__kingzone__的专栏 学习数据挖掘~

个人资料

kingzone_2008

访问： 296233次
积分： 5002
等级： BLOG 6
排名： 第2375名

原创： 109篇 转载： 15篇
译文： 19篇 评论： 86条

BitBlaze（二） - BitBlaze架构

分类： Security&Privacy 2013-04-21 15:22 1491人阅读 评论(0) 收藏 举报

安全 代码分析 二进制 计算机安全

2 BitBlaze二进制代码分析平台的架构

在这一节中，我们首先讨论对安全问题的二进制分析的难点，然后介绍一个满足安全应用的二进制分析平台需要的条件，最后勾勒出Bitblaze平台的架构。

2.1挑战

二进制代码分析面临几个主要挑战，有一些还是需要具体分析的。

复杂性。二进制代码分析的第一个主要难点在于二进制代码是很复杂的。二进制分析必须对这些复杂的代码进行精确的建模以保证分析的精确性。然而，现代计算机体系结构中指令巨大的规模和复杂度使精确建模编程了一项浩大的工程。常见的现代计算机体系架构中有成百上千条指令，并在新处理器中不断增加。而且每条指令都可以有复杂的语义，比如单指令循环 定执行效果的指令，以及隐藏的副作用（如设置处理器的标志位）。例如，介绍x86指令语义的IA-32手册里还11傍。

举个例子，考虑一下下列x86汇编程序的控制流：

```
// instruction dst, src

add a, b // a = a + b

shl a, x // a << x

jz target //jump if zero to address target
```

第一条指令add a, b，相当于a:=a+b。第二条指令shl a, x，相当于a:=a<<x。最后一条，jz target，表示当零标志位置位时跳转到target。

这里有一个问题，即add和shl指令都有隐含的副作用。这两条指令还可能修改六个处理器状态标志位。这六个标志位分别指示计算结果是否为零，结果的奇偶性，是否存在辅助进位，结果是带符号数还是无符号数，结果是否产生溢出。

条件转移语句，如jz指令，由这些隐含计算的标志位决定。因此，add或shl都可能会计算zero标志位。然而，二者谁会最终决定转移分支呢？这个问题很难直接回答。Shl指令根据操作数的不同产生不同效果：它只在操作数非零时更新zero标志位。

缺乏高级语义。第二个难点在于二进制代码缺乏源代码的高级语义。因此，我们需要开发适于二进制代码（通常难以获得调试信息）的程序分析技术和工具。另一方面，二进制代码缺乏抽象，这是源代码及其分析的基础，举例如下：

-没有函数。二进制代码中不存在函数。它们的控制流是通过跳转实现的。例如，x86指令call x实现以下操作：把当前指令指针（eip寄存器）存入esp寄存器所指向的地址，esp减小，然后把x赋值给eip。代码中会有调用一个“函数”中的某个位置，或者有某个单独的“函数”被隔离在非连续的片中，这在汇编中是合法的，有时候也会实际用到。

Security&Privacy (5)
笔试题 (7)
Python (0)
ACM (30)
Linux (3)
Maven (3)
汇编 (1)

文章存档

2015年05月 (1)
2015年04月 (2)
2015年03月 (4)
2015年01月 (1)
2014年12月 (1)
展开

阅读排行

GitHub入门：如何上传与 (16046)
Java：String、StringBut (11815)
掌握VS2010调试 -- 入门 (10390)
jQuery动态添加删除sele (8884)
Java：Date、Calendar、 (7928)
Java List与数组之间的转 (6680)
Java连接SQL Server：j (6627)
Spring MVC视图层：thy (6152)
数据挖掘（六）：预测 (5912)
Nexus创建本地Maven仓 (5527)

Favorites

Java 6 Doc
Java SE 6 API
Java 6 中文Doc
C++ Reference
POJ
ZOJ
HDOJ
九度OJ
LeetCode OJ

编程语言

性能排名
TIOBE 流行度
LangPop 流行度
透明排行榜
Github top lang
Computer Programming Language Statistics
DB Engines

数据库与数据挖掘

Oracle Online Documentation
ACOUG
Ask Tom
ERI - Data Mining & Predictive Analytics

-内存**vs.**缓冲区。二进制代码没有缓冲区，而只有内存。由于操作系统可以检测特定内存块是否非法，内存没有描述用户类型和大小的语义。缓冲区和内存的一个不同点在于在二进制代码中不存在缓冲区溢出问题。尽管我们认为一种特定的存储违背了源代码所支持的高级语义，但是这仅是出于遵守高级语义，而不是因为二进制代码本身。

-没有类型。因为二进制代码中没有类型构造器，所以二进制代码中不能创建和使用类型。仅有的类型是由硬件提供的：寄存器和内存。然而即使是寄存器类型也不一定提供可信的信息，因为我们经常将数值存在一个寄存器类型（如：32位寄存器）中，却使用另一种寄存器类型（如：8位寄存器）读取它。

总的来说，由于在现代复杂的指令系统的基础上开发分析工具非常乏味且容易出错，因此基于汇编的方法是不可行的。在一个如此巨大而且相当复杂的指令集上校验程序分析的正确与否是很困难的。而且，基于汇编的方法是平台相关的。我们希望已有的分析结果可以移植到任何新的平台上。因此，分析难以利用不同汇编语言之间的共同的语义。

全局观点。许多的安全应用要求包含分析操作系统内核操作和多个进程间的交互等功能，因此需要有全局的考量，这就比传统的单个程序的分析困难的多。

代码混淆（花指令）。一些安全应用要分析恶意代码。恶意代码可能使用一些诸如加壳，加密，混淆之类的反分析技术来防范代码分析，因而使其比分析良性代码更加困难。

2.2 设计原理

BitBlaze的目标是设计和开发一些技术以及相关工具以满足安全应用的共同需求，并为其他开发人员提供一个开发新工具的简便且高效的环境。鉴于前述难点，我们对BitBlaze平台的架构提出了一些准则：

精确性。我们希望通过精确的分析来建立准确的指令模型，从而使相关工具能够对程序执行进行建模。

可扩展性。鉴于二进制代码分析的复杂性，我们希望开发一些可以在各种复杂的二进制分析中重用和扩展的核心工具。

静态和动态分析的结合。静态分析和动态分析各有所长。因为能够遍历不同的执行路径，静态分析能够给出更完整的结果，但是这又因为指针混淆、非直接跳转的普遍使用、二进制代码缺乏对类型和其他高级抽象的支持等因素而变得相当困难。甚至静态地区别哪些是代码哪些是数据都是不可能完成的。其次，静态分析技术难以分析动态生成的代码和采用了反分析技术的恶意代码。另外，某些指令（如内核和浮点指令）难以精确地建模。另一方面，动态分析技术规避了静态分析技术面临的许多难点，当然，这是以每次分析一条路径为代价的。因此，我们将静态和动态分析技术结合起来以获取二者的优点。

2.3 架构

鉴于前述难点和设计原理，BitBlaze平台由三个部分组成：Vine，静态分析组件，TEMU，动态分析组件，Rudder，结合动态和静态分析进行具体和符号化分析的组件。

Vine将汇编语言翻译成一种中间语言（IL）并提供一系列在IL上进行静态分析的核心工具（包括控制流，数据流，优化，符号化执行，最弱前提计算）。

TEMU进行动态分析，以支持系统全局的细粒度监控和动态二进制仪表。它提供了一系列工具用于提取操作系统级语义，用户自定义的动态污点分析，以及一个用于用户自定义活动的插件接口。

Rudder利用Vine和TEMU提供的核心功能在二进制代码级进行具体及符号化混合的执行。对于一条指定的程序执行路径，它能给出满足要求的符号化输入参数。通过诸如决策过程的解算装置，它可以判定什么输入信息能使程序按给定的路径执行。因此，Rudder可以生成特定的输入以引导程序执行不同的路径。Rudder也为用户提供了一系列的工具和接口。

上一篇 评论: VMWare Workstation 3.1 vs Virtual PC 4.3.2 vs Bochs 1.4
下一篇 BitBlaze（三） - 静态分析组件Vine

主题推荐 代码分析 操作系统 汇编语言 开发人员 源代码

猜你在找

BitBlaze五 - 应用及相关工作
BitBlaze TEMUTracecap 在64位Ubuntu 1404 LTS上的编
我的AIX入门之路完整版
linux驱动学习1－ubuntu 内核源码下载及编译
反编译APK
【精品课程】开源信息安全管理平台OSSIM入门
【精品课程】.NET平台和C#编程从入门到精通
【精品课程】通俗易懂UML
【精品课程】零基础学HTML 5实战开发(第一季)
【精品课程】零基础学Java系列从入门到精通

准备好了么？跳吧 ！

更多职位尽在 CSDN JOB

系统架构师、asp.net高级开发	我要跳槽	前端架构师	我要跳槽
杭州常青网络科技有限公司	8-15K/月	上海乾才企业管理咨询有限公司	20-40K/月
JAVA架构师（数据监控和存储产品）	我要跳槽	PHP架构师	我要跳槽
北京星网锐捷网络技术有限公司	20-30K/月	玩咖欢聚文化传媒（北京）有限公司	25-30K/月

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap