

插件技术在漏洞分类扫描中的应用

谢惠琼, 凌捷

(广东工业大学 计算机学院, 广东 广州 510006)

摘要: 根据漏洞的威胁相关性,对网络安全漏洞进行重新分类.分析了主要的漏洞扫描技术及插件的开发方法.构建了分类漏洞插件,并给出了扫描插件的构造模型和主要的设计过程.实现了对漏洞的针对性扫描,为快速有效地更新扫描插件提供便利.

关键词: 信息安全; 威胁; 漏洞扫描; 插件

中图分类号: TP393.08

文献标志码: A

文章编号: 1007-7162(2011)01-0008-04

随着计算机技术的发展与网络的普及,包括政府在内的企事业单位对系统信息的依赖越来越强,信息系统的稳定对政府及企业发展影响深远.漏洞扫描作为信息安全的重要内容,可以对系统进行全方位的扫描,检查当前的系统是否存在漏洞,检测出系统的大量信息,如开放的端口、提供的服务、软件版本及系统漏洞等.

漏洞扫描已成为管理员评估系统的主要技术手段.通过漏洞扫描可以探测出信息系统存在一些问题,使管理员可以提前预知风险,采取相应的措施进行改进,从而避免安全事故的发生^[1].本文根据漏洞扫描的原理、技术,将漏洞按威胁相关性进行分类,并结合插件技术设计了分类漏洞插件,给出了插件的设计过程.

1 漏洞

漏洞也叫脆弱性,是计算机系统在硬件、软件、协议的具体实现或系统安全策略上存在的缺陷和不足.攻击者可以利用漏洞在未授权的情况下访问或破坏系统,从而威胁计算机系统安全.

1.1 漏洞分析

漏洞具有长久性、隐蔽性、客观存在性和必然发现性,目前还没有一个统一的规范来划分漏洞,不同的研究机构对漏洞的划分也不完全相同.单从漏洞产生的原因就可以归为:输入验证错误、造成缓冲区溢出、设计错误、意外情况处置错误、访问验证错误、配置错误、竞争条件、环境错误^[2].

系统脆弱性被利用后对信息系统产生的危害称

为威胁后果.威胁可能给信息系统带来不同层面的损害,如,常见的威胁后果包括“不能进行关键操作”、“无法访问”、“丢失关键信息”、“危害公共安全”等.许多漏洞之间都有一定的联系,当信息系统的一个弱点可以被威胁利用时,也可能被另外一个威胁利用,所以威胁的产生往往是因为相同的漏洞被利用.而且当一个漏洞被利用后,与之相关的漏洞被利用的可能性也将大大提高,所以漏洞与漏洞之间有很大的相关性.

1.2 漏洞扫描技术

漏洞扫描作为一种主要的网络安全防御技术,与防火墙技术、入侵检测技术、加密和认证技术处于同等重要的地位,通过收集目标系统的信息来检测系统和网络安全弱点.漏洞扫描的目的主要包括:获取主机名和IP地址;获取Telnet漏洞信息;获取SNMP漏洞信息;获取TCP端口信息;获取Finger漏洞信息;获取Web服务器信息;获取CGI信息;获取IIS漏洞信息等^[3].

漏洞扫描主要有知识匹配和模拟攻击两种方法.基于知识匹配的漏洞检测,就是根据每个漏洞存在的基本环境形成相应的漏洞知识上下文,比如操作系统类型和版本、所开放服务的应用程序和版本等,然后通过收集目标主机的相关基本信息,如果满足知识库中所描述的基本前提,那么就认为目标具有相应的安全漏洞,否则就认为没有.模拟攻击就是模拟黑客攻击手法对目标主机实施攻击,逐项检查目标系统可能存在的漏洞.如果模拟攻击达到了漏洞利用的结果,那么就认为目标具有相应的安全漏

收稿日期: 2010-08-20

基金项目: 广东省自然科学基金资助项目(9151009001000043); 广东省科技计划项目(2009B060700002)

作者简介: 谢惠琼(1986-),男,硕士研究生,主要研究方向为网络安全技术.

洞,反之就认为没有。常用的模拟攻击方法有:IP 欺骗、缓冲区溢出、DDOS、口令攻击等。

2 插件技术

插件就是很多通用或专用程序提供给应用层用户进行二次开发的一种程序接口方式,它是基于面向对象的思想设计和实现,可以提高软件的重用性和可扩展性,最大程度上延长了软件的生命周期^[4]。随着插件技术的推广,目前插件实现主要有两种方案:

(1) 动态链接库(Dynamic Link Library, DLL): 动态链接库是一种具有一定功能的可执行软件模块,虽然它本身不能独立运行,但是它可以输出函数或类,其它能独立的程序(宿主程序)可以调用它的内部功能。只需用户有在 Windows 下编写一般 DLL 的经历,再花少量的时间熟练有关插件的调用规则及编程规则,就可进行插件设计与开发。

(2) 组件对象模型(Component Object Model, COM): COM 是一种以组件为发布单元的对象模型,这种模型使各软件组件可以用一种统一的方式进行交互。COM 既提供了组件之间进行交互的规范,也提供了实现交互的环境,由于同类插件一般有统一的调用接口,因此也要使用 COM 技术作为开发插件的基础^[5]。应用 COM 技术编写插件时,涉及大量的 COM 技术及原理,导致开发者难以迅速地应用该项技术编写插件来解决实际问题。

3 漏洞分类扫描插件的构建

采用插件技术设计漏洞分类扫描插件,以方便扩展与维护。按照本文所提出的漏洞类别,选择相应的插件,并根据当前的安全需求,对系统进行针对性的扫描,对相关漏洞进行详细分析,针对存在的安全问题制定方案,可以让管理员尽快地了解信息系统所处的安全环境,避免安全问题的产生。

3.1 插件分类

在进行漏洞扫描时,根据用户选择获取用户对扫描的定制信息。然后收集扫描的必要信息并进行扫描的初始化工作,加载对某一类别漏洞的所有插件进行针对性扫描。完成扫描任务并对扫描中间结果进行综合分析,对主机安全状况进行评估。将扫描结果以各种直观的形式反映出来呈现给用户,并提供相应的解决方案,以辅助管理者做出决策。将漏洞扫描插件分类,用来存放漏洞扫描插件,扫描漏洞时

按安全需求与系统环境进行分类扫描。插件分类如图 1 所示。

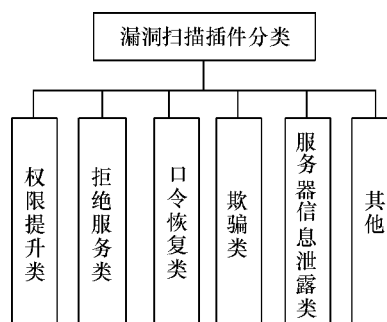


图 1 漏洞扫描插件分类

权限提升类插件扫描提升权限的漏洞,该类漏洞的威胁主要包括:获取普通用户访问权限、读取受限文件、远程非授权文件存取、获取远程管理员权限和获取本地管理员权限等。拒绝服务类插件扫描远程拒绝服务漏洞与本地拒绝服务漏洞。攻击者利用远程拒绝服务类漏洞,无须登录即可对系统发起拒绝服务攻击,使系统或相关的应用程序崩溃或失去响应能力。对于本地拒绝服务类漏洞,在攻击者登录到系统后,利用这类漏洞,可以使系统本身或应用程序崩溃。口令恢复类插件扫描因用户采用弱口令加密方式,使攻击者可以很容易地分析出口令的加密方法,并在得到密码后还原出明文的漏洞。欺骗类插件针对攻击者利用对目标系统实施某种形式的欺骗,从而获得一些敏感信息的漏洞,这种漏洞通常是由于系统的实现上存在某些缺陷。服务器信息泄露类插件针对攻击者利用系统程序存在的缺陷收集对攻击系统有用的信息的漏洞。其它类包括无法归类到以上几类类型扫描插件,例如刚刚出现的一些新的安全漏洞的扫描插件。

3.2 插件模型

在动态链接库中实现插件接口,在宿主程序中运用显示链接方式动态加载插件。整个插件模型的结构如图 2 所示。宿主程序实现的插件管理主要有:注册、反注册插件;启用、禁用插件;显示插件基本信息;配置插件参数等。插件的调用由宿主程序中的插件调用函数统一调度实现,这里必须注意的是,插件一定要遵循接口规范来实现,具体功能由插件方来实现。漏洞分类扫描插件以本文提出的漏洞分类为依据,按功能分为权限提升、拒绝服务、口令恢复、欺骗、服务器信息泄露、其他等 6 种插件类。插件的分类是依据插件实现的漏洞扫描功能,但对插件的初始化及插件的调用等操作,则基本是一致的。

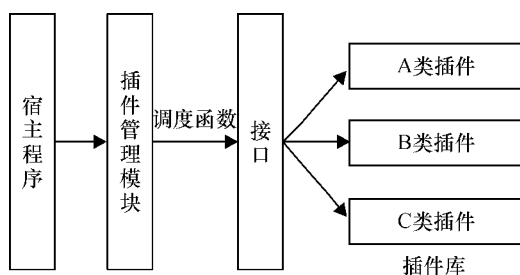


图2 插件模型

宿主程序根据扫描需求及插件的配置信息加载插件模块,然后获得插件的输出函数或者输出类的指针,并保存下来. 插件可以向宿主程序增加界面接口元素,如菜单、工具条按钮等. 当点击宿主程序中与插件相关联的接口元素(如菜单等)时,将会触发插件调度函数,在插件调度函数中使用宿主程序中所保存的插件信息调用插件中实现的功能. 在调用插件输出函数时也可以把宿主程序中实现的接口传递给插件方.

3.3 插件设计与实现

插件由两部分组成:初始化部分和执行部分. 初始化部分负责初始化插件的基本信息,例如:插件所属类别,插件编号,插件名,漏洞解决方案. 执行部分是对某一漏洞进行检测的具体代码. 插件被编译成动态链接库文件,由扫描宿主程序动态加载.

漏洞扫描是一种主动探测,需要通过目标主机进行连接,然后向对方主机发送包含漏洞探测码的数据包,然后等待接收对方的回应,才能检测系统的漏洞. 以主机所回应的数据包作为原始资料,依据漏洞特征来匹配,如果匹配了则表示目标系统存在该漏洞,否则就不存在. 本文的实例是在 Visual C++ 6.0 开发平台下实现的,其它的编程语言在具体实现上会有所不同,但是实现原理是相同的. 在宿主程序方,设计了一个结构体用于保存插件模块、插件对象的信息,并定义了接口函数.

(1) 插件数据结构

插件是一个对象,在宿主程序里将它的属性封装在一个数据结构之中,用于描述插件的基本信息和状态信息,根据漏洞类别定义如下:

```

typedef struct PLUGININFO{
    char * pClassName;          // 插件所属类别
    int pid;                    // 插件 ID
    char cve_id [50];          // CVE_ID 编号
    char * pName;              // 插件名称
    char Version [50];         // 插件版本

```

```

    char pParamsRequest [50];  // 插件所需参数
    char pDescription [500];   // 插件及漏洞描述
    char pAdvice [500];        // 漏洞解决方案
    int Single;                 // 是否为单一漏洞
    char pRisk [50];           // 漏洞风险级别
    int TimeOut;               // 超时限制,“0”表示不限时
} PLUGININFO

```

(2) 插件实现

插件通过一个特定的接口与宿主程序交互,接口是插件开发的重要类,任何需要被调用的插件都必须继承该类. 宿主程序与插件的信息流可能是双向的. 在插件方实现接口,在宿主程序中运用显式链接方式动态加载插件^[8]. 主要的接口函数简介如下:

// 初始化插件

```
BOOL InitializePlugin( struct Plugininfo * p );
```

// 加载插件

```
BOOL LoadPlugin( struct Plugininfo * p );
```

// 释放插件

```
BOOL DisablePlugin( struct Plugininfo * p );
```

// 执行插件扫描

```
BOOL RunPlugin( struct Plugininfo * p );
```

插件在扫描漏洞时,需要先确认远程主机的基本信息,在与主机建立连接的基础上,将根据漏洞的特征构造的数据包发送给主机,分析返回信息,判断是否需要继续构造数据发送,若可以判定漏洞的存在,则报告漏洞,并给出解决方案;否则继续发送数据,直到可以确认漏洞的存在或超时.

下面以具体实例来说明漏洞插件的实现,该插件是用来确定远程主机是否能够匿名登录 FTP 服务器,属于权限提升类的插件. 插件的部分功能代码如下:

// 插件基本信息部分

// 确定 21 端口是否打开,不打开则停止本次扫描

```

if( ! get_port_state( 21 ) exit( 0 );
con = open_tcp( 21 );
if( con )
{
    ftp_login( );
    // 进行匿名登录的测试,如果成功则返回结果,否则继续尝试.
}
Close( con );

```

Report(); // 报告漏洞

4 实验结果与分析

按本文的设计思路,进行了扫描系统的开发,并对原型系统进行了实验. 运行扫描系统后,添加新的扫描任务,插件设置模块会通过接口搜寻系统中所有的扫描插件,用户在选定要添加的插件后,主程序会对选中插件进行初始化,而对于没有被选定的插件将不会加载,这样避免了不必要的开销. 选取 SMTP、FTP、SQL、Telnet 等扫描插件,端口设定为 1-200、1080、1433、7000-7010、8080 等. 对 IP 地址为 192. 168. 1. 58 的目标进行漏洞扫描测试,它的操作系统是 Windows2000,共扫描出了 4 个漏洞,分别为 FTP 漏洞、服务器信息泄露漏洞、POP3 服务器漏洞、SMTP 漏洞. 测试结果表明主程序可以成功加载并执行所选中的扫描插件,并能通过预定义接口和宿主程序进行正确的通信,将扫描结果返回给主程序. 实验说明了本扫描系统的有效性.

另外,本文将原型系统与传统的 Nessus 扫描工具进行了对比实验. 目标主机仅开放了 21、23、25、80、135、139 等端口,在采用相同配置的情况下,通过比较扫描时间来比较扫描效率. 考虑到扫描速度不仅与扫描系统有关,还与网速有关,故做了 5 次扫描实验. 经实验,Nessus 扫描工具的平均扫描时间为 5 分 31 秒,而本原型系统的平均扫描时间为 5 分 23 秒,且所发现的漏洞相同. 因此从效率上看,本扫描系统因为针对性地调用漏洞扫描插件,仅执行需要的插件,在一定程度上提高了扫描的效率. 而且,当

目标主机开放端口较少时,扫描系统根据端口开放信息调用相关漏洞扫描插件,大大减少了调用许多无关扫描插件的开销,且能保证扫描的正确性,这也充分说明了扫描系统的效率.

5 结束语

安全漏洞层出不穷,本文将插件技术应用于漏洞分类扫描当中,不仅提高了软件的重用性,而且为用户及时地更新插件提供了便利,为有效阻止入侵事件争取了时间. 在漏洞分类基础上,用户可以更加清晰、直观、规范地开发插件,进行快速的、有针对性的漏洞扫描. 本文还介绍了插件的构建过程和一个弱口令插件的实例,为安全管理员及时开发出针对最新漏洞的插件提供强有力的支持,具有良好的理论和实用价值.

参考文献:

- [1] 孙钢锋,韩权印. 网络隐患扫描器的设计与实现 [J]. 计算机工程,2006,32(7): 253-255.
- [2] 葛先军,李志勇,何友. 漏洞信息数据挖掘系统设计 [J]. 计算机工程与设计,2009,30(4): 883-886.
- [3] 杨森,张新有. 网络安全漏洞扫描器设计与实现 [J]. 网络安全技术与应用,2008(11): 24-26.
- [4] 高慧萍,吕俊. 插件式开发技术研究与实现 [J]. 计算机工程与设计,2009,30(16): 3805-3807.
- [5] 彭永康,章义来. 插件及其接口的研究与应用 [J]. 计算机应用,2003,23(6): 122-123.
- [6] 肖辉,张玉清. Nessus 插件开发及实例 [J]. 计算机工程,2007,33(2): 241-243.

The Application of Plugin in Vulnerability Category Scanning

Xie Hui-qiong, Lin Jie

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: In the field of information security, Network security vulnerability scanning is essential to the assessment of the security of the system. The Network security vulnerability is reclassified according to the relevance of the vulnerability threat. It analyses the main technology of vulnerability scanning, advances the method to develop plugin, and proposes the way to classify vulnerability plugin. The plugin structural model and designing process are given, which can achieve the targeted vulnerability scanning, making it convenient to update quickly the new scanning plugin.

Key words: information security; threat; vulnerability scanning; plugin