doi :10.3969/j.issn.1671-1122.2011.08.026

基于漏洞模型检测的安全漏洞挖掘方法研究

徐有福[1]，文伟平[1]，万正苏[2]

（1. 北京大学软件与微电子学院，北京 102600；2. 湖南理工学院数学学院，湖南岳阳 414006）

摘 要：文章通过研究自动机原理，提出了基于漏洞模型检测的安全漏洞挖掘理论，为批量发掘未知漏洞提供一定的理论基础。

关键词：安全漏洞；自动机；模型检测

中图分类号：TP393.08 **文献标识码：**A **文章编号：**1671-1122（2011）08-0072-04

Vulnerability-based Model Checking of Security Vulnerabilities Mining Method

XU You-fu[1], WEN Wei-ping[1], WAN Zheng-su[2]

(1.*Department of Information Security, SSM, Peking University, Beijing* 102600*, China;*
2. *Hunan Institute of Science and Technology School of mathematics, Yueyang Hunan* 414006*, China*)

Abstract: The paper though studying automata theory, proposed a vulnerability-based model checking Mining vulnerability theory, as the bulk discover unknown vulnerabilities to provide a theoretical basis.

Key words: security vulnerabilities; automaton; model checking

0 引言

目前，Windows 系统漏洞层出不穷，每年发布的安全公告和漏洞补丁呈上升趋势，经统计，发现微软在 2010 年全年总共发布了 106 个安全公告，修补了 247 个漏洞，比 2009 年多 57 个，其中 41 个高危漏洞，60 个重要漏洞和 5 个一般等级漏洞[1]。Windows 漏洞如此之多，对漏洞分析和漏洞挖掘工作带来了巨大的挑战。

笔者在北京大学软件与微电子学院软件安全研究小组[2]从事软件安全漏洞研究工作，通过对大量的漏洞分析，发现其中的规律和共性，通过研究自动机原理，提出了基于漏洞模型检测的安全漏洞挖掘理论，使用该理论指导实际的漏洞挖掘工作。相比传统的软件安全漏洞挖掘方法，该方法基于模型检测理论，对发掘与已知漏洞类似的未知漏洞的挖掘工作具有特殊意义。

1 理论基础

1.1 有限状态自动机

所有的计算机高级语言，比如：C、C++、Pascal 等，都遵循 BNF（巴科斯－诺尔范式），可以看作是一种上下文无关文法。文法和自动机分别从产生和识别的角度来定义计算机语言，两种方式是等价的，两种方式之间也可以进行相互转化，而上下文无关文法对应的是下推自动机[3]。所以，计算机中各种高级语言实现的软件系统都遵循下推自动机的相关性质，可以理解为不同程序状态之间的转移，可以利用自动机相关的性质来对软件的安全性进行检测。

以自动机的理论来分析一个软件系统，其实就是程序从输入开始，之后通过在不同的中间状态之间转移，最后执行到输出结果的一种自动机，如图 1 所示。

从图 1 可以看出，如果软件系统处在安全状态下，也就是程序状态的转移都是按照程序设计者的设计思路进行变化的，那么所有的程序状态也是在程序状态的空间之内的，最后执行到正常结束的状态。如果程序状态转移到了不在程序本身所设置的状态空间时，也就是超出了程序本身的状态空间，可能导致程序最后的停机状态不是结束在程序的原先设计的结束状态，那么这个程序执行流程是存在风险的。

收稿时间：2011-07-12

作者简介：徐有福（1985-），男，江西，硕士研究生，主要研究方向：网络安全；文伟平（1976-），男，湖南，副教授，博士，主要研究方向：网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等；万正苏（1976-），男，湖南，副教授，博士，主要研究方向：微分方程数值解等。

图1 自动机状态转移

可以使用自动机对软件程序进行抽象的模型表述，针对软件系统本身属性做到直接的安全性分析。虽然下推状态自动机可以较好地解释软件程序的执行流程，但是这种自动机对系统的属性限制比较少，表现范围太大，对程序安全性分析的需求并不能有效满足。因为在实际软件系统中各个模块之间的相互依赖关系比较强，而各个程序状态的转移也是存在很多条件限制的，所以对程序描述的形式语言需要具有更强的限制性和依赖性。有限状态自动机（Finite State Automaton，FSA）就是为了研究计算机内存执行过程和某些高级语言类而抽象得出的一种计算机模型。

有限状态自动机的执行状态数量是有限的，可以从一个状态转移到零个或多个状态，输入的限制条件用于确定状态的转移选择。有限状态自动机可以对应的形式语言是正则语言，正则语言相对上下文无关语言的约束性更强，具有条件判断等。有限状态自动机可以表示为一个抽象的模型：$FSA=(Q,\Sigma,\delta,q_0,F)$，变量说明如下：$Q$：有限状态的集合；$\Sigma$：输入字符集合；$q_0$：程序开始状态 $q_0 \in Q$；F：接受和终止状态集合$F \subset Q$；δ：$Q\times\Sigma\to Q$表示状态转化函数，$\delta(q_i,x)=q_j$表示在状态 q_i 时接受输入 x 后状态转移到 q_j。

有限状态自动机很好的抽象了软件程序在计算机中的执行过程。可以对软件系统进行静态分析，对功能和流程进行抽象解析，然后通过有限状态自动机进行模拟软件系统的执行流程，这样可以使用静态漏洞检测具有一定的动态漏洞检测的部分模型推导功能（状态转移模型推导实际程序执行过程中的状态转移）。

1.2 模式识别

由给定的具体模式的特征来识别它属于哪类的问题称为模式识别。软件漏洞挖掘过程实际上就是一个模式识别的过程，通过对软件的执行输入、软件调用流程、功能集进行状态推导，从而判断状态转移是否会超出安全的范围。对于已公开的漏洞可以进行特征码扫描和特定功能或者特定流程检测来确定该软件系统是否存在该漏洞。对于未知漏洞一般是不能通过直接的检测得到软件系统是否存在漏洞的，因为未知漏洞往往针对不安全的代码的特定执行流程和 / 或特定的输入条件才会导致漏洞的产生，而直接检测到的信息具有随机性、重叠性和无关性等特征，使得检测的结果具有模糊性。模糊识别对这种状况具有比较有效的处理方式[4]。

不失一般性，设所分类的状态具有 n 个可以提取的特征，对于某个对象，对应于这些可提取的特征的隶属度为：μ_1，μ_1，$\cdots\mu_n$。而每个特征对于模式识别的作用是不同的，所以针对不同隶属度分别赋予不同的权值：a_1，a_1，$\cdots a_n$，然后给出限制阀值 θ，如果$F\geq\theta$则该检测对象属于 θ 类，最终通过设置值完成分类的目的。

1.3 模型检测

模型检测是一项比较简单、自动化程度较高且非常适用于安全漏洞的实际检测的理论。模型检测能够比较精确的通过形式化描述证明软件系统的执行，并能够以自动机的形式化语言对软件程序进行形式化建模，可以合理地描述模型中各个模块的不同属性和属性之间的依赖关系，方便分析人员对软件系统的检测和分析。整个模型检测过程如图 2 所示。

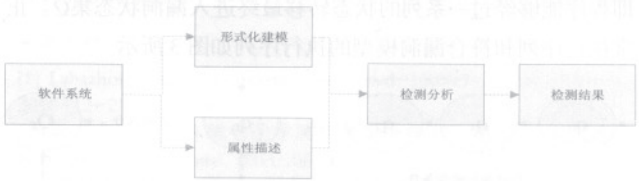

图2 模型检测过程

使用自动机理论对软件模型检测的建模和自动机本身的模型比较类似，软件系统的模型也是一个五元组，$M=(V,\Sigma,V_0,\Delta,L)$，其中：$V$：系统状态集合；$\Sigma$：系统输入；$V_0\subseteq V$：系统的开始状态；$\Delta\subseteq V\times\Sigma\times V$：状态转移关系；$L$：系统属性描述。

使用上面的模型可以将软件系统的执行过程通过 V 中状态的转移序列进行$v_0v_1\ldots v_i$描述，所以 M 就可以作为软件系统的理论分析模型。另外，模型中属性对模型分析也起到非常重要的作用，特别是对当系统中各个模块具有关联和依赖性的时候，属性的影响将比较明显。

1.4 基于漏洞模型检测的安全漏洞挖掘理论

本课题通过对以上理论基础的研究提出了基于漏洞模型检测的安全漏洞挖掘理论，该理论主要用于指导软件安全漏洞自动 / 半自动化挖掘过程，是一种以静态检测为基础，并通过漏洞模型匹配找出脆弱点，然后通过人工分析脆弱点并进行动态调试，最终验证已知漏洞和挖掘未知漏洞的一套方法理论。

基于 Windows 内核漏洞挖掘的分析对象一般是 Windows 的系统 dll、sys 以及系统组件 exe 等文件，模型所检测的通常不是整个操作系统的完全执行流程，而是分析和检测待分析目标文件中的执行流程，所以本检测中所指的软件系统是 Windows 的某个模块的执行流程，当然也适用于真正的软件系统。

系统抽象为一个有限状态自动机，模型完全按照前面的有限状态自动机模型$F_{sa}=(Q,\Sigma,\delta,Q_0,Q_f,Q')$，其中：

Q：系统状态集合，程序计算、异常处理、执行终止都属于这个集合；Σ：输入字符集合，这里不仅包括程序输入还包括漏洞模型输入；Q_0：程序开始状态集$Q_0\subset Q$，由于程序进入分析模块的状态不止一种情况，所以这里开始状态也以集合表示；Q_f：接受和终止状态集合$Q_f\subset Q$，其中程序终止状态可以是程序正常执行终止，也可以是程序运行出错但是按照程序所设计的异常处理流程执行的终止状态；Q'：表示符合

漏洞模型的状态集，Q'可以满足$Q' \subseteq Q$也可以$Q' \not\subset Q$，可以使程序执行到正常情况没法处理的状态，或者正常情况不应该转移到的状态；$\delta : Q \times \Sigma \to Q$表示状态转化函数，$\delta(q_i, x) = q_j$表示在状态$q_i$时接受输入 x 后状态转移到q_j，$q_i, q_j \in Q$。

软件系统正常的实际执行流程就可以抽象成状态转换序列$q_0 \to q_1 \to \dots \to q_m \to q_f$，其中所有的状态$q_x \in Q$，其中程序执行结束到状态$q_f$，$q_f \in Q_f$，$Q_f$表示终止状态集合。

漏洞模型执行序列可以抽象表示为$q_0' \to q_1' \to \dots \to q_m' \to q_f'$，其中$q_0', q_1', q_m' \in Q$，符合漏洞模型的程序执行终止状态$q_f' \in Q'$。即程序能够经过一系列的状态转移最终进入漏洞状态集Q'。正常执行序列和符合漏洞模型的执行序列如图 3 所示。

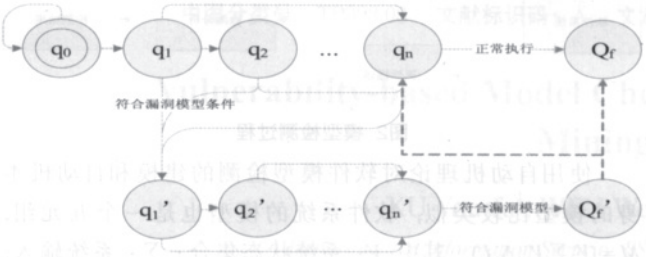

图3 正常和符合漏洞模型的执行序列

如图 3，如果程序正常执行，最终将结束在正常终止集合Q_f中，如果程序符合漏洞模型则执行结束在或者经过漏洞状态集Q'，如图 3，如果终止在Q'，则程序结束，如果是经过漏洞状态集Q'则程序可能会通过虚线回到正常状态转移序列。并且出现漏洞条件符合情况可以是上面的任何一个状态转移到下面的任何一个状态，图中只画了q_1, q_2, q_n到q_1'的转移，没有画出q_1, q_2, q_n都是可以转移到q_2'到q_n'的。

所以检测一个系统是否具有特定漏洞模型的漏洞，其实就是检测正常状态转移序列能否通过特殊的状态属性的改变或者其他一些输入条件，使得正常状态转移到漏洞状态序列。

而一个状态下不同的属性对状态的转移的影响程度是不一样的，也就是说一个状态下每个属性 s_i 对该状态转移的决定性是一个具有概率性 [0,1], 0 表示不可能；1 表示肯定转移。比如：如果是一个字符串拷贝操作，某状态下如果目的缓冲区比源字符串缓冲区小，那么这个拷贝发生状态转移到缓冲区溢出的概率就是 1。这里使用模糊识别中的隶属度 μ 来表示某状态 q_i 下通过其属性 s_i 能够转移到漏洞状态q_i'的概率，表示为$\mu_i(q_i, s_i) \in [0,1]$。则软件系统下程序执行过程模糊集合可以使用隶属度序列表示为 U：$(\mu_1, \mu_2, \dots, \mu_n)$。同理，漏洞状态执行模糊集可以表示为$U'$：$(\mu_1', \mu_2', \dots, \mu_n')$，注意这里的$\mu_i$和$\mu_i'$表示的是某状态下其某个属性 s_i 对其影响概率。

所以，检测一个软件系统是否存在某个漏洞模型的漏洞，其实就是检测该软件系统的隶属度序列 U 和该漏洞模型隶属度序列U'的贴近度度$\sigma(U, U') \in [0,1]$。在模糊数学中求两个模糊集合的贴近度使用内积和外积来表示，通过“格贴近度”进行贴近度描述，·表示内积，× 表示外积 [5]。

2 漏洞建模与验证

2.1 漏洞建模

本文以最常见的缓冲区溢出为基础，进行简单的漏洞模型建立。

缓冲区溢出的根本原因是程序所分配的内存空间被所拷贝的内容超过导致的，而根据不同的内存操作方式，缓冲区溢出漏洞可以分为两种：一、本身调用不安全的漏洞导致出错。二、循环对内存进行复制操作，超出缓冲区边界，导致内存出错。

不安全函数调用模型：不安全函数主要包括 Windows 的一些没有判断输入长度的内存和字符串操作函数，比如：strcpy、strcat、sprintf 等。而这里又分参数数量固定和不固定的情况，strcpy、strcat 就是参数数量确定的，而 sprintf 则为参数数量不确定的。

第一类：针对参数数量确定的，下面以两个参数的 strcpy 为例建立模型：

算法输入：待分析的文件。

算法输出：存在安全问题的地址。

算法描述：$n_i[i \in N^*]$表示函数名$n_i \in F$，F 表示反汇编代码集合，N 需要分析的函数名，$a_i[i \in N^*]$表示函数 n_i 的函数地址。

1）通过需要分析的函数名 N 获取$if(n_i == N)$，获得 n_i 的地址 a_i；

2）从 a_i 开始分析，获取目标地址缓冲区大小 destminsize，获取源数据缓冲区大小 srcminsize；

3）对缓冲区大小进行判断（具体判断参考 4.5 节中的脚本描述）：（1）if(((destminsize == –1)||(destminsize == 0)) && (srcminsize != –1))，提示目标区大小未知，返回 a_i；（2）if((destminsize != –1) && ((srcminsize == –1) || (srcminsize == 0))，提示源区大小未知，返回 a_i；（3）if(((destminsize == –1)||(destminsize==0)) && ((srcminsize == –1)||(srcminsize==0))，提示目标区和源区大小未知，返回 a_i；（4）if(destmaxsize < srcminsize)，提示存在溢出情况，返回 a_i；（5）if(destminsize < srcmaxsize)，提示存在溢出情况，返回 a_i；（6）if(destminsize < srcmaxsize)，提示存在溢出情况，返回 a_i；；

4）继续分析的函数名 N 获取$if(n_{i+1} == N)$，获得n_{i+1}的地址 a_{i+1}，直到遍历 F 结束。

5）输出分析结果a_i和其对应的提示信息。

第二类：针对参数数量不确定的，下面以 sprintf 为例 sprintf(dest,“%s…%s”,pram1,pram2,…,pramn):

算法输入：待分析的文件。

算法输出：存在安全问题的地址。

算法描述：$n_i[i \in N^*]$表示函数名$n_i \in F$，F 表示反汇编代码集合，N 需要分析的函数名，$a_i[i \in N^*]$表示函数 n_i 的函数地址。

1）通过需要分析的函数名 N 获取$if(n_i == N)$，获得 n_i 的地址 a_i；2）从 a_i 开始分析，获取目标地址缓冲区第二个参数

内容为字符串S，先去除“%%”得到s，因为两个%表示转译，表示字符串本身是%，而“%s”表示格式化字符串，后面需要跟参数；3）循环查找后面的参数pram1,pram2,…,pramn（如果没有参数则赋值hasstring=0），并获得每个参数的缓冲区大小$len(i)[i\in(0,n)]$，并计算所有参数缓冲区大小$srclen=\sum_{0}^{n}len(i)+len(s)$，获取第一个参数即目标缓冲区大小tgtsize；4）对缓冲区大小进行判断：（1）if(（tgtsize == 0）‖（tgtsize == -1））），提示目标区大小未知，返回a_i；（2）if(（tgtsize < srclen）&&（hasstring != 0）)，提示存在溢出风险，返回a_i；5）继续分析的函数名N获取$if(n_{i+1}==N)$，获得n_{i+1}的地址a_{i+1}，直到遍历F结束；6）输出分析结果a_i和其对应的提示信息。

2.2 实例验证

图4为通过缓冲区溢出模型验证某即时通讯软件的更新程序存在的漏洞的分析结果，成功发现存在漏洞溢出风险的位置。

图4 实验结果

3 结束语

本文通过研究和分析Windows已公开的漏洞和研究有限状态自动机理论和模糊模式识别理论原理，在此基础上提出了基于漏洞模型检测的安全漏洞挖掘理论。并基于该理论设计了部分验证的漏洞模型，通过实例检测验证该理论的合理性，为批量发掘软件安全漏洞指出了一个方向。（责编 杨晨）

参考文献：

[1] Labazhou. http://bbs.pxysm.com/thread-130999-1-1.html[EB/OL], 2010-12-21/2011-06-12.
[2] 北京大学软件与微电子学院软件安全研究小组 [EB/OL]. http://www.pku-exploit.com/, 2011-06-12.
[3] 陈文宇，欧齐，程炼．形式语言与自动机 [M]. 北京：人民邮电出版社，2005. 11-12.
[4] 梁保松，曹殿立．模糊数学及其应用 [M]. 北京：科学出版社，2007. 103-104.
[5] 张晛譻．基于Windows平台的软件安全漏洞发掘技术研究 [D]. 成都：电子科技大学，2010.

上接第62页

息均应受到管理和约束。同时，结合国务院的行政法规，更加明确省、自治区、直辖市通信管理部门的责任，采取属地管理原则和“谁主管谁负责”的管理方式，也就是说，服务器在何地，当地的通信管理部门就应当承担起管理的责任。

3.3 实体法向虚拟社会延伸

随着微博工具的到来，许多传统的实体法和互联网规则无法进行覆盖和涉及，以致造成了在管理这些工具和系统的时候往往处于无法可依的窘境。笔者认为：虚拟社会只是现实社会在互联网领域的延伸，许多在现实社会的法律法规可以向互联网世界进行覆盖和延伸。在我国，如果违反法律发表危害国家安全、分裂国家的言论，情节严重的话可以依照刑法以煽动国家分裂罪或者煽动颠覆国家政权罪处罚。那么如果在互联网上发表这些危害国家的言论是否需要追究刑事责任呢？答案是肯定的。那么运用互联网上的微博工具发表这些危害国家的言论，笔者认为也应当根据相关实体法进行责任的追究。另外，目前针对在互联网上出现的诈骗犯罪高发，我国司法机关单独为网络诈骗犯罪进行了司法解释。笔者认为：这非常必要，因为传统的诈骗犯罪需要造成一定的经济损失，且损失的金额必须达到刑法规定的法定数标准才会对其进行刑责的追究，而在互联网上的诈骗往往不针对特定主体，群发邮件、群发QQ、群发微博，谁相信谁就上当受骗，而且这种诈骗活动往往取证困难和落地困难，故此司法机关针对互联网诈骗犯罪进行了司法解释，规定群发5000条信息即可构成犯罪，这种做法也就是笔者所提到实体法向虚拟社会延伸的具体办法。

3.4 运用实名制制约信息发布

笔者建议通过对电子论坛版主管理方式来制约信息发布。第一步：对于新出现的博客、微博、个人空间必须采用网络实名制的方式进行管理。目前，申请博客、微博、个人空间不需要任何手续，只要有一个经常使用的邮箱地址即可以申请成功。笔者建议：国家管理部门应当规定新申请的博客、微博、个人空间必须采用实名制的管理方式，与真实姓名、身份证号、移动电话、IP地址进行审核，审核通过后，才允许开通博客、微博和个人空间；第二步：对以往已开通的博客、微博、个人空间进行溯源管理，明确实名制管理方式。目前，已开通使用的博客、微博、个人空间没有与现实身份绑定，无法核实用户的真实身份，从而给管理工作带来很大的困难。笔者建议：对以往已开通的博客、微博、个人空间的实名制管理设定时间表，在一定期限内规定已开通的用户必须进行实名登记，否则将终止其微博的使用，这样一来，就可以解决在微博类系统发布信息无法溯源的问题出现；第三步：针对微博上信息的发布，明确规定互联网服务提供商具有相应的管理责任，规定网民在其博客、微博、个人空间中发布信息必须受到审查才可以发布，以减少和杜绝互联网上的一些危害言论的出现。（责编 杨晨）

参考文献：

[1] 中华人民共和国国务院令第292号《互联网信息服务管理办法》.2000-09-20.
[2] 中华人民共和国信息产业部第三号令《互联网电子公告服务管理规定》，2000-10-08.
[3] 网易科技．Facebook即时通讯系统每天发送信息10亿条 [EB/OL]. http://tech.163.com/09/0616/15/5BUIFHKI000915BF.html, 2009-09-16/2011-04-01.
[4] 凤凰网．欧盟警告Facebook和Google：应遵守隐私新规 [EB/OL]. http://www.enet.com.cn/article/2011/0317/A20110317839979.shtml, 2011-03-17/2011-04-01.