第36卷 第21期 计 算 机 工 程 2010年11月
Vol.36 No.21 Computer Engineering November 2010
·安全技术· 文章编号：1000—3428(2010)21—0159—03 文献标识码：A 中图分类号：TP301.6

基于节点分值的攻击图量化评估算法

陈其云[1]，陈秀真[1]，陈依群[2]，范 磊[1]
(1. 上海交通大学信息安全工程学院，上海 200240；2. 61398部队，上海 200137)

摘 要：提出一种基于节点分值的攻击图量化评估算法NSBRank，用于对攻击图中所有节点的重要性进行量化计算。通过求解节点初始等级值，得到各节点的最终等级值，使各节点间的相对重要性更明确，从而提高网络安全事件的处理能力。实验结果证明，NSBRank计算得出的攻击图节点等级值更加客观准确。

关键词：攻击图；AssetRank算法；NSBRank算法；节点分值

Quantization Evaluation Algorithm for Attack Graph Based on Node Score

CHEN Qi-yun[1], CHEN Xiu-zhen[1], CHEN Yi-qun[2], FAN Lei[1]
(1. School of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200240, China; 2. Unit 61398, Shanghai 200137, China)

【Abstract】 This paper proposes a quantization evaluation algorithm for attack graph based on node score named NSBRank, which is used to quantize the importance of all nodes in the attack graphs. It computes the initial ranks of all nodes, and works out the final ranks on the basis of the initial values, so that the relative importance between nodes is more clear, and the ability of response for network security is enhanced. Experimental results prove that NSBRank is more reasonable and effective.

【Key words】 attack graph; AssetRank algorithm; NSBRank algorithm; node score

1 概述

计算机网络的广泛应用使得网络安全问题的影响范围越来越大。但是，相对于网络管理员，攻击者总是处于暗处，这就造成了网络攻击防不胜防。攻击图是对某一网络某一既定攻击目标的所有可能攻击途径的图形抽象，其为实现主动防御提供了一种重要手段。目前主要有2种类型的攻击图：状态枚举型和条件依赖型[1-2]。然而，攻击图规模往往非常庞大，节点间相互关系十分复杂，难以辨识，更无从确定哪些需要重点关注。

文献[3]提出AssetRank算法用于对大规模网络攻击图进行分析，并对攻击图中所有节点在攻击中的重要性进行了量化并归类，依据其重要性等级标以不同的颜色，使管理员能够集中精力优先处理重要性更高的问题。但AssetRank对软件漏洞节点的重视程度不够，在计算漏洞节点等级值时也只考虑其在攻击图中的相互依赖关系及初始赋值，并未考虑漏洞本身的属性对其等级值的影响。而且，AssetRank强调了访问控制策略的重要性，例如路由器或者防火墙的ACL策略，网络管理员一般会避免这些配置上的漏洞，但软件漏洞是不可避免的。进一步地，不同的漏洞由于其被攻击者成功利用的难度系数不同，它们的重要性也不相同。漏洞的本质属性反映了利用漏洞的难度系数，包括漏洞是否可远程利用、是否需要通过认证、该漏洞被成功利用后可能造成的后果等。再者，漏洞本身有一个生命周期，从漏洞的诞生、被发现、被用来攻击到官方补丁的发布以及漏洞随着时间的推移而消失[4]。漏洞在生命周期不同阶段的重要性是不同的，因此，时间因素也应被考虑在内。

基于上述分析，本文以条件依赖型攻击图为基础，进行节点的量化评估分析，并提出了一种计算攻击图节点重要性等级的新算法NSBRank。该算法改进了AssetRank的不足之处，重新计算了攻击图中节点的重要性等级，可以优先对高级别的攻击事件采取有针对性的预防措施。在计算软件漏洞节点等级值时考虑了漏洞自身属性以及随时间的推移漏洞重要性等级的变化情况。

2 NSBRank算法设计

NSBRank算法提升了软件漏洞在攻击中的重要性，并且在计算漏洞等级值时，兼顾可能影响其结果的若干因素，这使NSBRank计算出的结果更加合理。

2.1 攻击图表示

条件依赖型攻击图可表示为 $G=(V,A,f,g,h)$，其中，V 为节点集；A 为边集，例如边(u,v)表示节点 u 依赖于节点 v；f 表示节点值函数；g 为边值函数；h 为节点类型函数，表示AND型或OR型节点。而攻击图中节点的等级值由两部分组成：固有值和流入值。固有值是节点本身重要性的体现，流入值则是上一节点对下一节点的贡献(就像被重要性高的网页所链接的网页也较重要一样)。

2.2 NSBRank算法

该算法首先求出节点的初始等级值，然后计算出最终的

基金项目：国家自然科学基金资助项目(60605019, 60772098)；国家“863”计划基金资助项目(2007AA01Z473)；教育部博士点基金资助项目(20070248002)；上海市曙光学者基金资助项目

作者简介：陈其云(1981－)，男，硕士研究生，主研方向：信息安全；陈秀真，讲师、博士；陈依群，高级工程师、博士；范 磊，副教授、博士

收稿日期：2010-05-05 **E-mail：**chenqiyun@yahoo.cn

节点等级值。攻击图中节点 v 的初始等级值计算如下：

$$x_v = (1-\delta)f(v) + \delta \sum_{u \in N_(v)} g(u,v)x_u \quad (1)$$

其中，δ 是影响因子，体现了节点初始等级值中固有值与流入值对最终值的影响程度；$f(v)$是节点 v 的固有值；$g(u, v)$为边值；x_u为节点 u 的初始等级值；$N_(v) = \{u \in V:(u, v) \in A\}$，即依赖于节点 v 的所有节点的集合。

本文把所有节点的初始等级值放在一起组成一个向量 $\boldsymbol{X}$。相应地，边值 $g(u, v)$组成邻接矩阵 $\boldsymbol{D}$，反映节点间的依赖关系，$D_{vu}=g(u, v)$。当 $g(u, v) \notin A$ 时，$g(u, v)= 0$。同样，所有节点的固有值组成向量 $\boldsymbol{IV} = f(V)$。式(1)便可转化为下面的线性方程：

$$\boldsymbol{X} = (1-\delta)\boldsymbol{IV} + \delta \boldsymbol{DX} \quad (2)$$

解得的 $\boldsymbol{X}$ 即对应的各节点初始等级值所组成的向量。该线性方程可用 Jacobi 方法迭代下式求得：

$$\boldsymbol{X}_t = \delta \boldsymbol{D}\boldsymbol{X}_{t-1} + (1-\delta)\boldsymbol{IV} \quad (3)$$

软件漏洞在攻击图中的重要性受本身属性及时间因素的影响，本文利用 CVSS(Common Vulnerability Scoring System)对软件漏洞进行评估，以期客观准确地反映漏洞的重要性等级[4]。CVSS 为软件漏洞的评估量化提供了开放的统一标准，解决了不同软件厂商对软件漏洞评估有各自不同的标准这一难题。CVSS 对漏洞的打分包括基本分(BaseScore)、时间分(TemporalScore)和环境分(EnvironmentalScore)3 项。后一项在前一项的基础上综合累进得出，每一项的分值范围均为[0,10]。基本分是 3 项中最重要的部分，代表了漏洞本身的属性，不随时间的变化而改变。而时间分体现的是漏洞重要性随时间而产生的变化。环境分涉及具体环境情况，因此，本文舍弃了环境分计算这一步。NVD(National Vulnerability Database)提供了所有已知漏洞的基本分，为漏洞评级带来了方便。各漏洞的绝对分值意义不大，因此，本文漏洞取相对分值，称为 CVSS 相对分值，用 S 表示，范围相应设置为[0,1]。

NSBRank 算法在初始等级值的基础上引入了 e^S，以调整节点的等级值。在调整的过程中只涉及攻击图中漏洞节点评分，权限节点的等级值保持不变，即此时 S=0。因此，定义节点 i 的最终等级值为：

$$r_i = x_i e^{S_i} \Big/ \sum_{j=1}^{N} x_j e^{S_j} \quad (4)$$

其中，N 为所有节点的个数；s_i 为漏洞节点 i 对应的 CVSS 相对分值；x_i 为节点 i 的初始等级值。式(4)较大地提升了漏洞节点的等级值，突出了漏洞在攻击中的重要作用。

3 实验测试与分析

实验利用 NSBRank 算法计算反映攻击图中各节点重要性的等级值。实验网络环境如图 1 所示，该环境的拓扑结构及网络配置情况十分典型。攻击者可以通过 Internet 访问 Web 服务器，Web 服务器可以访问数据库，内网用户可同时访问 Web 服务器和 Internet。而攻击者的目标是获取数据库服务器用户权限。实验所涉及的漏洞信息如表 1 所示。

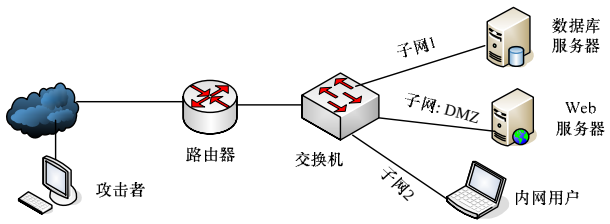


图 1 实验网络环境

表 1 实验涉及的漏洞信息

CVE 号	CVSS 基本分值	描述
cve-2002-0869	7.5	IIS 服务器漏洞，允许远程攻击者获得本地用户权限
cve-2005-4832	7.5	Oracle 服务器漏洞，允许远程攻击者执行任意 SQL 命令，获得权限提升
cve-2006-2784	5.1	Firefox 浏览器漏洞，攻击者诱使用户访问恶意网站，并利用该漏洞达到获取系统用户权限

图 2 由攻击图生成工具 MulVAL[2]生成。图中节点颜色代表其重要性级别，颜色越深代表重要性越高。用式(3)进行迭代计算时，取 δ=0.85，这是 Google 数据所证明的合理值。在邻接矩阵 $\boldsymbol{D}$ 中，对任意的 $\boldsymbol{X}_0$，式(2)总是收敛于式(1)的解[5]。此处，$\boldsymbol{X}_0$=$\boldsymbol{IV}$。由于攻击者的主要目标是数据库服务器，因此赋予节点“ execCode(databaseServer, oracle_user) ”总 $\boldsymbol{IV}$ 值的 15%，剩余 85%平均分给剩余节点。图 2 中菱形代表 OR 节点，表明该节点所依赖的任何一个节点成立均会导致该菱形节点成立；椭圆节点代表 AND 节点，即它所依赖的所有节点同时成立，此节点才成立；矩形节点代表末端节点，不依赖其他节点。

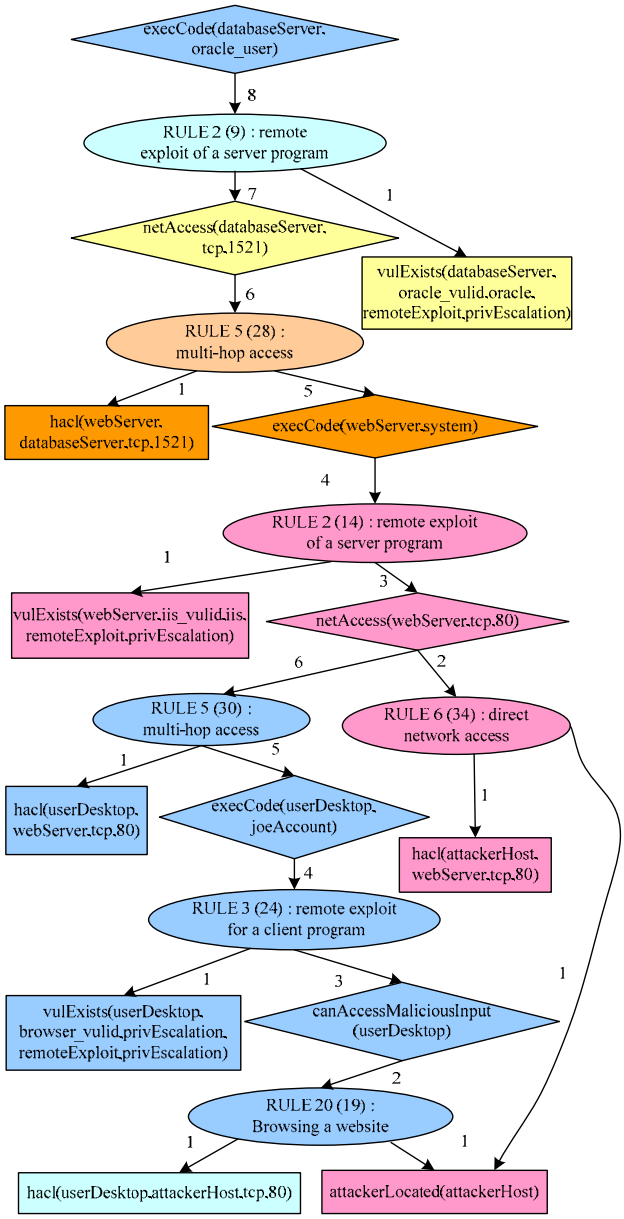


图 2 实验生成的攻击图

利用 AssetRank 计算上述攻击图节点等级值可得表 2，此处只列出了部分攻击图节点的等级值。其中一些漏洞节点的等级值很低，例如“ vulExists(userDesktop,browser_vulid, privEscalation, remote Exploit, privEscalation)”只有 0.025 31，而该漏洞经常被利用，因此，有必要提高它们的等级。表 3 则是利用 NSBRank 算法重新计算节点等级值后的数据表。

表 2　AssetRank 生成的节点等级值

Name	Rank×100
execCode(webServer,system)	4.535
execCode(databaseServer,oracle_user)	2.839
execCode(userDesktop,joeAccount)	1.566
hacl(attackerHost,webServer,tcp,80)	5.513
hacl(webServer,databaseServer,tcp,1521)	4.535
hacl(userDesktop,attackerHost,tcp,80)	3.431
…	…
vulExists(webServer,iis_vulid, …)	5.297
vulExists(databaseServer,oracle_vulid, …)	3.717
vulExists(userDesktop,browser_vulid, …)	2.531

表 3　NSBRank 生成的节点等级值

Name	Rank×100
execCode(webServer,system)	4.134
execCode(databaseServer,oracle_user)	2.588
execCode(userDesktop,joeAccount)	1.427
hacl(attackerHost,webServer,tcp,80)	5.025
hacl(webServer,databaseServer,tcp,1521)	4.134
hacl(userDesktop,attackerHost,tcp,80)	3.127
…	…
vulExists(webServer,iis_vulid, …)	9.271
vulExists(databaseServer,oracle_vulid, …)	6.505
vulExists(userDesktop,browser_vulid, …)	3.594

对比表 2 与表 3 可以看出，上述 2 个漏洞节点的等级值提高至 0.092 71 和 0.035 94，体现了其在攻击中的重要作用。因为前者 CVSS 基本分为 7.5，危险级别高，可远程实施攻击，且成功利用该漏洞的难度系数较小，所以该漏洞较危险，应该提高其等级值。后者 CVSS 基本分值虽然为 5.1，属于中等危险级别，但该漏洞会造成服务的中断，这对于 Web 服务器来说是个大问题，因此，该漏洞的等级值也应该提高。

以上实验结果证明，本文提出的 NSBRank 算法计算得出的攻击图节点等级值更客观准确，更符合实际情况。

4　结束语

本文针对在分析网络攻击图时存在的节点繁多、不便识别问题，提出了量化各节点重要性的算法 NSBRank。实验结果证明该算法客观准确，可以对攻击图中的节点进行合理有效的优先级排序。但算法有待进一步完善，例如，对攻击图中的权限节点也进行评估，而不是全部均等看待，使结果更加全面。

参考文献

[1] Noel S, Jajodia S, O’Berry B, et al. Efficient Minimum-cost Network Hardening via Exploit Dependency Graphs[C]//Proc. of the 19th Annual Computer Security Applications Conference. Las Vegas, Nevada, USA: [s. n.], 2003.

[2] Ou Xinming, Boyer W F, McQueen M A. A Scalable Approach to Attack Graph Generation[C]//Proc. of the 13th ACM Conference on Computer and Communications Security. Alexandria, Virginia, USA: ACM Press, 2006.

[3] Sawilla R, Ou Xinming. Googling Attack Graphs[R]. Defence R & D Canada——Ottawa, Tech. Rep: TM 2007-205, 2007.

[4] Mell P, Scarfone K, Romanosky S. Common Vulnerability Scoring System[DB/OL]. [2009-10-11]. http://www.romanosky.net/pubs/ieee-cvss-12-06.pdf.

[5] Bianchini M, Gori M, Scarselli F. Inside PageRank[J]. ACM Transactions on Internet Technology, 2005, 5(1): 92-128.

编辑　张　帆

（上接第 158 页）

MU 与 TA 的交互都是在离线条件下完成，所以从认证过程开始计算用户的接入时间。假设信息传播延迟忽略不计，记 $T_{H()}$ 为一次哈希运算时间，$T_{\otimes}$ 为一次模乘，$T_{\oplus}$ 一次模加，$T_{E\oplus}$ 为 ECC 上的一次加法运算，$T_{E\otimes}$ 为一次 ECC 上的乘法运算，T_{EXP} 为一次求解平方根的指数运算，计算花费如下：

MU：$T_{H()}+2T_{\otimes}+2T_{E\oplus}+T_{E\otimes}$

MP：$T_{EXP}+T_{H()}+2T_{\otimes}+(t-1)T_{E\oplus}+4T_{E\otimes}+(t-1)T_{\oplus}$

AS_i：$T_{H()}+2T_{\otimes}+T_{E\otimes}+2T_{\oplus}$

5　结束语

本文分析了文献[6]提出的认证方案，考虑实际 AP 也会遭受攻击的情况，设计了一个防失效的门限认证方案并对方案进行了安全性分析与性能分析。经过分析，本方案安全性高、接入延迟更短，便于用户快速接入。

参考文献

[1] Lamport L. Password Authentication with Insecure Communication[J]. Commun. of the ACM, 1981, 24(11): 770-772.

[2] Hwang M S, Li L H. A New Remote User Authentication Scheme Using Smart Cards[J]. IEEE Trans. on Consumer Electron., 2000, 46(1): 28-30.

[3] Chan C K, Cheng L M. Cryptanalysis of a Remote User Authentication Scheme Using Smart Cards[J]. IEEE Trans. on Consumer Electron., 2000, 46(4): 992-993.

[4] Zhu Jianming, Ma Jianfeng. A New Authentication Scheme with Anonymous for Wireless Environments[J]. IEEE Trans. on Consumer Electron., 2004, 50(1): 231-235.

[5] Lin Xiaodong, Zhu Haojin, Ho Pin-Han, et al. Two-factor Localized Authentication Scheme for WLAN Roaming[C]//Proc. of ICC’07. [S. l.]: IEEE Press, 2007: 1172-1178.

[6] Yang Yatao, Gu Yonghao, Tan Xi, et al. A New Wireless Mesh Network Authentication Scheme Based on Threshold Method[C]//Proc. of the 9th International Conference for Young Computer Scientists. Zhangjiajie, China: [s. n.], 2008.

[7] 杨君辉，戴宗铎，杨栋毅，等. 一种椭圆曲线签名方案与基于身份的签名协议[J]. 软件学报, 2000, 11(10): 1303-1306.

编辑　金胡考