北京理工大学

本科生毕业设计（论文）开题报告

**学 院：** 计算机学院

**专 业：** 软件工程

**班 级：** 08111605

**姓 名：** 李博

**指导教师：** 孙建伟

**校外指导教师：** 黄庆佳

二○二○年一月十三日

毕业设计（论文）开题报告评审表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | | 李博 | | 学号 | 1120162015 | 班级 | 08111605 | 专业 | | 软件工程 |
| 导师 | | 孙建伟 | | 校外导师（职称） | | 黄庆佳（高级工程师） | | 校外导师单位 | | 中国科学院  信息工程研究所 |
| 论文选题 | 题目名称 | | | ­­基于二次汇编的ROP攻击防御技术研究与实现 | | | | | | |
| 题目性质 | | | 软件开发（ ） 理论研究（ ）  工程设计（ ） 技术科学研究与工程技术研究（ ） | | | | | | |
| 题目来源 | | | 结合科研 （ ） 结合生产实际（ ）  结合实验室建设（ ） 自拟题目 （ ） | | | | | | |
| 评审组成员 | 姓 名 | | | 职 称 | | 工作单位及职务 | | | 签 字 | |
|  | | |  | |  | | |  | |
|  | | |  | |  | | |  | |
|  | | |  | |  | | |  | |
| 评  审  意  见 |  | | | | | | | | | |
| 成 绩 | | |  | | | | | | | |
| 评审组长签字： 年 月 日 | | | | | | | | | | |

**基于二次汇编的ROP攻击防御技术研究与实现**

# 毕业设计（论文）选题的内容

随着计算机技术的不断发展，基于代码重用的漏洞利用攻击越来越普遍，如何能够有效地防御ROP攻击成为了许多人关注的方向。现有的大部分工作是在有源码的基础上对源代码或中间代码进行的修改。然而许多软件并不开源，于是在没有源码的基础上对程序进行ROP攻击防御成为了一个关键的问题。

本课题将二次汇编技术应用于ROP攻击防御，实现自动化对可执行文件进行ROP攻击防御，并对其进行ROP攻击防御测试。基于二次汇编技术，可以对可执行文件进行反汇编，在汇编层面对指令进行替换和静态插桩操作，最后对汇编文件进行重编译得到新的可执行文件，整个过程不需要源码，并且可以避免对可执行文件混淆时面临的重定位问题。

1. **研究方案**
   1. **主要任务**
2. 了解软件安全技术相关领域背景知识，了解二次汇编技术、ROP攻击防御技术的实现原理与方法，了解国内外ROP攻击防御技术发展趋势，了解相关行业的政策和法律法规；
3. 在指导教师指导下阅读国内外文献，深入学习二次汇编技术的原理、ROP攻击防御技术的方法，对已有的ROP攻击防御技术进行分析和研究，并利用二次汇编技术对ROP攻击防御技术方法进行改进，实现自动化的ROP攻击防御系统，最后在实际系统环境下进行分析与验证；
   1. **技术方案的分析、选择**

近几十年来，许多研究人员对ROP攻击提出了防御方法。G-Free论文通过在编译器层面对中间代码进行修改，实现指令替换等操作，从而减少gadgets数量，缓解ROP攻击，该方法需要程序源码并对其进行重新编译。ROPecker论文通过对程序进行预运行获取所有指令的信息，如偏移、类型和对齐方式，从而确定可能被攻击者利用的gadgets，并为这些gadgets构建一个数据库，在之后程序运行时检测gadgets是否被利用。此外该论文还使用外部监控的方式，通过滑动窗口机制来限制程序中使用gadgets的数量。以上两个例子分别代表了现今ROP攻击防御方法中的了两个限制，一是需要程序源码进行重新编译，然而现在很多程序并不开源，获取程序源码是非常困难的；二是需要使用外部监控的方式对程序运行状态进行限制，这种方法在性能方面的开销较大，会在一定程度上影响程序的性能。

基于此，本文提出基于二次汇编的ROP攻击防御技术，通过二次汇编技术可以对可执行文件进行反汇编生成汇编文件，并对汇编指令进行混淆和插桩，然后将汇编文件重编译成新的可执行文件。此过程不需要程序的源代码，并且可以在指令级别进行处理，在不对程序进行外部监控的情况下自动化的完成ROP攻击防御，减少了性能开销。

在x86架构中，gadgets不限于现有指令的序列。 实际上，由于IA-32指令集没有固定长度的指令，因此将要执行的操作码取决于内存中执行的起点。 因此，攻击者可以通过跳入现有指令来构建不同的gadgets。接下来对本文提出的基于二次汇编的ROP攻击防御方法进行介绍，包括指令替换、静态插桩和重编译三部分。

### 2.2.1 指令替换

利用angr的反汇编模块，得到程序的汇编代码。主要使用angr提供的接口reassembler 实现对汇编代码的提取。指令替换模块主要负责对机器码中含有ret指令机器码的指令进行替换。通过对特定类型的指令设计等效的指令模板，在发现符合条件的指令时用等效指令对其进行替换，能够消除立即数被解析为跳转指令的可能，并减少gadgets数量。

### 静态插桩

静态插桩模块包括无效指令填充和自由跳转保护两部分。无效指令填充是指在指令的机器码中发现ret指令的机器码但没有相应的指令替换的模板时，在指令的前面填充无效指令，如nop指令，达到防止攻击者将其解析为可利用的gadgets的目的，并且需要在无效指令前添加相对跳转指令，减少因执行无效指令产生的性能开销。自由跳转保护是指在函数入口处使用canary对返回地址进行加密，并在函数结束时对返回地址进行解密，在间接跳转（jmp）时校验返回地址是否加密。canary的值位于寄存器fs偏移0x28处，在程序的每次运行时都会随之改变。并且canary由内核维护，无法直接获取，从而达到保护控制流完整性的目的，防止攻击者直接调用跳转指令进行控制流转移。

1. 无效指令填充

a) 通过对指令进行分析，若当前指令存在被攻击者任意构建gadgets的可能，则在原有指令序列之间插入若干nop指令，将原有指令序列分隔开来，达到防御任意构建gadgets的效果。为了不因执行nop指令而增大程序执行开销，在nop指令序列前插入一个jmp指令指向nop指令序列的下一个地址。

b) 对于跳转指令（jmp，call等），若使用立即数指定目的地，则跳转指令可能包含自由分支操作码。此时在跳转指令后添加一个nop指令，并将立即数加一，就可消除这种隐患。

2. 自由跳转保护

本文提出基于canary机制的方法对自由跳转指令进行保护，防止攻击者直接调用跳转指令进行操作。

对于直接跳转指令（ret），在每个函数的入口处添加如下指令，对返回地址与一个随机值（来自fs:0x28）进行异或加密。

1. 64 4c 8b 1c 25 28 00 00 00   mov    %fs:0x28,%r11
2. 4c 31 1c 24                  xor    %r11,(%rsp)

并在ret之前进行异或解密。

1. 64 4c 8b 1c 25 28 00 00 00   mov    %fs:0x28,%r11
2. 4c 31 1c 24                  xor    %r11,(%rsp)
3. c3                           retq

对于间接跳转指令（call eax），本方法保证在每个函数入口处至少有一个生成校验值的指令序列，如下。

1. 49 bb 47 b8 1f 44 ee 03 97 52  movabs $0x529703ee441fb847,%r11
2. 64 4c 33 1c 25 28 00 00 00     xor    %fs:0x28,%r11
3. 4c 89 5d d0                    mov    %r11,-0x30(%rbp)

并在每一个间接跳转之前进行校验，保护跳转指令不被直接调用，如下所示。

1. 49 bb 47 b8 1f 44 ee 03 97 52  movabs $0x529703ee441fb847,%r11
2. 4c 33 5d d0                    xor    -0x30(%rbp),%r11
3. 64 4c 3b 1c 25 28 00 00 00     cmp    %fs:0x28,%r11
4. 0f 84 01 00 00 00              je     400638 <main+0x98>
5. f4                             hlt
6. ff 55 e8                       callq  \*-0x18(%rbp)

### 2.2.3 重编译

重编译模块负责对二次汇编框架产生的结果输出并保存到一个新的汇编文件，并对汇编文件进行编译，产生新的二进制文件。新的二进制文件需要保持和原文件相同的架构类型，由于本文实现的方法暂时只支持64位可执行文件，所以重新编译的可执行文件也应为64位。并且也要与原文件采取一样的保护方式，例如，如果原文件开启了堆栈不可执行机制，重新编译得到的二进制文件也需开启。

* 1. **实施技术方案所需的条件**

操作系统：Deepin15.9

开发语言：Python3.6

二次汇编工具：angr，patcherex，compilerex

* 1. **存在的主要问题和技术关键**

目前的主要问题是，对于自由跳转指令的保护方法有一定局限性，若canary被攻击者获取，则保护方法失效，故在后续过程中会结合前沿的ROP攻击防御技术进行一定的改进。

技术关键：

* 如何寻找可能被攻击者利用的gadget。
* 如何设计指令模板使得替换后的指令不含有自由跳转指令的机器码。
* 如何保护自由跳转指令不被攻击者任意调用，并减少性能开销。
  1. **预期能够达到的研究目标**

这些技术需要实现的目标为：

1. 自动化地对程序进行反汇编，指令混淆，静态插桩和重编译操作；
2. 混淆后的可执行文件功能上与原始文件完全相同；
3. 混淆后的可执行文件的大小较原始文件的增幅在35%以内；
4. 混淆后的可执行文件能够有效地抵抗ROP攻击。

# 课题计划进度表

a) 学习并掌握ROP攻击与防御技术原理。（第1周-第2周）

b) 配置angr等相关依赖库，熟悉二次汇编框架（第3周-第4周）

c) 结合已有的ROP攻击防御技术，使用angr等工具实现自动化对实际环境中的可执行文件的反汇编，修改汇编以及重编译操作，修改汇编过程主要包括指令替换，nop填充以及自由跳转保护。（第5周-第12周）

d) 运行实现的自动化ROP攻击防御系统，并执行修改后的程序，分析系统执行时间，程序大小，程序功能的完整性以及程序改动前后的执行时间变化，撰写实验分析文档。（第13周-第14周）

e) 完成毕业论文，提交软件及相关文档。（第14周-第15周）

f) 完成本科生毕业设计（论文）外文翻译；（第1周-第15周）

g) 完成本科生毕业设计（论文）答辩；（第1周-第15周）

# 参考文献

[1] Wang R, Shoshitaishvili Y, Bianchi A, et al. Ramblr: Making Reassembly Great Again[C]//NDSS. 2017.

[2] Cheng Y, Zhou Z, Miao Y, et al. ROPecker: A generic and practical approach for defending against ROP attack[J]. 2014.

[3]Onarlioglu K, Bilge L, Lanzi A, et al. G-Free: defeating return-oriented programming through gadget-less binaries[C]//Proceedings of the 26th Annual Computer Security Applications Conference. 2010: 49-58.

[4]Cheng, Yueqiang, et al. "ROPecker: A generic and practical approach for defending against ROP attack." (2014): 1.