**软件漏洞分析技术读书报告**

姓名：仝志欣

|  |
| --- |
| **本周预期完成任务：**阅读软件漏洞分析技术的第九章内容 |
| **本周实际完成进度：**了解了模糊测试的一些关键技术，及两个典型的工具。 |
| **详细内容：**   1. **模糊测试基本原理**   **1. 基本概念**  基本思想：向待测程序提供大量特殊构造的或者是随机的数据作为输入，监视程序运行时的异常记录并记录导致异常的输入数据，辅助以人工分析，基于导致异常的输入数据进行进一步定位软件中漏洞的位置。  **2. 基本过程**  模糊测试的具体方法随着测试元素的不同而变化，没有一种对所有测试都适用的模糊测试方法。完全取决于目标应用程序、研究者的技能，以及被测试数据的。  五个基本阶段：  识别目标  识别输入：发往目标应用程序的任何输入都应该被认为是输入向量，因此都应该是可能的模糊测试变量。  构建模糊测试用例：如何使用预先确定的值、如何变异已有的数据或者动态生成数据。这些决策将取决于目标应用程序和数据格式。  检测执行并过滤异常  确定可利用性   1. **模糊测试的方法实现** 2. **输入数据的关联分析**   通过分析输入到程序的数据对象的结构以及其组成元素之间的依赖关系,构造符合格式要求的测试用例，从而绕过程序格式检查流程,是提高模糊测试发现程序漏洞可能性的一个重要步骤。   1. 输入数据结构化分析：在模糊测试过程中，对网络数据包或文件格式的结构进行分析，识别出特定的可能引起应用程序解析错误的字段，有针对的通过变异或生成的方式构建测试用例。通常，对于模糊测试有意义的字段主要有以下几种：表示字段长度、表示偏移的字段、可能一年期应用程序执行不同逻辑的字段、可变长度的数据。书中以HTTP的数据包为例 2. 数据块关联模型   数据块关联模型是目前解决模糊测试中处理复杂数据对象的一条有效途径。该模型以数据块为基本元素,以数据块之间的关联性为纽带生成畸形测试数据。其中,数据块是数据块关联模型的基础。通常,根据数据对象的内部结构,一个数据对象可以划分为几个数据块。不管是网络数据包还是文件数据,其内部各数据块之间都有比较紧密的依赖关系。数据块之间的依赖关系称为数据关联。  数据块关联模型主要由三部分组成：关联方式、关联强度、评价标准。（可见p220页）   1. **测试用例集的构建方法**   根据测试数据生成策略,可以将现有模糊测试方法分为两大类:①基于变异的模糊测试,该类模糊测试方法采用对已有数据进行变异技术来创建新的测试用例;②基于生成的模糊测试,该类模糊测试方法通过对文件格式或网络协议建模从头开始产生测试用例。  随机方法、强制性测试、预先生成测试用例、遗传算法、错误注入与模糊启发式  预先生成测试用例：这是 PROTOS模糊测试框架所采用的方法。测试用例的生成开始于对一个专门规约研究,其目的是为了理解所有被支持的数据结构和每种数据结构可接受的取值范围。然生成用于测试边界条件或迫使规约发生违例的硬编码的数据包或文件。这些测试用例可于检验目标系统实现规约的精确程度。创建这些测试用例需要事先完成大量的工作,但其优点是在测试相同协议的多个实现或相同文件格式时测试用例能够被多次重用。预先成测试用例的一个缺点是这种方式下的模糊测试存在固有的局限性:由于没有引入随机制,一旦测试用例表中的用例被用完,模糊测试只能结束。  遗传算法：通过不断提高适应值的大小，来使基因逐渐优良化。  错误注入与模糊启发式：与生成测试数据的方式同等重要的是生成测试数据的标准。软件错误注入指按照特定的故障模型,用人为的、有意识的方式产生故障,并施加特定故障于待测软件系统中以加速该系统错误和失效的发生。通常,可注入的错误类型有:①内存错误;②处理器错误;③通信错误;④进程错误,例如死锁、活锁、进程循环、进程挂起及使用过多的系统资源等:⑤消息错误,例如失去消息、已损坏的消息、无序的信息、信息复制和组件间等待消息时间超时等;⑥网络错误;⑦操作程序错误;程序代码错误,如语句、变量等。在模糊测试中,基于错误注入思想生成的测试数据通常是包含特定潜在危险值的模字串或者模糊数值列表。通常,将模糊字串或者模糊数值列表中包含的特定潜在危险值称作模糊启发式( Fuzz Heuristics,几种常见的模糊启发式包括以下几种。  (1)边界整型值。在整型测试用例列表中,选择极限边界作为测试用例的意图是很明显的。整型值的上溢、下溢以及符号溢出都可能会导致代码行中出现潜在的安全问题。  (2)字符串重复。字符串重复中,除了最常见的 AAAA…外,也应该使用由的ASCH可见字符构成的重复串,例如: BBBBBBBB…。这样做的原因很多其一,堆结构被改写为AAAA……和BBBB…在触发某些堆溢出漏洞时会具有不同行为,例如,微软 Windows操作系统中的堆溢出漏洞;其二,很多软件中已经专门针对AAAAAAAA…进行搜索并阻止。  (3)字段分隔符。包含非字母数字字符(包括空格和制表符等)同样很重要。这些字符通常被用做字段分隔符和终止符。  (4)格式化字符串  (5)字符转换和翻译  (6)目录遍历  (7)命令注入   1. **测试异常分析**   在模糊测试过程中，首先通过分类过滤并获取程序的某些运行信息（尤其是异常发生时的运行信息），然后对记录下来的信息中我们感兴趣的部分进行重点分析从而提高漏洞的挖掘效率。这个过程就是测试异常分析。  在程序动态分析过程中，相关信息获取途径有五个方面：通过程序的正常输出获取、通过静态代码插装获取信息、通过静态二进制插装获取信息、通过调试接口或者调试器、通过虚拟机获取信息  常用的方法：调试接口或者调试器   1. 调试接口监控   调试接口监控是指在程序调试执行状态下，通过操作系统提供的调试接口或者调试器控制应用程序执行，同时对执行状态进行监控。UNIX系列的ptrace系统调用和Windows系统的调试API都是典型的调试接口。   1. 异常及时过滤   模糊测试过程中,通常为每一个测试用例启动待测程序的一个进程。对于每一个在调试接口下运行的进程,调试接口均会捕获到大量的异常。在这些异常中,有些可能随后会被程序自身的异常处理逻辑捕获,而有些确实是会引起程序崩溃。模糊测试器应该能够识别每一种异常,并正确地过滤需要由调试接口处理的异常,并记录确实会引起程序崩溃的异常供后续分析使用。  在模糊测试中，主要记录的就是读内存异常、写内存异常和访问内存异常。   1. 异常影响分析和漏洞危害判定   对于会引起程序崩溃的异常,模糊测试平台会记录异常发生时的寄存器、堆栈等异常环境、引起程序崩溃的测试用例。不同的异常会对系统安全造成不同的影响,根据这些记录的异常信息,通过使用基于模式匹配的自动化工具可以快速地对每一个异常对系统的影响进行评估,并确定真实的漏洞。对于每一个已确认的漏洞,通常需要具有一定漏洞利用经验的安全分析人员手工对每个漏洞进行精细分析,确定漏洞可能造成的危害,以及可能的利用方法。   1. **模糊测试框架**   模糊测试框架是一个通用的模糊器,可以对不同类型的目标进行模糊测试,它将些单调的工作抽象化,并且将这些工作减少到最低程度,从这一点来看,所有模糊测试框架的共同目标都是相同的,即为模糊器的开发者提供一个快速的、灵活的、可重用的以及同构的开发环境。采用模糊测试框架,研究者就可以导入大量的经验数据,并且将其关注的重点放在更加适合于人工完成的工作上。  一款通用且支持拓展的模糊测试框架应该由测试数据生成、动态调试、执行监控、自动化控制脚本、异常过滤、测试结果管理以及可扩展插件等主要模块构成。（具体可见p231页）   1. **典型工具**   **1. Peach**  **2. Sulley** |
| **相关文件及参考资料：** 软件漏洞分析技术的第九章内容 |
| **遇到的问题及该阶段学习的想法：从第九章开始就进入了二进制漏洞分析技术的层面，跟之前的是不太一样的，并且难度也有所增加，很多内容不容易理解。我想这需要多实践才能更加深入理解。** |
| **接下来的进度计划：** 阅读第十章 |