看雪 2018 安全开发者峰会

Kanxue 2018 Security Developer Summit

2000-2018

自 动 逆 向 机 器 人

弗为@阿里安全

自我介绍

弗为

- 阿里巴巴安全部-猎户座攻防实验室安全专家
- 主要关注：二进制程序分析与漏洞挖掘、软件供应链安全
- 结合工程界传统人工分析经验与学术界方法论成果，多种尝试、工具规模化自动化。

向“逆向分析”致敬

第4章

逆向分析技术

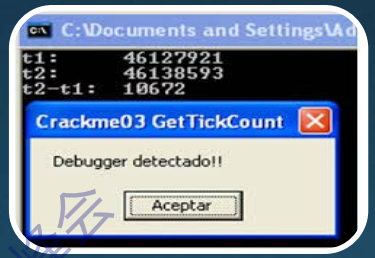
将可执行程序反汇编，通过分析反汇编代码来理解其代码功能，如各接口的数据结构等，然后用高级语言重新描述这段代码，逆向分析原软件的思路。这个过程被称做“逆向工程（Reverse Engineering）”，或者有时只是简单地称作“逆向（Reversing）”。这是一个很重要的技能，需要扎实的编程功底和汇编知识。逆向分析的首选工具就是 IDA，其中它的一款插件 Hex-Rays Decompiler 能完成许多代码反编译的工作，逆向时可以作为一款辅助工具参考。

逆向工程可以让人们了解程序的结构以及程序的逻辑，因此利用逆向工程可以深入洞察程序的运行过程。一般所谓的“软件破解”只是逆向工程中非常初级的一部分。本节探讨的代码分析技术是基于 IA-32 处理器体系结构的。

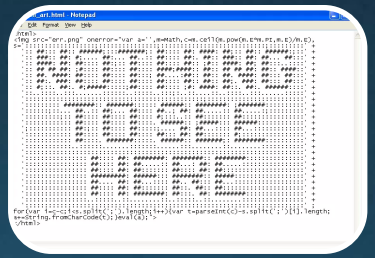
- 我们很多人从这里入门，
- 从此走上了逆向“痛并快乐”的生涯

从狭义“逆向”的痛点说起

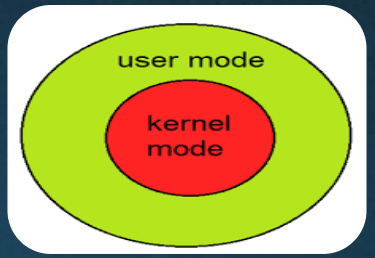
机械重复



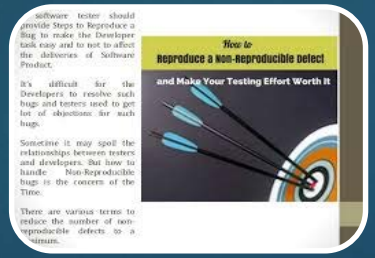
反调试



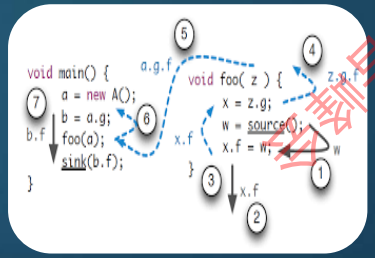
混淆



多体系/架构



不稳定重现

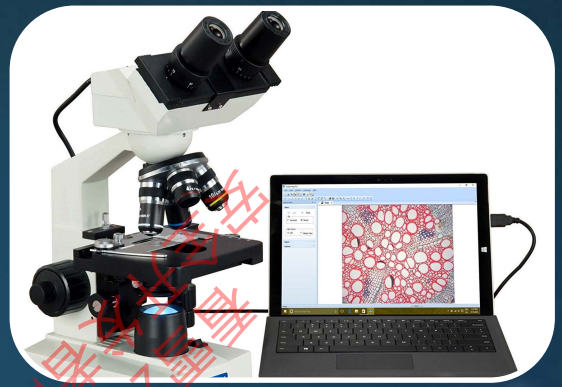


数据流跟踪

理想的基础工具能力

录像机

播放机



显微镜

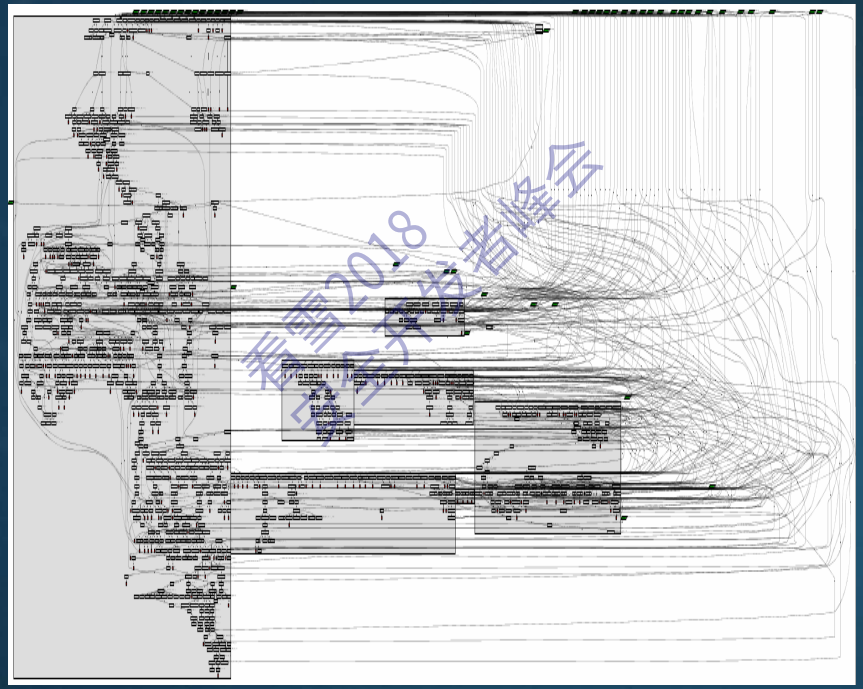
实现准确录放

- 全系统模拟
- 变量控制
  - 初始状态：CPU、内存、外设
  - 运行过程：中断、IO输入、DMA等
- 确定性与变化
  - 对确定的重放，可**在线**或**离线**分析
  - 可修改上下文

全系统与细粒度分析

- 细粒度：处理器视角
  - 每条指令，每瞬时内存上下文
- 架构与系统支持
  - 虚拟机自省：进程/线程，文件系统，网络，IPC，内存管理
- 上帝模式
  - 对抗反调试反虚机
  - 形成“关联搜索”

符号化污点数据流分析

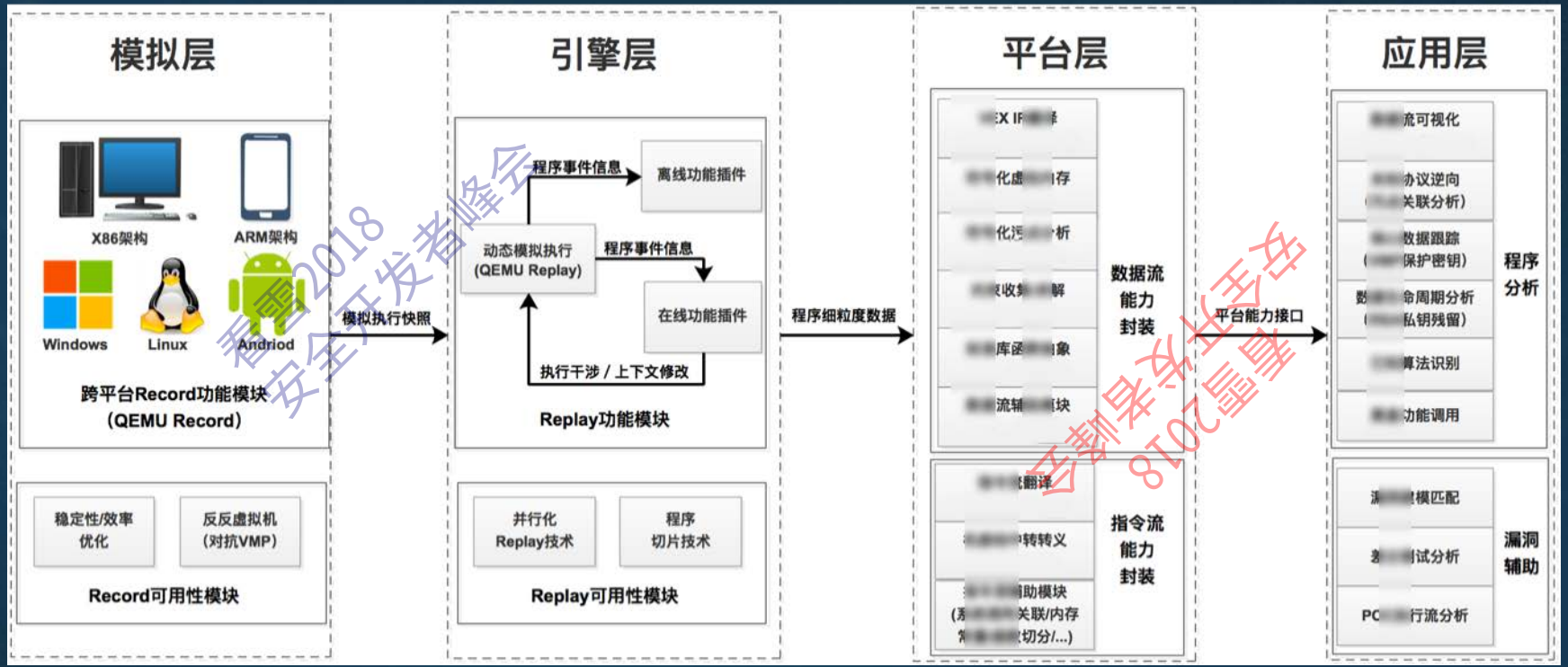


- 符号执行+污点分析 扬长避短
  - 标定-跟踪-表示
  - 以符号运算表示指令语义
- 还原程序、函数的数据本质：变换

性能

- 运算架构
  - 流水线-并行-归并
- 录制性能损失 < 30x
  - 纯CPU密集型约30x

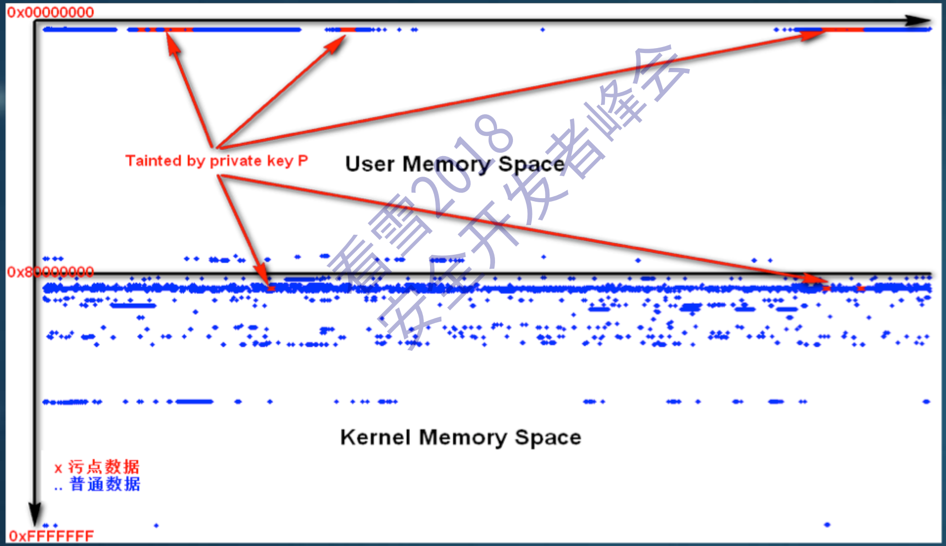
自动逆向机器人TimePlayer



应用1：WannaCry勒索软件RSA私钥恢复

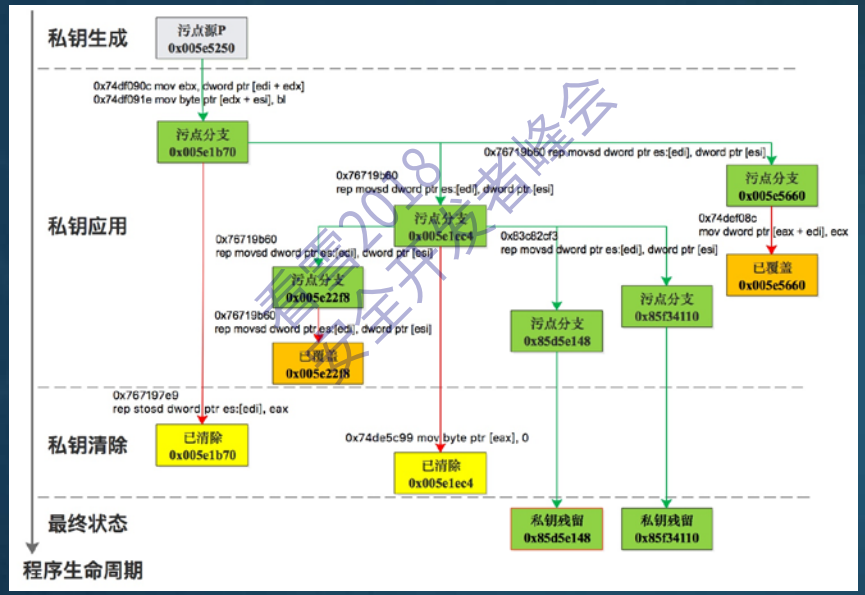
- 寻找甜点：残留私钥？
  - 未得到有目的性擦除的内存
- 如果有，有哪些地方、在什么时机？
- 密码学组件，是否可能“故意”残留了敏感数据？

应用1： WannaCry勒索软件RSA私钥恢复



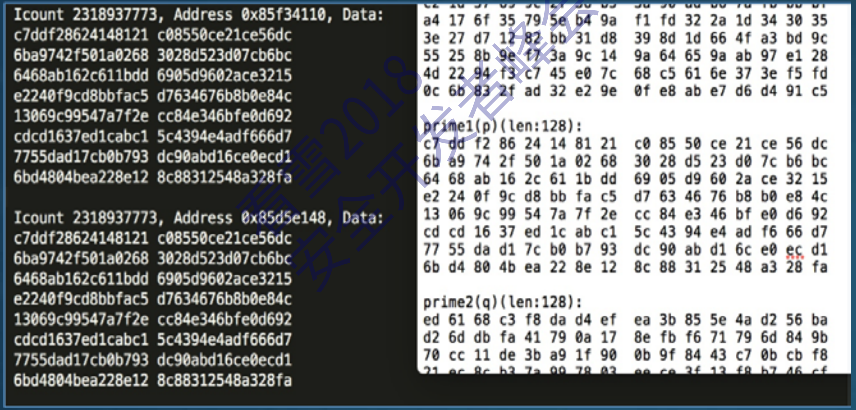
- 针对正常的加解密全过程，应用符号化污点分析，分析全内存空间的数据痕迹

应用1： WannaCry勒索软件RSA私钥恢复



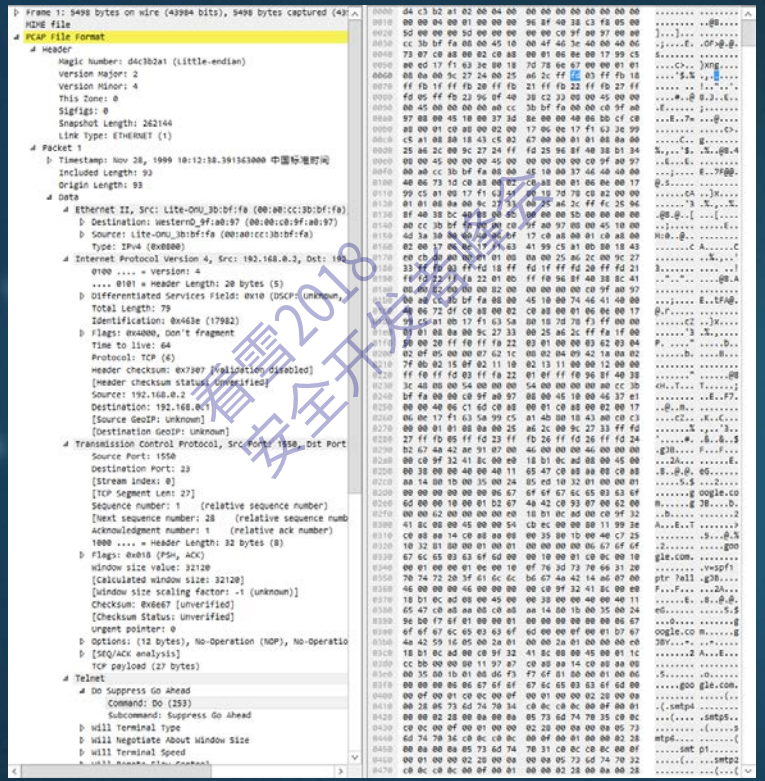
- 私钥参数P的救赎
  - 结合符号化污点分析结果和程序运行周期，整理私钥P局部传播图
  - 从用户态到内核态

应用1： WannaCry勒索软件RSA私钥恢复



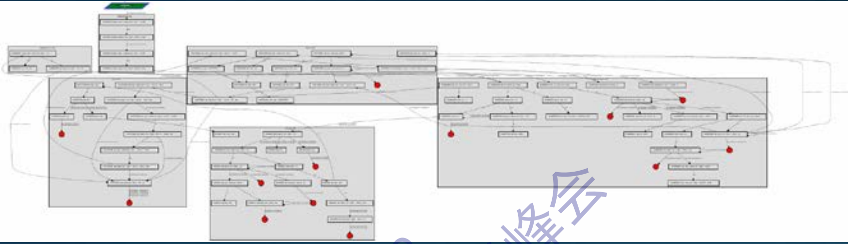
- 在程序调用CryptDestroyKey和CryptReleaseContext之后提取地址0x85d5e148和0x85f34110的内存数据，与P的值做比较

应用2：自动化私有网络协议逆向

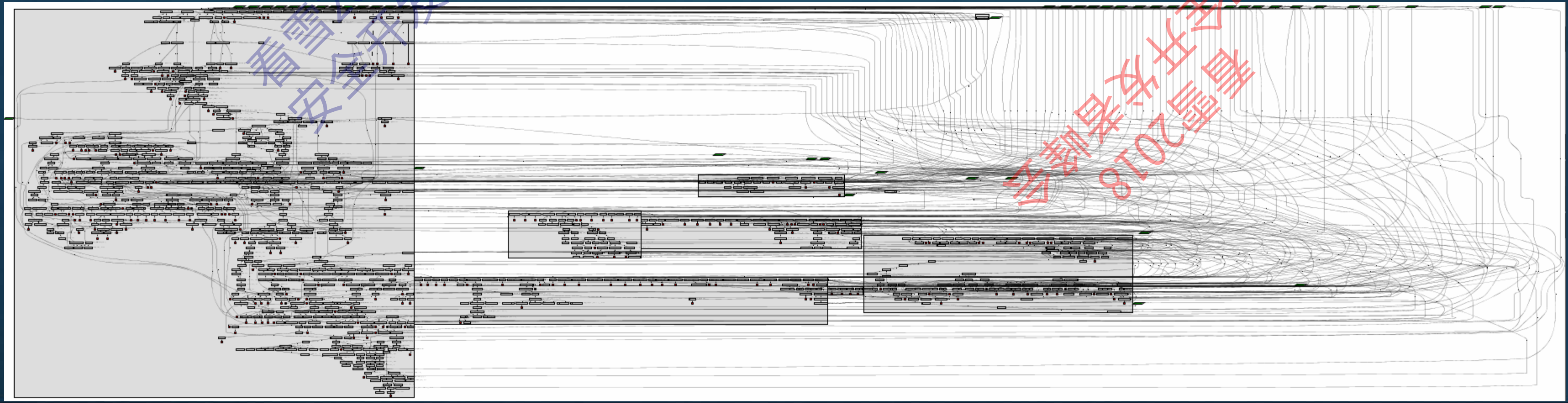


- 拿得到程序与数据，却对数据格式一头雾水？
  - 以往依据不断的调试、猜测从中获取有用信息
- 掌握了程序对数据的处理过程，就能归纳推测出程序处理过程的“语义”，进而对输入数据的“语法”做推断。

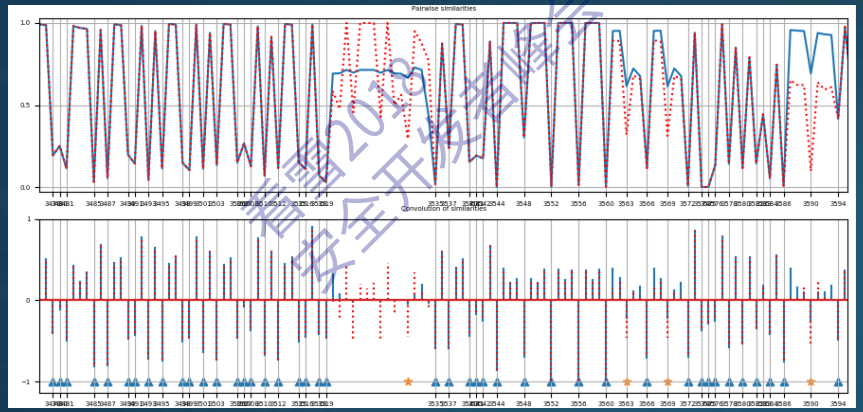
应用2：自动化私有网络协议逆向



- 字节粒度/带符号信息的准确描述，保证尽量消除了过传播与欠传播，并保证全量覆盖。
- 对每个字节做污点数据流图，可获得主观的结构认知和相似印象。

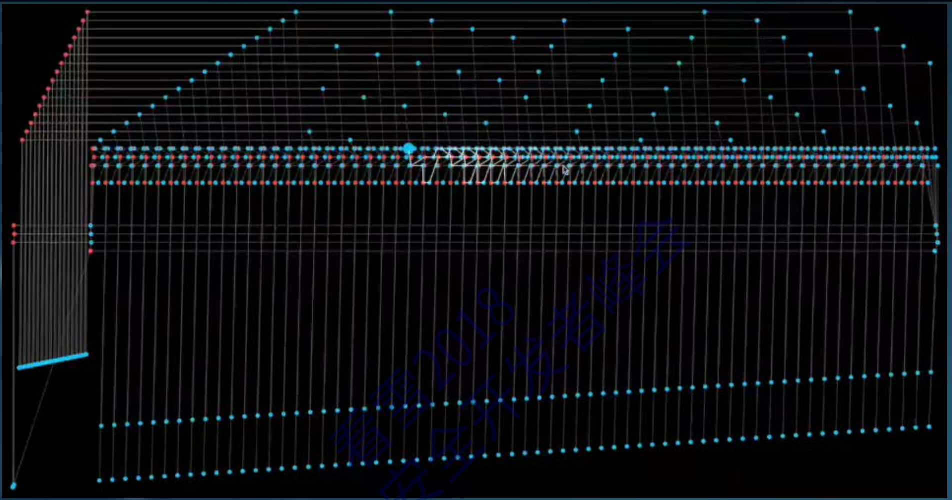


应用2：自动化私有网络协议逆向

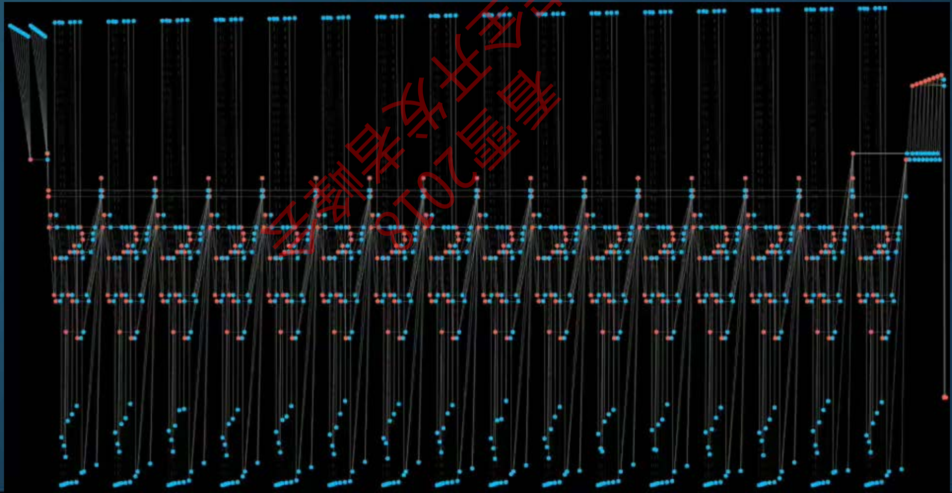


- 字段边界判定
- 字段聚类
  - 考察时间代码相关性，暴露字段相似度，聚类得到上层协议结构；
- 典型模式学习匹配，获取特殊属性字段推断
  - 加密、压缩、序列化数据块
  - 校验和、哈希字段
  - 长度、偏移量、间隔符、终止符字段等

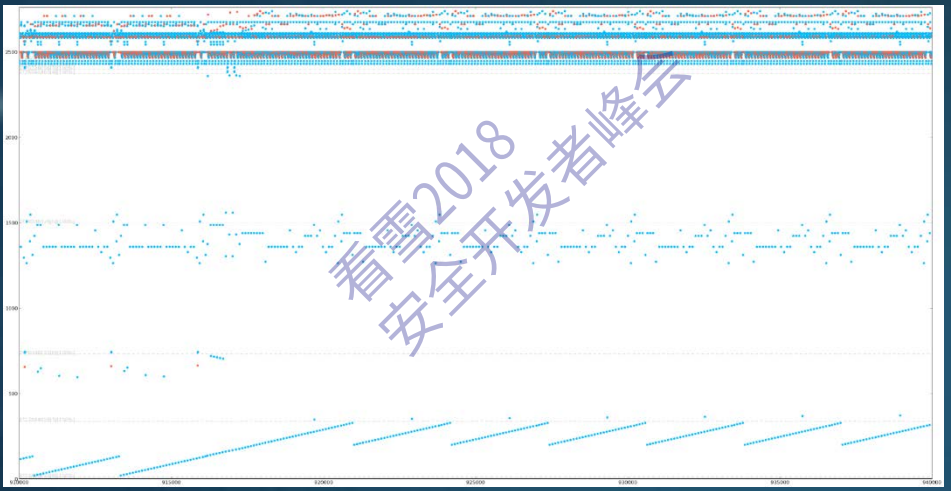
应用3：虚拟机壳等混淆下的算法识别



- 任何函数、过程、算法，从数据维度总有特定的输入输出模式

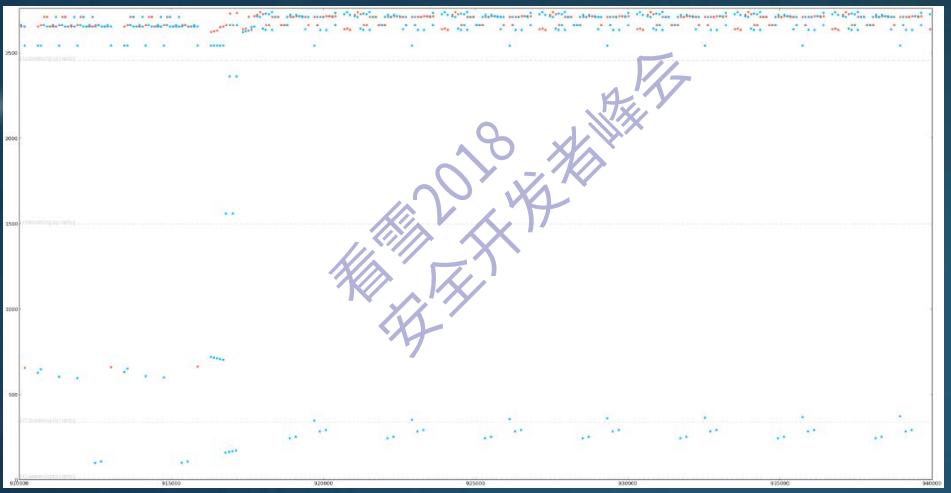


应用3：虚拟机壳等混淆下的算法识别



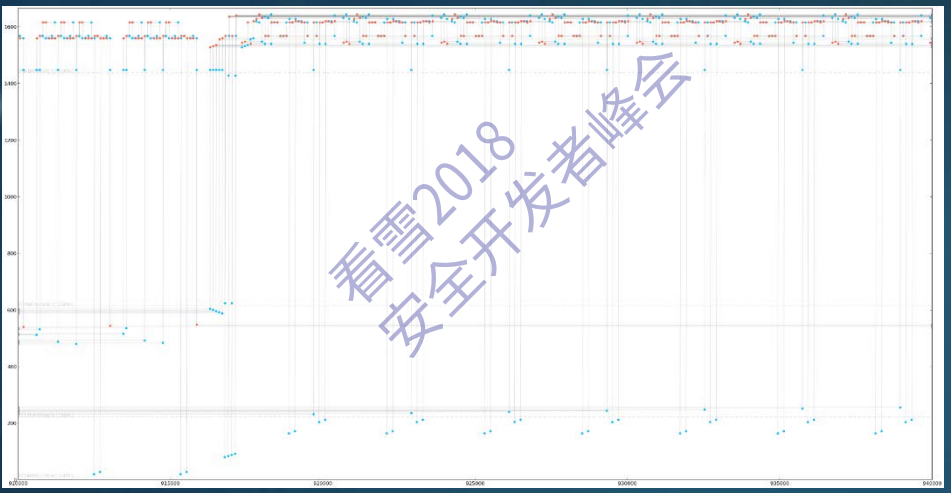
- 通过**逆向污染分析**，筛选出算法相关的指令，排除掉混淆部分的无关指令，指令数量大幅缩减，再通过数据流可视化，勾勒出数据计算之间的关系，方便进行抽丝剥茧的分析。

应用3：虚拟机壳等混淆下的算法识别



- 通过逆向污染分析，筛选出算法相关的指令，**排除掉混淆部分的无关指令**，指令数量大幅缩减，再通过数据流可视化，勾勒出数据计算之间的关系，方便进行抽丝剥茧的分析。

应用3：虚拟机壳等混淆下的算法识别



- 通过逆向污染分析，筛选出算法相关的指令，排除掉混淆部分的无关指令，指令数量大幅缩减，再通过**数据流可视化**，勾勒出数据计算之间的关系，方便进行抽丝剥茧的分析。

总结&思考

- 重新定义“逆向工程”😢
  - 逆向的不只是代码碎片，更是全局、逻辑、功能
  - 为工具开发者打call，自动化、规模化是大势所趋
- 二进制的黑盒性质，代码混淆的障眼法，总有失效的一天
  - 人力+经验vs固定模式的机器语言混淆，在两方面都将改变
  - 将出现的密码学强度可证明的混淆模型
- 业界圈子需要吸纳学术成果，理论创新需要工程实践改造