基于不确定攻击图的攻击路径的网络安全分析

曾赛文[1] 文中华[2,3] 戴良伟[1] 袁 润[1]
(湘潭大学信息工程学院 湘潭 411105)[1]
(湖南工程学院湖南省风电装备与电能协同变换创新中心 湘潭 411104)[2]
(湘潭大学智能计算与信息处理教育部重点实验室 湘潭 411105)[3]

摘 要 随着科学技术的发展,现有攻击图生成算法在描述突发网络拥塞、网络断开、网络延迟等意外情况时存在不足;并且对于在攻击图中同样可以达到目标状态的攻击路径,哪一条路径网络更可靠等问题还未开始研究。通过不确定图模型提出了一种攻击图的生成算法,从攻击者的目标出发,逆向模拟生成攻击图,可以较好地模拟现实攻击情况并找出最可靠攻击路径,而且可以避免在大规模网络中使用模型检测方法出现状态空间爆炸的问题,以帮助防御者更好地防御网络漏洞攻击。实验结果表明,该方法能够正确生成攻击图,并且对大型网络的模拟也很实用。

关键词 不确定图,攻击图,模型检测,网络漏洞攻击
中图法分类号 TP393 **文献标识码** A

Analysis of Network Security Based on Uncertain Attack Graph Path

ZENG Sai-wen[1] WEN Zhong-hua[2,3] DAI Liang-wei[1] YUAN Run[1]
(College of Information Engineering, Xiangtan University, Xiangtan 411105, China)[1]
(Hunan Province Cooperative Innovation Center for Wind Power Equipment and Energy Conversion, Hunan Institute of Engineering, Xiangtan 411104, China)[2]
(Key Laboratory of Intelligent Computing & Information Processing, Ministry of Education, Xiangtan University, Xiangtan 411105, China)[3]

Abstract With the development of science and technology, the existing attack graph generation algorithm has deficiencies in describing of network congestion, network disconnection, network delays and other unforeseen circumstances. And in pathing out which route network will be more reliable when all the routes can achieve the same target state has not keen studied in pathing out. Researches nowadays about the uncertain graph have delicate descriptions about the real network. Therefore, this thesis will put forward a new algorithm through uncertain graph model, and we can simulate the reality of attacks by reverse simulation to generate attack graph from the target of attackers and we can also avoid the troubles of space explosion to help defenders against the risks of network vulnerabilities. Through experiments we fond that our approach can generate the attack graph correctly and it is also practical for the simulation of large networks.

Keywords Uncertain graph, Attack graph, Model check, Network vulnerabilities attack

1 引言

随着 Internet 和各种网络服务的迅速发展,据中国互联网信息中心(CNNIC)[1]最新网络发展统计报告[2]显示,截至 2016 年 6 月,我国网民规模达 7.10 亿,且根据信息化发展评价报告,中国信息化发展指数排名位列全球 25 名,首次超过了"G20 国家"的平均水平。当我们越来越依赖网络的时候,安全问题也随之而来。据国家应急中心(CNCERT/CC)[3]统计,2015 年共发现 10.5 万余个木马和僵尸网络控制端,控制了我国境内 1978 万余台主机。网络漏洞使得我们存储在电脑上的很多资料都变得不安全,会使得我们的重要信息在不经意间就被人盗走,因此加强漏洞分析对我们而言是非常必要的。

漏洞分析首先需要对网络攻击进行建模,尤其是对组合型的网络攻击的建模更是当前研究的一个难点。到目前为止,大多数网络攻击模型的研究都集中在对漏洞以及漏洞利用工具的归类上[4]。目前对组合型网络攻击行为的建模方法主要有 3 种:1)攻击树方法[5];2)攻击网方法[6];3)攻击图方法。攻击图是描述攻击者从攻击起始点到其攻击目标的所有路径的方法。攻击图提供了一种表示攻击过程场景的可视化方法。因其有利于帮助防御者进行分析,所以本文重点研究攻击图。已有一些学者利用确定的攻击图方法来进行网络漏洞分析,现有图论中的较多理论也可以支持我们的研究。例如:文献[7]提出了一种特权图的方法进行网络攻击链分析。文献[8-10]均基于模型检测然后得到攻击图。但是确定的攻击图有一定的局限性,对实时网络描述存在不足,所以本文通过不确定图来进行攻击图的分析。虽然模型检测的方法在自

本文受国家自然科学基金(61272295,61105039,61202398),湘潭大学智能计算与信息处理教育部重点实验室,湖南省重点学科建设项目(0812)资助。

曾赛文(1991-),男,硕士生,主要研究方向为智能规划,E-mail:z13618482773@126.com;文中华(1966-),男,博士,教授,博士生导师,主要研究方向为智能规划、图论及算法,E-mail:zhonghua@xtu.edu.cn;戴良伟(1989-),男,硕士生,主要研究方向为智能规划;袁 润(1992-),女,硕士生,主要研究方向为智能规划。

动生成攻击图方面有很大的便利，但是它都是从源攻击者出发搜索到达目标主机，与本文方法不同，因此不能直接借用已有的模型检测方法来生成本文的不确定攻击图。

根据现有基于模型检测生成攻击图时状态空间爆炸以及描述实时网络状况的不足，我们提出了一个基于不确定图从目标主机回溯的新方法。该方法的主要思想是首先把目标主机加入目标集，然后搜索网络中所有能成功攻击目标集中的主机作为上一层，依次递归，直到找到源攻击者或者找不到攻击者，方法结束。使用该方法找到的主机都是确定能直接攻击到目标的主机，不需要借助其他中间节点来达到攻击效果，因此不会盲目地搜索攻击不到目标主机集的主机。如果从源攻击者出发搜索，我们就不能排除哪个主机是确定不能到达目标主机的，因此需要保留所有的攻击信息，从而浪费了很多不必要的计算资源，提高了时间复杂度。使用该方法不需要考虑主机所处的状态，并且减少了搜索其他非目标主机的计算开销，可以较快地生成一个攻击图。在不确定图中，利用漏洞进行攻击需要满足漏洞利用的相关条件。根据目标主机的漏洞情况，回溯可能的攻击者需要满足：1）主机之间的连通性；2）攻击者、目标主机的漏洞利用的前提条件；3）主机之间的可信关系。并且假设攻击者一旦满足要求就一定会攻击，且攻击成功。

漏洞的检测有多种方法，主要是根据网络状态或主机进行检测。基于网络的检测主要是利用端口扫描以及尝试攻击来发现漏洞，该方法是破坏性的搜索；基于主机的检测主要是通过系统配置或者用户权限等方面检测，该方法是非破坏性的搜索。本文使用第二种方法。常用的漏洞检测工具有Openvas和 ISS securityscanner 等工具。

本文方法基于两个假设：1）单调性假设为前提，单调性假设是指在一次漏洞攻击中，一旦它的前提条件满足，那么它就一直有效，不会因为其他条件改变而变得不可用，并且一定可以攻击成功，攻击效果也一定会达到预期效果[11]；2）攻击中是以主机为中心，而不是以网络为中心。因此，每一台主机的属性都是独立的，而不是表现为在网络中所有主机的属性。下面介绍主机的属性，网络中每一台主机的属性抽象为：1）主机 A 提供的服务，例如：web services；2）软件脆弱性，主机 A 提供的软件有漏洞，例如：Apache Chunked-Code for web service provided by A；3）配置脆弱性，主机 A 服务相对应端口的漏洞，例如：Apache 的默认端口是可用的；4）连接性，其他主机能连接到 A；⑤可信关系，其他主机相信主机 A，例如：如果主机 B 相信主机 A。

在我们的模型中，每一台主机都有一个访问权限的有序集，范围从无权限和用户权限到管理员权限，本文方法中使用 0，1，2 代替。例如：如果主机 B 运行了一个 ssh 服务，我们在主机 A 上可以通过利用主机 B 的 ssh 服务缓存溢出漏洞获得主机 B 的管理员权限。实验中，不考虑此时其他攻击对主机 B 的影响，例如从主机 A 到主机 B 的 ssh 利用不再有效，由于当主机 A 到主机 B 的连接被中断（可能通过防火墙断开），或者 ssh 服务从主机 B 上被移除时，其他漏洞又要重新分析，因此不考虑这种情况。

2 相关定义

漏洞分析最主要的是攻击图的生成。关于攻击图生成过程中涉及几个内在联系的重要概念，包括不确定图、不确定攻击图，以及攻击路径的可靠性等。图中每一个节点表示一个主机，主机属性包括：主机的系统配置、连通性、可信关系、提供的服务、运行的软件、用户权限。边表示一次漏洞攻击，攻击者利用漏洞提升在受害主机的访问权限。不确定攻击图是结合不确定图和攻击图定义的。其中不确定图可以用来模拟实时网络，通过建立网络不确定函数来计算不确定图中边的不确定度（概率）[12]。

定义 1（不确定图）[13] 设不确定图 $G=\{V,\vec{E},P\}$，其中：V 是节点集合；$\vec{E}$ 是一个有向边集；P 表示边存在的不确定测度集。$P(\vec{e})$ 表示边 $\vec{e}$ 存在的不确定测度。其中 $0\leqslant P(\vec{e})\leqslant 1$，$P(\vec{e})=1$ 表示边 $\vec{e}$ 一定存在，$P(\vec{e})=0$ 表示边 $\vec{e}$ 一定不存在，$0<P(\vec{e})<1$ 称为不确定边。

定义 2（不确定攻击图） 不确定攻击图是一个六元组 $UAG=\{V,\vec{E},P,\tau,I,F,T\}$，$V$ 是一个顶点集；边集 $\vec{E}$ 是一个有向边集；$P=\{p(\vec{e})\mid \vec{e}\in\vec{E},p(\vec{e})\in(0,1]\}$ 表示边存在的不确定测度集；τ 是一个漏洞利用或渗透，表示攻击者在物理上一个权限的转移；$I\subseteq V$ 是一个初始顶点集合（源攻击者）；$F\subseteq V$ 是一个终止顶点集合（攻击者的目标）；T 表示一个可信关系集合。

例 1 图 1 给出了一个不确定攻击图示例，其中，hostA 表示攻击者，hostG 表示目标主机，边中间的两个字符分别表示边存在的概率和攻击者利用的漏洞编号，从图 1 可以清晰地看出攻击者到目标主机的所有路径。

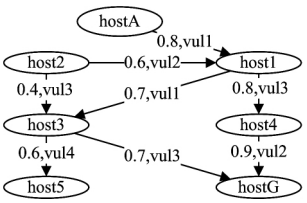

图 1 一个不确定攻击图示例

最后，根据不确定攻击图，定义攻击路径的可靠性。

定义 3（可靠性） 假设一条攻击路径由 n 台主机构成，即：$h_1\rightarrow h_2\rightarrow\cdots\rightarrow h_n$，边 $e_i(h_i,h_j)\in\vec{E}(1\leqslant i,j\leqslant n)$，$P(e_i)\in P$，则整个攻击路径的可靠性（Reliability）定义为：

$$R=\prod_{i=1}^{n}M(\vec{e_i}) \tag{1}$$

例 2 根据图 1，可以简单找到一条从攻击者到目标主机的路径，如 Path1＝hostA→ host1 → host3 → hostG。根据定义 3，可以得出 Path 的可靠性为 $R_{Path1}=0.392$。

定义 4（最可靠攻击路径） 当某两台主机存在多条路径时，可靠性最高的那条为最可靠路径。

例 3 根据图 1，可以找到与例 2 不同的一条路径 Path2＝hostA→host1 → host4→ hostG。可以计算出该路径的可靠性 $R_{Path2}=0.576$。因此该条路径就是最可靠攻击路径，攻击者很有可能选用这条路径。

3 算法的实现以及复杂性分析

利用主机漏洞进行不确定网络脆弱性分析，最主要的是生成不确定攻击图。

3.1 不确定攻击图生成算法

3.1.1 算法的基本思想

利用主机的漏洞来进行攻击图生成分析，首先要把目标主机加入到目标主机集，根据当前主机所拥有的漏洞与当前主机相连的主机进行判断，如果在当前情况下，谁攻击当前主机成功，就把该主机记录在当前主机的上一层（NewLayer）中，然后一起加入到 S 中。如果该主机就是攻击者，算法结

束；如果不是，继续按照前面的步骤递归。如果找到真正的攻击者或者下一层为空，算法结束。生成攻击图伪码见算法 1。其中，子算法 2 用来判断两个相连主机是否能攻击成功，首先判断攻击者是否满足前提条件：1）在主机 A 上至少拥有普通权限（user）。2）在主机 B 上信任主机 A，且还没有获取请求的权限或者最高权限（root），其次判断网络是否满足前提条件：①主机 A 与主机 B 在网络上的连通可能性要大于 ε（网络连通性的最低要求）；②主机上要运行某一项服务（例如：ssh），如果主机上没有已知漏洞，则无法攻击成功；③从主机 A 可以到达主机 B 的某一个开放端口（ssh 服务的默认端口），如果不开放或者被防火墙屏蔽，则攻击不能成功。3）最后攻击结果：①在主机 B 上获得请求的权限；②从主机 A 到主机 B 不再需要原来的服务就能访问主机 B（例如：使用 FTP 的溢出漏洞可以获得远程 shell）。随着科学技术的发展，可能发现更多的漏洞，使用该算法只需要扩充漏洞库，从而具有良好的可扩充性。具体判断主机 A 能否成功攻击主机 B 的伪码见算法 2。

3.1.2　算法描述

（1）算法输入

1）Input：一个不确定图 UG；

2）Input：一个主机集合 $hosts$；

3）Input：一个主机之间的连通性集合；

4）Input：攻击者 $S0$ 和目标主机 Sg。

（2）算法输出

1）Output：一个所有主机的动作集合 $actS$；

2）Output：一个记录攻击成功边的集合 $RecordEdge$；

3）Output：一个记录所有能攻击到的目标主机集合 S。

算法 1　生成攻击图

```
1. Function GenerateUncertainAttackGraph(S0,Sg)
2. 根据 UG 初始化 actS 集合；
3. S1=Sg;
4. layer= 1;
5. while(NewLayer≠∅∧S0∉(S_layer∪S_layer−1∪…∪S_1))
6. layer= layer+1;
7.    for each i∈hosts
8.       if i∉S
9.          for each j∈actS{host_i}
10. if(actS{host_j}∈S∧Attack(host_i,host_j))
11.             Append i to the end of NewLayer set;
12.             Append e→(host_i,host_j) to RecordEdge;
13.             end if;
14.          end for;
15.       end if;
16.    end for;
17.    Append NewLayer set to the end of S set;
18. end while;
19. return actS,RecordEdge,S;
20. end Function
```

算法第 1－3 行为初始化，需遍历所有的边，所以该行的时间复杂度为 $O(m)$（m 为不确定图中的边数），第 4－16 行对攻击成功的主机进行分层。如果出现极端情况，即所有主机最终只分为两层，假设分层函数第一次给第一层分了 x 台主机，那么第二层分的主机数为 $n-1-x$，其中 n 为主机总数，假设平均每台主机拥有的边数为 m/n，此时具有最好的时间复杂度 $O((n-1-x)m/n+xm/n)\approx O(m)$。而当出现 n 个分层时，因为每个层次只有一台主机，但是搜索判断时，剩下的主机都要判断一次，判断次数依次为 $(n-1)m/n$，$(n-2)m/n$，…，1，所以最坏的时间复杂度为 $O(((n-1)+(n-2)+\cdots+1)m/n)\approx O(nm)$。其余情况下，复杂度介于这两者之间。

算法 2　判断主机 A 能否攻击成功主机 B

```
1. Function Attck(A,B)
2. if (P_AB)>ε and ∃ access((A,B)∈T)>requested privilege)
3.    return ture;
4. end if;
5. for each i(i∈ The number of B′ vulnerabilities services)
6. if(P_AB<ε or The precondition of B′ service are not met)
7. continue;
8. end if;
9. if(B request privilege >Obtained Privilege)
10. continue;
11. end if;
12. return true;
13. end for;
14. return false;
15. end Function
```

若一个主机 A 要攻击另一个主机 B，为了简化攻击图的规模，找出最可能的攻击路径。首先判断两个主机之间的连通性是否大于连通阈值 ε，如果小于阈值，我们认为该主机之间不连接，也就不存在攻击。第 1－4 行用来判断主机 A 与主机 B 的可信关系可以获得的权限，如果满足，返回成功，并且记录使用哪一种方式取得成功；否则，判断主机 B 上的所有漏洞，如果存在满足要求的漏洞，记录漏洞编号，返回成功，如果所有漏洞都不符合要求，就返回失败。该方法的最坏时间复杂度为扫描该机所得的所有的漏洞 O（Vulnerabilities Counts），目前每一个主机漏洞都不多，所以在计算总时间复杂度时，忽略该算法时间损耗。

3.2　从攻击图中找出源攻击者到目标主机的所有路径方法

一般来说，攻击图得出的攻击路径都不止一条，特别是在网络节点较多时，这一情况尤为突出。因此，防御者更加希望找到攻击者最有可能采取的攻击路径，从而找到最关键的路径或者最主要的主机和漏洞。换句话说就是找到可靠性最高的那条路径。因此，进一步基于不确定攻击图研究攻击路径具有重要的理论和现实意义。

3.2.1　算法的基本思想

从上一步得出的攻击图中可以看出，能够找到很多可以攻击到目标主机的与特定攻击者在同一层的其他攻击者，因此攻击路径相当多，当规模网络增大后更加明显。因此本算法从攻击图中找出所有特定攻击者到目标主机的攻击路径，并且验证出了每条路径的可靠性。生成过程见算法 3。

3.2.2　算法描述

（1）算法输入

1）Input：一个攻击图集合 S；

2）Input：一个所有主机的动作集合 $actS$；

3）Input：一个攻击图中攻击成功所使用的路径集合 $RecordEdge$；

4）Input：$S0$ 攻击者，Sg 目标主机。

（2）算法输出

输出所有从攻击者到目标主机的路径集合 $S1$。

算法 3　生成所有从攻击者到目标主机的路径

```
1. Function GenerateAttackPath(S0,Sg,&path,&P)
2. Append S0 to path;
3. if(S0=Sg)
```

4. print(path,P * p($host_{S0}$,$host_{Sg}$))
5. end if;
6. P=P * p($host_{S0}$,$host_{Sg}$);
7. while($host_i$∈actS{$host_{S0}$})
8. GenerateAttackPath($host_i$,Sg,path,P);
9. next $host_i$;
10. end while;
11. pop S0 from path;
12. end Function;

算法 3 的复杂度分析与算法 2 的分析类似,取得最好时间复杂度时,所有主机只分为二层,只要找到目标主机在第二层中即可,在此情况下,复杂度为 $O(x)<O(n)$;算法 3 取得最坏时间复杂度时,所有主机分为 n 层,由于每一台主机都要遍历它能攻击到的主机,因此它的时间复杂度为 $O(nm/n)=O(m)$。其他情况介于两者之间。

4 算法实例分析

4.1 实例网络分析

为了证明所提方法的有效性,我们设计了一个小型网络,与文献[11]类似,其包括一个文件服务器(FTP server)、一个数据库(Database server)、一个 Web 服务器、一个攻击者(Attacker),拓扑结构如图 2 所示。

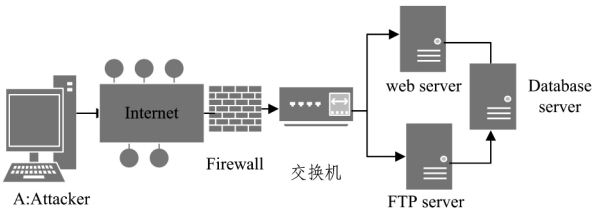

图 2 实验网络拓扑

防火墙(Firewall)详细过滤规则如表 1 所列。表 2 详细列出了主机中的漏洞信息,通过这些漏洞信息可以使用扫描工具判断能否攻击成功,漏洞信息详见文献[14]。表 3 列出了主机之间的可信关系。在本实验中,攻击者的目标是"获取数据库服务器 Database server 上的 root 权限"。

表 1 防火墙规则

Source	Destination	Service	Action
All	Web	http	Allow
All	Web	ftp	Allow
All	File	ftp	Allow
All	Database	Oracle	Allow
All	Database	ftp	Allow
All	All	All	Deny

表 2 主机漏洞

Host	Vulnerability	Bugtraq	Access
Web	Apache Chunked-Enc	5033	admin
Web	Wu-Ftpd SockPrintf()	8668	admin
File	FTP Bounce	126	user
Database	Oracle TNS Listener	4033	admin

表 3 主机之间的可信关系

HT	A	W	F	D
A	—	none	none	none
W	none	—	user	none
F	none	user	—	none
D	none	none	none	—

运用算法 2,可得到如图 3 所示的攻击图。在已生成的攻击图上,利用算法 3 生成从攻击者到目标主机的可能攻击路径,如表 4 所列。

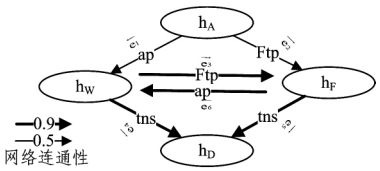

图 3 攻击图

表 4 从攻击者到目标主机的可能攻击路径

Path	Reliability
$h_A \rightarrow h_W \rightarrow h_D$	0.45
$h_A \rightarrow h_F \rightarrow h_D$	0.45
$h_A \rightarrow h_W \rightarrow h_F \rightarrow h_D$	0.405
$h_A \rightarrow h_F \rightarrow h_W \rightarrow h_D$	0.405

4.2 仿真实验分析

因为实验室不能模拟大规模网络,所以我们只能使用实验室的网络模拟器模拟一个大规模网络来进行实验。基于经典的 Waxman 模型[15]来生成全流通的随机网络拓扑。在该方法中,顶点 u 到顶点 v 的边存在的概率函数为:

$$P(u,v)=\alpha e^{-d/L\beta} \tag{2}$$

其中,建模参数 $\alpha,\beta\in(0,1]$,d 是顶点 u 和顶点 v 之间的距离,L 是平面内所有顶点中相距最远的两点之间的距离。α 值越大,图中边越多;β 值越大,图中长边比短边的比值越大。使用 C++程序在 15×15 的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 个节点为攻击节点,生成的最后一个节点为目标节点;同时,设置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。不同规模的随机网络拓扑参数如表 5 所列。

表 5 随机网络拓扑参数列表

No	网络规模	最大链路概率参数(α)	长链路比率参数(β)	节点平均度
1	20	1	0.15	4
2	500	1	0.15	9
3	800	1	0.15	7
4	1000	1	0.15	10
5	1600	1	0.15	8
6	2000	1	0.15	10
7	3000	1	0.15	10
8	4000	1	0.15	11
9	5000	1	0.15	9

现以网络规模为 20 个节点为例来说明本文方法的有效性。使用 Waxman 模型生成的 20 个节点网络拓扑如图 4 所示,共有 61 条边,其中 1 表示相交的两节点存在边;0 表示不存在。

0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1
0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0
0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1 0
0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0

图 4 20 个节点网络拓扑图

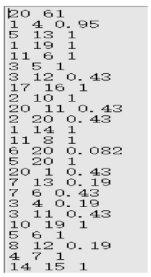

图 5　20 个节点 61 边详情图

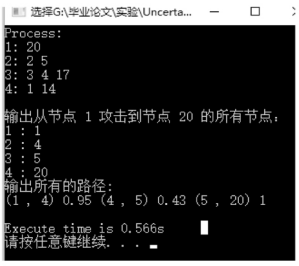

图 6　运行本文算法得到的结果

使用式(2)得出两节点存在边时边的概率(见图 5，因文件较长，只截取了一部分)。通过运行本文算法 1－算法 3，得到如图 6 所示的攻击结果，其中得到一条攻击路径 p＝host1→host4→host5→host20；其可靠性 $Rp=0.95*0.43*1=0.41$。其中生成的主机漏洞都是 4.1 节表 2 的漏洞，若要补充更多的漏洞请参考文献[13]。将集中的漏洞随机地分配给每一个主机。设置不确定度阈值 ε 为 0.2。实验环境为 Windows 10 专业版＋Inter(R) Core(TM) i3-3220 CPU @ 3.3GHz＋4.0GB 内存＋VS2013。运行本方法的运行时间比较如表 6 所列。

表 6　运行时间比较

主机数	边数	运行时间/s	边数	运行时间/s	边数	运行时间/s	边数	运行时间/s
500	1000	0.385	1500	0.385	2000	0.332	2500	0.4
800	1600	0.357	2400	0.408	3200	0.605	4000	0.785
1000	2000	0.323	3000	0.369	4000	0.437	5000	0.672
1500	3000	0.348	4500	0.355	6000	0.316	7500	1.142
2000	4000	0.347	6000	0.554	8000	0.398	10000	0.67
3000	6000	0.354	9000	0.39	12000	0.322	15000	0.301
5000	10000	0.322	15000	0.585	20000	1.536	25000	0.323

图 7 是在同样配置环境下使用模型检测方法获得的攻击路径图，使用模型检测方法的时间复杂度约为 $O(N^4)$(N 表示图中节点数)，详见文献[11]。

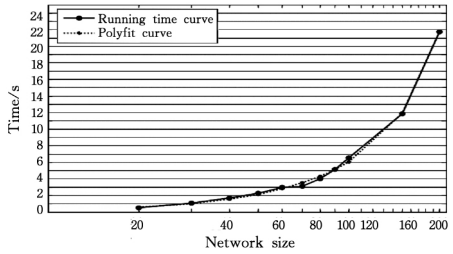

图 7　基于模型检测的时间规模图

根据表 6 可以看出，本文方法对大规模网络是适用的，而且可以在较快的时间内正确地生成攻击图。从图 7 可以看出，当网络规模较小时优势还不是很明显，随着网络规模的增大，模型检测方法所需要记录的中间状态数呈指数型增长，会导致它的状态搜索占据较多时间，使其生成攻击图的时间呈多项式时间增长。新方法直接从目标结果出发，不需要搜索中间过程，使得时间复杂度降低，减少了计算时间。

结束语　本文设计了一种生成攻击图的算法。该方法主要从两个方面进行了优化：1)从目标主机回溯到攻击者，减少了很多从攻击者出发搜索的中间状态计算的花费；2)改进搜索策略，避免对已经加入攻击路径的主机进行重复搜索。同时，从理论上分析了算法的时间复杂度的范围，通过实验验证了该方法的效果。今后还可以从以下两个方面进行研究：

1)对攻击路径进行预测，对一定会加入的主机和一定不会加入的主机进行预处理，从而减少计算花费，提高效率。

2)优化不确定图的不确定函数，可以根据文献[12]的不确定函数对网络进行实时建模，以此来优化网络的不确定性，并且验证其可行性。

参 考 文 献

[1] 中国互联网网络信息中心[OL]. [2016-08-03]. http://www.cnnic.net.cn.

[2] CNNIC. 中国互联网网络发展状况统计报告(2016)[R]. 北京：中国互联网络信息中心，2016.

[3] 国家互联网应急中心[OL]. [2016-06]. http://www.cert.org.cn.

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

[5] SCHNEIER B. Secrets and Lies[M]. John Wiley and Sons，2000：318-333.

[6] MCDERMOTT J P. Attack net penetration testing[C]// Proceedings of the 2000 Workshop on New Security Paradigms. ACM，200：15-21.

[7] PHILLIPS C，SWILER L P. A graph-based system for network-vulnerability analysis[C]// Proceedings of the 1998 Workshop on New Security Paradigms. ACM，1998：71-79.

[8] HEWETT R，KIJSANAYOTHIN P. Host-centric model checking for network vulnerability analysis[C]// Computer Security Applications Conference，2008(ACSAC 2008). Annual. IEEE，2008：225-234.

[9] MAGGI P，POZZA D，SISTO R. Vulnerability modelling for the analysis of network attacks[C]// Third International Conference on Dependability of Computer Systems，2008(DepCos-RELCOMEX'08). IEEE，2008：15-22.

[10] MALHOTRA S，BHATTACHARYA S，GHOSH S K. A vulnerability and exploit independent approach for attack path prediction[C]// IEEE 8th International Conference on Computer and Information Technology Workshops，2008(CIT Workshops 2008). IEEE，2008：282-287.

[11] 刘强，殷建平，蔡志平，等. 基于不确定图的网络漏洞分析方法[J]. 软件学报，2011，22(6)：1398-1412.

[12] 高原. 不确定图与不确定网络[D]. 北京：清华大学，2013.

[13] GAO X L. Regularity index of uncertain graph[J]. Journal of Intelligent & Fuzzy Systems，2014，27(4)：1671-1678.

[14] BUGTRAQ. The security vulnerabilities mailing list[OL]. http://www.securityfocus.com.

[15] WAXMAN B M. Routing of multipoint connections[J]. IEEE Journal on Selected Areas in Communications，1988，6(9)：1617-1622.