软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2011,22(6):1398−1412 [doi: 10.3724/SP.J.1001.2011.03819]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于不确定图的网络漏洞分析方法*

刘 强[1+], 殷建平[1], 蔡志平[1], 程杰仁[1,2]

[1](国防科学技术大学 计算机学院,湖南 长沙 410073)

[2](湘南学院 数学系,湖南 郴州 423000)

Uncertain-Graph Based Method for Network Vulnerability Analysis

LIU Qiang[1+], YIN Jian-Ping[1], CAI Zhi-Ping[1], CHENG Jie-Ren[1,2]

[1](School of Computer, National University of Defense Technology, Changsha 410073, China)

[2](Department of Mathematics, Xiangnan University, Chenzhou 423000, China)

+ Corresponding author: E-mail: libra6032009@gmail.com

Liu Q, Yin JP, Cai ZP, Cheng JR. Uncertain-Graph based method for network vulnerability analysis. *Journal of Software*, 2011,22(6):1398−1412. http://www.jos.org.cn/1000-9825/3819.htm

Abstract: Network vulnerability analysis is one of the irreplaceable foundations of network security. Host- centric methods of vulnerability analysis can generate an attack graph in polynomial time, whereas the inherent link uncertainty has not been of a concern. An uncertain-graph based method for network vulnerability analysis is proposed in this paper, which uses link uncertainties to describe link states accurately. In this way, finding an optimal exploit chain becomes feasible. An algorithm for generating an uncertain attack graph (UAG) is proposed, whose running time is $O(n^4)$. Next, a heuristic algorithm to that can generate the optimal exploit chain, on the basis of UAG, is proposed, which runs in $O(n^3)$ time. Experimental results show that this method can generate UAG in an acceptable amount time and find a vulnerability exploit chain with a maximum attack benefit.

Key words: vulnerability analysis; uncertain attack graph; vulnerability exploit chain; attack benefit

摘 要: 网络漏洞分析是提高网络安全性的重要基础之一.以主机为中心的漏洞分析方法可在多项式时间内生成攻击图,但是没有考虑网络链路本身存在的不确定性.提出了一种基于不确定图的网络漏洞分析方法,采用链路不确定度以准确地描述网络链路状态,使得求解最佳利用链成为可能.在此基础上,提出了一种时间复杂度为 $O(n^4)$的不确定攻击图生成算法;基于不确定攻击图提出了一种时间复杂度为 $O(n^3)$的最佳利用链生成启发式算法.实验结果表明,该方法能在可接受的时间内生成不确定攻击图,找到一条攻击效益最佳的漏洞利用链.

关键词: 漏洞分析;不确定攻击图;漏洞利用链;攻击效益

中图法分类号: TP393 **文献标识码**: A

网络漏洞使得网络安全面临严峻挑战,有效的网络漏洞分析对于提高网络安全性具有很重要的现实意义,它能够为攻击路径的预测、僵尸网络的预警等提供良好的技术支持.网络漏洞分析研究主要包括攻击建模、攻

基金项目: 国家自然科学基金(60970034, 61070198, 60903040); 湖南省自然科学基金(06JJ3035); 湖南省教育厅资助科研项目(07C718)

收稿时间: 2009-07-05; 定稿时间: 2009-12-25

击图自动生成和网络漏洞风险评估方法.

攻击建模是一种采用攻击图、形式化描述语言等多种数学工具描述网络中的攻击行为,从而为攻击的检测、预测、评估等提供模型支持的技术.其主要研究点包括:

(1) 传统攻击图模型改进和完善.攻击图是一种描述攻击者从攻击初始状态到攻击目标状态的所有攻击路径的方法,它提供了一种形式化描述攻击行为的途径.文献[1]针对攻击图模型存在可扩展性问题提出了一个权限图的概念,用于网络漏洞分析.作者提出一种最小化方法挖掘网络中的权限提升路径,进而构造出完整的权限图,有效提高了攻击模型的可扩展性;

(2) 在攻击建模中引入人工智能方法.文献[2]提出了一种通用的逻辑框架,用于建模网络配置和网络拓扑.在框架定义基础上,作者把网络中广泛存在的漏洞建模成为通用推理规则,用于专家系统自动推理出攻击者可能采取的攻击路径;

(3) 从形式化建模语言的角度研究更高效的形式化语言来描述网络漏洞.文献[3]提出了一种漏洞描述语言,用于方便自动推理工具处理漏洞信息.通过扩展开放漏洞描述与评估语言(open vulnerability and assessment language)的功能,能够为包括计算攻击可能性在内的更多工作提供支持.

攻击图自动生成技术采用高效的生成算法自动地分析出所有可能的攻击路径,并能有效解决传统人工方式查找最佳攻击路径面临的复杂性高和工作量大的问题.其研究重点包括:攻击图自动生成算法和最佳攻击路径查找两个方面.在攻击图自动生成算法方面,文献[4]提出了一种以网络为中心的模型检验分析方法,通过构造破坏网络正常状态的反实例,最终得出所有可能的漏洞利用路径.以网络为中心的模型检验方法能够得出所有可能的攻击路径,但是由于状态空间大小随网络规模呈指数级别增长,它也面临着可扩展性差的问题.为此,文献[5]提出了一种以主机为中心的模型检验分析方法.该方法基于单调性假设,证明了其耗费随网络规模呈多项式级别增长.在最佳攻击路径查找方面,文献[6]提出了一种有效的、网络规模可变的攻击路径识别方法,它使用攻击面作为系统安全量化度量指标.结果表明,方法适用于大规模网络,并且能够识别出一条最优的攻击路径.文献[7]提出了一种基于攻击能力增长的网络安全分析模型,模型以攻击能力增长为主导,参考网络环境配置,从模拟攻击的角度对网络安全性进行分析,进而生成攻击图.同时,基于攻击图的最小攻击代价分析和最小环境改变分析,预测攻击者最有可能采取的攻击路径.

网络漏洞风险评估是网络漏洞分析的重要组成部分,用于量化评估网络安全状况和攻击威胁程度.如何找到一种简单有效的网络漏洞风险评估方法,一直是安全研究人员关注的焦点之一.文献[8]综述了计算机网络安全评估建模的工作,考察了系统中的各个安全实体与安全要素,不但适用于单机系统的安全分析,也可用于网络信息系统的安全评估;既可采用定性评估的方式,也可通过引入弱点攻击复杂性的概念来进行定量的分析.文献[9]提出了基于 Agent 的漏洞分析框架,框架主要包括用于收集事件的 Agent、用于汇聚事件的事件关联模块、用于构建网络节点连接关系的依赖模块、用于评估漏洞的漏洞分析模块和用于响应的问题分析模块.此外,作者按照节点级、连接级和系统级的顺序分层次地计算漏洞指数,用于评估系统状态.文献[10]也提出了一种量化主机安全性的框架,它分析已观察攻击场景上下文中的漏洞特征以生成主机安全属性.文献[11]研究了网络攻击图模型的建模方法,给出了一种攻击图生成算法.同时,研究了基于攻击图的网络安全评估方法,包括攻击序列成功概率分析方法和网络系统损失风险分析方法,并通过实验对算法的有效性进行了分析验证.文献[12]将基于攻击图的评估和依赖标准的评估结合起来,提出了一种基于安全状态域的网络安全评估模型.该模型通过安全状态域和安全状态域趋向指数反映进入不同安全状态难易程度,实现量化评估网络安全状况.文献[13]则在攻击图生成的基础上定义了两个用于计算总体风险的操作,提出了一种网络漏洞评估算法.

本文提出一种基于不确定图的网络漏洞分析方法,主要贡献是:(1) 考虑到网络链路本身存在的不确定性,本文将不确定图的概念引入到网络漏洞分析之中,可以更为准确地反映真实网络环境,为有效的网络漏洞分析提供前提;(2) 为了降低攻击图生成算法的复杂度,本文基于单调性假设,提出了一种不确定攻击图生成算法,该算法可在多项式时间内生成不确定攻击图;(3) 为了在不确定攻击图的基础上快速得到最佳攻击路径,本文提出一种最佳利用链生成启发式算法,该算法可在多项式时间内得出最佳漏洞利用链,通过实验验证了该算法的

有效性.

本文第 1 节介绍基于不确定图的网络漏洞分析方法,给出不确定攻击图模型的详细定义.第 2 节介绍不确定攻击图生成算法并分析其复杂度.第 3 节介绍最佳利用链生成启发式算法并分析其复杂度.第 4 节是实验部分,通过应用本文方法到不同网络环境之中以及与其他方法的比较来验证本文方法的有效性.第 5 节是总结和未来工作.

1 基于不确定图的网络漏洞分析方法

本文工作涉及到几个内在联系的重要概念,包括攻击图模型、不确定图、不确定攻击图模型等.因此,本节首先介绍攻击图模型的相关内容,其次介绍不确定图的相关概念,最后给出不确定攻击图模型的详细定义.

1.1 攻击图模型

攻击图被形式化为一个有向图 $G=\langle V,E\rangle$,其中:V 是节点集合,每一个节点表示主机的状态;E 是有向边集合,每一条有向边表示一次漏洞利用,攻击者利用特定漏洞提升其在受害主机上的访问权限.主机属性包括以下几个方面:主机提供的服务、软件漏洞、配置漏洞、连通性、可信关系和攻击者在该主机上的访问权限.在单调性假设下,前 5 个属性基本不变.在实现过程中,将每一个节点形式化为(h,a),其中,h 为主机名,a 为攻击者在主机 h 上的访问权限.访问权限分为 3 个级别:无权限、用户权限、Root 权限,分别取值为 0,1,2.一次漏洞利用包括以下几个属性:攻击前件、主机前件、攻击后件、主机后件和漏洞利用模式,其中:攻击前件表示攻击者在源主机上的能力,比如访问权限等;主机前件表示漏洞利用所需的条件,比如目标主机上的服务与漏洞、源主机与目标主机的连通性等;攻击后件表示漏洞利用后攻击者在目标主机上获得的能力,比如获得在目标主机上的访问权限等;主机后件表示漏洞利用后对目标主机的影响,比如关闭端口或停止服务等;漏洞利用模式表示攻击者利用漏洞的方式,分为本地和远程两种.在实现过程中,将每一条有向边形式化为$(name,pre_A,pre_H,post_A,post_H,mode)$,其中,$name$ 为漏洞利用名称,pre_A 为攻击者前件,pre_H 为主机前件,$post_A$ 为攻击者后件,$post_H$ 为主机后件,$mode$ 为漏洞利用模式.图 1 给出了一个攻击图例子,其中,实线表示利用漏洞,虚线表示利用可信关系.从图中可以方便得出到达攻击目标状态的所有攻击路径.

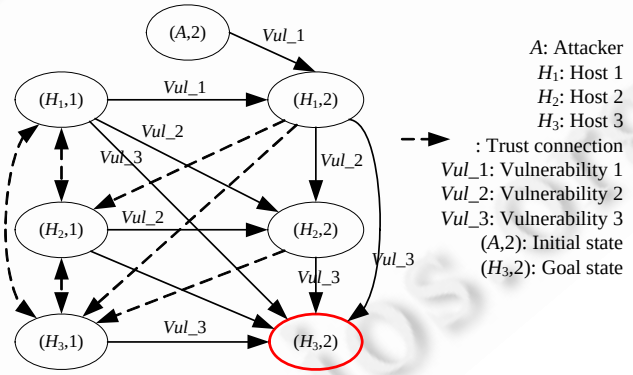


Fig.1 An example of attack graph

图 1 一个攻击图的例子

目前,攻击图建模一般以单调性假设为前提.单调性假设是指一次漏洞利用的前提条件一旦满足,那么其有效性恒为真.在这一假设下,攻击图的生成无需回溯,降低了验证前提条件的计算开销.此外,也无需动态地存储和更新网络状态.

1.2 不确定图

近年来,不确定数据研究受到广泛的关注.不确定图作为一类特殊的不确定性数据,既包括图中节点的不确

定性,也包括图中边的不确定性.本文首次将不确定图与网络漏洞分析技术相结合改善攻击图生成算法的性能.

实际网络环境是时刻变化的,一条链路在当时可用不代表其在未来也是可用的.现有攻击图存在的一个缺陷是只考虑两种链路状态:连接状态和非连接状态,描述网络环境的能力有限.而不确定图能够把链路状态从过去的两种离散状态扩展到部分可用的连续状态,这极大地增强了攻击图的描述能力.同时,在不确定攻击图的生成算法中,使用一个链路不确定度阈值,还可以裁剪确定攻击图中的部分有向连接,达到优化攻击图生成和节省算法开销的目的.以图 1 所示的攻击图为例,当主机 1 和主机 2 之间的链路不确定度下降超过某个阈值时,不确定攻击图生成算法将不考虑主机 1 和主机 2 之间的有向连接.裁剪过程如图 2 所示,由图 2 可以发现,攻击图中的有向边数目得到缩减,优化了性能.

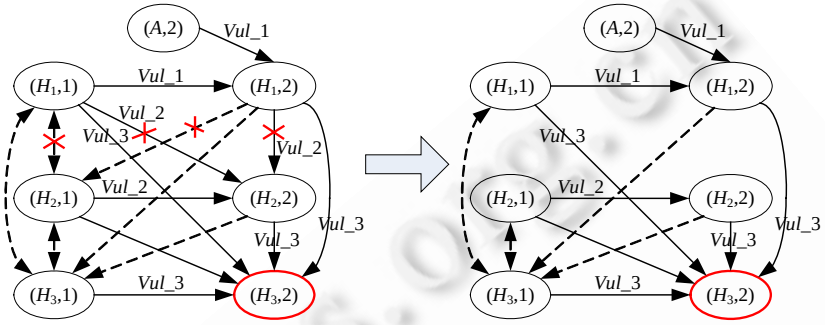


Fig.2 Prune attack graph with link uncertainty threshold

图 2 依据链路不确定度阈值裁剪确定攻击图

综上所述,可以得出以下几点结论:(1) 不确定性数据的应用范围不断扩大;(2) 传统网络漏洞分析中攻击图建模没有考虑图数据的不确定性,在实际应用中可能会影响模型的效果;(3) 考虑攻击图的不确定性可以更为准确地描述网络的实际情况,进而得出更贴近攻击者意图的最佳漏洞利用链.下一节将不确定图的概念与攻击图结合起来,提出了不确定攻击图模型.

1.3 不确定攻击图模型

不确定攻击图模型将不确定图概念与攻击图模型结合起来,用于更精确地建模网络漏洞利用攻击.不确定攻击图模型同样以单调性假设为前提,在实现过程中,将每个节点形式化为(h,a),其中,h 为主机名,a 为攻击者在主机 h 上的访问权限.由于目前不确定图模型主要以概率图作为解决方案,本模型为每一条有向边赋予一个概率属性,表征这两个节点所在主机之间的链路不确定度,链路不确定度受到网络拓扑、链路延迟、链路带宽和网络拥塞情况的影响.在实现过程中,将每条有向边形式化为$(vul,exploit,P)$,其中,vul 为利用的漏洞,$exploit$ 为漏洞利用信息,P 为两个主机之间的链路不确定度.有向边属性 vul 形式化为$(cve,harm)$,其中,cve 为漏洞 ID,$harm$ 为漏洞危害值.另一个有向边属性 $exploit$ 形式化为$(pre_A,pre_H,post_A,post_H,mode)$,其中,$pre_A$ 为攻击者前件,pre_H 为主机前件,$post_A$ 为攻击者后件,$post_H$ 为主机后件,$mode$ 为漏洞利用模式.下面将详细定义不确定攻击图模型 UAG (uncertain attack graph).

定义 1. UAG 模型定义为一个七元组$\langle V,E,VUL,EXP,HV,HC,HT\rangle$,其中:

(1) V 是 UAG 中的节点集合.对于任意的 $v_i\in V,v_i=(h_i,a_i)$,分别表示主机名和攻击者在主机 h_i 上的访问权限,且满足 $a_i\in\{0,1,2\}$;

(2) E 是 UAG 中的有向边集合.对于任意的 $e_i\in E,e_i=(vul_i,exploit_i)$,分别表示漏洞及其利用.其中,$vul_i\in VUL$ 且 $exploit_i\in EXP$;

(3) VUL 是漏洞集合.对于任意的 $vul_i\in VUL,vul_i=(cve_i,harm_i)$,分别表示漏洞编号和漏洞危害.目前,国际上一种公认的漏洞评价体系为 CVSS,因此本文使用漏洞的 CVSS 值度量其危害程度;

(4) EXP 是与 VUL 对应的漏洞利用集合.对于任意的 $exploit_i\in EXP,exploit_i=(pre_A^i,pre_H^i,post_A^i,post_H^i,$

$mode_i$),分别表示利用漏洞 vul_i 的攻击者前件、主机前件、攻击者后件、主机后件和利用模式;

(5) HV 是主机-漏洞映射矩阵.对于任意的 $hv_{ij}\in\{0,1\}$,取值为 1 表示主机 h_i 存在漏洞 vul_j;反之,取值为 0 表示主机 h_i 不存在漏洞 vul_j;

(6) HC 是主机连通性矩阵.对于任意的 $hc_{ij}\in[0,1]$,表示主机 h_i 与主机 h_j 的链路不确定度,且满足 $hc_{ii}=1$.链路不确定度可以设定为链路延迟、链路带宽和网络拥塞度等多个因素的函数;

(7) HT 是主机可信关系矩阵.对于任意的 $ht_{ij}\in\{0,1\}$,取值为 1 表示主机 h_i 与主机 h_j 之间存在可信关系;反之,取值为 0 表示不存在可信关系.

在 UAG 模型的基础上,本文提出了不确定攻击图生成算法和最佳利用链生成启发式算法.

2 不确定攻击图生成算法

2.1 算法基本思想

利用漏洞进行权限提升,需要保证漏洞利用的相关条件得到满足.为此,判断是否生成下一个节点需要综合考虑漏洞利用的前置条件、主机之间的连通性以及主机之间的可信关系.

假设攻击图的生成过程满足单调性,用于缓解攻击图生成过程的可扩展性问题;为了进一步关注问题本身,假设攻击者一旦有条件发起攻击,即可成功地利用漏洞达到阶段性目的.

算法采用宽度优先策略进行攻击图的生成.基于以上两点假设可以发现,不确定攻击图生成过程满足偏序关系,即单条利用链上已扩展节点不会再次被重复扩展;进一步发现,不同利用链之间有可能扩展出相同节点.但是这不影响算法的执行,因为宽度优先策略能够保证重复节点仅出现在待扩展节点集合内,而不会出现在已扩展节点集合内.

2.2 算法描述

2.2.1 算法输入

(1) 已扩展节点集合 *Expanded*,其元素形如$\langle h,a,lv\rangle$,其中,h 表示主机名,a 表示攻击者在主机 h 上的访问权限,lv 表示攻击图生成之后节点所处扩展层的编号.初始为空集合;

(2) 待扩展节点集合 *Expanding*,其元素形如$\langle h,a,lv\rangle$,符号的物理含义同上.初始为空集合;

(3) UAG 模型;

(4) 链路不确定度阈值ε.

2.2.2 算法输出

元组集合表示的不确定攻击图,其元素形如$\langle\langle h_{src},a_{src},lv_{src}\rangle,\langle h_{dst},a_{dst},lv_{dst}\rangle,vul,exploit,HC[src,dst]\rangle$,表示从源节点$\langle h_{src},a_{src},lv_{src}\rangle$到目的节点$\langle h_{dst},a_{dst},lv_{dst}\rangle$的一条有向边;$vul$,$exploit$ 和 $P_{src\text{-}dst}$ 是有向边的属性,分别表示漏洞、漏洞利用和源、目的主机之间的链路不确定度;lv 表示攻击图生成之后节点所处扩展层的编号,初始攻击者节点处于 1 层.

2.2.3 算法伪码

不确定攻击图生成算法流程见算法 1.

算法 1. An algorithm for generating uncertain attack graph (UAG).

Begin GenUAG

1. Append $\langle attacker,2,1\rangle$ to *Expanding* set;
2. Clear *Expanded* set;
3. Initialize *V*, *VUL*, *EXP*, *HV*, *HC*, *HT*;
4. **While** *Expanding* Set is not empty **Do**
5. **Begin**
6. Remove the first element ($\langle h_{src},a_{src},lv_{src}\rangle$) from *Expanding* set;

7. **Set** $hosts=\{host\neq attacker, P_{src\text{-}dst}\geqslant\varepsilon|$uncertain connect to $h_{src}\}$;
8. Append $\langle h_{src},a_{src},lv_{src}\rangle$ to the end of *Expanded* set;
9. **Set** $t_hosts=\{host\in hosts|$trust to $h_{src}\}$;
10. **For each** $\langle h_{dst},a_{dst},lv_{src}+1\rangle\in t\_hosts$ **Do**
11. **Begin**
12. Append to $Graph\langle\langle h_{src},a_{src},lv_{src}\rangle,\langle h_{dst},a_{dst},lv_{src}+1\rangle,-,-,HC[src,dst]\rangle$;
13. **If** $\langle h_{dst},a_{dst},\text{“–”}\rangle\notin Expanding$ **And** $\langle h_{dst},a_{dst},\text{“–”}\rangle\notin Expanded$ **Then**
14. Append $\langle h_{dst},a_{dst},lv_{src}+1\rangle$ to the end of *Expanding* set;
15. **Else If** $\langle h_{dst},a_{dst},\text{“–”}\rangle\notin Expanded$ **Then**
16. Update $\langle h_{dst},a_{dst},lv_{src}+1\rangle$ in *Expanding* set;
17. **End**
18. **For each** $\langle h_{dst},a_{dst},lv_{src}+1\rangle\in hosts$ **Do**
19. **Begin**
20. **For each** v_k exists in h_{dst} **Do**
21. **Begin**
22. **If** a_{src} satisfies $exp_k.pre_A$ **Then**
23. **Begin**
24. **If** $\langle h_{src},a_{src},\text{“–”}\rangle\neq\langle h_{dst},exp_k.post_A,\text{“–”}\rangle$ **Then**
25. Append to $Graph\langle\langle h_{src},a_{src},lv_{src}\rangle,\langle h_{dst},exp_k.post_A,lv_{src}+1\rangle,v_k,exp_k,HC[src,dst]\rangle$;
26. **If** $\langle h_{dst},exp_k.post_A,\text{“–”}\rangle\notin Expanding$ **And** $\langle h_{dst},exp_k.post_A,\text{“–”}\rangle\notin Expanded$ **Then**
27. Append $\langle h_{dst},exp_k.post_A,lv_{src}+1\rangle$ to the end of *Expanding* set;
28. **Else If** $\langle h_{dst},exp_k.post_A,\text{“–”}\rangle\notin Expanded$ **Then**
29. Update $\langle h_{dst},exp_k.post_A,lv_{src}+1\rangle$ in *Expanding* set;
30. **End**
31. **End**
32. **End**
33. Move the goal node to the end of *Expanding* set if it exists;
34. **End**
35. **Return** *Graph*;

End GenUAG

注:符号“–”表示在算法实现中不考虑这一部分的取值.

2.3 算法复杂性分析

算法 1 第 1 行~第 3 行是初始化过程,其时间复杂度是 $O(n^2)$.算法第 4 行~第 34 行是主体部分,下面将详细分析其时间复杂性,已扩展节点集、待扩展节点集均使用队列数据结构.

第 4 行的最外层循环中,判断待扩展节点集是否为空可在常数时间内完成,因此最外层循环时间量级是 $O(n)$.第 6 行移除待扩展节点集队列中的第 1 个元素,这可在常数时间内完成.第 7 行获取与 $host_{src}$ 不确定连通的主机集合,利用索引最坏情况下可在 $O(n^2)$时间内完成.第 8 行添加元素到已扩展节点集队列,可在常数时间内完成.同理,第 9 行的时间复杂度为 $O(n^2)$.算法第 10 行~第 17 行生成使用可信连接后的节点集合:第 10 行循环次数是 $O(n)$量级;第 12 行将生成的攻击图节点加入到结果集中,常数时间内完成;第 13 行判断待扩展节点集和已扩展节点集是否已经包含 $host_{dst}$,最坏情况下的时间复杂度是 $O(n)$.因此,这一段代码总的时间复杂度最坏情况下为 $O(n^2)$.第 18 行~第 32 行生成利用漏洞后的节点集合:第 18 行循环次数是 $O(n)$量级;第 20 行循环次数是 $O(n)$量级;算法第 22 行~第 30 行最坏情况下的时间复杂度是 $O(n)$.因此,这一段代码总的时间复杂度最坏情况下

为 $O(n^3)$.第 33 行进行移动操作,最坏情况下的时间复杂度为 $O(n)$.

综上所述,算法最坏情况下的时间复杂度为 $O(n^4)$,空间复杂度为 $O(n^2)$,均为多项式时间复杂度.这里,n 为节点数目.

3 最佳利用链生成启发式算法

一般来说,通过攻击图得出的攻击路径数量是比较多的,在大规模网络环境下,这一情况尤为突出.在防御方看来,更希望找出攻击者最有可能采取的攻击路径,进而定位到威胁最大的主机和漏洞.换句话说,最佳漏洞利用链会比大量可能的漏洞利用链更具安全指导意义.因此,进一步研究如何基于不确定攻击图快速生成最佳漏洞利用链具有重要的理论和现实意义.

3.1 算法基本思想

在不确定攻击图基础上,需要找出最佳漏洞利用链,使得攻击效益最大.本文采用启发式方法搜索攻击图,通过计算并比较各个状态节点的启发式函数值,判断下一步最优扩展节点,最终得到一条最佳攻击路径.首先给出几个定义.

定义 2. 有向边攻击效益定义为攻击者沿着该有向边实施攻击所获得的利益.假设一条利用链上两个相邻节点分别是(h_i,a_i)和(h_j,a_j),连接两个节点的有向边为 e_{ij},则定义有向边攻击效益 $benefit_{ij}$ 为

$$benefit_{ij}=a_j\cdot HC_{ij}\cdot e_{ij}.vul.harm \tag{1}$$

其中,a_j 表示在主机 h_j 上获取的权限,HC_{ij} 表示有向边 e_{ij} 的不确定度,$e_{ij}.vul.harm$ 表示有向边 e_{ij} 关联的漏洞的危害值.

定义 3. 假如一条利用链上两个相邻节点之间存在多条有向边,每条有向边均有相应的攻击效益.不妨设两个节点分别是(h_i,a_i)和(h_j,a_j),则从节点 i 到节点 j 的单步攻击效益定义为

$$Benefit_{ij}=\max_{k\geqslant 1} benefit_{ij}^{k} \tag{2}$$

定义 4. 一条漏洞利用链上有多个节点,不妨设为 $v_1\rightarrow v_2\rightarrow v_3\rightarrow\ldots$,假如两个相邻节点之间的单步效益已知,那么该漏洞利用链的攻击效益定义为

$$Benefit=\sum_{k\geqslant 2} Benefit_{(k-1)k} \tag{3}$$

定义 5. 最佳利用链是一条漏洞利用路径,攻击者按照该路径展开攻击能够获得最大的总体攻击效益.

启发式算法的核心是定义启发式函数.基于最佳利用链生成的需求,本文将启发式函数定义为 $f(n)=g(n)+h(n)$,其中,$g(n)$表示从源节点到当前节点的攻击效益,$h(n)$表示估计的从当前节点到目标节点的攻击效益.显然,$g(n)$函数比较容易定义,只需要计算攻击进行到节点 v_n 时的总体效益即可;关键是如何定义估计函数 $h(n)$,由于攻击者追求攻击效益最大化,所以攻击者总是倾向于向预期攻击效益最大的方向展开进一步攻击.同时发现,不确定攻击图生成算法执行的最后一步过程是攻击目标节点从待扩展节点集合取出加入到已扩展节点集合,因此,算法结束时已扩展节点集合 *Expanded* 最后一个元素必为攻击目标节点.因此,估计函数 $h(n)$定义为已扩展节点集合中节点 v_n 与攻击目标节点的“距离”.

综上所述,公式(4)和公式(5)分别给出了 $g(n)$和 $h(n)$的定义.

$$g(n)=\max_{t\geqslant 1}\sum_{2\leqslant k\leqslant n}(a_k\cdot HC_{(k-1)k}\cdot\max_{l\geqslant 1} e^{l}_{(k-1)k}.vul.harm) \tag{4}$$

$$h(n)=\|v_nv_{goal}\|/v_{goal} \tag{5}$$

公式(4)中的 t 表示到节点 v_n 的不同利用链数量,n 表示某条到节点 v_n 的利用链的长度;公式(5)中的范数定义为 v_{goal} 与 v_n 所处扩展层编号之差.

3.2 算法描述

3.2.1 算法输入

(1) “活跃”节点集合 *ActiveNodes*,其元素形如$\langle\langle h,a,lv\rangle,f,forward\rangle$,其中:$lv$ 表示节点扩展层编号;f 表示节点

⟨*h*,*a*⟩的启发式函数值;*forward* 是一个指向前驱节点的引用,用于构造利用链.初始为空集合;

(2) “死亡”节点集合 *DeadNodes*,其元素形如⟨⟨*h*,*a*,*lv*⟩,*f*,*forward*⟩,符号的物理含义同上.初始为空集合;

(3) 算法 1 的输出结果.

3.2.2 算法输出

到达攻击目标的最佳漏洞利用链(最佳攻击路径).

3.2.3 算法伪码

最佳利用链生成启发式算法流程见算法 2.

算法 2. A heuristic algorithm to generate the optimal exploit chain.

Begin FindOptimalPath

1. Append ⟨⟨*attacker*,2,1⟩,lv_{goal}–1,“–”⟩ to *ActiveNodes* set;
2. Clear *DeadNodes* set;
3. **While** the first element of *ActiveNodes* set is not goal node **Do**
4. **Begin**
5. Remove the first element (⟨⟨h_{src},a_{src},lv_{src}⟩,f_{src},$forward_{src}$⟩) of *ActiveNodes* set;
6. Append ⟨⟨h_{src},a_{src},lv_{src}⟩,f_{src},$forward_{src}$⟩ to the end of *DeadNodes* set;
7. **For each** ⟨h_{conn},a_{conn},lv_{conn}⟩ which has edge from ⟨h_{src},a_{src},lv_{src}⟩ **Do**
8. **Begin**
9. Calculate the heuristic value of ⟨h_{conn},a_{conn},lv_{conn}⟩;
10. **If** ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,“–”,“–”⟩∉*DeadNodes* **And** ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,“–”,“–”⟩∉*ActiveNodes* **Then**
11. Append ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,f_{conn},h_{src}⟩ to the end of *ActiveNodes* set;
12. **Else If** ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,“–”,“–”⟩∉*ActiveNodes* **Then**
13. **If** $f_{conn} \geqslant f_{conn}^{old}$ **Then**
14. **Begin**
15. Update ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,f_{conn},h_{src}⟩ in *DeadNodes* set;
16. Append a copy of ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,f_{conn},h_{src}⟩ to *ActiveNodes* set;
17. **End**
18. **Else If** ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,“–”,“–”⟩∉*DeadNodes* **Then**
19. **If** $f_{conn} \geqslant f_{conn}^{old}$ **Then**
20. Update ⟨⟨h_{conn},a_{conn},lv_{conn}⟩,f_{conn},h_{src}⟩ in *ActiveNodes* set;
21. **End**
22. Move the element which has the maximum heuristic value to the head of *ActiveNodes* set;
23. **End**
24. Print the optimal path according to *forward* section of the element in *DeadNodes* set;

End FindOptimalPath

注:符号“–”表示在算法实现中不考虑这一部分的取值.

3.3 算法复杂性分析

算法第 1 行、第 2 行是初始化过程,可在常数时间内完成;算法第 24 行是输出结果部分,可在 $O(n)$时间内完成;算法第 3 行~第 23 行是主体部分,下面将给出详细的算法分析.

第 3 行判断 *ActiveNodes* 集合第 1 个元素是否目标节点,判断次数是 $O(n)$量级.算法第 5 行~第 22 行是外层循环的循环体.第 5 行的移除元素操作和第 6 行的添加元素操作可在常数时间内完成.第 7 行的循环次数是 $O(n)$

量级,算法第 9 行~第 20 行是内层循环的循环体.第 9 行计算节点启发式函数值,当采用增量计算方法可在常数时间内完成.第 10 行~第 20 行最坏情况下可在 $O(n)$时间内完成.因此,算法第 7 行~第 21 行总的时间复杂度为 $O(n^2)$.第 22 行进行移动操作,其中,得出启发式值最大的节点需要 $O(n)$的时间完成,而移动操作最坏情况下的时间复杂度为 $O(n)$.

综上所述,最佳利用链生成启发式算法总的时间复杂度为 $O(n^3)$.这里,n 为节点数目.

4 实验结果与分析

4.1 实例网络分析

本实验搭建了一个实际网络环境,与文献[5]类似,包括一个 Web 服务器(*W*)、一个文件服务器(*F*)、一个 Oracle 数据库服务器(*D*)、一个攻击主机(*A*)和运行本文方法的监控主机(*M*),拓扑结构图如图 3 所示.防火墙(*FW*)规则遵循的原则是“一切未被允许的都是被禁止的”,详细规则见表 1.表 2 详细列出了网络中的漏洞及其利用信息,包括利用条件和利用模式.

本实验中,攻击者的攻击目标是“获取数据库服务器 *D* 上的 Root 访问权限”.根据用户使用网络的一般经验,内网链路可用性一般高于外网链路可用性.为此,在实验过程中通过向外网和内网分别加入一些背景流量,使得外网链路可用性稳定在 0.5 附近、内网链路可用性稳定在 0.9 附近.

基于防火墙规则、链路不确定度和漏洞利用信息,主机连通性矩阵和主机可信关系矩阵见表 3 和表 4.链路不确定度阈值ε设置为 0.2.

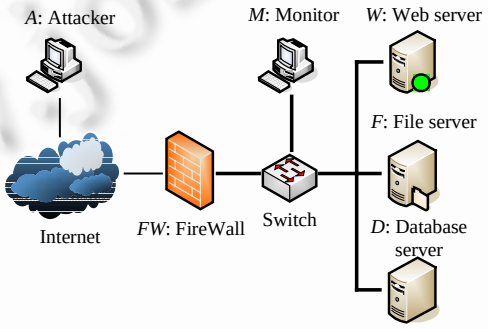


Fig.3 Experimental network topology

图 3 实验网络拓扑

Table 1 Firewall rules

表 1 防火墙规则

Source	Destination	Service	Policy
All	*W*	http	Allow
All	*W*	ftp	Allow
All	*F*	ftp	Allow
W	*D*	Oracle	Allow
F	*D*	ftp	Allow

Table 2 Vulnerability list of experimental network

表 2 实验网络中的漏洞列表

Vulnerability/Trust relationship		Impact	Victim host	Precondition onattack host	Post condition on victim host	Exploit mode
ap	*Apa.Chunked Code buff. Ovf.* [CVE-2002-0392]	7.5	*W*	*Access*≥1	*Access* 2	Remote
wu	*WuFtpd sockprintf buff. Ovf.* [CVE-2003-1327]	9.3	*W*	*Access*≥1	*Access* 2	Remote
tns	*Oracle TNS listen buff. Ovf.* [CVE-2001-0499]	10.0	*D*	*Access*≥1 $host_{src}$=*W*/*D*	*Access* 2	Remote
t_1	Trust remote login (*F* to *D*)	10.0	*D*	*Access*≥1	*Access* 1	Remote
t_2	Trust remote login (*F* to *W*)	10.0	*W*	*Access*≥1	*Access* 1	Remote
t_3	Trust remote login (*W* to *D*)	10.0	*D*	*Access*≥1	*Access* 1	Remote
t_4	Trust remote login (*W* to *F*)	10.0	*F*	*Access*≥1	*Access* 1	Remote

Table 3 Host connectivity matrix

表 3 主机连通性矩阵

HC	*A*	*W*	*F*	*D*
A	1	0.5	0.5	0
W	0	1	0.9	0.9
F	0	0.9	1	0.9
D	0	0.9	0.9	1

Table 4 Host trust relationship matrix

表 4 主机可信关系矩阵

HT	*A*	*W*	*F*	*D*
A	0	0	0	0
W	0	0	1	1
F	0	1	0	1
D	0	1	1	0

应用算法 1,得出如图 4 所示的不确定攻击图.在已生成的不确定攻击图基础上,应用算法 2,得出最佳漏洞利用链为 $(A,2)\xrightarrow{wu}(W,2)\xrightarrow{tns}(D,2)$,如图 4 中空心箭头标识的路径所示.

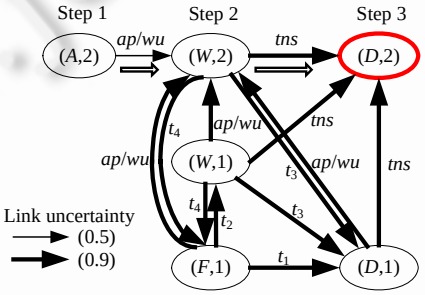


Fig.4 Uncertain attack graph

图 4 不确定攻击图

从图 4 可以看出,本文方法生成的不确定攻击图能够展现所有的攻击路径,攻击者通过结合漏洞利用和可信关系利用,可以通过多种途径达到其攻击目标(*D*,2),如(*A*,2)→(*W*,2)→(*D*,2),(*A*,2)→(*W*,2)→(*F*,1)→(*D*,1)→(*D*,2)等.同时,攻击图中每条有向边赋予了概率值链路不确定度,用于表征链路不确定度,该属性也是算法 2 需要用到的一个重要属性.

算法 2 综合了主机连通信息、主机可信关系信息、主机漏洞信息、已生成的不确定攻击图等多源数据,通过启发式算法得出了最佳漏洞利用链.实验结果表明,最有效的攻击方式是:首先利用 Web 服务器上的漏洞取得 Web 服务器的 Root 访问权限,然后利用 Oracle 服务器中的漏洞取得服务器的 Root 访问权限.虽然其他途径也能够达到攻击目标,但是效果却没有这条攻击路径好.这个结果与真实网络环境中攻击者攻击内网服务器的操作过程也是一致的,从而验证了本文方法的有效性.

4.2 仿真实验分析

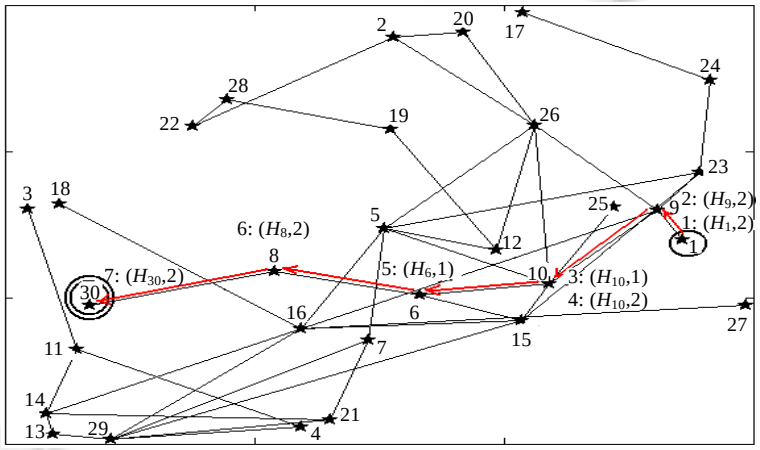
基于经典的 Waxman 模型[14]来生成全流通的随机网络拓扑,它使用 3 个参数来表征生成拓扑的特性:网络节点数目 *n*、最大链路概率参数α和长链路比率参数β.在实验中,使用 MATLAB 程序在 15×15 范围内生成实验

所需的随机拓扑.针对本文研究内容,设定生成的第 1 个网络节点为攻击节点,生成的最后一个网络节点为攻击目标节点.为了进行统计实验,设定一个漏洞集合并随机地为每个网络节点指定漏洞集合中的漏洞;同时,对于连接度大于平均节点连接度的网络节点,具有到达相连节点的可信关系.最后,随机生成网络链路不确定度.

本实验设置漏洞集合包括两个可用于获取 Root 权限的漏洞,危害值分别为 10.0 和 5.0;设置不确定度阈值ε为 0.1,不同规模的随机网络拓扑参数见表 5.现以网络规模为 30 个节点和 40 个节点为例,说明本文方法的有效性.通过运行本文算法 1 和算法 2,得到如图 5 所示的结果,其中,单线圆圈标识了攻击节点,双线圆圈标识了攻击目标节点.生成的最佳漏洞利用链分别是:

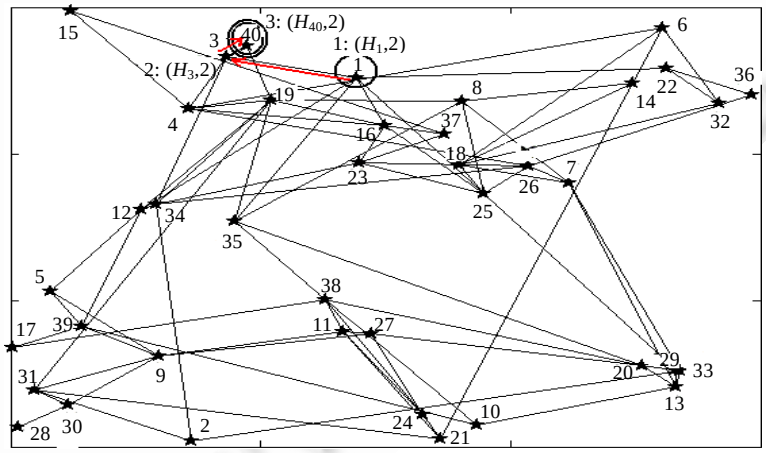
(1) 在参数配置为 n=30/α=1/β=0.15 的情况下,得到的最佳漏洞利用链为$(H_1,2)\rightarrow(H_9,2)\rightarrow(H_{10},1)\rightarrow(H_{10},2)\rightarrow(H_6,1)\rightarrow(H_8,2)\rightarrow(H_{30},2)$,如图 5(a)所示;

(2) 在参数配置为 n=40/α=1/β=0.15 的情况下,得到的最佳漏洞利用链为$(H_1,2)\rightarrow(H_3,2)\rightarrow(H_{40},2)$,如图 5(b)所示.



(a) Metrics: n=30/α=1/β=0.15

(a) 参数设置:n=30/α=1/β=0.15



(b) Metrics: n=40/α=1/β=0.15

(b) 参数设置:n=40/α=1/β=0.15

Fig.5 Optimal vulnerability exploit-chains under different metrics

图 5 不同参数条件下生成的最佳漏洞利用链

Table 5 Metric list of random network topologies

表 5 随机网络拓扑参数列表

No.	Network size (n)	Maximal link probability (α)	Ratio of long edges to short edges (β)	Average degree
1	20	1	0.15	2.20
2	30	1	0.15	3.00
3	40	1	0.15	4.35
4	50	1	0.15	6.08
5	60	1	0.15	7.03
6	70	1	0.15	7.14
7	80	1	0.15	9.98
8	90	1	0.15	10.64
9	100	1	0.15	11.90
10	150	1	0.15	19.48
11	200	1	0.15	28.43

从实验结果可以看出,当攻击者不能直接获取到攻击目标节点的 Root 权限时,将会逐步攻陷网络中的其他关联节点,最终达到攻击目标.本实验验证了本文方法在更大规模网络环境下仍然有效,并且生成的最佳漏洞利用链也是比较合理的.

为了验证本文算法分析的结论,通过在不同网络规模下运行算法,得到如图 6 所示的运行时间统计结果(主机配置:双核 1.8GHz CPU,2G 内存).其中,横坐标为网络规模的对数坐标,纵坐标为生成最佳漏洞利用链耗费的时间,实折线图表示实际运行时间随网络规模的变化情况,虚线表示多项式拟合后的曲线.从图 6 可以发现,实验结果验证了前面算法分析的结果.

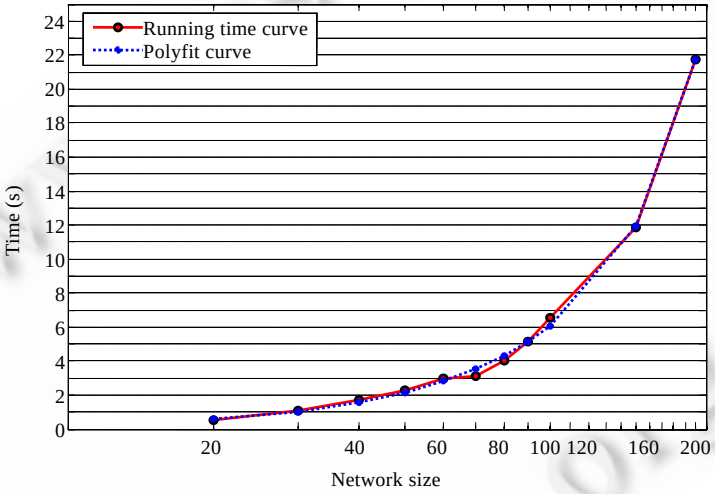


Fig.6 Generation time of the optimal exploit chain vs. network size

图 6 最佳利用链的生成时间随网络规模变化情况

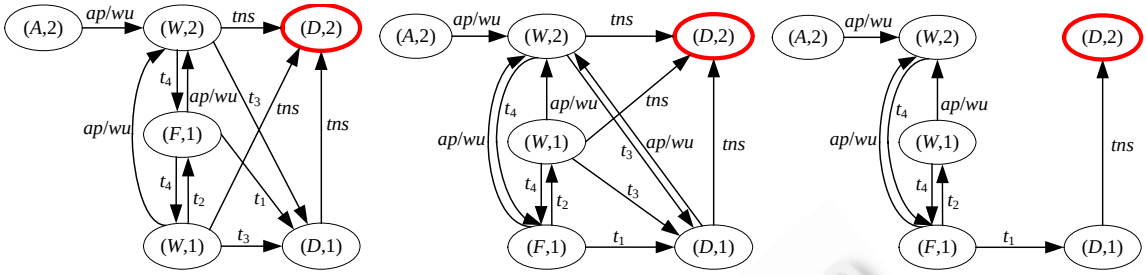
4.3 与其他方法的比较

为了进一步验证本文方法的有效性,现将本文方法与文献[5]中的方法进行比较.本节包括两个部分:一是第 4.1 节所设置的网络环境下的结果比较;二是仿真实验结果比较.

在第 4.1 节所设置的网络环境中,为了说明链路不确定性对攻击图生成结果和最佳漏洞利用链生成结果的影响,实验包括: (1) $P_{W\text{-}D}\geq\varepsilon$,其他链路不确定度不变;(2) $P_{W\text{-}D}<\varepsilon$,其他链路不确定度不变.结果如图 7 所示.

文献[5]方法生成的攻击图如图 7(a)所示,最佳漏洞利用链为$(A,2)\rightarrow_{ap}(W,2)\rightarrow_{tns}(D,2)$.当 $P_{W\text{-}D}\geq\varepsilon$时,主机 W 与主机 D 之间的链路不确定度不小于链路不确定度阈值,运行算法 1 得到如图 7(b)所示的不确定攻击图,运行算法 2 得到的最佳漏洞利用链为$(A,2)\rightarrow_{wu}(W,2)\rightarrow_{tns}(D,2)$.当 $P_{W\text{-}D}<\varepsilon$时,主机 W 与主机 D 之间的链路不确定度小于链路不确定度阈值,运行算法 1 得到如图 7(c)所示的不确定攻击图,运行算法 2 得到的最佳漏洞利用链为

$$(A,2)\rightarrow_{wu}(W,2)\rightarrow_{t_4}(F,1)\rightarrow_{t_1}(D,1)\rightarrow_{tns}(D,2).$$



(a) Host-Centric attack graph
(a) 以主机为中心的攻击图

(b) UAG under $P_{W\text{-}D}\geq\varepsilon$
(b) $P_{W\text{-}D}\geq\varepsilon$时生成的不确定攻击图

(c) UAG under $P_{W\text{-}D}<\varepsilon$
(c) $P_{W\text{-}D}<\varepsilon$时生成的不确定攻击图

Fig.7 Attack graphs generated by different methods
图 7 不同方法生成的攻击图

从图 7 可以看到,当主机 W 和主机 D 之间链路不确定度不小于设定的阈值时,算法 1 能够生成与文献[5]方法同样完备的攻击图,算法 2 生成的最佳漏洞利用链与比较方法生成结果也是一致的;当主机 W 和主机 D 之间链路不确定度小于设定的阈值时,算法 1 生成的不确定攻击图比文献[5]方法生成的攻击图更加简洁,算法 2 生成的最佳漏洞利用链与比较方法结果相比有了很大不同.算法 2 结果表明,攻击者在知道主机 W 与主机 D 之间链路基本不可用的情况下更倾向于借助主机 F 实现其攻击目标,更贴近实际网络攻击场景,与比较方法结果相比更为合理.

在仿真实验环境中,为了说明最佳漏洞利用链生成算法的正确性,基于本文定义的攻击效益,分别计算出不同方法生成的漏洞利用链的攻击效益,得到如表 6 所示的结果.可以发现,本文方法生成的漏洞利用链的攻击效益普遍高于比较方法生成的漏洞利用链的攻击效益,从而验证了本文算法的正确性.在更大规模网络环境下也能得到类似结论,限于篇幅,未列出比较结果.

Table 6 Comparisons of attack benefits for exploit chains generated by different methods
表 6 不同方法生成的利用链的攻击效益比较

Network size	The exploit chain generated by our method and its link uncertainties	Benefit of our method	The exploit chain generated by host-centric model checking and its link uncertainties	Benefit of host -centric method
20	$(H_1,2)\rightarrow(H_{11},2)$: 0.68 $(H_{11},2)\rightarrow(H_{15},2)$: 0.80 $(H_{15},2)\rightarrow(H_{20},2)$: 0.84	**52.34**	$(H_1,2)\rightarrow(H_{15},2)$: 0.65 $(H_{15},2)\rightarrow(H_{20},2)$: 0.84	29.80
30	$(H_1,2)\rightarrow(H_9,2)$: 0.85 $(H_9,2)\rightarrow(H_{10},1)$: 0.97 $(H_{10},1)\rightarrow(H_{10},2)$: 1.00 $(H_{10},2)\rightarrow(H_6,1)$: 0.87 $(H_6,1)\rightarrow(H_8,2)$: 0.33 $(H_8,2)\rightarrow(H_{30},2)$: 0.46	**71.30**	$(H_1,2)\rightarrow(H_9,2)$: 0.85 $(H_9,2)\rightarrow(H_{10},1)$: 0.97 $(H_{10},1)\rightarrow(H_6,1)$: 0.87 $(H_6,1)\rightarrow(H_8,2)$: 0.33 $(H_8,2)\rightarrow(H_{30},2)$: 0.46	51.30
40	$(H_1,2)\rightarrow(H_3,2)$: 0.86 $(H_3,2)\rightarrow(H_{40},2)$: 0.91	35.38	$(H_1,2)\rightarrow(H_3,2)$: 0.86 $(H_3,2)\rightarrow(H_{40},2)$: 0.91	35.38
50	$(H_1,2)\rightarrow(H_{33},2)$: 0.66 $(H_{33},2)\rightarrow(H_7,1)$: 0.86 $(H_7,1)\rightarrow(H_{50},2)$: 0.77	**44.60**	$(H_1,2)\rightarrow(H_{17},2)$: 0.87 $(H_{17},2)\rightarrow(H_{39},1)$: 0.33 $(H_{39},1)\rightarrow(H_{50},2)$: 0.09	12.89
60	$(H_1,2)\rightarrow(H_{12},2)$: 0.88 $(H_{12},2)\rightarrow(H_{34},2)$: 0.94 $(H_{34},2)\rightarrow(H_{60},2)$: 0.81	**53.09**	$(H_1,2)\rightarrow(H_{34},2)$: 0.64 $(H_{34},2)\rightarrow(H_{60},2)$: 0.81	14.55
70	$(H_1,2)\rightarrow(H_{14},2)$: 0.89 $(H_{14},2)\rightarrow(H_{24},2)$: 0.63 $(H_{24},2)\rightarrow(H_{32},1)$: 0.34 $(H_{32},1)\rightarrow(H_{56},1)$: 0.42 $(H_{56},1)\rightarrow(H_{70},2)$: 0.75	**71.18**	$(H_1,2)\rightarrow(H_{62},2)$: 0.58 $(H_{62},2)\rightarrow(H_{33},1)$: 0.01 $(H_{33},1)\rightarrow(H_{70}.2)$: 0.51	21.76

复杂性分析比较得出,本文算法的时间复杂度为 $O(n^4)$,文献[5]方法的时间复杂度为 $O(n^2)$,均为多项式时间复杂度,但是本文算法的复杂度量级稍高,这是下一步需要改进和优化的地方.

5 总结和未来工作

高效准确的网络漏洞分析能够为提高网络安全性提供有力的支持.通过结合不确定图概念和攻击图模型,本文提出了一种基于不确定攻击图的网络漏洞分析方法——一个多项式时间复杂度的不确定攻击图生成算法和一个多项式时间复杂度的最佳利用链生成启发式算法.实验结果表明,本文方法能够真实地反映网络漏洞之间的相互联系、准确地预知攻击者最可能使用的攻击路径.

链路不确定度可以设定为链路延迟、链路带宽和网络拥塞度等多个因素的函数,但是这样设计的攻击模型更加复杂,所需的计算复杂性也更高.

下一步需要做的工作主要包括:改进不确定攻击图生成算法,进一步降低时间复杂度;优化最佳利用链生成算法中的启发式函数,更为快速准确得出最佳漏洞利用链;考虑更多链路状态,将生成算法应用到更大规模的网络中,验证其有效性.

References:

[1] Zhang BW, Zhu WL, Xue Z. Mining privilege escalation paths for network vulnerability analysis. In: Lei JS, Yu J, Zhou SG, eds. Proc. of the 4th Int'l Conf. on Fuzzy Systems and Knowledge Discovery. Haikou: IEEE Computer Society, 2007. 56−60. [doi: 10.1109/FSKD.2007.406]

[2] Shahriari HR, Ganjisaffar Y, Jalili R, Habibi J. Topological analysis of multi-phase attacks using expert systems. Journal of Information Science and Engineering, 2008,24(3):743−767.

[3] Maggi P, Pozza D, Sisto R. Vulnerability modelling for the analysis of network attacks. In: Zamojski W, Mazurkiewicz J, Sugier J, eds. Proc. of the Int'l Conf. on Dependability of Computer Systems. Szklarska: IEEE Computer Society, 2008. 15−22. [doi: 10.1109/DepCoS-RELCOMEX.2008.49]

[4] Sheyner O, Haines JW, Jha S, Lippmann R, Wing JM. Automated generation and analysis of attack graphs. In: Proc. of the 2002 IEEE Symp. on Security and Privacy. Berkeley: IEEE Computer Society, 2002. 273−284. [doi: 10.1109/SECPRI.2002.1004377]

[5] Hewett R, Kijsanayothin P. Host-Centric model checking for network vulnerability analysis. In: Proc. of the 24th Annual Computer Security Applications Conf. Anaheim: IEEE Computer Society, 2008. 225−234. [doi: 10.1109/ACSAC.2008.15]

[6] Malhotra S, Bhattacharya S, Ghosh SK. A vulnerability and exploit independent approach for attack path prediction. In: He X, Wu Q, Nguyen QV, eds. Proc. of the 8th IEEE Int'l Conf. on Computer and Information Technology. Sydney: IEEE Computer Society, 2008. 282−287. [doi: 10.1109/CIT.2008.Workshops.73]

[7] Zhang HX, Su PR, Feng DG. A network security analysis model based on the increase in attack ability. Journal of Computer Research and Development, 2007,44(12):2012−2019 (in Chinese with English abstract).

[8] Zhang T, Hu MZ, Yun XC, Zhang YZ. Research on computer network security analysis model. Journal on Communications, 2005,26(12): 100−109 (in Chinese with English abstract).

[9] Qu GZ, Rudraraju J, Modukuri R, Hariri S, Raghavendra CS. A framework for network vulnerability analysis. In: Hamza MH, Technol DI, Telecommun IT, eds. Proc. of the IASTED Int'l Conf. on Communications, Internet and Information Technology. St Thomas, VI: Acta Press, 2002. 289−294.

[10] Scarfone K, Grance T. A framework for measuring the vulnerability of hosts. In: Stepnowski A, Moszynski M, Kochanski T, eds. Proc. of the 1st Int'l Conf. on Information Technology (IT 2008). Gdansk: IEEE, 2008. 145−148. [doi: 10.1109/INFTECH.2008.4621610]

[11] Wang YJ, Xian M, Liu J, Wang GY. Study of network security evaluation based on attack graph model. Journal on Communications, 2007,28(3):29−34 (in Chinese with English abstract).

[12] Zhang HX, Lian YF, Su PR, Feng DG. Security-State-Region-Based model of network security evaluation. Journal of Software, 2009,20(2):451−461 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/20/451.htm [doi: 10.3724/SP.J.1001.2009.03172]

[13] Vu HL, Khaw KK, Chen TY, Kuo FC. A new approach for network vulnerability analysis. In: Proc. of the 33rd Annual IEEE Conf. on Local Computer Networks. Montreal: IEEE, 2008. 189−195. [doi: 10.1109/LCN.2008.4664170]

[14] Waxman BM. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 1988,6(9):1617−1622. [doi: 10.1109/49.12889]

附中文参考文献:

[7] 张海霞,苏璞睿,冯登国.基于攻击能力增长的网络安全分析模型.计算机研究与发展,2007,44(12):2012−2019.

[8] 张涛,胡铭曾,云晓春,张永铮.计算机网络安全性分析建模研究.通信学报,2005,26(12):100−109.

[11] 王永杰,鲜明,刘进,王国玉.基于攻击图模型的网络安全评估研究.通信学报,2007,28(3):29−34.

[12] 张海霞,连一峰,苏璞睿,冯登国.基于安全状态域的网络评估模型.软件学报,2009,20(2):451−461. http://www.jos.org.cn/1000-9825/20/ 451.htm [doi: 10.3724/SP.J.1001.2009.03172]

刘强(1986－),男,江西临川人,博士生,主要研究领域为网络安全.

蔡志平(1975－),男,博士,副教授,CCF 高级会员,主要研究领域为网络安全,网络测量,虚拟化.

殷建平(1963－),男,博士,教授,博士生导师,主要研究领域为信息安全,模式识别,网络算法.

程杰仁(1975－),男,博士生,讲师,主要研究领域为网络安全,人工智能.