**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十二章内容 |
| **本周实际完成进度：**二进制代码比对技术的目的就是通过分析软件开发商提供的补丁的信息，定位出补丁所修补的软件漏洞。本周了解基于二进制代码比对的漏洞分析，以及三个工具。 |
| **详细内容：**   1. **基本原理**   **1. 研究现状**  补丁及其所发布的信息相当于；漏洞研究者的指南针，所以软件厂商近年来向公众封闭与补丁相关的漏洞技术细节。漏洞研究者唯一能够得到的是软件系统在打补丁前后的二进制代码。  通过二进制比对分析技术,可以快速定位补丁发布前后代码的差异,从而确定原程序中代码被修补的位置,也就可以确定原代码中存在漏洞代码的精确位置,为后续进行漏洞的利用提供重要线索。所以,二进制比对分析技术是进行1day漏洞快速利用的基本方法。  但是,二进制比较有着诸多的难点:**由于信息的不对称,对源码简单修改后重新编译,就会导致逆向工程中对目标代码的分析需要完全重做来检测修改;一个补丁往往多个问题,使得代码变化较多;编译器对二进制代码的优化,如指令顺序的颠倒、寄存器选择的不同等,增加了比较的难度。**这些难点使得传统的基于文本比对的补丁比较方法不能很好地工作,如二进制字节对应比较只适用于若干字节变化的补丁比较。而通过对反汇编后的文本进行比较,由于缺乏对程序逻辑的理解,只适用于小文件和少量变化。  **2. 基本原理**  (1) **补丁比对技术原理**  补丁比对技术是一种通过比对打补丁前后的目标代码,分析发现两个版本的程序之间差异的技术。这里所说的差异是指语义上的差异,即程序在执行时所表现出的不同的逻辑行为。通过二进制代码比对定位出有差异的函数,再经过进一步的人工分析,可以确定出二进制补丁对程序执行逻辑上的修改,从而推测漏洞位置及成因,辅助漏洞挖掘工作。  目前在补丁对比中可以利用多种二进制比对技术来实现，其中，基于结构化的二进制比对技术是目前采用最多的一种。  (2) **二进制比对技术原理**  基于文本的比对  基于图同构的比对  基于结构化的比对  综合比对技术   1. **方法实现**   **在这三种技术中，基于结构化的二进制对比方法是目前最适合补丁比对技术的方法。**   1. **基于文本的比对** 2. **二进制字节的对应比较**   二进制字节比较并不涉及逻辑分析,仅适用于查找文件中极少量字节差异。字节级别的比对方法一般不需要对二进制文件进行处理,可以直接从二进制文件中读入数据进行比较。较为常用的算法是最长公共子序列算法。  这种方法很直接，但是缺乏比对方法缺乏对语义和程序整体逻辑的理解，比对的效率很低。当比对对象体积较小、逻辑简单时，该算法能够发挥其简单快速的优势。   1. **反汇编文本比较**   字面意思理解，就是将二进制代码进行反汇编得到的文本进行比对。该比对实际上是一种指令级别的比对方法，它把汇编指令作为分析对象，研究指令之间相似性和差异性，指令之间比较关系可以分为“相似”“相近”“可忽略”“不同”。具体说明可参考p310底部。   1. **基于图同构的比对**   图形化比对技术是以图论为基础，通过对可执行程序的图进行抽象，将可执行程序转化为一个有向图，从而把二进制代码比对问题转化为了图的同构问题。在进行问题的转化后，也需要进行相应的同构问题调整，根据二进制文件进行以下假设:  (1)不同版本的两个目标文件从本质上是不同构的,算法的目标是找到一个“最佳”匹配映射,而不需要穷尽所有匹配。  (2)可执行文件提供的基本信息可作为匹配的起点,例如程序入口点、DLL文件的导出函数等。  (3) 生成的有向图中,大部分顶点只有一个入口和一个出口。  (4)图中的顶点有可比较的属性,包括操作码、操作数等  (5)不对整个图进行同构匹配,而是寻找图中某一部分的同构匹配,例如以函数为单位进行同构匹配。  有了上述的前提假设,就可以对具体的二进制代码进行图同构比对。按照比对的层次,基于图同构比对的二进制比对技术可以分为两种,即指令级图同构比较和函数级图同构比较。   1. **指令级图同构比对算法**   指令级图同构比对的基本思路是:两个需要比较的可执行文件分别构造成图,以指、数据常量、函数调用指令等作为顶点,以控制流图的边作为图的边。对生成的两个图做同构识别,用同构算法找到最相似的两个部分作为同构部分。然后,对两个图的非同构部分继续识别，直至结束。  分析二进制文件，获得函数、引用表、字符串等。  识别比较的开始点。对于PE文件，可以用IMAGE\_OPTIONAL\_HEADER结构中的成员EntryPoint，也可以分析导出函数表，匹配相应的导出函数，作为比较的开始点。  处理队列   1. 函数级图形化比对算法 2. 基于结构化的比对   结构化比对技术区别于指令比对技术,其注重点是可执行文件逻辑结构上的变化,而不是某一条反汇编指令的改变,这自然也就规避了上述问题。一般的,描述二进制代码文件需要用到如下两种类型的图  函数调用图,描述函数之间的调用关系。可执行文件的二进制码通常由多个函数所组成。假设其函数集合为:{(f1…f},在函数调用图里面,每一个结点代表一个函数从到f的边则代表了调用关系,说明f是f的子函数。函数调用图描述了整个二进制文件的函数调用关系,其中的每一个节点代表了一个函数。  具体描述一个函数的控制流图(CFG)。控制流图是一个有向图,它被用来描述进制文件中的一个函数。一个函数所有的语句序列可以被分为若干基本块。基本块是一个连续的语句序列,控制流从它的开始进入,从它的末尾离开,不可能出现中断或者分支(末尾除外)。控制流图的节点由基本块组成,表示计算;节点之间的边表示控制流向。每一个函数控制流图都只有一个入口,但可以有多个出口。   1. 基于结构化签名的二进制比对   不安全函数主要包含c函数中的一些没有判断输入长度的内存和字符串操作函数，比如：strcpy、strcat、sprintf等，针对参数量固定和不固定，又分为两类（详细见于p290）   1. 基于可信基点的结构化签名二进制比对 2. **软件补丁的二进制比对技术**   相比较而言,在用于软件补丁比对方面,结构化的比对技术更适合软件补丁。软件的补丁往往修复的是软件逻辑上的漏洞,因此,软件补丁与软件之间的差异更多是结构层面上的。结构化比对技术的关键在于函数的签名技术,而函数签名的设计、实现都相对容易。但是,结构化的比对算法也存在着一定的问题需要解决。对于函数签名的碰撞问题,方面在设计上需要尽力避免碰撞,另一方面可以通过算法方面来解决碰撞问题,提高比对的准确度。对于代码耦合度较低、结构化特征不是很明显的可执行文件,可以通过算法的选择子,选择合适的算法来进行比对,以达到算法与文件特征相适应。   1. **典型工具**   **1. Bindiff**  **2. Eye Binary Diffing Suite** |
| **相关文件及参考资料：** 软件漏洞分析技术的第十二章内容 |
| **遇到的问题及该阶段学习的想法：新学期第一章，遇到了很多术语，尤其认为结构化签名和最小素数积认为比较难以理解** |
| **接下来的进度计划：** 阅读第十三章 |