通过模糊测试实现百万行代码的漏洞挖掘

介绍

模糊测试是最强大的漏洞检测技术[1-5]之一，并已被广泛用于检测各种软件安全问题的行业。例如，由于2016年，存在由OSS-起毛在widelyused第三方库[6]发现超过9000漏洞。 Windows 7的WEX安全漏洞中，有三分之一是由微软模糊服务发现数百万美元被保存[7]。心脏出血漏洞，这威胁全球网络空间在2014年，现在可以起毛[8]的10秒内被检测到。由于起毛可以生成具体的输入来触发该漏洞，很容易形成概念验证（PoC）证明开发人员了解和解决这些问题。使用快捷随机突变产生的评估目标程序的输入数量巨大的简单直觉使起毛能够对所有类型的项目部署。

然而，模糊是无效的检测大型程序的漏洞。有在大型程序的大小带来的三大挑战。首先，它难以产生含有不同语义wellformat输入。一些大型项目如MySQL需要输入遵循其独特的语法标准。格式错误的输入导致执行停止在初期阶段不运行该程序的任何核心功能。此外，很难生产具有不同语义输入，用于全面评估的所有功能。从巨大的项目，这些不同的规格使它不可能为了模糊自动生成有效的投入。其次，它是低效率的，以产生满足大而复杂的路径的条件的输入。随着日益增加的程序大小，复杂的路径的条件缩小可行输入空间。这降低了随机突变，从而产生满足路径条件输入的可能性。因此，效率低输入的生成结果在许多无意义的执行。当程序尺寸越大该问题加剧。第三个挑战来自于执行开销。起毛使用仪器来记录执行信息来指导以后的输入生成。根据最近的研究[9-10]，我们注意到，在现有的仪器方法的开销可能超过600％以上。同时，状态国税发技术模糊测试框架运行目标程序数百万次，以增加检测的脆弱性的概率。这种积累的开销使得模糊化低效的大型程序部署。为了解决模糊这些挑战，我们提出我们的技术，以改善起毛的效率，使其能够更好地支持大型程序。

相关工作

强化学习的语义推理

这是很难生成包含漏洞语义输入。关键原因是缺乏对语义输入生成有效的指导。现有的作品多数仍然使用程序覆盖率来判断的输入值。没有明确的标准，基于语义的投入的优先次序。虽然他们尝试学习语义无论是从众多的输入[11]或手动给定规格的[12]，对于大规模方案，精确地描述整个可能的语义与可行的输入的数量有限的挑战。同时，它也很难有一个适合于不同规格的通用模型。

我们建议我们的方法来解决基于强化学习的灵感这个问题。强化学习[13]是一种机器学习技术，它能够利用自己的行动和经验反馈通过试错在一个互动的环境中学习的代理。不像其他的机器学习技术寻求的分类或相似的结果，强化学习算法旨在寻找最大化所述预定奖励的最优解。根据目标的目标，奖励可以是不同的。如果我们考虑满足特定属性的脆弱性，某些类型的程序语义的，那么我们可以设计的学习算法的奖励标准，最终生成满足这种性质的正确输入。例如，如果我们想检测内存泄漏漏洞，那么它是更明智地使用，可能会导致更多的内存使用情况的操作。由于起毛需要一个适应度函数来指导输入生成，这个奖励标准应以提高质量投入新的有效的指导。与现有的作品相比，强化学习与深度学习相结合，更能够处理大规模的投入或不同的规格。

符号分析

符号执行是最有力的方法来检测弱点之一[14-

16]

。它可以精确地产生，导致执行某些程序点投入。例如，单一的条件X×X = 100可能是非常困难的随机突变，以产生可行的值x = 10。的可能性小于10〜20的整数变量x。大型程序加剧，因为道路条件的数量增加了这个问题。约束求解弥补了随机突变在处理复杂的路径条件的弱点。这两种方法的组合[17，4]增强漏洞检测的能力。

然而，效果主要是由约束求解著名性能问题的限制。在国家的最先进的模糊测试框架还需要随机突变，产生的大部分基础上，求解结果的输入。云等人。 [5]使用较少的约束来生成输入同时利用起毛验证输入的有效性。杜特拉等。 [18]提出，以产生多路输入，以增加更多的满足条件的路径的可能性。这些方法集中在用于从解算器直接产生模糊测试更具体的输入的效率。我们提出的方法需要从这些方法的好处，同时利用约束为起毛提供额外的指导。约束本身包含相关的输入变量的信息。简单生成输入不能很好地代表在约束的全部信息。我们的方法分析了制约，提高变异的有效性，从而使其成为大型程序更有效。

成分起毛

直接分析整个程序时，程序的大小是非常大的不实用。此外，易受片段仅表示该程序的一小部分。这是没有必要的程序全面分析的全过程。因此，模糊应采用成分分析变得更具可扩展性。状态的最先进的组成分析主要集中在符号执行[19]和单元测试[14]。我们充分利用现有的方法发现本地的漏洞。我们结合语义推理的方法来产生输入满足当地语法同时使用约束求解提供更可行的投入。该成分模糊设置为整个程序的模糊化，以提高效率的潜在目标。

符号分析可以帮助整个程序的模糊来验证组成模糊设定的目标。目前已在编程语言社区[20-25]上前提推论显著的作品。主要的区别在于，他们瞄准的声音验证，而我们的目标就加快模糊测试。他们推断程序安全性的充分前提，并以展示所有程序路径举办条件证明的正确性。在我们的方法，我们的目的不是来推断象征性的条件为所有可能的程序执行。相反，我们推断目标程序点多的必要条件。如果相关的变量违反在早期阶段的必要条件，没有必要继续执行，使他们无法对目标程序点去。

提议的技术

在本节中，我们描述了我们模型的高层次的意见，以及我们提出的方法的工作流程。

成分起毛

给定图1和图2显示了我们所提出的组成框架的基本模型。我们利用成分起毛提供整个程序的验证目标。

另外两个引擎，语义推理和符号分析，使输入与设计语义和值。现在我们讲解每个引擎背后的一些细节。

图1第一阶段：成分起毛

图2第二阶段：整个程序验证

语义推理引擎

图3显示了语义推理引擎的工作流程。我们采用的是强化学习，以产生含有所需的语义投入。为了使它更实用，我们通过学习由语法定义的符号和操作带来了不同的影响开始。了解这些语义是更合理地生成满足由漏洞描述中提供的属性的输入。还有就是模糊与强化学习相结合的另一个好处。执行的众多时间提供精确的预言丰富的数据用于学习理解语义。学习模型保持起毛，其提供了可以触发漏洞的更有效的输入时更新。有迹象表明，可以收集甲骨文没有太多的努力，如执行时间和内存状态的许多数据。状态的最先进的模糊测试框架[26-27]使用这些状态来指导输入生成。然而，如何正确地混合不同的标准，影响输入生成。我们的强化学习方法是由目标的漏洞，因此能够发现，最大限度地为所有标准的结果输入驱动。因此，这应该是解决语义问题的正确方向。

对于起毛指导图4符号分析

执行（图5）。符号分析需要组成评价的概念和分裂朝向目标程序点分成多个小单元的路径。我们为每一个单元的必要条件，并把它们插入到程序。如果相关的变量违反了这些条件，执行直接停止，并与其他投入开始。因此，起毛可以评估更可能的输入，并找到潜在的

符号分析引擎

正如在第2节前面提到的，象征性的分析引擎有两个主要功能。首先，符号分析提供了用于起毛输入生成（图4）的额外指导。我们注意到，约束包含很多有用的信息，比如相关的变量，可行的数值范围。只有产生可行的输入不能提供此信息起毛。为了提供从约束求解更精确和有益的指导，我们利用求解器来分析约束提取额外的信息，并将其发送到模糊。与精确的导向，这两个组合物和整个程序起毛可以成为在生成输入更有效。

其次，符号分析动态仪器断言停止不必要

评估计划

所有的评价都有两个指标：代码覆盖率和相同的时间内发现了几个安全漏洞。没有为每个方法的一些其他补充评价标准。

比较与其他基于语法的模糊器。为了评估语义推理的有效性，我们进行与其它基于语法的模糊[11-12]的项目如MySQL，FFmpeg的，Libav实验。所有这些项目都需要满足一定的语法输入。我们还评估由不同的奖励标准所带来的效益。此外，我们需要优化的参数，使模型实用于所有类型的项目。

比较基于约束的模糊器。在这个实验中，我们研究FuzzFuzzFuzz对标准的熔岩-M的数据集和真实世界的项目，如在Binutils，OpenSSL的代码覆盖和错误的狩猎能力。我们比较与不分析语义程序诸如AFL [1]，AFLFast [28]，QSYM所述状态的最先进的模糊器我们的方法[5]，钻床[4]。

我们证明了我们对熔岩-M [29]数据集的方法，如表1所示的效果左边的数字显示由模糊发现的缺陷数量。另外一个是错误的总数。

进展与展望

目前，我们已经实现了一个原型，象征性的分析相结合。在第3节，我们展示了与熔岩-M基准的国家的最先进的模糊比较的评估结果。结果证明文中提出的方法的有效性。我们要提高原型和评估实际项目的结果。

随后，我们将通过动态仪表符号分析结果

上熔岩-M表1实验结果

这个评估结果表明，我们的优化方法优于其他模糊测试框架。调查甚至发现，在基准以前的一些看不见的错误。对于节目“谁”，FuzzFuzzFuzz显著优于所有其他的fuzzer。因为“谁”是比其他3个项目要大得多，这也证明了我们方法的有效性。

成分模糊的效果。为了评价新框架的分析中，我们直接比较与其他框架我们的方法。我们用同样量的不同实际项目时间进行实验。算作真正正的所有漏洞，必须由开发人员进行确认。

案例分析。我们选择通过我们的新的模糊器发现证明我们的工作有效性的漏洞情况。如果我们的方法不能胜过其他框架来讨论我们的方法的缺点我们也分析程序。

技术如Dyninst [30]。我们的原型首先应用成分分析收集潜在的易受攻击点的信息。这些项目点设置为以后验证的目标。象征性分析优化仪器的方式，并产生对这些点的电压输入。这项工作的目的是有效引导起毛执行对目标程序分和减少因执行无关的开销。这些优点使模糊的大规模方案更有效。

最后，我们将采用不同的语法，如SQL，HTTP打造grammarbased模糊测试框架的强化学习算法。它还将与构图象征性分析相结合来调节输入特定值要求的小部分。所有的方法来实现，我们最后的框架应该是能够检测各种漏洞在万行代码的项目。

资金

本文是由国家密钥R（E S）é资助的R C h的Ñd发展项目（2016YFF0204002）

参考

2014年AFL：美国模糊LOP。 http://lcamt​​uf.coredump.cx/afl/。 （2014）。访问时间：2014年。

P.陈和H.陈。 2018年安哥拉：由原则性搜索高效起毛。 2018年IEEE研讨会安全和隐私（SP），卷。 00. 711-725。 https://doi.org/10.1109/

SP.2018.00046

W.约翰森，M. Svensson的，U. E.拉尔森，M.阿尔姆格伦和V.

Gulisano。 2014 T-起毛：基于模型的起毛健壮性

电信协议的测试。 2014年IEEE第七

国际会议软件测试，验证和确认。 323-332。 https://doi.org/10.1109/ICST.2014.45

尼克·斯蒂芬斯，约翰Grosen，克里斯托弗Salls，安德鲁·达彻，

若欲王，雅格布科尔贝塔，严Shoshitaishvili，克里斯托弗

Kruegel，和Giovanni豇豆。 2016年钻床：增广起毛通过选择性符号执行。在NDSS，卷。 16. 1-16。

INSU芸，李桑戈，孟旭Yeongjin张，并Taesoo

金。 2018年QSYM：实用Concolic执行引擎

专为混合起毛。在第27届USENIX安全

研讨会（USENIX安全18）。 USENIX协会，巴尔的摩，MD，745-761。 https://www.usenix.org/conference/ usenixsecurity18 /展示/云

2016年OSS-FUZZ报告。 HTTPS：//security.googleblog。

COM /十一分之二千零十八/ A-新章换OSS-fuzz.html。 （2016）。访问：

2016年

帕特里斯戈德弗鲁瓦，迈克尔·莱文Y.，大卫莫尔纳。

2012年SAGE：白箱起毛的安全测试。

队列10，如图1所示，第20条（2012年1月），8页。 https：//开头DOI。

组织/ 10.1145 / 2090147.2094081

2015年libfuzzer。 https://llvm.org/docs/LibFuzzer.html。 （2015年）。

访问：2015年。

斯特凡·纳吉和马修·希克斯。 2019年全速起毛：

通过覆盖制导跟踪减少起毛开销。在IEEE研讨会安全和隐私（奥克兰）。

徐雯，Sanidhya卡什亚普，Changwoo敏和Taesoo

金。 2017年设计新的工作基元提高

起毛性能。在计算机和通信安全2017年ACMSIGSAC会议论文集（CCS

'17）。 ACM，纽约，NY，USA，2313年至2328年。 https：//开头DOI。组织/ 10.1145 / 3133956.3134046

J.王，陈B.，L炜和刘Y.。 2017年天火：数据驱动的种子生成的模糊化。在2017年IEEE研讨会安全和隐私（SP）。 579-594。 https://doi.org/10.1109/

SP.2017.23

魏友，培元宗，启辰，王晓峰，小京辽，潘卞和梁彬。 2017年SemFuzz：Semanticsbased证明的概念漏洞的自动生成。在计算机和通信安全的ACM 2017年SIGSAC会议论文集。 ACM，2139年至2154年。

沃洛Mnih，科瑞Kavukcuoglu，银鸿，安德烈

A.鲁苏，乔尔俯伏，马克·G.·贝勒马尔，亚历克斯·格雷夫斯，马丁

Riedmiller，安德烈亚斯K. Fidjeland，乔治Ostrovski，斯蒂格·彼得森，

查尔斯·比蒂，阿米尔·萨迪克，扬Antonoglou，海伦·金，

Dharshan库马兰，大安Wierstra，巴蒂尔莱格和德米斯·哈萨维斯。通过深入的强化学习人类2015级控制。自然518，7540（2015年2月），529-533。 HTTP：//dx.doi。

组织/ 10.1038 / nature14236

帕特里斯戈德弗鲁瓦，尼尔斯Klarlund和科希克参议员2005年DART：

检测的自动随机测试。 SIGPLAN没有。 40，6（2005年6月），213-223。 https://doi.org/10.1145/1064978.1065036

帕特里斯戈德弗鲁瓦，迈克尔·莱文Y.，大卫莫尔纳。 2008年自动化白盒测试模糊测试。 https://www.microsoft.com/ EN-US /科研/出版/自动-白牌 - 模糊测试/

健强马，邱攀YIT，杰弗里·福斯特小号，和迈克尔·

希克斯。 2011年导演的符号执行。在国际静态分析研讨会。施普林格，95-111。

Rupak马宗达和科希克参议员2007年混合Concolic测试。在第29届国际会议论文集

软件工程（ICSE '07）。 IEEE计算机学会，华盛顿特区，美国，​​416-426。 https://doi.org/10.1109/ICSE。 2007.41

拉斐尔·杜特拉，凯文Laeufer，乔纳森·巴克拉克和科希克

2018年参议员试验SAT解决方案的有效采样。在软件工程的第40届国际会议论文集（ICSE '18）。 ACM，纽约，NY，USA，549-559。 https://doi.org/10.1145/3180155.3180248

大卫Trabish，安德烈Mattavelli，诺姆Rinetzky和克里斯蒂安Cadar。 2018年切碎符号执行。在软件工程的第40届国际会议论文集

（ICSE '18）。 ACM，纽约，NY，USA，350-360。 https：//开头DOI。组织/ 10.1145 / 3180155.3180251

AWS Albarghouthi，ISIL Dillig，和阿里Gurfinkel。 2016年最大规格合成。在ACM SIGPLAN通告，卷。 51。

ACM，789-801。

萨蒂什钱德拉，斯蒂芬·芬克Ĵ和马努Sridharan说。 2009年。

Snugglebug：一个强大的方法来最弱的前提条件。

ACM SIGPLAN声明44，6（2009），363-374。

帕特里克·库赞特，拉德希亚·库赞特，曼努埃尔Fähndrich，和Francesco Logozzo。必要的先决条件在2013年自动推理。在国际研讨会验证，模型检验与抽象解释。施普林格，128到148。

帕特里克·库赞特，拉德希亚·库赞特和弗朗西斯Logozzo。从断断续续的断言和应用程序在收集合同2011年前提推论。在国际研讨会验证，模型检验与抽象解释。施普林格，150-168。

弗朗西斯Logozzo和托马斯球。 2012模块化和验证程序自动修复。在ACM SIGPLAN通告，卷。 47. ACM，133-146。

弗朗西斯Logozzo，Shuvenduķ拉赫瑞，曼努埃尔Fähndrich，和Sam布莱克希尔。 2014年验证模版本：迈向可用的验证。在ACM SIGPLAN通告，卷。 49. ACM，294-304。

卡罗琳·勒米厄，罗汉Padhye，科希克森，和黎明

歌曲。 2018年PerfFuzz：自动生成病理

输入。在第27届ACM国际SIGSOFT论文集

研讨会上的软件测试和分析（ISSTA

2018）。 ACM，纽约，NY，USA，254-265。 https：//开头DOI。组织/ 10.1145 / 3213846.3213874

Theofilos Petsios，杰森赵，安吉洛D. Keromytis，和

苏曼亚娜。 2017年SlowFuzz：自动化领域无关

算法复杂性漏洞的检测。在计算机和通信安全的ACM 2017年SIGSAC会议论文集（CCS '17）。 ACM，纽约，NY，USA，2155年至2168年。 https://doi.org/10.1145/3133956.3134073

马塞尔·鲍姆（Böhme），范 - 顺范和Abhik Roychoudhury。

2016年Coveragebased Greybox的模糊马尔科夫链。在计算机和通信安全2016年ACM SIGSAC会议论文集（CCS '16）。 ACM，纽约，NY，USA，1032至1043年。 https://doi.org/10.1145/2976749.2978428

B.多兰-Gavitt，P.虎林，E. Kirda，T.韭菜，A. Mambretti，W。

罗伯逊F.乌尔里希和R.惠兰。 2016年LAVA：大型自动化漏洞加成。在2016年IEEE研讨会安全和隐私（SP）。 110-121。 https://doi.org/10.1109/

SP.2016.15

2014年Dyninst。 https://dyninst.org/dyninst。 （2014）。访问：

2014年

作者简介：

张川（1974-），男，中国香港人，香港科技大学副教授。

研究方向：软件安全，代码安全。

电子信箱：charlesz@cse.ust.hk

毛慧（1988-），通信作者，女，湖北天门人，高级工程师研究方向：代码安全。

电子邮箱：576467379@qq.com

王忠（1974-），男，北京人，高级工程师研究方向：软件和信息安全能力验证。

电子信箱：wangzh@cnas.org.cn

靳冬（1982-），男，山西人，研究员研究方向：软件，信息安全检测和认可。

电子信箱：jind@cnas.org.cn

黄河清（1993-），男，中国香港人，香港科技大学在读博士生研究方向：软件安全，信息安全。

电子信箱：hhuangaz@cse.ust.hk

杨尚沅（1994-），男，广东珠海人，工程师研究方向：代码安全。

电子邮箱：1533231845@qq.com

（收稿日期：2019年7月5日）

生成的PoC经由起毛万行代码的程序的

张Charles1，毛辉2，王Zhong3，金董，黄荷清1，杨上元2

（1.网络安全实验室，计算机科学与工程学院，香港科技大学的系，香港999077，中国;

珠海南方软件和网络评估和测试中心，广东珠海519085，中国;

中国实验室国家认可合格评定，北京100062，中国）

摘要：模糊化大型项目总是在时下漏洞检测挑战。这是很难生成的时间以覆盖万行代码的程序的限制量内同时满足输入语法和指数路径条件的输入。此外，用于触发该漏洞的输入是复杂得多比实现程序的覆盖范围。因此，满足所有要求的有效输入生成方法成为一种迫切的需要。为了解决这个问题，我们采用强化学习文法推断和提高输入生成的效果优化符号分析。这个效果可以显著减少执行的时间与随机输入。在此期间，我们优化通过组成分析的启发，以减少所分析的程序大小国家的最先进的起毛的工作流。小程序的大小，保证效率的同时，修剪出不必要的程序片段供以后验证。本文的目标是开发一种能够有效地产生输入触发大规模程序的脆弱性的新颖输入的生成方法。我们已经做了在现有的基准一些实验，显示了我们方法的有效性。

关键词：起毛;输入生成;语法推理;符号分析