中国科学: 信息科学 2020 年 第 50 卷 第 12 期: 1944–1960
SCIENTIA SINICA Informationis
论文

融合广义随机 Petri 网的二维拟态安全评估模型

杨昕[1,2], 李挥[1,2*], 邬江兴[3], 伊鹏[3]

1. 北京大学深圳研究生院, 深圳 518055
2. 鹏程实验室, 深圳 518055
3. 国家数字交换系统工程技术研究中心, 郑州 450002
* 通信作者. E-mail: lih64@pkusz.edu.cn

收稿日期: 2019–10–09; 修回日期: 2019–12–12; 接受日期: 2020–03–17; 网络出版日期: 2020–10–21

国家重点研发计划 (批准号: 2017YFB0803200)、先进防御设备与系统研制 (批准号: 2017YFB0803204)、鹏城实验室 – 大湾区未来网络试验与应用环境 (批准号: LZC0019)、国家自然科学基金创新研究群体项目 (批准号: 61521003)、广东省重点领域研发计划 (批准号: 2019B010137001)、深圳市科创委资助研究计划 (批准号: JCYJ20190808155607340, JCYJ20170306092030521, JSGG20170824095858416) 资助项目

摘要 为应对网络空间中的未知安全漏洞, 拟态防御系统作为一种动态异构冗余的新型防御架构破茧而出. 拟态系统根据网络环境自发进行重配置, 扭转了传统静态网络攻防不对等的局面. 然而目前仍缺乏有说服力的能够定量评估并比较不同的安全防御系统有效性的实用方案. 本文深入研究拟态架构, 提出了一种二维分析模型, 该模型将系统配置细节计算为量化结果, 以比较不同动态网络的可靠性, 且该模型在不同网络配置间可保持良好的可扩展性. 具体来说, 在分析的第 1 维度即单节点攻击分析时, 我们详细介绍了系统配置, 使用广义随机 Petri 网模型对攻击者和防御者的行为分别进行描述建模, 刻画其对系统状态的影响. 结合泊松过程、常见漏洞和暴露以及常见漏洞评分系统, 我们对其影响设计函数进行赋值、量化计算. 在分析的第 2 个维度即链路攻击中, 我们采用马尔可夫 (Markov) 链, 并用靶理论进行计算, 量化表达了攻击难度即攻击得手时长和网络配置之间的关系. 最后, 给出日常防御下和攻击情景下的安全度量方案, 验证了拟态防御的有效性, 为拟态系统的设计提供了理论指导.

关键词 拟态网络防御, 广义随机 Petri 网络, 通用漏洞评分系统, 安全分析, 安全模型

1 背景介绍

近年来, 网络安全事故频发, 人们逐步将注意力集中于网络安全领域. 在传统网络中, 攻击者和防御者处于不对等的地位. 在防御者完成初始配置后, 攻击者可以随时、持续地收集防御方的信息, 并选择合适的时机筹划发动攻击, 甚至可以在获得特权后保持很长一段时间内不被清除 [1]. 这极大地威胁

引用格式: 杨昕, 李挥, 邬江兴, 等. 融合广义随机 Petri 网的二维拟态安全评估模型. 中国科学: 信息科学, 2020, 50: 1944–1960, doi: 10.1360/SSI-2019-0224
Yang X, Li H, Wu J X, et al. A two-dimension security assessing model for CMDs combined with Generalized Stochastic Petri net (in Chinese). Sci Sin Inform, 2020, 50: 1944–1960, doi: 10.1360/SSI-2019-0224

着网络系统的安全性. 为了改变安全博弈中攻守双方不对等的局面, 国内外提出了许多新型防御架构, 这些新型防御技术具有动态和自适应的特点, 通称为主动防御.

美国国土安全部 (Department of Homeland Security, DHS) 提出的移动目标防御 (moving target defense, MTD) [2] 是动态防御思想的典型应用, 被视为 "改变游戏规则" 的革命性防御技术 [3]. 由于不同的配置对应的攻击薄弱点也不相同, 移动目标防御通过定期主动地迁移网络配置以限定同一弱点的暴露时长, 增大攻击表面和不确定性, 提高攻击者的攻击难度, 降低系统被攻陷的概率. 移动目标防御用主动改变系统配置换取了系统安全性的提升, 但系统配置的频繁更改带来的性能损失不可忽略. 为了避免频繁配置变迁, 必须寻找新的有效安全机制. 拟态安全防御 (cyberspace mimic defense, CMD) 是其中一种, 它将多种不安全组件聚合成为一个可靠系统 [4]. 其核心是异构冗余机制和多模表决机制. 其中, 动态异构冗余 (dynamic heterogeneous redundancy, DHR) 架构 [5] 是实现拟态防御的重要原理性方法之一. 通过 DHR 结构, 拟态防御降低了传统移动目标防御中周期性动态迁移带来的巨大开销.

主动防御与存储编码、Web 服务器、虚拟网络、加密算法等技术结合, 已在不同领域展开了应用. 值得注意的是, 移动目标防御和拟态安全防御都牺牲部分性能以满足系统的安全性需求. 如何定性、定量地分析它们的有效性, 成为了一项亟待解决的工作. 主流网络安全评估方法大多针对静态网络, 例如基于安全标准和漏洞检测等规则的评估方法, 基于故障树、攻击图等的评估方法. 这些方法难以适用于动态、主动的攻防博弈分析, 所以对主动防御网络安全性的评估方法还处于探索阶段 [6]. 目前的分析工作主要可以分为两种思路: 用实验模拟攻击过程, 测试其有效性; 用数学工具对攻防过程进行建模, 并计算其安全指标, 如攻击成功率、平均失效时间、稳态可用性等.

实验测量方法一般是构造一个具体的系统, 对系统进行攻防实验等仿真操作, 进行数据收集、测量. 这种方法由于保留了大量系统细节更加具有真实性, 但很少给出定量结果以及安全性和系统配置的关系, 所以该方案难以在不同系统间迁移. Zhuang 等首先在文献 [7] 主动改变网络配置, 将 MTD 应用在网络中, 并且在文献 [8] 中比较了简单的随机适应的 MTD 系统与智能的基于攻击检测的 MTD 系统的有效性. Hong 等 [9] 将 MTD 技术分为 3 类: 混乱、多样和冗余, 然后使用分层攻击表示模型和重要性度量技术, 从而提高系统的可扩展性. Richard 等 [10] 用机器学习的方法, 用博弈模型描述攻击和防御两方的交互, 分析了电子邮件应用中 MTD 防御的有效性.

与实验仿真不同, 使用数学方法推算有效性的研究显得非常稀少. 数学分析通常将攻防问题进行抽象, 借助现有数学模型, 例如 Petri 网络、Markov 链、博弈论、随机过程等进行分析. Yang 等 [11] 融合了随机过程的鞅和 Markov 链等理论, 对 MTD 安全性进行了量化描述. Maleki 等 [12] 介绍了一个基于 Markov 链的分析框架, 推算出了攻击者成功击溃一个 MTD 系统的概率和防御者花费的时间成本间的关系. 此外, 在详细刻画系统状态方面, Petri 网络是非常有力的工具. Wu [5] 用基于 Petri 网的建模方法 [13] 描述了拟态防御架构 (mimic defense architecture, MDA), 并研究其安全性. Mitchell 等 [14] 将网络状态划分为 5 类部件, 包括中央控制器、传感器、执行器、分布式管理器和网络连接, 以及 3 种攻击状态: 磨损、扩散、攻击逃逸, 用随机 Petri 网络记录其中的状态转移, 计算网络失效时间和各种配置参数的关系. Cai 等 [15] 用随机 Petri 网络 (stochastic Petri nets, SPN) 初步分析了 MTD 服务器的安全性. Moody 等 [16] 将 MTD 节点状态分为工作、空闲和诱捕 3 类, 并用随机 Petri 网络分析系统的状态停留时间变化时对应的安全性, 定性说明状态转换频率和系统有效性的关系. 对比实验方法, 数学方法通常更加抽象, 能够得到定量分析结果, 迁移性比较强, 但是由于舍弃了系统细节, 可信度方面有一定缺陷.

本文对存在判决时延的分布式 MDA 的攻防过程进行抽象, 假设攻击者从外部进入网络, 对网络链路中某个特定节点发动攻击. 在接近目标节点的过程中, 攻击者需要对链路上的单个节点逐个攻击.

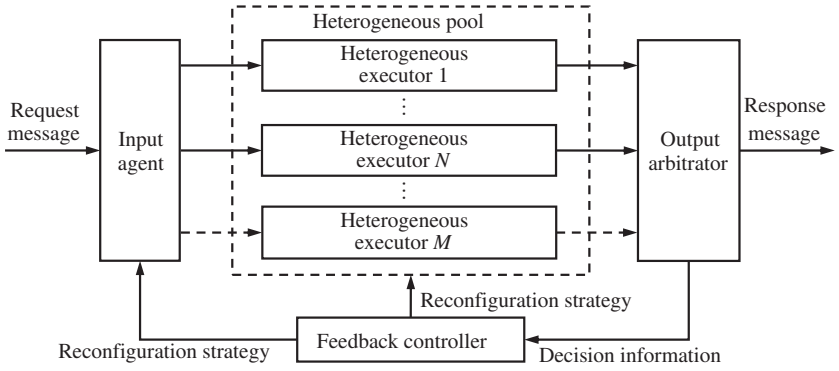


图 1　拟态防御总体结构

Figure 1　The CMD framework

这个攻击过程按照攻击粒度, 划分为两个层次: 单节点攻击和链路攻击, 其中, 同一条攻击链上的顺序单节点攻击, 组成了网络链路攻击. 本文使用广义随机 Petri 网络 (generalized stochastic Petri net, GSPN) [17], 结合 Markov 链 [18] 和鞅 [19] 等数学工具分析拟态防御模型, 由节点到链路, 最终分析整个系统的安全性, 包括单节点攻击成功率、节点稳态时间、目标节点失效时间. 分析时, 攻击者针对节点某个执行体特定漏洞的一次攻击动作作为整个攻击过程的原子动作, 是分析的最小粒度. 具体地, 本文首先考虑单节点的安全性, 即攻击者在一定时间内攻击单节点成功的概率和用时, 然后将其传递给链路攻击层, 作为输入的攻击强度参数, 进一步计算链路攻击成功的概率. 通过层次化分析, 本模型先捕捉系统细节保证建模可信度, 再进行抽象数学计算, 保证可迁移性.

本文的其余部分安排如下. 第 2 节介绍使用的拟态系统模型、假设和符号; 第 3 节介绍两层次安全分析数学模型和方法; 第 4 节, 以一个典型的拟态系统为例, 对其安全性进行分析; 然后在第 5 节以具体系统为例应用该分析方法, 并用 SPNP 软件进行模拟; 最后在第 6 节得出结论和阐述进一步研究方向.

2　问题描述

拟态旨在建立一个相对安全的攻击防御系统, 这个系统具有动态、冗余、异构的特点. 本文以拟态防御的经典实现结构 —— MDA 结构为例, 分析拟态防御的安全性, 再由多个 MDA 结构组成最终的拟态防御网络. MDA 的基本结构如图 1 所示.

MDA 结构由输入代理器、执行体异构池、执行体集、反馈控制器和输出裁决器即多模表决器构成. 多个功能等价且实现方式相异的执行体即异构功能等价体构成该结构的执行体异构池. 异构池中的异构体不需要同时持续处于工作状态, 输入代理选择 N 个 (一般为 3 个) 执行体组成一个当前工作的执行体集合, 其他执行体则作为备用. 当系统接收到请求消息, 输入代理下发指令, 为当前执行体集合分配相同的功能需求, 由工作状态的执行体执行任务, 经过一定执行时间, 分别将输出矢量作为执行结果发送给输出裁决器. 输出裁决器收集到足够数量的结果以后进行择多判决. 若收集到的输出矢量完全一致, 多模表决器将该结果视为正确并输出; 反之, 输出裁决器将激活防御机制. 以三执行体为例, 本文定义 4 种系统行为如下.

定义1 (驱逐 (d))　当某个执行体发送与其他执行体不一致的输出矢量时, 系统将该执行体标为可疑执行体, 停止其工作任务清洗后放入异构池, 并从拟态异构池中重新选择一个没有被使用过或确

认未遭受攻击的执行体继续工作. 如果该执行体是出错执行体, 那么这个动作称为驱逐.

定义2 (误驱逐 (m)) 当某个执行体输出与其他执行体不一致的输出矢量, 但这种情况是由多数执行体被攻击且输出矢量一致, 与正常执行体相异造成的, 此时系统将正常执行体当作可疑执行体进行停用, 这种防御动作称为误驱逐. 误驱逐会导致拟态异构池中的资源消耗. 值得注意的是, 若连续两次对同一异构组件执行了误驱逐操作, 即在驱逐动作完成后, 新上线执行体与被驱逐执行体结果一致且与原多数相同结果的执行体输出不一致, 系统标记原多数执行体为可疑执行体, 并执行驱逐操作.

定义3 (停用 (s)) 当 3 个执行体的结果各不相同, 裁决器无法输出结果, 此时系统标记全部 3 个执行体为可疑执行体并进行停用, 从异构池中选择 3 个新的执行体代替他们的工作.

定义4 (判决 (j)) 当 3 个执行体对同一个任务都执行完毕, 裁决器将对比收到的 3 个输出矢量, 若一致, 则直接输出结果; 若两个一致, 另一个不同, 那么判断两个一致的结果为真, 并将输出不同结果的执行体标为可疑执行体, 并执行驱逐.

不同的攻防行为会使系统进入不同的状态, 本文根据不同的攻击结果定义 5 种系统状态如下.

定义5 (正常工作 (A)) 攻击者并未发动攻击或无任何攻击奏效, 所有执行体都正常运行.

定义6 (非特异性感知 (B)) 攻击者攻击一个执行体成功, 但系统在择多判决时发现该执行体与其他执行体的输出结果不一致, 并替换被入侵执行体, 攻击失败.

定义7 (磨损 (C)) 攻击者攻击两个执行体成功, 但无法控制其出现相同的错误输出; 或者攻击者攻击 3 个执行体成功, 但其输出两两不一致. 此时, 系统得到的 3 个输出结果各不相同, 无法判决. 虽然进入本状态没有使更多的在线执行体失效, 但是消耗了大量异构池中的资源.

定义8 (攻击扩散 (D)) 攻击者成功攻击大部分执行体, 并使得超过半数执行体输出相同的错误矢量, 这时, 判决器驱逐其他执行体. 但系统的驱逐操作不仅没有把攻击者有效地清理出系统, 还额外消耗了异构池中的资源, 造成攻击扩散.

定义9 (攻击逃逸 (E)) 如果攻击者的攻击能力足够强且攻击速度足够快, 在拟态防御系统进行择多判决前攻击全部执行体成功且均产生相同的错误输出, 那么攻击者攻击逃逸成功, 即在不被防御方发现的情况下取得了本节点的控制权. 此时裁决器判断该输出正确, 并允许这些被入侵的执行体继续工作.

显然, 在上述 5 种系统状态中, 攻击逃逸在本轮结果输出后无法及时被系统捕捉到, 会长期在系统中潜伏下去. 直到由于随机扰动替换掉其中部分被攻破的执行体, 该状态才会退化成攻击扩散状态, 被防守者觉察. 所以, 该状态对防守者来说是最危险的情况.

3 系统模型

基于上述网络模型, 本文对 MDA 的攻防过程进行抽象. 图 2 展示了本文的双层分析模型的结构.

在一个完整的拟态局域网中, 攻击者首先需要从外网进入拟态网络, 攻击某个目标节点. 在攻击目标节点前, 攻击者需要逐个攻击入侵节点到目标节点之间的链路节点, 这些链路节点构成了攻击链, 均为拟态结构节点. 在攻击每个链路节点时, 攻击者需要攻破其多个异执行体, 并产生相同的结果. 所以将攻击网络中某个特定节点的过程依据粒度划分为两个层次: 单节点攻击和链路攻击. 其中, 同一条攻击链上的顺序单节点攻击构成了网络链路攻击. 本文依次对这两个攻击层次建立模型并计算安全参数, 进而分析整个拟态防御系统的安全性, 包括单节点攻击成功率、节点稳态时间和特定目标失效时间.

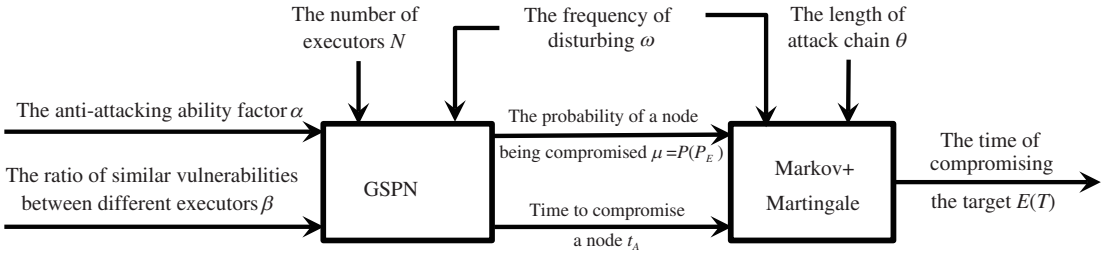


图 2　分析模型结构

Figure 2　The framework of analysis model

3.1　单节点攻击

本文将攻击过程中的某个独立功能部件看作一个节点. 一个节点可能是一台服务器, 一个主机, 一个软件功能进程, 或一个功能集群.

因为实现一个节点功能的 3 个执行体有各自的漏洞, 攻击者若想攻击整个节点, 则必须分别攻击这 3 个执行体. 也就是说, 攻击者可以对工作执行体的配置信息进行探测, 利用每个执行体的漏洞, 针对性地采取扫描、重放、字典攻击、DoS、命令注入等攻击, 提升自身的权限, 进一步发动更多攻击, 直到取得该执行体的控制权. 对于单个节点来说, 取得节点控制权的前提是攻克其多数执行体, 并控制它们的运行结果一致以通过择多判决. 在这一步中, 根据攻击能力的强弱, 本文将各个执行体遭受的单步攻击分为一般攻击 (general attacks) 和特殊攻击 (special attacks), 其在建模中的区别主要体现在赋值过程中, 在第 4 节具体论述.

防御者一方面进行输出矢量的择多判决, 标记、替换可疑执行体; 另一方面, 为了防止高频率大强度的快速攻击, 防御者采取低频率的随机性扰动机制. 随机扰动是指以随机命令方式选择一个执行体进行下线预清洗或策略性重构操作, 在其恢复后等待被调度加入当前工作集. 由于随机扰动机制的存在, 即使待机式协同攻击成功也无法维持可逃逸的状态, 即攻击不可维持.

攻击和防守在这一层形成了一场博弈, 双方不同的动作会对单个节点的输出状态产生不同的影响, 因此本文建立广义随机 Petri 网络模型, 刻画攻防双方不同动作对系统产生的影响, 计算单节点攻击成功的概率、攻击时间等安全评估参数.

通常情况下, 用一个 6 元组表示广义随机 Petri 网络:

一个 GSPN $= (S, T, F, W, M_0, \lambda)$, 其中

(1) S 表示网络位置集 (place), 一般表征系统状态;

(2) T 表示网络变迁集, 一般表征系统行为、动作, 并且分为两个子集: $T = T_t \cup T_i$, $T_t \cap T_i = \emptyset$, $T_t = t_1, t_2, \ldots, t_k$ 和 $T_i = t_{(k+1)}, \ldots, t_n$ 分别表示时延变迁集和瞬时变迁集.

(3) λ 表示变迁的时延速率, 与时延变迁集相关联的平均变迁实施速率集合为 $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_k$.

(4) F 表示 T 和 S 之间的连接弧, 并允许有抑制弧.

(5) W 表示弧对应的权值, 弧起始位置的标记数多于该值, 是该变迁可以实施的必要条件.

(6) M_0 表示初始标记位置.

本文先建立攻击者视角的 GSPN 模型, 如图 3 左图, 攻击方需要对该节点的执行体配置进行嗅探, 有针对性地攻破足够多的执行体, 才能取得该节点控制权. 在攻击者的角度, 单个节点系统具有以下状态: 正常工作、1/2/3 个执行体被攻破. 但只有在一轮攻击结束. 通过观察输出矢量, 攻击者才能知道自己攻击的结果. 另外, 攻击方的模型包括对执行体进行攻击时攻克的漏洞, 带来的权限提升, 所以

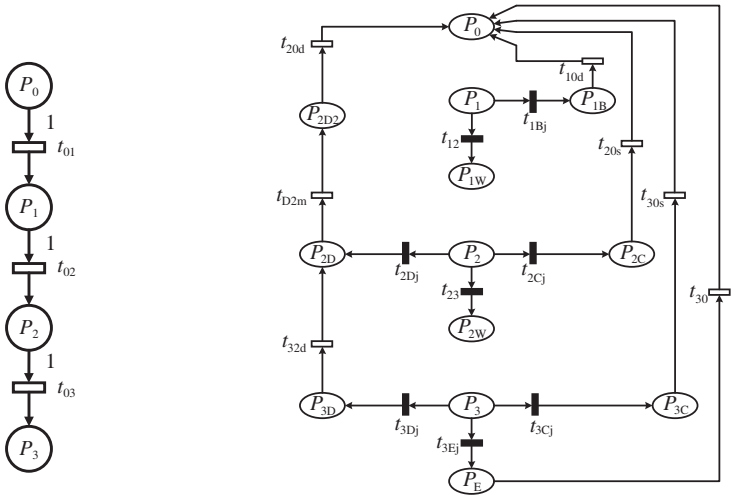


图 3 攻防视角的 GSPN 网络

Figure 3 The GSPN model in perspectives of the attacker and defender

它与每个执行体的具体配置以及攻击目标有关, 漏洞带来的影响我们在第 4 节进行讨论.

至于防守方, 单个节点系统具有以下状态: 正常工作、1/2/3 个执行体被攻破等待判决、经判决后被攻破的 2 个执行体输出同样/不同的错误矢量、经过判决后被攻破的 3 个执行体输出完全一致/部分一致/两两不同的错误矢量. 防守者通过拟态判决和后续反馈控制操作, 完成系统在不同状态间的转换. 防守方视角的 GSPN 模型如图 3 右图所示, 其中各位置和变迁的含义与左图的相同, 在下文中给出解释.

综合攻击者和防守者的视角, 可以得到综合的攻防 GSPN 模型. 攻击者可以对几个执行体并行发起攻击. 由于攻破各个执行体所需的时间不同, 所以按照攻击完成的时间, 可以对攻击者攻击成功的执行体进行排序. 以三执行体为例, 攻击者攻击成功的次序有 6 种排列. 为了简化分析过程, 本文忽略不同的攻击完成顺序带来的影响. 假设执行体 1∼3 被顺序攻破, 简化的 GSPN 网络结构图, 如图 4 所示.

对于图 4 中的 Petri 网, 位置用圆圈表示, 记为 P_{iX}, 其中 i 是数目位, 表示被攻破的执行体数目, X 是状态位, 表示被攻破的执行体整体呈现的状态. P_0 表示系统正常运行: $P_1/P_2/P_3$ 分别表示 1/2/3 个执行体被攻击成功; $P_{i\mathrm{B}}$ 表示非特异性感知状态, 也就是 i 个执行体受到了攻击, 但由于错误矢量不一致, 在经过择多判决后, 被全部捕捉到; $P_{i\mathrm{C}}$ 表示系统磨损, 也就是 i ($i \geqslant N/2$) 个执行体受到了攻击, 但是输出不一致, 导致系统无法判决, 从而把所有执行体都标为被攻击执行体; $P_{i\mathrm{D}}$ 表示攻击扩散, 也就是 i ($i \geqslant N/2$) 个执行体受到了攻击后, 控制了大部分执行体输出一致的矢量, 导致择多判决发生误判, 系统将正确的或者也被攻击但是出错不一致的少数执行体标为被攻击执行体; $P_{i\mathrm{W}}$ 表示攻击者对执行体 i 的攻击完成, 其攻击目标转移为执行体 $i+1$; P_E 表示攻击者控制了全部执行体, 并且输出同样的错误矢量, 导致系统无法通过择多判决找到出错的执行体.

图 4 中方块表示防御者的行为, 其中黑色实心方块表示瞬间变迁, 对应参数为转移概率 p; 黑色空心方块表示时延变迁, 对应参数为转移速率 λ. 攻击者和防御者行为的不同导致了系统在不同状态之间的转换. 变迁记为 t_{ijx}, i, j 分别表示变迁的起始位置和终止位置, 若两个位置间被攻破执行体数目发生了改变, 那么 i, j 分别表示改变前后的数目; 若没有发生改变, 则用该位置的末位即状态位表示. x 是动作位, 表示攻防方的主要动作, 可缺省. 若状态位缺省, 即 t_{ij}, 则表示攻击者已经攻击执行体 i 成功, 转而攻击执行体 j. 存在的状态位包含以下几种: d 表示驱赶 (drive), m 表示误驱赶 (miss

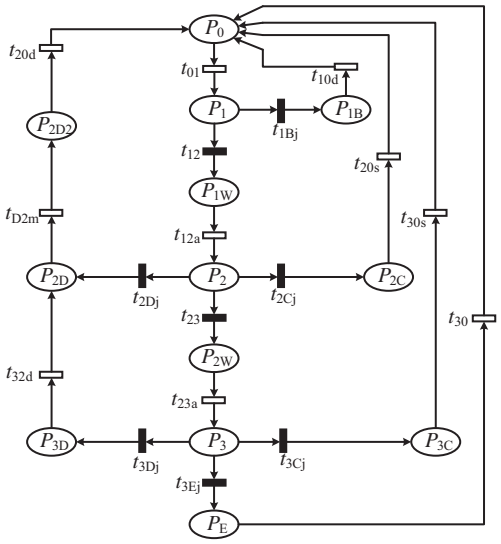


图 4　简化 GSPN 网络

Figure 4　The simplified GSPN model

drive), s 表示停用 (stop), j 表示判决 (judge), a 表示攻击 (attack). 例如, t_{20s}, 表示在两个执行体受到攻击后, 防御者对可疑执行体进行了停用, 将 3 个可疑执行体都替换成从异构池中取出的健康执行体, 从而使系统恢复到没有执行体受到攻击的状态; t_{ija} 处表示攻击者对每个执行体的具体行为, 这和执行体配置有关. 第 4 节会以一个典型系统为例进行分析.

建立了完整 GSPN 模型后, 本文可以设置各个变迁的参数, 参数的设置在第 4 节具体阐述. 根据完整的 GSPN 网络可以推导其可达树, 构造其同构 Markov 链, 从而推算得到最后的单节点攻击成功率和节点稳定时间.

3.2　链路攻击

完整的网络链路由许多节点构成, 因此若想攻击链路中的某个节点, 攻击者需要先依次攻击该链路上的各个节点. 本文取单个节点的稳态时间作为系统的攻击周期. 攻击者沿攻击链推进, 每攻击成功一个节点, 则攻击者沿攻击链下行一步, 若遭遇了拟态随机扰动, 便沿着攻击链退行一步. 这种知道上一步的状态和下一步的攻击范围, 找寻攻击停留位置的做法, 和 Markov 链的特征吻合. 所以采用 Markov 链和鞅对该部分建模求解.

令攻击单个节点成功的概率为 μ, 攻击链的节点总数为 θ, 系统在该时刻选择该节点进行随机扰动的概率为 ω. 假设当前时刻攻击者停留在第 k 个节点, 即已攻击成功 k 个节点, 那么攻击转移图如图 5 所示.

根据文献 [11], 可以将上述 Markov 链转换成鞅进行求解. 对于一条长度为 θ 个节点的攻击链, 如果攻击者攻击单个节点成功的概率为 μ, 单节点处遭遇主动随机扰动的概率为 ω, 那么攻击者成功攻击目标节点 (即 θ 点) 需要的步数期望为

$$\mathrm{E}[S]=\frac{\theta}{[(1-\omega)\mu-\omega]},\quad \omega\neq\mu/(\mu+1). \tag{1}$$

该计算结果有可能出现负值, 当 $\omega<\mu/(\mu+1)$ 时, 计算结果为正值, 表示攻击者沿攻击链路前进 θ 个节点时, 需要的攻击步数; 当 $\omega\geqslant\mu/(\mu+1)$ 时, 计算结果为负值, 表示攻击者由于受到随机扰动,

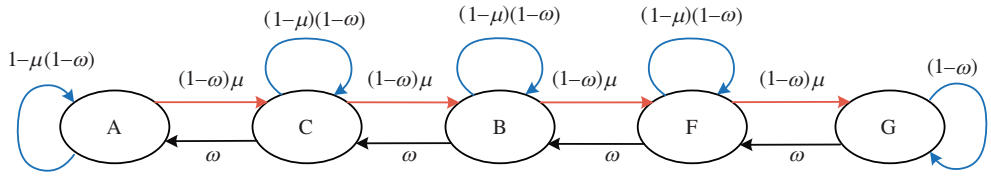


图 5 (网络版彩图) 马尔科夫链

Figure 5 (Color online) The Markov chain

沿攻击链路被退行 θ 个节点时, 需要经过的攻击步数.

根据攻击步数和系统周期, 可以求得目标节点失效时间.

4 系统分析

在第 4 节和第 5 节中本文以一个实际的攻击链为例, 对包括攻击中的各个节点以及各节点的执行体构成进行分析.

假设以下情况, 攻击从外部网络进入, 攻击目标是窃取一条链路中各节点数据库中的文件并安插后门, 一共有 10 个链路节点, 即 $\theta = 10$. 对于每个链路节点, 攻击者对节点的操作系统、服务器以及数据库分别展开攻击, 直到窃取数据并成功安插后门.

执行体以操作系统 * 前端语言 * 数据库为例, 攻击者根据扫描到的信息确定攻击执行体的顺序, 防守者根据失效执行体的信息选择替换的执行体, 二者之间形成博弈. 本文以广义随机 Petri 网络进行描述. 在设计时延函数的时候本文用到了一个很重要的数据库: 通用漏洞评分系统 (common vulnerability scoring system, CVSS) [20]. CVSS 是一个被设计用来评测漏洞的严重程度, 并帮助确定所需反应的紧急度和重要度的 “行业公开标准”, 一般与公共漏洞和暴露 (common vulnerabilities & exposures, CVE) [21] 一同由美国国家漏洞库 (National Vulnerability Database, NVD)[1)] 发布, 目标是为所有软件安全漏洞提供一个严重程度的评级, 建立衡量漏洞严重程度的标准, 使漏洞的严重程度可以互相比较, 从而确定处理它们的优先级.

CVSS 得分基于一系列维度上的测量结果, 这些测量维度被称为量度 (metrics). 漏洞的最终得分最大为 10, 最小为 0. 在基准分数 (base score, BS) 中有几个衡量要素: 攻击途径 (access vector, AV)、攻击复杂度 (access complexity, AC)、特权 (privileges required, PR)、交互 (user interaction, UI)、范围 (scope, S)、机密性 (confidentiality impact, CI)、完整性 (integrity impact, II)、可利用性 (availability impact, AI). 依据它们可以计算出两个核心评估值: 可探测性 (exploitability sub score, ES) 和可利用性 (impact sub score, ISC), 其中可探测性衡量该漏洞被发现的难易程度, 而可利用性表示该漏洞被利用后的影响力. 最终得到对该漏洞的评估分数, 基础得分具体计算公式如下:

$$\mathrm{ES} = 8.22 \times \mathrm{AV} \times \mathrm{AC} \times \mathrm{PR} \times \mathrm{UI}, \tag{2}$$

$$\mathrm{ISC} = 6.42 \times 1 - [(1 - \mathrm{CI}) \times (1 - \mathrm{II}) \times (1 - \mathrm{AI})], \tag{3}$$

$$\mathrm{BS} = \mathrm{Roundup}(\mathrm{Minimum}[(\mathrm{ISC} + \mathrm{ES}), 10]). \tag{4}$$

具体推算可参考文献 [15].

1) NVD. National vulnerability database v2. 2019. http://nvd.nist.gov/.

表 1　系统变迁表格

Table 1　Transitions in the GSPN model

Transitions	t_{01}	$t_{1\mathrm{Bj}}$	t_{12}	$t_{2\mathrm{Dj}}$	$t_{12\mathrm{a}}$	$t_{10\mathrm{d}}$	$t_{2\mathrm{Cj}}$	t_{23}	$t_{23\mathrm{a}}$
General attacks	1	0.4883	0.5117	9.093×10^{-5}	1	1	0.9093	0.0907	1
Special attacks	1	0.3032	0.6968	6.988×10^{-5}	1	1	0.6987	0.3012	1
Transitions	t_{D2m}	$t_{20\mathrm{d}}$	$t_{20\mathrm{s}}$	$t_{3\mathrm{Dj}}$	$t_{3\mathrm{Cj}}$	$t_{3\mathrm{Ej}}$	$t_{32\mathrm{d}}$	$t_{30\mathrm{s}}$	t_{30}
General attacks	1	1/2	1/3	2.9997×10^{-4}	0.9997	1×10^{-8}	1	1/3	0.0001
Special attacks	1	1/2	1/3	2.9997×10^{-4}	0.9997	1×10^{-8}	1	1/3	0.0001

4.1　GSPN 建立

本文现在描述一个拟态节点, 这个节点的异构有 3 个层级, $C1\times C2\times C3=\{\{$Redhat Linux 6.0, Windows 7$\}\times\{$Go 1.6, Python 3.0$\}\times\{$MySQL 5.1.7, PostgreSQL 9.6$\}\}$. 也就是说这个节点在操作系统、语言、数据库 3 个层面进行了异构, 其中, 操作系统在 Redhat Linux 6.0 (以下简称 linux), Windows 7 之间选择, 语言在 Go 1.6, Python 3.0 之间选择, 数据库在 MySQL 5.1.7, PostgreSQL 9.6 之间选择. 所以整个异构池中的有效配置可以表示成 $C=\{$(Linux, Go, MySQL), (Linux, Go, PostgreSQL), (Linux, Python, MySQL), (Linux, Python, PostgreSQL), (Windows 7, Go, MySQL), (Windows 7, Go, PostgreSQL), (Windows 7, Python, MySQL), (Windows 7, Python, PostgreSQL)$\}$, 共 8 种有效配置.

对于操作系统, 本文只考虑来自相邻网络不需要授权没有用户交互的严重漏洞. 其中, Windows 7 有 2 条, Linux 有 1 条. 对于程序语言和数据库, 本文考虑来自网络不需要授权并且不需要其他组件交互的漏洞, 其中, Python 3.0 有 4 条, Go 1.6 有 6 条, PostgreSQL 9.6 有 6 条, MySQL 5.1.7 有 2 条.

执行体之间结构差异越大, 整个系统越安全, 所以本文从异构池中选择异构体配置如下: $\{$(Linux, Go, MySQL), (Windows 7, Python, MySQL), (Linux, Python, PostgreSQL)$\}$.

所以根据第 3 节, 忽略不同攻击顺序带来的影响, 假设攻击完成顺序为执行体 1∼3, 本文得到整个拟态节点的攻防 GSPN 网络, 建立 GSPN 模型比较大, 如附录 A 所示.

4.2　GSPN 赋值

表 1 展示了部分变迁的参数, 具体漏洞攻击部分变迁数目过多, 篇幅限制以文字表述.

(1) 防守行为赋值. 对于防守者来说, 变迁的时延与变迁涉及的执行体个数有关, 例如, 误驱逐、驱逐动作一般只针对单个执行体, 所以时延为 1, 对应 $\lambda=1$; 停用动作一般针对 3 个执行体输出矢量各不相同, 裁决器无法进行结果输出的情况, 会将 3 个执行体都视为可疑执行体并全部停用, 所以时延为 3, 对应 $\lambda=1/3$. 根据文献 [5], 本文取两个异构执行体出现相同错误的概率为 0.0001.

(2) 随机开关赋值. 攻击者在攻击执行体的同时, 其他未被攻破的执行体在正常运行计算正确输出矢量, 攻击者的攻击时间越长, 其他执行体输出正确输出矢量的概率越大; 裁决器收集到的正确输出矢量越多, 收集到足够多输出矢量进行拟态裁决的概率也越大. 随着攻击的进行, 攻击成功的执行体数目与攻击时间也呈正相关关系.

对于未受攻击的执行体, 若执行任务时间为 t_w, 那么在接收任务到输出结果这段时间内, 各执行体的结果输出次数服从 Poisson 分布, 但在产生结果输出后就停止该过程. 易知参数 $\lambda=1/t_w$. 用 $N(t)$ 表示从任务分发开始的 t 时间间隔内执行体进行结果输出的次数 (输出 1 次后停止).

第 3 节提到, 以特殊攻击和一般攻击为例研究 CMD 节点在不同攻击强度下的抗攻击能力. 一般攻击的攻击能力较弱, 且无累加效应, 其攻击下一个执行体的耗时不会受到之前攻击行为的影响. 特殊

攻击是一种攻击能力较强, 且有累加效应的攻击, 如果在后续攻击中遇到之前攻击过的相同漏洞, 攻击速度会加快. 所以攻击者在攻击过程中, 攻击下一个执行体的耗时会降低. 我们设置参数 α 和 β, 分别用于刻画攻击者的攻击能力和学习能力. 首先不同类型的攻击操作复杂度不同, 木马攻击、注入攻击的引发速度和正常操作执行时间相近, 所以假设攻击一个执行体的时间为 αt_w, 其中 $\alpha \in [0.8, 1.2]$. 其次, 攻击者学习能力不同, 带来的攻击下一个执行体的耗时折扣不同, 如果攻击第一个执行体耗时为 t_w, 记攻击第 n 个执行体的攻击时间为 $\beta^{n-1}t_w$, 其中, $\beta \in [0.5, 1]$ [9]. 篇幅限制, 我们取极值为一般攻击和特殊攻击, 进行对比实验. 对于一般攻击取 $(\alpha = 1.2, \beta = 1)$, 特殊攻击取 $(\alpha = 0.8, \beta = 0.5)$.

对于一般攻击, 在攻击者攻克了一个执行体后, 正常工作的执行体已完成结果输出的概率为

$$P_{1M} = P\{N(1.2t_w) - N(0) \geqslant 0\} = 1 - \mathrm{e}^{-1.2t_w \times \frac{1}{t_w} \frac{(\lambda t)^0}{0!}} \approx 0.6988. \tag{5}$$

所以, 一般攻击在攻击一个执行体完成后, 其他两个正常工作执行体都已经完成结果输出, 即系统开展拟态判决 ($t_{1\mathrm{Bj}}$) 的概率为

$$P_{1M}^{J} = P_{1M} \times P_{1M} \approx 0.4883. \tag{6}$$

类似地, 一般攻击在攻击完成两个执行体 (用时 $2.4t_w$) 后, 另一个正常工作的执行体已完成结果输出, 即系统开展拟态判决 ($t_{2\mathrm{Cj}} + t_{2\mathrm{Dj}}$) 的概率为

$$P_{2M}^{J} = P\{N(2.4t_w) - N(0) \geqslant 0\} = 1 - \mathrm{e}^{-2.4t_w \times \frac{1}{t_w} \frac{(\lambda t)^0}{0!}} \approx 0.9093. \tag{7}$$

由于特殊攻击强大的攻击能力, 和攻击累加效应, 攻击者在攻击过程中, 未被攻击的执行体输出正确结果, 导致其遇到拟态判决的概率相较于一般攻击有所降低. 根据前文的论述, 特殊攻击攻破第 1 个、第 2 个执行体的时间分别为 $0.8t_w$ 和 $1.2t_w$. 对于特殊攻击, 在攻击者攻克一个执行体后, 正常工作的执行体已完成结果输出的概率为

$$P_{1M}^{S} = P\{N(0.8t_w) - N(0) \geqslant 0\} = 1 - \mathrm{e}^{-0.8t_w \times \frac{1}{t_w} \frac{(\lambda t)^0}{0!}} \approx 0.55067. \tag{8}$$

所以, 特殊攻击在攻击一个执行体完成后, 其他两个正常工作执行体都已经完成结果输出, 即系统开展拟态判决 ($t_{1\mathrm{Bj}}$) 的概率为

$$P_{1M}^{JS} = P_{1M}^{S} \times P_{1M}^{S} \approx 0.3032. \tag{9}$$

类似地, 特殊攻击在攻击完成两个执行体 (用时 $1.2t_w$) 后, 另一个正常工作的执行体已完成结果输出, 即系统开展拟态判决 ($t_{2\mathrm{Cj}} + t_{2\mathrm{Dj}}$) 的概率为

$$P_{2M}^{JS} = P\{N(1.2t_w) - N(0) \geqslant 0\} = 1 - \mathrm{e}^{-1.2t_w \times \frac{1}{t_w} \frac{(\lambda t)^0}{0!}} \approx 0.6988. \tag{10}$$

再结合各种情况的判断概率, 可以完整表 1 中随机开关的概率参数.

(3) 攻击行为赋值. 广义随机 Petri 网络防守者可以根据自己对安全性的需求灵活设计时延函数, 比如如果防守者比较重视漏洞的影响力, 可以将时延设为 1/ISC; 如果防守者要把影响力和可探测性同时纳入考虑, 可以设计时延函数为 1/BS; 还可以根据需求添加对环境、时间的影响, 灵活调控.

对于攻击具体执行体漏洞的速率, 也分攻击类型进行讨论. 对于一般攻击, 因为攻击者在 T 时刻的攻击对在 $T + X$ 时刻的攻击无协同累积的影响, 所以本文设一般攻击的变迁时延只与攻击难度 ES

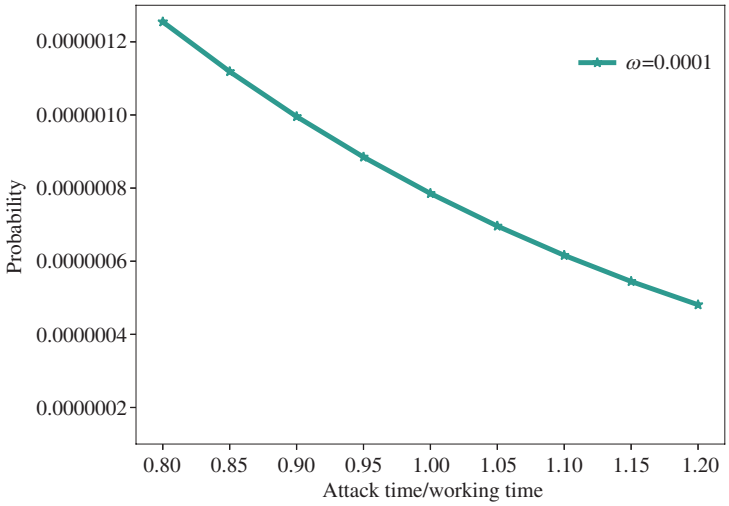


图 6 (网络版彩图) 攻击时间/执行时间倍率 vs. 攻击逃逸概率

Figure 6 (Color online) Attacking time/working time vs. probability of exfiltration

成反比, 与攻击阶段无关, 所以这里时延变迁的参数 $\lambda = 1/\mathrm{ES}$. 对于特殊攻击, 由于攻击者在 T 时刻的攻击对在 $T+X$ 时刻的攻击的协同累积影响, 攻击者在攻击到相同的漏洞时, 会吸取上一次的经验, 导致攻击速度增快, 攻击成功率升高, 本文设其第 1 次遇到的漏洞实验速率为 $\lambda_1 = 1/\mathrm{ES}$, 接下来每遇到一次该漏洞, 攻击速度提升一倍, 即 $\lambda_2 = 2/\mathrm{ES}$, $\lambda_3 = 4/\mathrm{ES}$.

一般攻击和特殊攻击的区别主要体现在攻击过程和攻击速度以及遭遇系统结果输出的概率上, 也就是说一般攻击和特殊攻击对 GSPN 模型的影响是结构及参数的设定不同.

5 实验仿真及结果分析

本文用开源 GSPN 分析工具随机 Petri 网软件包 (stochastic Petri nets package, SPNP) [22], 对上述例子进行模拟分析.

本节先以一般攻击为例, 探索前期条件假设对拟态单节点安全性的影响, 然后探寻随机扰动频率对一般攻击和特殊攻击下的攻击逃逸概率、系统到达稳定的时间带来的影响, 最后选取不同的攻击场景, 分析系统总体安全性即目标节点失效时间, 并给出在日常防御下和受到攻击场景下的应对措施建议.

5.1 单节点攻击成功率分析

本文取随机扰动频率 $\omega = 0.0001$, 对前期假设条件攻击倍率 (α) 在 $0.8t_w \sim 1.2t_w$ 之间每隔 $0.05t_w$ 取点, 对应的攻击成功率变换情况, 如下所述.

根据图 6, 攻击成功率随攻击速度的下降而下降, 但下降幅度不大, 也就是说攻击的攻击速度、学习能力的增强, 带来的攻击逃逸概率增大, 只是同一量级上的细微变化, 而不会带来总体安全性量级的改变. 这也侧面说明, 系统安全性的增强是由拟态系统的结构带来的, 前文的假设条件带来的细微差距可以忽略不计.

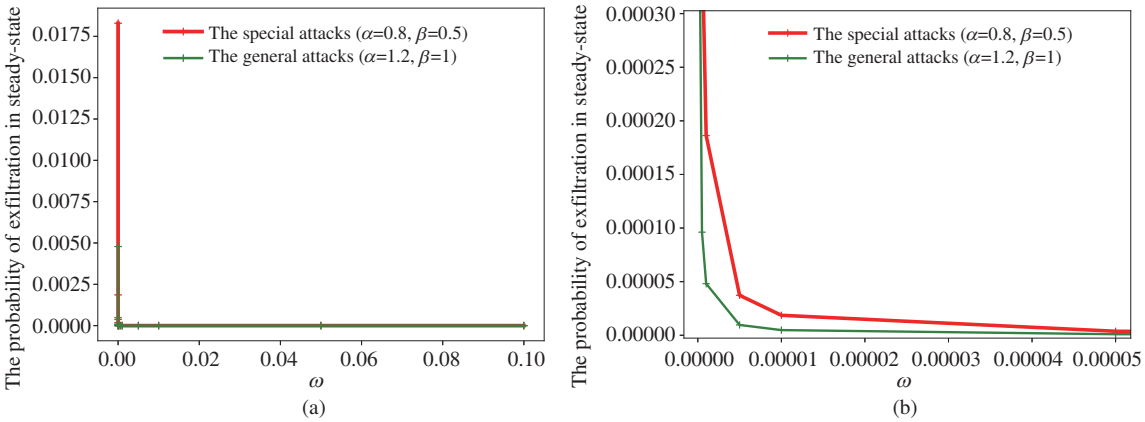


图 7 (网络版彩图) 随机扰动频率 vs. 攻击逃逸概率. (a) 总图; (b) 细节图

Figure 7 (Color online) Frequency of disturbing & probability of exfiltration. (a) General figure; (b) detailed figure

5.2 一般攻击 & 特殊攻击下的单节点攻击成功率分析

考虑到只要没有发生攻击逃逸, 防守者都可以发现攻击行为, 攻击者无法毫无声息地在系统中潜伏, 因此我们改变随机扰动频率, 探索系统安全性和变换频率之间的关系, 以供防御者根据安全性和系统性能的不同需求灵活选择参数.

变换曲线如图 7 所示, 其中红线表示特殊攻击, 绿线表示一般攻击, 因为变换频率数值非常小且选点密集, 所以图 7(a) 拐角处非常尖锐, 把拐角处局部放大, 得到图 7(b).

观察图 7, 可以看出, 对于同样的随机扰动频率, 特殊攻击因为学习能力, 攻击力更强, 攻击速度更快, 所以攻击逃逸概率比一般攻击要高. 总体上看, 无论一般攻击还是特殊攻击, 攻击逃逸概率随随机扰动频率升高而降低.

5.3 单节点稳态时间分析

我们取不同的随机扰动频率, 测试在一般攻击和特殊攻击下系统达到稳定的时间, 作为试验, 本文取 $\omega = 0.0000001$, 根据以上假设, 选取 “秒” 作为随机扰动的最小时钟周期单位, 即每隔 1 秒系统以概率 ω 进行一次随机扰动, 那么此时系统平均每年采取 3 次主动随机扰动. 其逃逸概率在一般攻击、特殊攻击两种应用场景下, 随时间变换规律如图 8 所示. 根据图 8, 我们可以得到系统到达稳态概率的时间, 其中, 绿色线为一般攻击下的场景, 红色线为特殊攻击下的场景.

一般攻击本文执行到 4×10^7 秒时陷入 P_E 的概率达到此时的稳态概率. 特殊攻击系统达到稳态的时间约为 3.5×10^7 秒.

取 $\omega = 0.00005$ 即每隔 5.56 个小时采取一次主动随机扰动, 此时一般攻击 P_E 稳态概率为 9.6190×10^{-7}, 特殊攻击的 P_E 稳态概率为 3.7267×10^{-6}.

在这种情况下, $P(P_E)$ 随时间变化曲线如图 9 所示. 在这个例子中, 在受到一般攻击时大约需要 90000 秒后系统达到稳定状态, 以 9.62×10^{-7} 的概率陷入状态 P_E; 在受到特殊攻击时, 大约需要 70000 秒后系统达到稳定状态, 以 3.72×10^{-6} 的概率陷入状态 P_E.

由这两个例子可知, 在同样的随机扰动频率下, 一般攻击比特殊攻击的攻击逃逸概率要低, 系统进入稳定的时间要更长.

掌握了这个规律, 我们根据受到的攻击强度, 将攻击场景分为日常防御和被强力攻击下的防御两

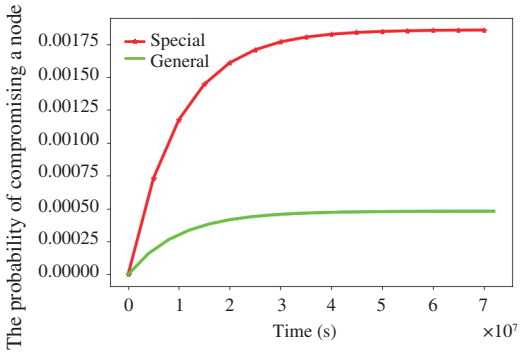


图 8 (网络版彩图) 攻击逃逸概率 ($\omega = 0.0000001$)

Figure 8 (Color online) Probability of exfiltration ($\omega = 0.0000001$)

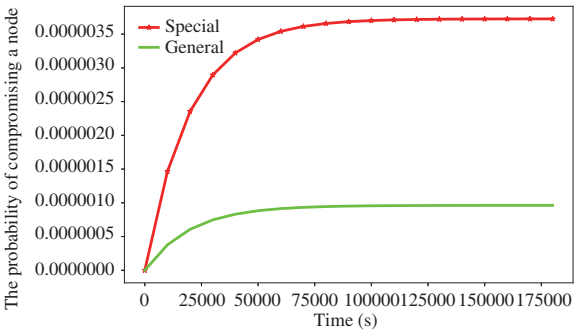


图 9 (网络版彩图) 攻击逃逸概率 ($\omega = 0.00005$)

Figure 9 (Color online) Probability of exfiltration ($\omega = 0.00005$)

种情况, 分别对应受到一般攻击和特殊攻击, 来分析随机扰动频率对系统总体安全性, 即目标节点攻击时间的影响.

5.4 目标节点失效时间分析

5.4.1 日常防御

当系统没有在被攻击或受到低频率低强度一般攻击时, 我们选取比较低的随机扰动频率, 换取一般的安全性, 这里我们取一般攻击, 其 $\omega = 0.000001$, 此时对应的攻击下行概率为 $P(P_E) = 0.000048$. 在这种情况下, 大概经过 4000000 个系统时钟, P_E 可以达到稳定状态. 也就是 $\mu = 0.000048$, $\omega = 0.000001$, 假设攻击链长度为 10, 那么

$$\mathrm{E}[S] = \frac{10}{[(1-0.000001)\times 0.000048 - 0.000001]} = 212766.17. \tag{11}$$

攻击目标节点成功的时间期望为 $\mathrm{E}[T] = 212766.17 \times 4000000 = 8.51 \times 10^{12}$ 个系统时钟.

也就是说对于一条 10 个节点长的攻击链, 攻击者想要成功攻击目标节点, 需要对单个节点完成约 212766 次一般攻击, 而在攻击者攻击单节点的过程中又会遇到很多次被拟态判决发现并且驱逐的可能. 如果只考虑系统达到最大攻击成功率 (P_E 达到稳态概率) 的攻击, 并且防御方不加强防御力度, 一直以 0.0003 的主动随机扰动频率进行变换, 攻击者将完成约 212766 次一般攻击后才能成功攻击被攻击目标.

5.4.2 危机防御

当系统受到高频特殊攻击或有其他需要加强防御时, 我们选取比较高的随机扰动频率, 换取更高的安全性, 这里我们取特殊攻击, 其 $\omega = 0.0005$, 即每 0.55 个小时随机扰动一次, 此时攻击下行概率为 $P(P_E) = 0.00000037$.

在这种情况下, 大概经过约 8000 个系统时钟, P_E 可以达到稳定状态.

也就是 $\mu = 0.00000037, \omega = 0.0005$, 假设攻击链长度为 10, 那么

$$\mathrm{E}[S] = \frac{10}{[(1-0.0005)\times 0.00000037 - 0.0005]} = -20015. \tag{12}$$

攻击目标节点成功的时间期望为 $\mathrm{E}[T] = -20015 \times 8000 = -1.6 \times 10^8$ 个系统时钟.

这时, 可以看到步数出现了负值, 负值表示在攻击链中, 攻击下行概率比上行概率更低, 也就是说理论上讲, 攻击无法沿攻击链下行, 反而会由于一次次随机扰动, 被清理出系统. 这个例子的意思是, 对于一条 10 个节点长的攻击链, 防御者想要将攻击者清理出攻击链, 需要经过攻击者对单个节点完成约 20015 次特殊攻击. 如果只考虑系统达到最大攻击成功率 (P_E 达到稳态概率) 的攻击, 攻击者不可能沿攻击链下行, 并且终将被移出攻击链.

6 总结与展望

本文提出了一种统一又灵活的拟态系统安全性分析模型, 将攻击过程分为攻击者对单节点的攻击和沿着网络链路对特定节点的攻击. 在对单节点攻击的过程中, 我们用广义随机 Petri 网描述攻防双方动作对系统的影响, 并结合 CVSS 和 NVD 数据库, 给出了根据实际系统配置分析其安全性的方法. 在网络链路安全中, 我们采用 Markov 链和鞅寻求攻击难度和网络配置之间的关系. 本文首先通过该模型建立了网络配置和攻击时间之间的量化关系, 可应用于各种单个拟态设备及多个拟态设备组成靶场的安全量化分析, 然后以一个典型的网络场景为例给出参数选择方案, 并分析了其在日常防御下和受到攻击场景下的度量方案.

本文提出了一套理论分析方案, 参数选择方面大多以假设或特例为主, 用户在用以实际系统评测时, 可以针对特定应用场景及特定攻击方法, 合理选择参数. 我们的后续研究将以单个具体拟态设备, 如拟态防火墙、拟态路由器、拟态 Web 服务器、拟态分布式存储及由他们不同配置构成的成套拟态设备为目标进行分析, 一方面展示实际系统中的参数选择方案, 一方面验证本建模的有效性.

参考文献

1 Mandiant Intelligence Center. APT1: Exposing One of China's Cyber Espionage Units. Mandiant, Technical Report, 2013

2 Jajodia S, Ghosh A K, Swarup V, et al. Moving Target Defense. New York: Springer, 2011

3 Zhang Y H. Analysis and research on moving target defense system based on MUTE. Modern Computer, 2015, 4: 15–19 [张艺衡. 基于易变网络的动态化目标防御系统分析研究. 现代计算机, 2015, 4: 15–19]

4 Wu C R, Yan M, Jin H L, et al. A self-transforming proactive defense network framework based on "carrier". J Cyber Secur, 2016, 1: 11–28 [吴承荣, 严明, 金蒿林, 等. 一种基于托架的自蜕变主动防御网络框架. 信息安全学报, 2016, 1: 11–28]

5 Wu J X. Introduction to Cyberspace Mimic Defense. Beijing: Science Press, 2017 [邬江兴. 网络空间拟态防御导论. 北京: 科学出版社, 2017]

6 Zhang Y, Zhang B W. Research progress of security assessment methods for moving target defense systems. Commun Technol, 2018, 51: 1–6 [张莹, 张保稳. 移动目标防御系统安全评估方法的研究进展. 通信技术, 2018, 51: 1–6]

7 Zhuang R, Zhang S, Deloach S, et al. Simulationbased approaches to studying effectiveness of moving-target network defense. In: Proceedings of National Symposium on Moving Target Research, 2012

8 Zhuang R, Zhang S, Bardas A, et al. Investigating the application of moving target defenses to network security. In: Proceedings of the 6th International Symposium on Resilient Control Systems, 2013. 162–169

9 Hong J B, Kim D S. Assessing the effectiveness of moving target defenses using security models. IEEE Trans Depend Secure Comput, 2016, 13: 163–177

10 Richard C, Glass K. Predictive Moving Target Defense. No. SAND2012-4007C. Albuquerque: Sandia National Lab.(SNL-NM), 2012

11 Yang X, Li H, Wang H. NPM: an anti-attacking analysis model of the MTD system based on martingale theory. In: Proceedings of IEEE Symposium on Computers and Communications, 2018

12 Maleki H, Valizadeh S, Koch W, et al. Markov modeling of moving target defense games. In: Proceedings of ACM Workshop on Moving Target Defense, 2016. 81–92

13 German R. Markov regenerative stochastic Petri nets with general execution policies: supplementary variable analysis and a prototype tool. Performance Evaluation, 2000, 39: 165–188
14 Mitchell R, Chen I R. Modeling and analysis of attacks and counter defense mechanisms for cyber physical systems. IEEE Trans Rel, 2016, 65: 350–358
15 Cai G, Wang B, Luo Y, et al. A model for evaluating and comparing moving target defense techniques based on generalized stochastic Petri net. In: Advanced Computer Architecture. Singapore: Springer, 2016. 184–197
16 Moody W C, Hu H, Apon A. Defensive maneuver cyber platform modeling with stochastic Petri nets. In: Proceedings of IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2014. 531–538
17 Lin C, Wang Y Z, Wang Y. Analysis and Evaluation for Network Security Based on Stochastic Game Model. Beijing: Tsinghua University Press, 2011 [林闯, 王元卓, 汪洋. 基于随机博弈模型的网络安全分析与评价. 北京: 清华大学出版社, 2011]
18 Ross S M. Stochastic Processes. Hoboken: Wiley, 1983
19 Li S Y R. A martingale approach to the study of occurrence of sequence patterns in repeated experiments. Ann Probab, 1980, 8: 1171–1176
20 Mell P, Scarfone K, Romanosky S. Common vulnerability scoring system. IEEE Secur Priv Mag, 2007, 4: 85–89
21 Mell P, Scarfone K, Romanosky S. A complete guide to the common vulnerability scoring system, version 2.0. FIRST Forum of Incident Response and Security Teams, 2007. 1–23
22 Hirel C, Tuffin B, Trivedi K S. SPNP: stochastic Petri nets. Version 6.0. In: Lecture Notes in Computer Science. Berlin: Springer, 2000. 1786: 354–357

附录 A

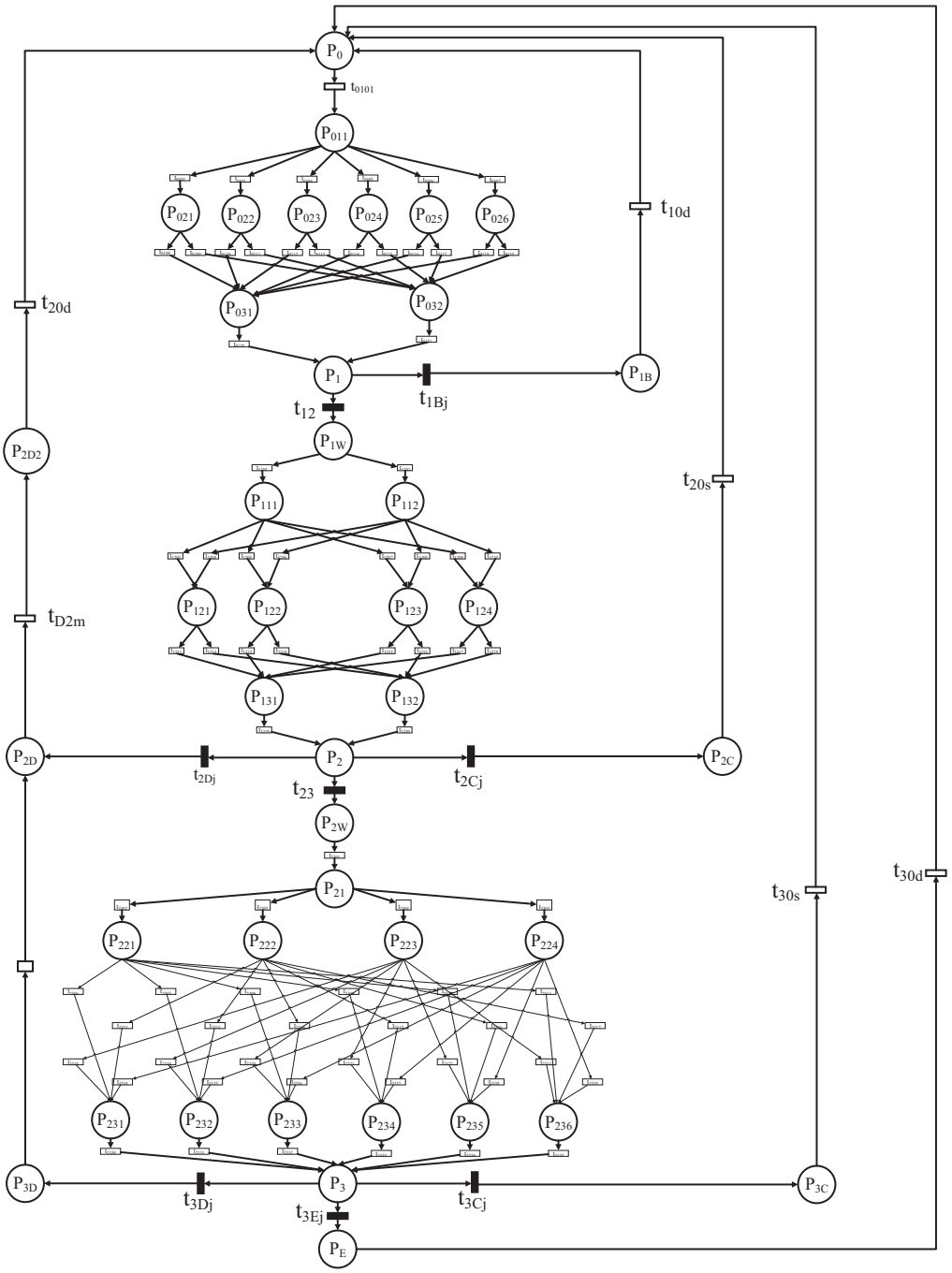


图 A1 拟态节点攻防 GSPN

Figure A1 GSPN model

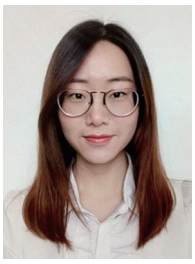
A two-dimension security assessing model for CMDs combined with Generalized Stochastic Petri net

Xin YANG[1,2], Hui LI[1,2*], Jiangxing WU[3] & Peng YI[3]

1. *Shenzhen Graduate School, Peking University, Shenzhen* 518055*, China*;
2. *Peng Cheng Laboratory, Shenzhen* 518055*, China*;
3. *National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou* 450002*, China*
* Corresponding author. E-mail: lih64@pkusz.edu.cn

Abstract Cyber mimic defenses have recently emerged as a dynamic heterogeneity redundancy architecture, which adjust the asymmetry between defenders and attackers by reconfiguring the system according to the network scenario. Some studies have investigated the effectiveness of security models, however, there is still a lack of convincing and practical methods to assess CMD networks quantitatively. Thus, in this paper, we propose a two-dimension model that calculates those details as a digital result to compare different CMD networks. In addition, the proposed method demonstrates good scalability in different networks. Specifically, in the first dimension, i.e., attacking a single node, we elaborate on system configurations and employ the Generalized Stochastic Petri net model to capture the effectiveness of different behaviors from gamers. To quantify the impacts of those behaviors, we parameterized them using a Poisson process, common vulnerabilities and exposures, and the common vulnerability scoring system. In the second dimension, we adopt Markov chains and the Martingale theory to analyze the attack process along the attack chain. Finally, security metrics and countermeasures under different scenarios are presented to verify the effectiveness of CMD, which provides some guidance for designing future systems with acceptable cost.

Keywords cyber mimic defense, generalized stochastic Petri net, common vulnerability scoring system, security analysis, security model



Xin YANG received her B.E. degree from the Department of Computer Science and Engineering, South China University of Technology in 2016. She is currently working toward a Ph.D. degree at the School of Electronics Engineering and Computer Science, Peking University. She is also a student at the Peng Cheng Laboratory. Her research interests include cybersecurity, future network architectures, and distributed storage systems.

Hui LI received his B.E. and M.S. degrees from the School of Information Electronic News Gathering, Tsinghua University, Beijing, China in 1986 and 1989, respectively, and his Ph.D. degree from the Department of Information Engineering, The Chinese University of Hong Kong in 2000. He is currently a Full Professor at the Shenzhen Graduate School, Peking University. His research interests include future network architectures, cyberspace security, distributed storage, and blockchain technologies.

Jiangxing WU was born in Jiaxing, Zhejiang, China in 1953. He received his B.S. degree from the Institute of Engineering and Technology of the PLA in 1982. Since 2003, he is a member of the Chinese Academy of Engineering. He is currently the Lead Director of the National Digital Switching System Engineering and Technological Research and Development Center (NDSC) and a professor at NDSC. His research interests include cybersecurity, high-performance computing, and future internet architectures.

Peng YI is currently a professor at the National Digital Switching System Engineering and Technological Research and Development Center. His contributions encompass security, network architecture, and signal processing.