基于 Fuzzing 的 Web 控件漏洞检测改进模型*

周美秀[1],俞 洁[1],姚国祥[2]
(1.暨南大学 信息科学技术学院,广东 广州 510632;
2.暨南大学 网络中心,广东 广州 510632)

摘 要:Web 软件安全漏洞层出不穷,攻击手段日益变化,为分析现有的 Web 控件漏洞检测方法,提出基于 Fuzzing 测试方法的 Web 控件漏洞检测改进模型。该系统从功能上分为五大模块进行设计和实现,并结合静态分析与动态分析技术检测 Web ActiveX 控件模型的漏洞,给出"启发式规则"来优化测试数据生成引擎。实例测试结果表明,Web 控件漏洞的 Fuzzing 测试模型是有效和可行的,并能妥善处理好交互性问题。

关键词:Fuzzing 测试;Web 控件;漏洞检测;漏洞分析
中图分类号:TP393 **文献标识码**:A **文章编号**:1674-7720(2012)04-0085-04

Improved Web controls vulnerability detection model based on Fuzzing

Zhou Meixiu[1], Yu jie[1], Yao Guoxiang[2]
(1.College of Computer Science and Technology, Jinan University, Guangzhou 510632, China;
2.Network Center, Jinan University, Guangzhou 510632, China)

Abstract: The number of security vulnerabilities in the ActiveX controls has increased a lot in resent years. Once illegally used, these vulnerabilities could lead to serious consequences, in order to improve the existing vulnerabilities detection method, we present a new Fuzzing-based Web controls vulnerability detection model. The system is divided into five modules for design and implementation. We combines static analysis and dynamic analysis techniques to detect Web ActiveX control vulnerability, gives "heuristic rules" to optimize the test data generation engine. Example test results show that our Fuzzing-based Web control vulnerability testing model is effective, feasible, and can properly deal with the interaction problem.

Key words: Fuzzing test; Web controls; vulnerabilities detection; vulnerability analysis

"漏洞"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安全方面的缺陷。漏洞一旦被发现,攻击者就可使用这个漏洞获得计算机系统的额外权限,使在未授权的情况下访问或破坏系统[1]。Web 控件漏洞数量在近几年呈现迅速增长的趋势,根据著名的软件安全公司赛门铁克关于 Web 控件安全漏洞的统计:2007 年至 2009 年的检测 Web 漏洞呈几何数量级增长[2]。针对日益增多的 ActiveX 控件漏洞,一方面操作系统安全工程师被动地研究新的防御机制,另一方面许多软件安全研究人员主动地探究 ActiveX 控件漏洞的发掘技术。目前常用的漏洞挖掘技术主要有 Fuzzing 技术、静态分析技术、动态调试技术、补丁比较技术等[3]。

1 Fuzzing 漏洞检测模型

Fuzzing 漏洞测试方法的选择依赖不同的因素,如目标程序、需要测试的数据所采用的格式、研究者的技能等,但其步骤相对一致。故此,可以抽象其模型,图 1 是 Fuzzing 漏洞检测模型示意图[4]。

首要步骤是构造有可能触发漏洞的畸形测试数据。此模块完成测试目标各种信息收集的功能,收集的这些信息为接下来的进一步测试服务,对 Fuzzing 测试能否检测到漏洞起决定性的作用。新的测试数据可以用预先设定的值,也可以通过改变已有的测试数据来动态生成。

* 基金项目:国家自然科学基金项目(61070164);广东省省部产学研基金项目(2008B090500201);
广东省教育厅广东高校科技成果转化重大项目(cgzhzd0807);广东省科技计划项目(2009B010800023)

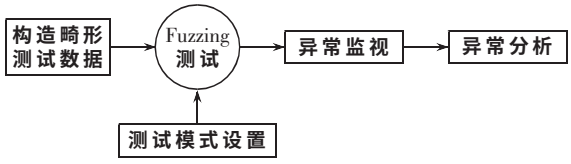

图 1 Fuzzing 漏洞检测模型示意图

Fuzzing 漏洞检测模型的核心步骤是 Fuzzing 测试。一般都可设置测试模式，即指定该次测试用的是组合字段测试、等价类测试、边界值测试等模式。也可以在后续异常分析时进行数据构造方式以及测试模式的调整，从而使测试过程更加高效。

异常监视也是 Fuzzing 测试中较为重要的步骤。异常监视可以使我们确切知道测试数据包或者文档在送往 Fuzzing 系统测试的过程中，哪些数据包和文档触发了系统异常。再者，异常监视可以记录下异常情况下的一些重要信息，如寄存器状态、CPU 状态以及堆栈状态等，以供下一步的异常分析做参考。

异常分析的任务是确定该次异常是否有可利用性以及是否是漏洞。因为触发系统异常的原因繁多，不只是软件漏洞，软件的 bug 也可能导致系统崩溃。异常分析几乎是现阶段每一套 Fuzzing 系统的弱点，因为其涉及的人工分析工作太多，所以这一环节很难引入自动化机制。

2 传统的 Web 控件漏洞检测模型

传统模型主要由测试前准备、测试阶段、测试后报告三个阶段组成。测试前准备阶段包括分析 Web 控件属性、分析函数及其参数、生成测试数据等；测试阶段主要包括模拟用户打开测试实例、动态异常监测和异常记录。图 2 所示为传统的 ActiveX 控件漏洞检测模型。

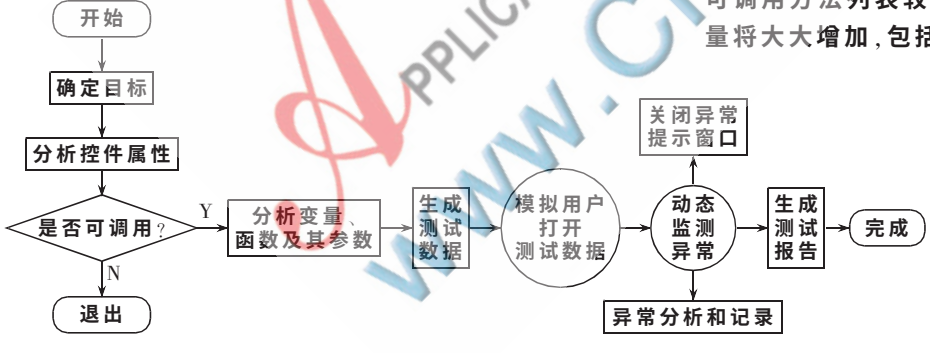

图 2 传统的 ActiveX 控件漏洞检测模型

2.1 测试前准备阶段

首先，确定目标 Web ActiveX 控件，即找出 Web 控件的 CLSID。在注册表相应的表项信息中根据控件的 CLSID，确认其是否实现 IObjectSafety 安全接口、是否是脚本安全、是否被设置了 killbit 位，只有在确认控件实现了 IObjectSafety 安全接口、脚本安全、没有被设置 KillBit 位的情况下，才继续后续的步骤。

最后，还要分析出 ActivieX 控件的属性列表、函数列表以及函数参数列表，只有明确这些列表之后，才能根据不同的漏洞有针对性地构造 Fuzzing 测试数据。这也是 Web ActiveX 控件 Fuzzing 漏洞测试的特殊性之一。

2.2 测试阶段

在测试阶段，将每个测试实例送往目标程序进行测试。在 Web 控件的漏洞检测过程中，测试实例往往以 htm、html、wsf、PDF 等文档格式存在，将所有的测试文档打开，使其可以用程序模拟用户手工点击实现。

打开测试文档后，IE 弹出对话框提示是否加载相应的 Web 控件。Web 控件加载、Web 控件初始化、测试实例调用控件方法等过程，都可能出现异常，可以利用 Windows 提供的调试接口和 Windows 的结构化异常处理机制 SEH，将 Fuzzing 工具作为调试器附加在 IE 浏览器之上，这样便可以接收、处理、记录 IE 的各种调试、异常事件。当异常出现时，Windows 系统总会弹出错误提示窗口，一个较为完善的 Fuzzing 测试工具，必须能够模拟用户点击关闭提示窗口，该功能可以用 Windows 提供的 HOOK 技术实现，在以后的研究中将更为详细地分析和解决这个问题。

2.3 测试报告阶段

在捕获并保存服务器进程内部的异常之后，原型系统将对异常信息进行自动分析，从而智能地给出异常分析报告。该模型系统所实现的异常自动分析算法能够有效地分析出导致异常命令、异常类型以及异常风险程度等。根据该算法生成的异常分析报告在一定程度上能减少漏洞分析人员的后期分析工作，缩短漏洞分析时间，从而提高漏洞发掘的效率。

2.4 传统 Web 控件漏洞检测模型的不足

传统模型存在以下几方面不足：(1)若 ActiveX 控件的可调用方法列表较长，这对于 Fuzzing 漏洞测试的工作量将大大增加，包括测试样本数据的数量增加和测试过程的时间增加。本文应用代码扫描分析技术解决这个问题，有效地减少了测试的工作量。(2)生成测试数据的随机性。对此，本文采用启发式生成测试数据的方式来解决，提高了漏洞发掘效率和自动化程度。(3)测试方法较为单一，动态分析可在代码执行过程中查看代码，善于发现运行时错误；而静态分析通过算法检查代码的错误。传统的模型比较着重静态分析，很难发现复杂交互中产生的缺陷，本文采用静态分析和动态分析结合的方式。

3 Web 控件漏洞检测模型的改进

针对 Web 控件漏洞检测模型的不足，本文采用一种融合代码扫描和 Fuzzing 测试的改进模型，采用静态分析和动态分析相结合的方法，如图 3 所示。改进后的模型增加了代码扫描分析模块、启发式生成测试数据模

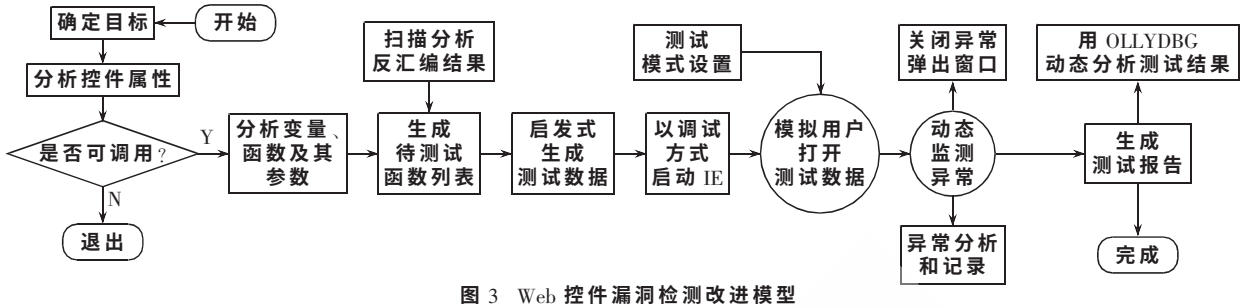

图 3 Web 控件漏洞检测改进模型

块、OllyDbg 分析测试模块等模块。

3.1 代码扫描分析模块

本模块的目的是减小测试函数列表。模块搜索函数地址空间中的“目标字”,为测试数据的构造提供支持。另外,用 IDA 等反汇编工具对目标程序的二进制文件进行反汇编,若文件扫描到运用了不安全方法,则将此控件函数标记为待测试函数。

3.2 启发式生成测试数据模块

借鉴人工智能中的遗传变异和启发式算法来生成测试数据,使测试数据更具针对性,从而提高漏洞发掘效率和自动化程度。

3.3 OLLYDBG 分析测试结果模块

利用 Fuzzing 进行漏洞发掘时,漏洞发掘人员往往需要第三方调试器配合才能有效地发掘软件漏洞。本模型系统采用 OLLYDBG 对服务器进程进行全程监控。OLLYDBG 是一个新的动态追踪工具,它将 IDA 与 SoftICE 相结合,成为目前最强大的调试工具。

4 测试结果及分析

为了验证本文提出的 Web 控件漏洞 Fuzzing 测试模型的可行性和有效性,选择用户量较大的“暴风影音”播放器进行测试(主要针对其 mps.dll 控件进行测试)。测试环境如下:Windows 系统的 Internet Explore:V7.0,测试对象:暴风影音(Stormplayer)V3.11。

4.1 枚举 ActiveX 控件属性、方法及方法参数

安装“暴风影音 V3.11”,然后用改进的 Fuzzing 测试模型针对暴风影音的 mps.dll 控件进行测试,枚举 ActiveX 控件的属性、方法和方法参数。

4.2 构造测试文档

本文的 Fuzzing 测试模型,在枚举出来的属性和方法的参数列表中,右击任意一个属性或方法,便可生成对应的测试数据。模型以 VB 脚本文档形式生成测试文档,便于 wscript.exe 的直接调用。下面代码是暴风影音 mps.dll 控件的导出函数 OnBeforeVideoDownload()的 Fuzzing 测试文档之一。

```
<?XML version=´1.0´ standalone=´yes´ ?>
<package ><job id =´DoneInVBS´ debug =´false´ error =´true´>
<object classid =´clsid:6BE52E1D -E586 -474F -A6E2 -1A85A9B4D9FB´ id=´target´ />
<script language=´vbscript´>
´Wscript.echo typename(target)
targetFile = "C:\Program Files\StormPlayer \mps.dll"
prototype = "Sub OnBeforeVideoDownload ( ByVal URL As String )"
memberName = "OnBeforeVideoDownload"
progid="MPSLib.StormPlayer"
argCount=1
arg1=String(8212, "A")
target.OnBeforeVideoDownload arg1
</script></job></package>
```

4.3 漏洞测试及结果分析

构造了一系列的测试文档之后,下面要进行漏洞测试。当 ActiveX 控件的属性或方法较多时,耗时较长。于是 Fuzzing 测试自动化的好处便得到体现。因为能够处理好交互性问题,所以漏洞测试过程便可做到无人值守[5]。本文的 Fuzzing 测试模型便能妥善处理好交互性问题。

4.3.1 异常列表

对 Web ActiveX 控件的某一属性或者方法进行 Fuzzing 测试,用列表的方式呈现其结果,从该表中便可以方便地获知导致异常发生的具体测试文档、异常发生的数目等信息。图 4 是暴风影音 mps.dll 控件的导出函数 OnBeforeVideoDownload()Fuzzing 测试的异常列表。

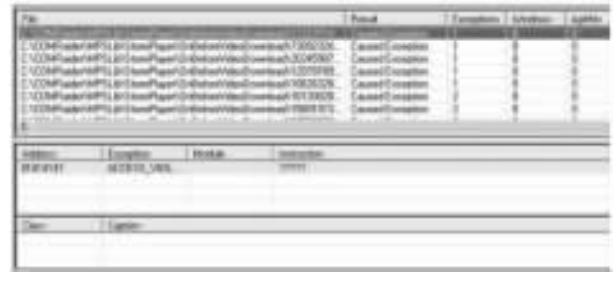

图 4 Web 控件漏洞测试异常列表

4.3.2 异常详细报告

对异常列表中的每一项可以查看其详细报告。利用论文提出的 Fuzzing 测试模型对暴风影音 mps.dll 控件的导出函数 OnBeforeVideoDownload()进行 Fuzzing 测试得到测试结果异常报告。图 5 是其中的一项详细报告。从中可以方便地获知异常发生的情况下的各种重要信息,包括异常类型、当前线程状态、当前 SEH 链表的状态、当

图 5 异常详细报告

前堆栈内存状态、当前各重要寄存器内容以及当前 EIP 附近的指令等。

从这份详细报告中可以看到 EIP 寄存器的值为“41414141”，这是构造测试文档时所采用的经典长字符串“AAAA”的十六进制码。充分说明了若攻击者加入设计的输入参数，EIP 可以被劫持，从而可以引导系统去执行攻击者设计的特殊代码。因此可以确定 mps.dll 控件的导出函数 OnBeforeVideoDownload()确实存在着漏洞。有经验的测试者还可以根据异常报告中崩溃地址、崩溃时的栈空间状态和崩溃时的栈空间状态指令状态等信息，进一步分析出 mps.dll 控件漏洞出现的具体位置。

由于 Web ActiveX 控件具有特殊性、变化快等特点，Web 应用的安全检测变得十分困难，而 Web 安全防范日益重要。本文针对 Web 的安全漏洞检测的模型和算法做了一些探讨，所提出的方案在解决当前 Web ActiveX 控件漏洞检测问题上有所突破。未来，将进一步考虑增强测试数据的针对性，从而提高测试数据触发漏洞的比率；搜索 ActiveX 各导出函数地址空间中的“特征字”，搜索调用非安全方法的 ActiveX 导出函数的具体实现方法[6]。

参考文献

[1] 王雨晨.系统漏洞原理与常见攻击方法[J].计算机工程与应用，2001，38(8)：62-64.

[2] SUTTON M，GREENE A，AMINI P.Fuzzing：brute vulnerability discovery(1st ed)[M].Pearson Education，2007.

[3] 邵林，张小松，苏恩标.一种基于 Fuzzing 技术的漏洞发掘新思路[J].计算机应用研究，2009，26(3)：1086-1088.

[4] 吴毓书，周安民，吴少华.基于 Fuzzing 的 ActiveX 控件漏洞发掘技术[J].计算机应用，2008，28(9)：2252-2254.

[5] 张美超，曾凡平，黄奕.基于漏洞库的 fuzzing 测试技术[J].小型微型计算机系统，2011，4(4)：651-655.

[6] 高峻，徐志犬，李健.漏洞自动挖掘技术研究进展[J].计算机与数字工程，2009，37(1)：100-105.

(收稿日期：2011-11-18)

作者简介：

周美秀，女，1988 年生，硕士，主要研究方向：网络安全、密钥学。

俞洁，女，1986 年生，硕士，主要研究方向：密码学、网络安全。

姚国祥，男，1963 年生，博导，主要研究方向：网络安全、数据加密、信息系统安全。