北京工业大学学生开题报告表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课题名称 | 基于二进制动态翻译的ROP攻击检测方法研究与实现 | | | | |
| 课题来源 |  | 课题类型 | CX | 导 师 | 詹静 |
| 学生姓名 | 刘天祺 | 学 号 | 15074707 | 专 业 | 物联网工程 |
| 开题报告内容：（调研资料的准备，设计目的、要求、思路与预期成果；任务完成的阶段  内容及时间安排；完成设计（论文）所具备的条件因素等。）   1. 调研资料的准备   在开题前，我对即将研究的题目进行深入的调查，熟悉并掌握了相关的背景知识，并且完成了一篇相关论文的翻译工作，与此同时，我查阅了与ROP攻击相关的十余篇文章，增进我对该领域及我所研究的题目的理解。  目前的ROP攻击防御机制可粗略分为5种：1.控制流完整性检测；2.动态/静态指令检测；3.用户输入扫描；4.指令重构；5.内存随机化。我所设计的题目属于第二类中的动态指令检测，即：在程序运行时动态地检测指令，并根据获取的信息判断程序行为是否为ROP攻击。  与本题目相关的背景资料如下：   1. ROP攻击   返回导向编程（ROP，Return-Oriented Programming）是另一种代码复用攻击技术，它执行称为gadget的短指令序列，而不是执行一整个函数。它最初由Shacham提出并应用于x86平台，随后被拓展到其他体系结构。ROP已被证明可实现图灵完备计算。此外，一些允许攻击者使用ROP自动构造任意恶意程序的工具已被开发出。  ROP攻击是对栈中的返回地址进行利用的一种攻击方式。ROP攻击即通过返回导向编程思想，将具有不同功能的gadget链接在一起，从而达到攻击目的。Gadget是程序中的短指令序列，它执行一些小的计算操作（例如：将两个寄存器的值相加或将某个值加载到内存）并以返回指令为结尾。攻击者通过利用返回指令，将这些gadget链接在一起，并通过向栈中写入适当的参数，便能实现将控制流从一个gadget转至下一个gadget。  一次具体的ROP攻击流程如下：   1. 搜集可用gadget，该步骤可通过gadget搜索工具ROPGadget或ropper自动完成。 2. 利用程序存在的缓冲区溢出或其他漏洞，达到覆盖栈中程序返回地址的目的。 3. 构造gadget链，并将返回地址覆写为gadget链中首个gadget的地址。 4. 当程序执行指向gadget链的*ret*指令后，程序流即被引至gadget链中，于是攻击者构造的恶意代码将被执行。   主要资料：   1. Shacham, H.: The geometry of innocent flesh on the bone: return-into-libc without function calls (on the x86). In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 552-561. ACM (2007) 2. Buchanan, E., Roemer, R., Shacham, H., Savage, S.: When good instructions go bad: generalizing return-oriented programming to risc. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 27-38. ACM (2008) 3. hDullien, T., Kornau, T., Weinmann, R.P.: A framework for automated architecture-independent gadget search. In: WOOT (2010) | | | | | |
| 1. Roemer, R.G.: Finding the bad in good code: automated return-oriented programming exploit discovery (2009) 2. ROP变种   ROP攻击的核心思想是构造gadget链，也就是将具有特殊功能的gadget链接在一起。利用返回指令*ret*链接gadget的方法（即返回导向编程，Return-Oriented Programming）最先被提出，并广泛的得到应用。因此专门检测*ret*指令的防御机制被提出，于是攻击者只好另寻他法，不再使用*ret*链接gadget，于是各种ROP攻击变种被提出。  Checkoway等人发现可以通过搜寻尾随有间接跳转的pop指令（例如*pop edx; jmp [edx]*）来进行返回导向编程。这种指令序列的行为类似于返回指令，亦可用于将gadget链接在一起。  跳转导向编程（Jump-Oriented Programming, JOP）是ROP攻击的另一种变种，它使用寄存器间接跳转代替了返回指令。JOP使用调度程序表来保存每个gadget的地址。每个gadget对应一个调度程序。调度程序是一段可以控制程序控制流的指令序列。调度程序用作虚拟程序计数器(PC)，将程序控制流转换为调度表中存储的地址条目，其中这些地址是特殊的、具有跳转导向功能的gadget的地址。在这些gadget的结尾，攻击者通过间接跳使程序控制流回归至调度程序。随后，调度程序将指针指向下一个gadget。一个简单的调度程序如下：*add edx,4; jmp [edx]*。  调用导向编程（Call Oriented Programming, COP）由Nicholas Carlini和David Wagner于2014年提出。攻击者用以间接调用指令为结尾的gadget代替以返回指令为结尾的gadget。COP攻击不需要调度程序，它通过依次将内存间接位置指向下一个gadget的方法，来将gadget链接在一起。  主要资料：   1. Bletsch, T., Jiang, X., Freeh, V.W., Liang, Z.: Jump-oriented programming: a new class of code-reuse attack. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, pp. 30-40. ACM (2011) 2. Carlini, N., Wagner, D.: ROP is still dangerous: breaking modern defenses. In: 23rd USENIX Security Symposium (USENIX Security 2014), pp. 385-399 (2014) 3. Checkoway, S., Davi, L., Dmitrienko, A., Sadeghi, A.R., Shacham, H., Winandy, M.: Return-oriented programming without returns. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 559-572. ACM (2010) 4. 二进制插桩框架PIN   PIN是二进制插桩框架，它能够检测程序所有实际执行的指令。它有两种工作模式：探针模式和即时（Just-In-Time, JIT）模式。在JIT模式下，PIN能够在处理器执行每条指令前将其拦截，如果程序被攻击，非正常的程序流中的指令将被拦截，于是通过分析指令特征，便可以检测攻击。其中起到分析检测功能的代码被称为Pintool。PIN框架提供链各种各样的插桩检测API。用户可以使用使用这些API来编写Pintool。  PIN框架由三部分组成：虚拟机（VM）、代码缓存和供Pintool调用的API。其中，虚拟机包含JIT编译器、模拟器和调度程序。当程序开始运行时，各条指令先经JIT编译并检测，再交由调度程序激活并执行。经过编译的指令存储在代码缓存中，以便在多次调用代码段时降低性能开销。模拟器用于解释那些无法被直接执行的指令。  主要资料：   1. <https://software.intel.com/sites/landingpage/pintool/docs/97619/Pin/html/index.html> | | | | | |
| 二、设计目的  目前ROP攻击是一种有力的攻击方式，它能够绕过绝大多数的现有防御机制，即使程序开启了最新的地址随机化（ASLR, Address Space Layout Randomization）保护，也无法反全的防御ROP攻击。因此，本题目中采取了基于二进制插桩框架的动态检测方法，目的在于检测并防御所有类型的ROP攻击，并在程序遭到ROP攻击时及时关闭程序，防止损失。此外，由于动态检测的缘故，必然会导致程序运行放缓，因此本题目设计的另外一个目的是尽可能的减小这种副作用。   1. 设计要求 2. 能够检测现有的各种ROP攻击及其变种。 3. 检测到ROP攻击后，采取相应的防御措施。 4. 不出现误报，以免影响程序的正常工作。 5. 设计思路   我目前的思路是结合静态和动态检测，选取恰当的ROP特征对其进行识别。结合静态分析的原因在于，动态检测会导致程序放缓，为了减少检测带来的这种副作用，我决定引入静态分析，利用静态分析获取尽可能多的信息，从而减少动态检测时获取的信息量以达到整体优化的目的。  具体的候选检测方法如下：   1. 静态收集程序所用的所有函数（名称、地址），即合法地址集合，记为*setF*，程序运行时，动态检测*ret,jmp,call*的目标地址，若不在*setF*中，即不是正常函数入口，则判定为gadget。这种检测方法利用了gadget首地址必然不是正常函数入口地址的特征。 2. 提取gadget集合，记作*setG*，动态检测时比对程序流的二进制代码，若是gadget链，则会连续或频繁击中gadget集合*setG*。不过这种思路存在缺陷，因为正常代码量远大于gadget的代码量，比对所有代码会资源大量消耗。后续会想办法加入其他gedget特征来克服该缺陷。 3. 预期成果 4. 实现上述系统，并取得尽可能小的系统开销。 5. ROP攻击防御效果的界面展示。   六、任务完成的阶段内容及时间安排  任务内容：  1. 查阅资料，了解ROP攻击及其变种，并归纳总结其特点。  2. 查阅资料，了解现有的ROP攻击及其变种的检测方法。  3. 熟悉动态插桩工具PIN的使用方法。  4. 设计ROP攻击及其变种检测的新方案并实现。  5. 通过界面展示攻击检测结果。  时间安排：  2018.12 - 2019.3 完成工作前两项，查阅资料，加深对课题的理解。  2019.3 - 2019.4 开题，提出检测方案。进行初步的系统实现，并完成70%以上的系统实现。3月10日前，完成论文背景现状部分，撰写需求和概要设计文档。  2019.4 - 2019.5 对系统进行调试、优化，完成原型系统。  2019.5 - 2019.6 撰写论文，准备答辩。 | | | | | |
| 1. 完成设计（论文）所具备的条件因素   毕业设计主要内容可以分为以下三个部分：  1. 搭建实验所需的环境。  2. 设计动态插桩检测工具。  3. 演示ROP攻防御效果。  其中，实验所需的环境，可以通过搭建虚拟机实现。检测工具可由C/C++语言编写并在PIN框架下运行。演示ROP攻防御效果，需要准备攻击素材，我在本科期间曾参加过信息安全类的竞赛，专攻二进制安全，在此期间，我积累了丰富的ROP攻击案例，可作为展示素材。 | | | | | |

课题类型：（1）A—工程设计；B—技术开发；C—软件工程；D—理论研究；

（2）X—真实课题；Y—模拟课题；Z—虚拟课题

（1）、（2）均要填，如AY、BX等。