基于符合执行和模糊测试的混合模糊测试技术综述

**摘要：**模糊测试现在已经发展成为一种高效的漏洞挖掘方法，符号执行也是当前流行的软件漏洞挖掘技术，两者都是网络与信息安全领域的研究热点。混合模糊测试是在传统模糊测试的基础上加入了符号执行技术，目前已经发展成为模糊测试的新分支。文章对现有的混合模糊测试方法进行研究，综述了混合模糊测试的发展演变过程和技术核心，并通过一种基于符号执行的实验方法对目前知名的混合模糊测试进行性能对比。最后讨论了目前混合模糊测试领域研究存在的问题，并尝试对其未来发展趋势进行了展望。

**关键词：混合模糊测试、模糊测试、符号执行、覆盖率**

**1. 引言**

近年来，漏洞已经成为非常严重的安全问题，据CNVD（国家信息安全漏洞共享平台）报告，2010年至2019年十年间漏洞数量由5,223个暴增为16,016个[1]，已经对人们的日常生活产生了严重影响。加之物联网时代的兴起，各类软件数量的井喷式增长，以及恶意漏洞利用事件的相继爆发，导致随处可见的设备时刻运行着存在漏洞的软件[2]。自动化软件漏洞挖掘技术变得尤为重要，已经成为安全领域比较热门的研究领域，无论是工业界还是学术界都得到了足够的重视。在自动化软件漏洞挖掘领域中，有两项发展迅猛的技术：模糊测试技术和符号执行技术。

模糊测试，通过将大量的测试用例输入到特殊处理的测试程序，获取测试信息以及可能会出现的崩溃信息。模糊测试具有自动化程度高、消耗率低、计算资源利用率高等优点，已经被很多实践证实效果良好[5],[6],[7]，得到了学术界和工业界广泛应用。在漏洞挖掘方面，模糊测试已经发挥了非常惊艳的效果，仅拿AFL（America Fuzzy Lop，以下简称AFL）[5]这一典型基于变异的模糊测试工具举例，从2014年开发至今，已经发现了数百CVE，这还不包括诸多基于AFL的论文所发现的CVE漏洞[8][9][10]，以及未被披露的漏洞。但是高度自动化同时也导致模糊测试做不到对待测试程序的所有可能路径进行测试，并不能覆盖所有的路径，因此面对分支复杂的应用程序只能做到尽快达到路径的深度，做不到全覆盖路径的广度。

符号执行，【对符号执行的介绍】。如果按照符号执行树的形式将待测试程序进行直观展示，就可以很容易的理解，模糊测试只能尽可能到达较深的位置，但是难以应对复杂分支，而符号执行可以突破浅位置的复杂分支，路径全覆盖，但是因为计算量限制等问题并不能到达更深的路径。显然模糊测试和符号执行各有优缺点，混合模糊测试技术就是结合了双方的优点，弥补了各自的不足，从而到达更深的路径，获得更高的覆盖率。混合模糊测试比单独使用符号执行或者模糊测试效果都要更好[3],[4]。

**2. 技术概述**

2.1 漏洞挖掘技术

当前用于漏洞挖掘的技术有很多，最主要的有三种：静态分析、动态分析和符号执行[12]，这三项技术各有优缺点。

静态分析是一种在不执行待测试程序的情况下对其进行分析的软件测试技术，常用的分析技术有词法分析、语法分析、抽象语法树分析、语义分析、控制流分析、数据流分析、污点分析、无效代码分析等，广泛应用于漏洞挖掘。静态分析的优势在于对上万行代码可以进行快速分析，并用形式化的方法给出了可证明的、完整的分析结果【引用】。静态分析的缺点也很明显，分析结果未能给出精准的POC（漏洞触发证明样例），分析的目的是挖掘软件的漏洞，得到的分析结果却不能产生可触发漏洞的输入，比如对于PNG图片读取型待测试程序，即使用静态分析得到了结果，但是不能得到可以触发某漏洞的PNG图片，需要根据分析结果自己构建PNG文件。

动态分析可以执行待测试程序，通常利用插桩方式来获得各类路径，通过路径可以获得覆盖率信息。不同于静态分析，动态分析可以监控待测试程序的运行，当程序出现运行错误时，将触发此错误的输入保存下来，如果同样运用到PNG图片读取型待测试程序，就可以保存得到能触发错误的相应PNG图片文件。动态分析还可以检测静态分析无法检测到的依赖项，比如多态的动态依赖关系。动态测试的缺点主要是来自于插桩技术，插桩技术是在待测试程序中插入特定操作，依此来探测程序达到测试的目的，因此插入的此部分代码（高级语言或编程语言）一定会对原来的待测试程序产生一定的负面性能影响。另外一个缺点是无法得到全部的覆盖率，因为待测试程序的运行是基于用户交互或自动化测试，并不能保证程序所有可能位置的全部覆盖。

符号执行【论文引用】，

虽然漏洞挖掘的主流是以上三类，但现在的研究已经不再单纯使用某一种技术，而是倾向于多种技术结合。模糊测试直观上是属于动态分析，通过插桩技术对待测试程序进行动态监控，但是目前主流的模糊测试工具中也加入了静态分析技术，比如基于模糊测试和污点分析技术的【】。

2.2 模糊测试技术

模糊测试技术最早于1995年由M等人提出，针对UNIX系统进行安全测试并发现组件中诸多漏洞[11]。模糊测试的核心在于【】。从模糊测试的

2.3 符号执行技术

符号执行技术出现在

2.4 混合模糊测试技术（优缺点）

混合模糊测试技术结合了模糊测试技术和符号执行技术。

**3. 混合模糊测试研究现状**

2007年，【】等人首次提出混合测试的概念，将符号执行和自动化软件测试结合起来。【简介论文核心】。虽然当时作者并没有明确提出混合模糊测试，但是其中用到的测试思想跟模糊测试一致，所以此项研究成为后来众多混合模糊测试的概念原型。

在混合测试提出之后，计算机的算力得到不断的提升，符号执行技术在约束求解等等方面得到了一定的突破，由于模糊测试没有标志性的进展，混合模糊测试也一直没有得到突破性的进展。2015年，AFL[5]的出现给工业界和学术界带来了新思路，是模糊测试里程碑式发展，也是混合模糊测试技术的【】，覆盖率为导向的思想给混合模糊测试带来了新的思路。

2016年Driller

2018年QSYM

2019年SAVIOR

2020年PAGOLIN

1. **关键技术探究**

4.1 覆盖率为导向的模糊测试

从1995年第一次提出模糊测试的概念到现如今，模糊测试经过多番改进，主要出现了两类分支：基于变异的模糊测试【引用】和基于生成的模糊测试【引用】。直到AFL的出现，让覆盖率为导向的模糊测试技术（属于基于变异的模糊测试）成为当前最主流的模糊测试技术。

4.2 fork-server 技术

4.3 混合模糊测试技术框架

**5. 性能比对**

1. **模糊测试的发展方向**

6.1 符号执行指导模糊测试过程细分阶段

6.2 基于泛覆盖率为导向的混合模糊测试

**7. 结束语**

[1]. <https://www.cnvd.org.cn/flaw/statistic>

[2]. Shoshitaishvili Y , Wang R , Hauser C , et al. Firmalice - Automatic Detection of Authentication Bypass Vulnerabilities in Binary Firmware[C]// Network & Distributed System Security Symposium. 2015.

[3]. Majumdar, Rupak , and K. Sen . "Hybrid Concolic Testing." International Conference on Software Engineering IEEE, 2007. 第一次提出混合测试

[4]. 谢肖飞, XIE XiaoFei, 李晓红,等. 基于符号执行与模糊测试的混合测试方法[J]. 软件学报, 2019, Vol.30Issue(10):3071-3089.

[5]. afl

[6].honggfuzz

[7].ossfuzz

[8]. collafl

[9]. mopt-afl

[10]. aflgo?aflfast?

[11]. Fuzz Revisited: A Re-examination of the Reliability of UNIX Utilities and Services

[12]. driller

[13]. QSYM

[14]. SAVIOR

[15]. Pangolin