数量较小的情况下，3 种模型的防御效果相对比较接近，其中传统防御方法 Greedy-Minimax 的防御效果比 DPOF 和 DPCDG 的防御效果略差。随着系统数量大幅增加，Greedy-Minimax 的防御效果明显比其余两种方法差，攻击者获得的收益大幅增加。与之相比，DPOF 和 DPCDG 的防御策略下的攻击者收益增加速度相对较慢，其中 DPCDG 模型中攻击者收益增长更加迅速，即 DPOF 的防御效果略好于 DPCDG 的防御效果。这是因为传统防御方法是贪婪的指纹混淆策略，没有考虑网络中系统数量变化的情况。在网络规模较大且预算有限的情况下，部分增加的系统同样无法进行混淆，因此防御效果有限。而随着网络规模的扩大，DPCDG 中的部分系统因网络规模扩大导致预算不足而无法进行混淆，攻击者攻击这一部分系统能获得大量收益。改进后的混淆策略使防御者能够在预算受限的情况下对进行混淆的系统数量和系统配置进行决策，通过策略优化选择在最坏的情况下使攻击者收益最小的策略，因此防御效果相对更好。

综上所述，在问题规模较小的情况下，传统防御方法的防御效果比 DPOF 和 DPCDG 稍差；问题规模较大的情况下，传统防御方法的效果最差，其次是 DPCDG，DPOF 防御效果最好。也就是说，无论问题规模的大小，DPOF 的防御效果均优于其余两种模型，扩展性更强。

结束语 指纹探测是攻击者进行网络侦察的重要手段之一，基于网络欺骗防御的思想，网络指纹混淆能够通过构造虚假的指纹信息欺骗攻击者，使其形成错误的视图。然而，现有的指纹混淆机制仍具有较多的局限性，存在无法消除统计信息特征和在预算有限的情况下对抗探测效果欠佳的问题。为此，本文提出了面向网络侦察欺骗的差分隐私指纹混淆机制，建立了资源约束下的混淆策略优化模型，支持在成本和指纹变换能力等约束下进行最优混淆决策，达到在预算受限情况下尽可能减少攻击者收益的目的。在此基础上，提出了一种基于粒子群优化的混淆策略求解算法，并建立仿真实验环境验证了方法的有效性。

在未来的工作中，我们将研究降低防御者最优策略求解规模的方法，进一步增强主动欺骗攻击者指纹探测行为的能力。

参 考 文 献

[1] LYON G F. Nmap network scanning: The official Nmap project guide to network discovery and security scanning[M]. US: Insecure, 2008.

[2] AUFFRET P. SinFP, unification of active and passive operating system fingerprinting[J]. Journal in Computer Virology, 2010, 6(3): 197-205.

[3] KOHNO T, BROIDO A, CLAFFY K C. Remote physical device fingerprinting[J]. IEEE Transactions on Dependable and Secure Computing, 2005, 2(2): 93-108.

[4] BRYANT B D, SAIEDIAN H. A novel kill-chain framework for remote security log analysis with SIEM software[J]. Computers & Security, 2017, 67: 198-210.

[5] ZHU M, ANWAR A H, WAN Z L, et al. A survey of defensive deception: Approaches using game theory and machine learning [J]. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2460-2493.

[6] AKSOY A, LOUIS S, GUNES M H. Operating system fingerprinting via automated network traffic analysis[C]// 2017 IEEE Congress on Evolutionary Computation(CEC). IEEE, 2017: 2502-2509.

[7] HAGOS D H, YAZIDI A, KURE Ø, et al. A Machine-Learning-Based Tool for Passive OS Fingerprinting With TCP Variant as a Novel Feature[J]. IEEE Internet of Things Journal, 2020, 8(5): 3534-3553.

[8] HAGOS D H, LØLAND M, YAZIDI A, et al. Advanced Passive Operating System Fingerprinting Using Machine Learning and Deep Learning[C]// 2020 29th International Conference on Computer Communications and Networks (ICCCN). IEEE, 2020: 1-11.

[9] YE D Y, ZHU T Q, SHEN S, et al. A differentially private game theoretic approach for deceiving cyber adversaries[J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 569-584.

[10] SCHLENKER A, THAKOOR O, XU H F, et al. Deceiving cyber adversaries: A game theoretic approach[C]// International Conference on Autonomous Agents and Multiagent Systems. 2018: 892-900.

[11] ALBANESE M, BATTISTA E, JAJODIA S. A deception based approach for defeating OS and service fingerprinting[C]// 2015 IEEE Conference on Communications and Network Security (CNS). IEEE, 2015: 317-325.

[12] WANG Y L, GUO J, ZHANG J C, et al. Moving OS fingerprint adaptively in SDN network[C]// 2017 3rd IEEE International Conference on Computer and Communications (ICCC). IEEE, 2017: 438-442.

[13] SHI Y，ZHANG H G，WANG J，et al. Chaos：An SDN-based moving target defense system[J]. arXiv：1704.01482，2017.

[14] PAWLICK J，COLBERT E，ZHU Q Y. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy[J]. ACM Computing Surveys（CSUR），2019，52(4)：1-28.

[15] LIU J W，LIU J J，LU Y L，et al. Optimal Defense Strategy Selection Method Based on Network Attack-Defense Game Model[J]. Computer Science，2018，45(6)：117-123.

[16] LI S H，ZHANG G M，SONG L H，et al. Incomplete Information Game Theoretic Analysis to Defend Fingerprinting[J]. Computer Science，2021，48(8)：291-299.

[17] JAJODIA S，PARK N，PIERAZZI F，et al. A probabilistic logic of cyber deception[J]. IEEE Transactions on Information Forensics and Security，2017，12(11)：2532-2544.

[18] RAHMAN M A，HASAN M G M M，MANSHAEI M H，et al. A game-theoretic analysis to defend against remote operating system fingerprinting[J]. Journal of Information Security and Applications，2020，52：102456.

[19] PAWLICK J，COLBERT E，ZHU Q Y. Modeling and analysis of leaky deception using signaling games with evidence[J]. IEEE Transactions on Information Forensics and Security，2018，14(7)：1871-1886.

[20] BILINSKI M，FERGUSON-WALTER K，FUGATE S，et al. You only lie twice：A multi-round cyber deception game of questionable veracity[C]// International Conference on Decision and Game Theory for Security. Cham：Springer，2019：65-84.

[21] SUN P Y，ZHANG H W，MA J Q，et al. A Selection Strategy for Network Security Defense Based on a Time Game Model[C]// 2021 International Conference on Digital Society and Intelligent Systems(DSInS). IEEE，2021：223-228.

[22] WAN Z L，CHO J H，ZHU M，et al. Foureye：Defensive Deception Against Advanced Persistent Threats via Hypergame Theory[J]. IEEE Transactions on Network and Service Management，2021，19(1)：112-129.

[23] DWORK C. Differential privacy[C]// International Colloquium on Automata，Languages，and Programming. Berlin：Springer，2006：1-12.

[24] HASSAN M U，REHMANI M H，CHEN J J. Differential privacy techniques for cyber physical systems：a survey[J]. IEEE Communications Surveys & Tutorials，2019，22(1)：746-789.

[25] WEI K，LI J，DING M，et al. Federated learning with differential privacy：Algorithms and performance analysis[J]. IEEE Transactions on Information Forensics and Security，2020，15：3454-3469.

[26] JIANG B，LI J Q，YUE G H，et al. Differential Privacy for Industrial Internet of Things：Opportunities，Applications，and Challenges[J]. IEEE Internet of Things Journal，2021，8(13)：10430-10451.

HE Yuan，born in 1998，postgraduate. His main research interests include cyber deception defense and game theory.

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

（责任编辑：何杨）