网络安全新威胁与对抗技术论坛

恶意代码同源分析

唐勇
国防科技大学计算机学院网络与信息安全研究所
2014年1月16日

Who am I？

- 国防科技大学计算机学院网络与信息安全研究所
- 投身安全十余年（2000-）
  - 2年的IDS系统工程工作（2000-2002）
  - 3年的蜜罐技术研究，研究Linux内核、hack AIX （2002-2004）
  - 3年的攻击特征自动提取技术研究（2004-2008）
  - 5年的网络安全设备（十余款设备）研制经验（2008-2012）
  - 1年的反病毒、反APT技术研究（2013-）
- 曾徘徊在产业、学术、政府的三界边缘空间
  - 参与过973、863、科技支撑、自然科学基金、242
  - 推销产品、销售员
  - 搬过机器、蹲守机房
  - 写了不少paper
  - 战略研究报告
  - ……

提纲

- 前言
- 产业界的工作
- 学术界的研究
- 反思：恶意代码同源分析方法
- 结束语

恶意代码同源分析：从生物学得到的启发

- 同源（Homology）

若两个或多个物种具有相同的祖先，则称它们同源

- 同源和相似的区别

非同源相似：昆虫的翅膀和鸟类的翅膀

趋同演化：相似的结构有不同渠道演化而来

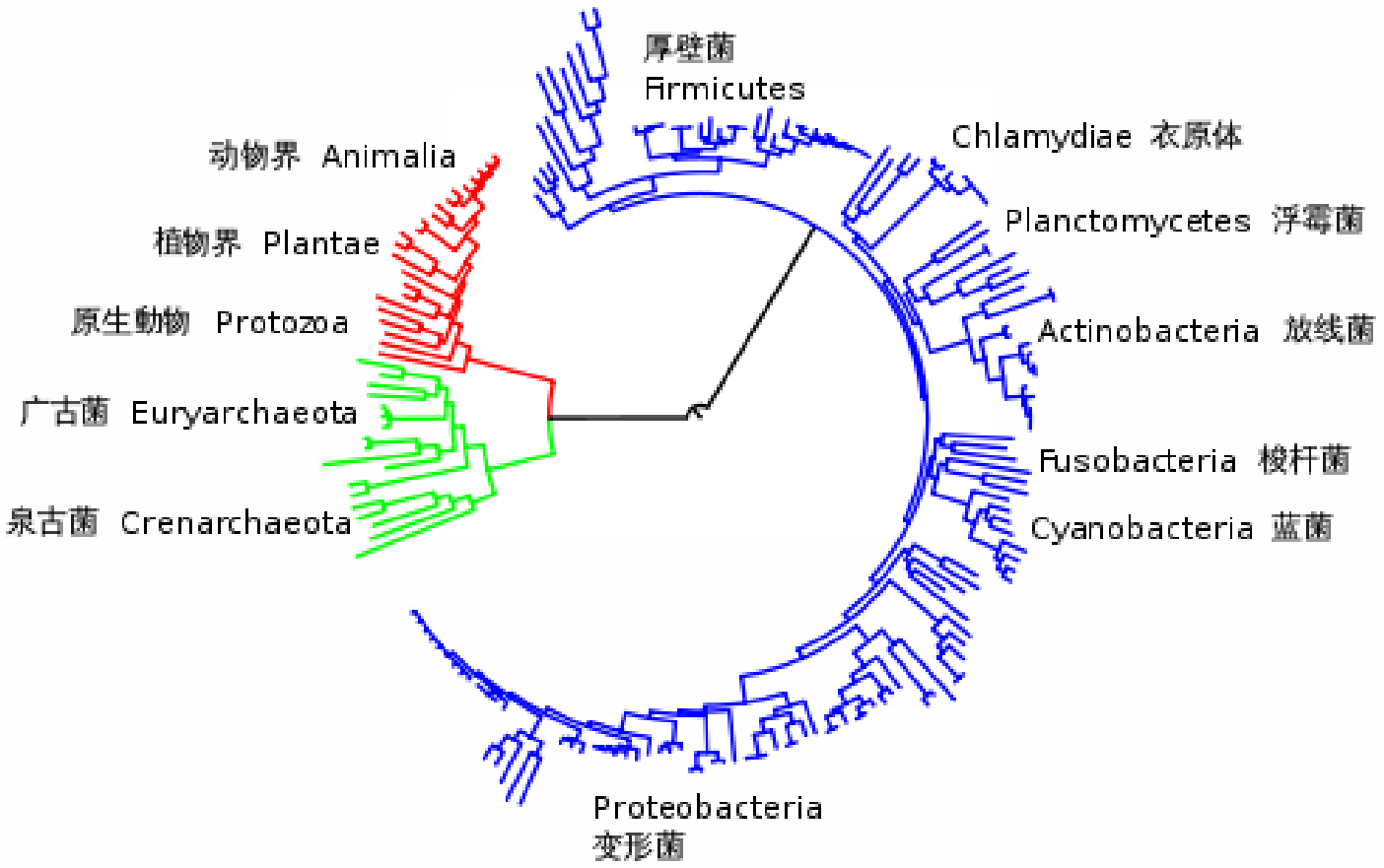
- 生物同源分析

生物学中，蛋白质和DNA的同源性常常通过它们序列的相似性来判定

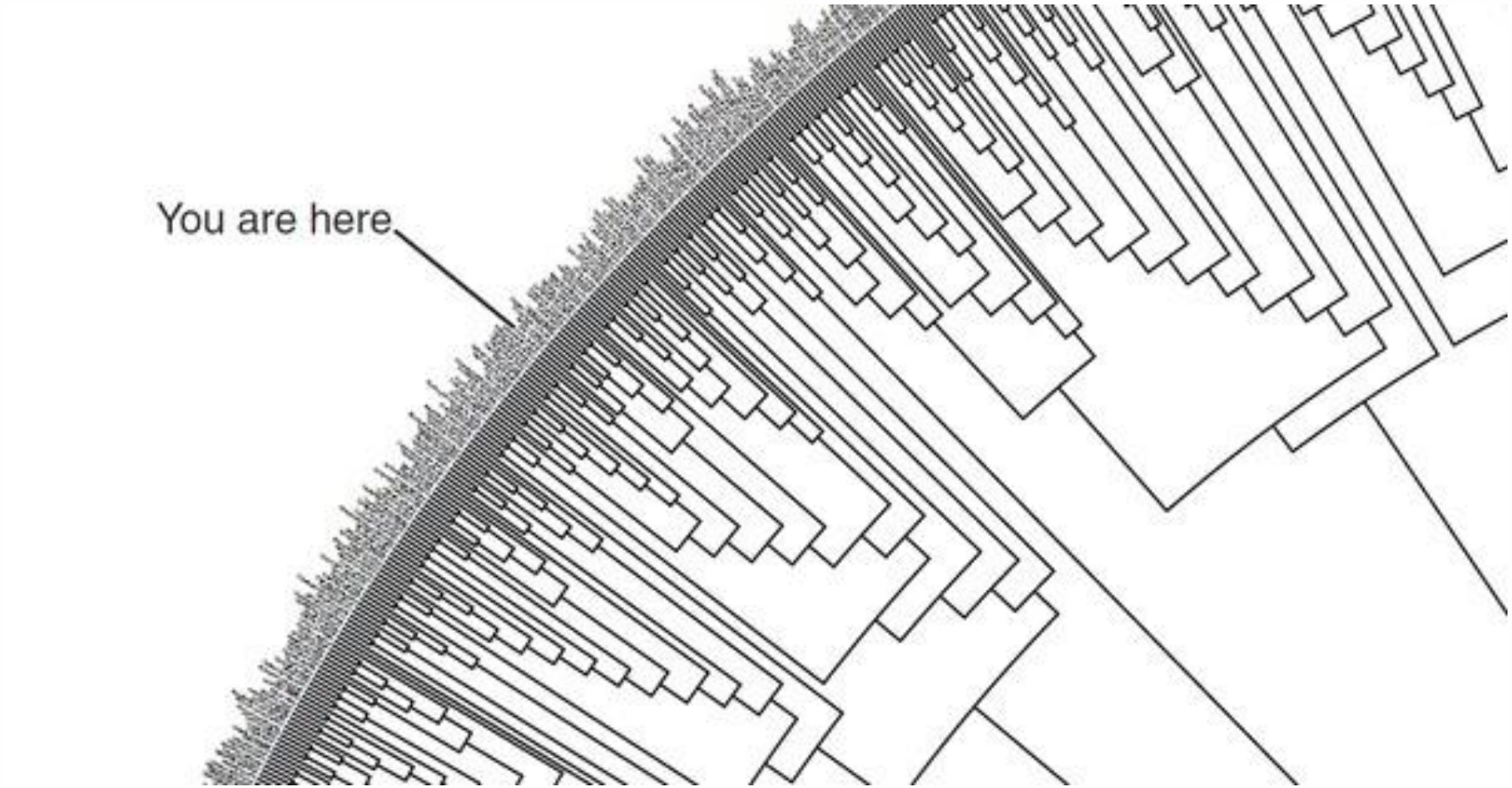
- 家族树（进化树，系统发生树）

三个或者更多个体之间进化关系的典型图示，不仅表示了数据集之间的关系，还体现了它们的分歧时间和它们共同祖先的特征

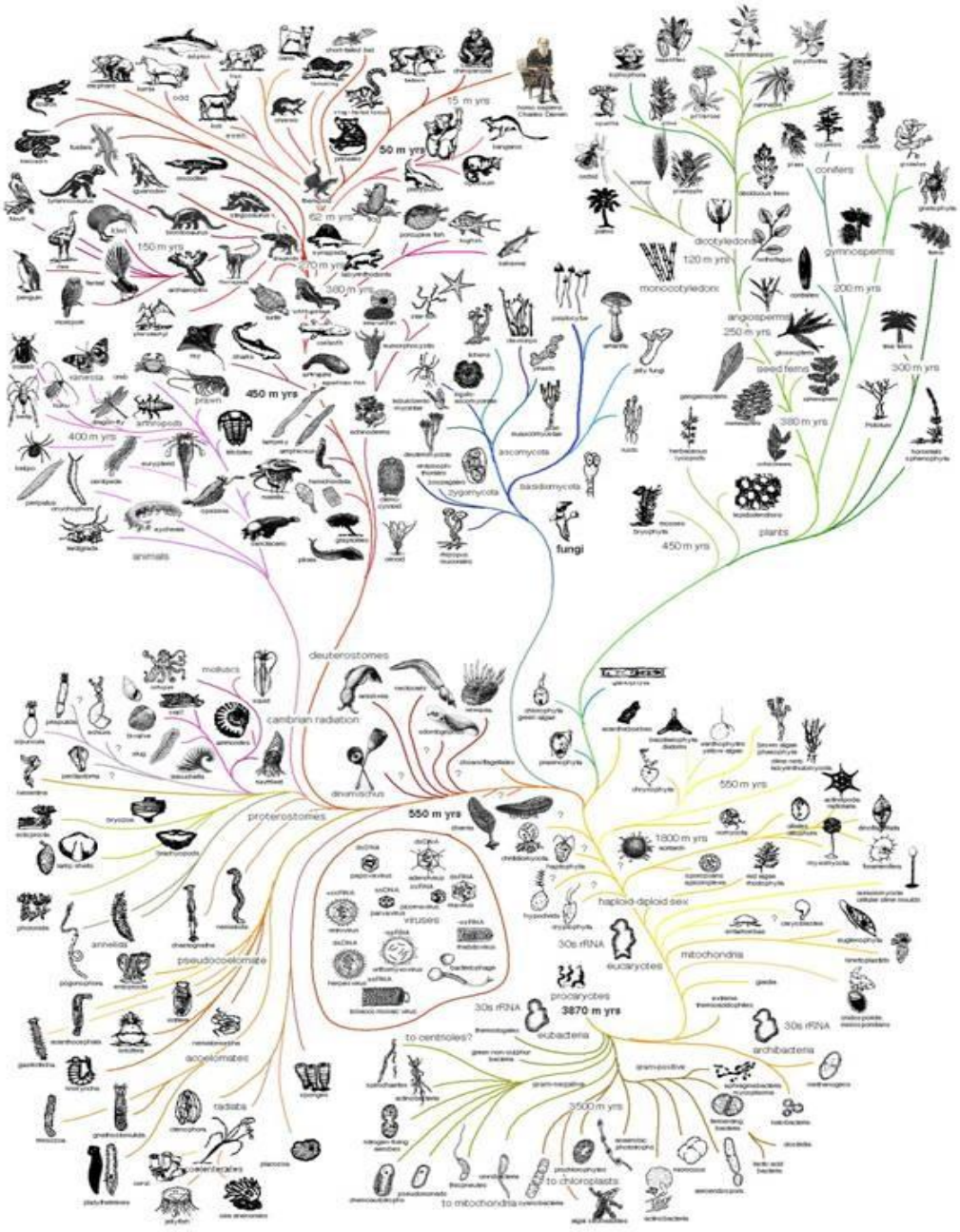
生物学进化树例子


厚壁菌
Firmicutes
动物界 Animalia
Chlamydiae 衣原体
植物界 Plantae
Planctomycetes 浮霉菌
原生動物 Protozoa
Actinobacteria 放线菌
广古菌 Euryarchaeota
Fusobacteria 梭杆菌
泉古菌 Crenarchaeota
Cyanobacteria 蓝菌
Proteobacteria
变形菌

生物学进化树

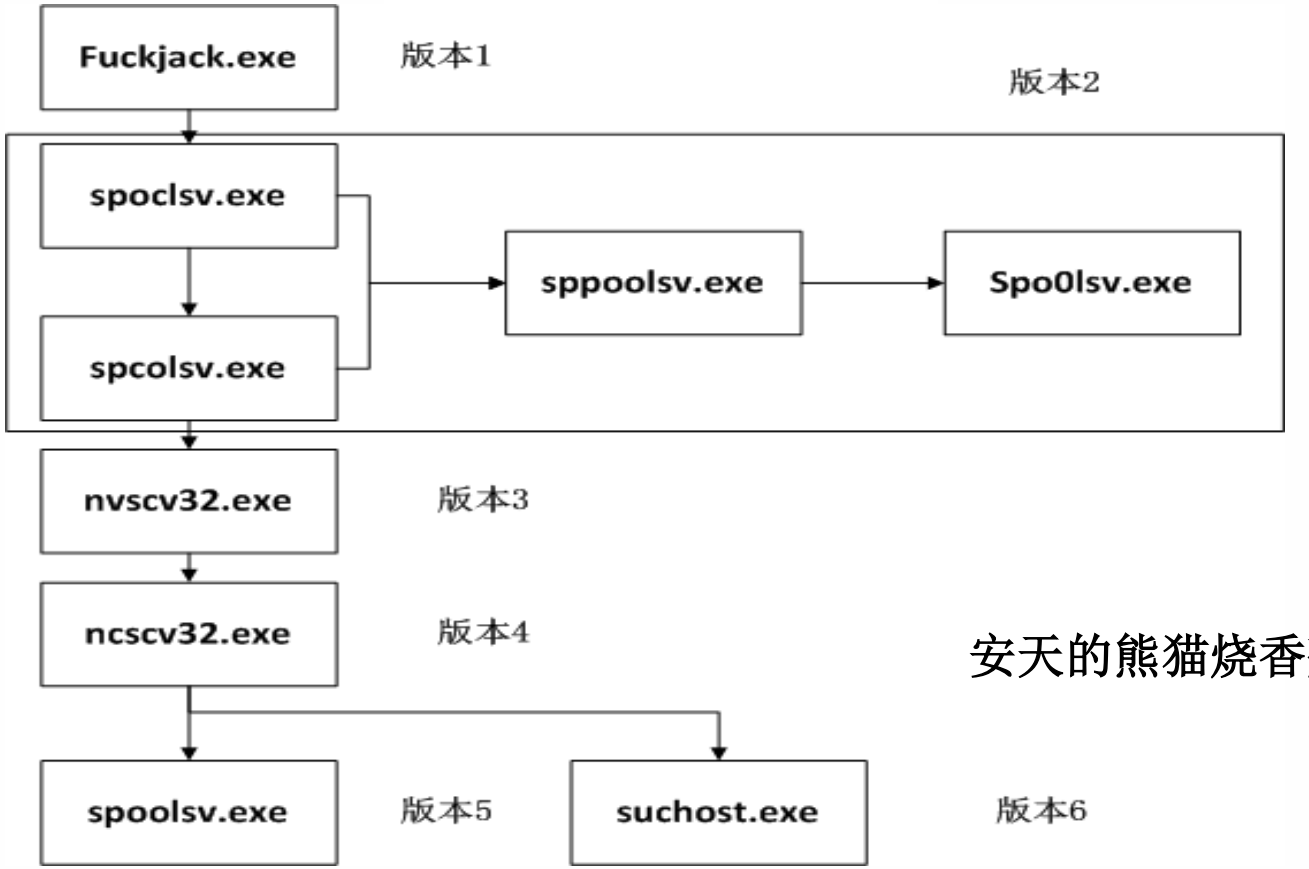


生物学进化树


50 m yrs
65 m yrs
150 m yrs
300 m yrs
380 m yrs
450 m yrs
400 m yrs
fungi
plants
conifers
dicotyledons
monocotyledons
gymnosperms
angiosperms
seed ferns
250 m yrs
120 m yrs
200 m yrs
300 m yrs
380 m yrs
450 m yrs
deuterostomes
cambrian radiation
proterostomes
550 m yrs
1800 m yrs
haploid-diploid sex
mitochondria
eucaryotes
prokaryotes
3870 m yrs
eubacteria
archibacteria
30s rRNA
viruses
pseudocoelomate
acoelomates
radiata
to centrioles?
to chloroplasts
to mitochondria
3500 m yrs

恶意代码代码同源

- 能否构建恶意代码的家族树？



安天的熊猫烧香变种数72个

恶意代码代码同源

- 恶意代码同源分析的动机
  - 恶意代码命名
  - 了解恶意代码如何演化，以及恶意代码之间的关系
  - 在APT时代，同源分析可聚合APT分析的线索
- 恶意代码同源分析定义
  - 两个或多个恶意代码从同一恶意代码的源代码经过趋异变化而形成，则同源
- 恶意代码规模增长的数字
  - 6000万样本数 vs 16万家族
  - 新增恶意代码家族速度 = 60+/天
  - 最大的家族？

恶意代码同源与生物同源的比较

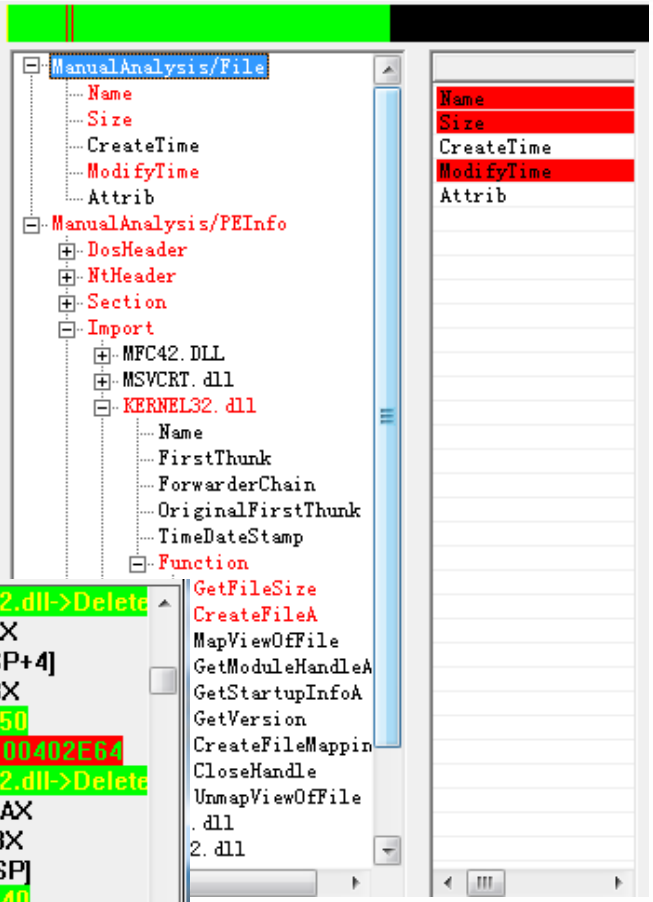
生物	恶意代码	备注
物种 (species)	恶意代码家族	生物学有非常好的科目分类依据。病毒厂商分类不统一，粒度粗
DNA/RNA	源代码	DNA/RNA一般是可测定的 源代码（除解释执行）一般难以完全逆向
基因	源代码片段	基因片段决定了生物的某一项“特质”，源代码片段同样
外表（结构）	可执行代码，行为	生物体观察，恶意代码静态/动态分析
基因型相似 Genotype	源代码相似	本质上的相似
显型相似 Phenotype	代码相似，行为相似（调用序列）	表象上的相似

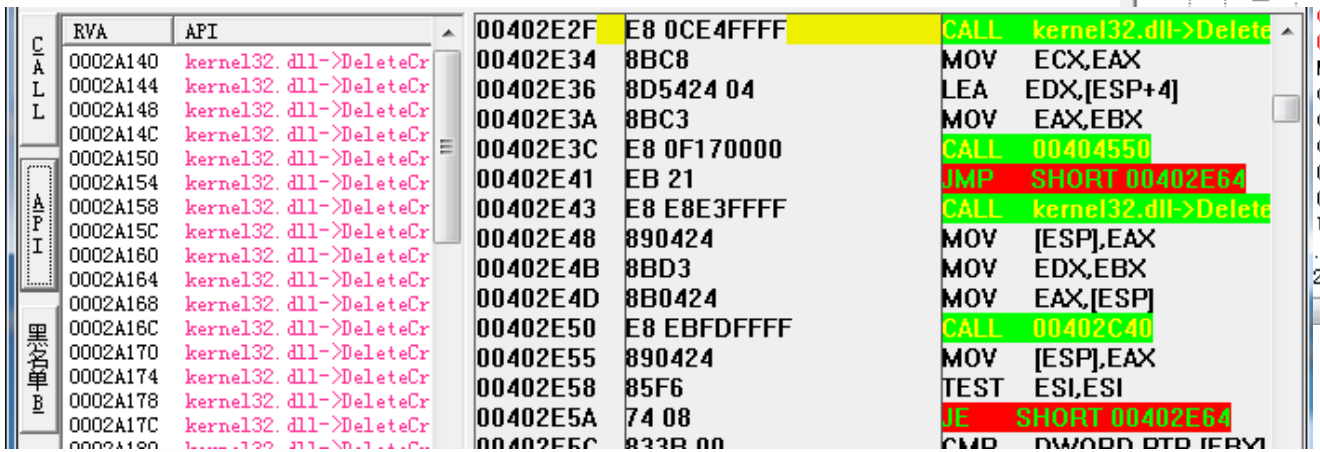
提纲

- 前言
- 产业界的工作
- 学术界的研究
- 反思：恶意代码同源分析模型
- 结束语

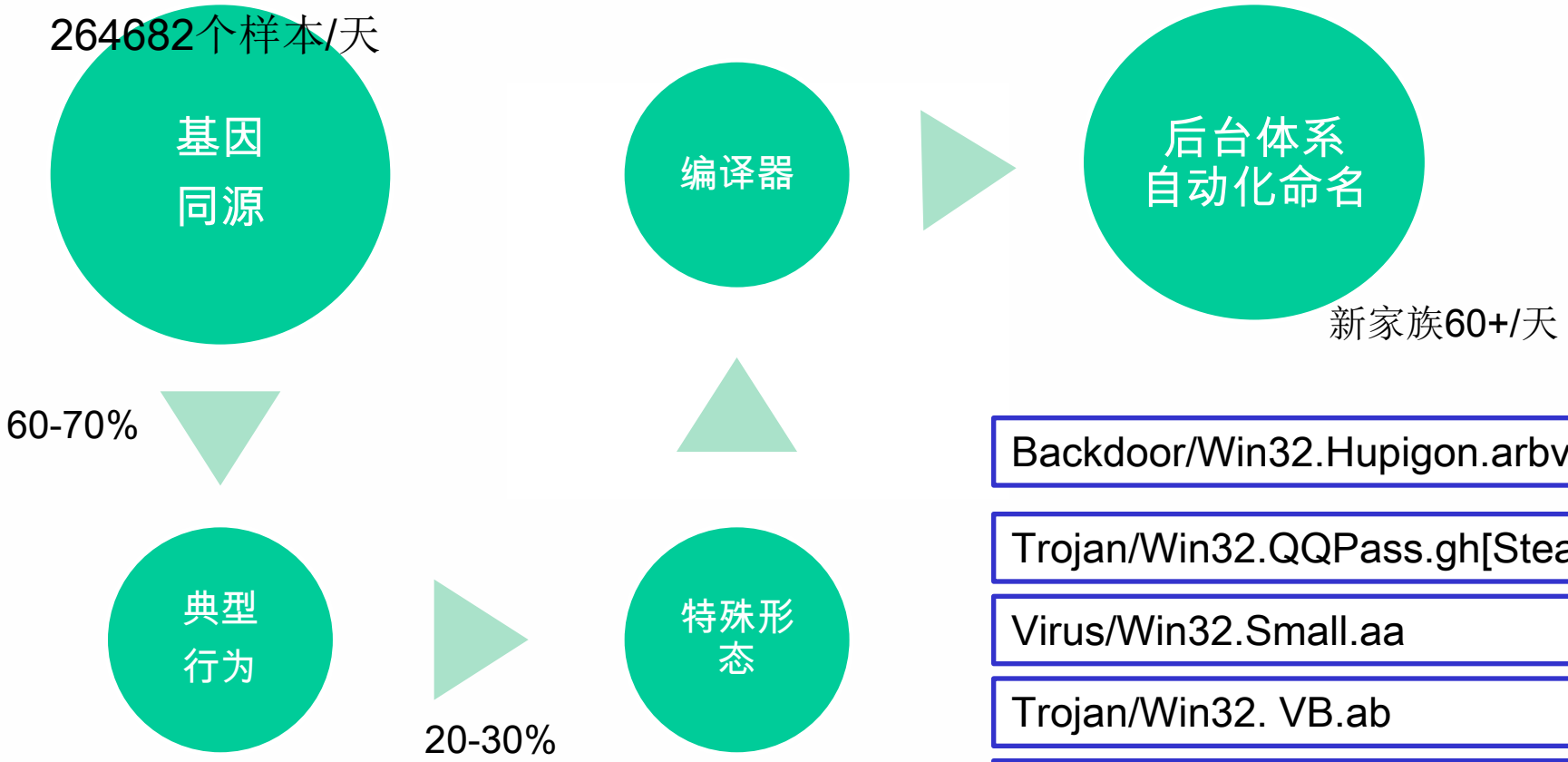
恶意代码自动化分析基础—基因提取

- 自动提取恶意代码的“基因”
  - ✓ 短代码片段
  - ✓ 典型入口特性（预处理后）
  - ✓ 导入/导出表
  - ✓ 互斥量
  - ✓ 典型字符串
- 实际达到100多维的特征向量





多层次自动化命名机制



Backdoor/Win32.Hupigon.arbv

Trojan/Win32.QQPass.gh[Stealer]

Virus/Win32.Small.aa

Trojan/Win32. VB.ab

Trojan/Win32. Genome.ak

人工同源分析

- 恶意代码自动分析和命名流程实际上是一个分类过程
- 安天对于同源分析方法的理解：

  计算机病毒分析工程师通过判断新病毒与已有病毒是否在编写上存在相互借鉴、衍生、复用等关系，找到新病毒的某些线索，最终确定两个病毒是否具有同源性。主要考查病毒样本的以下几个方面：

  - 模块结构相似性
  - 编译器架构相似性
  - 关键功能实现相似性
  - 数据结构相似性
  - 病毒作者编码心理特点
- 缺少自动化，依靠人工分析：

提纲

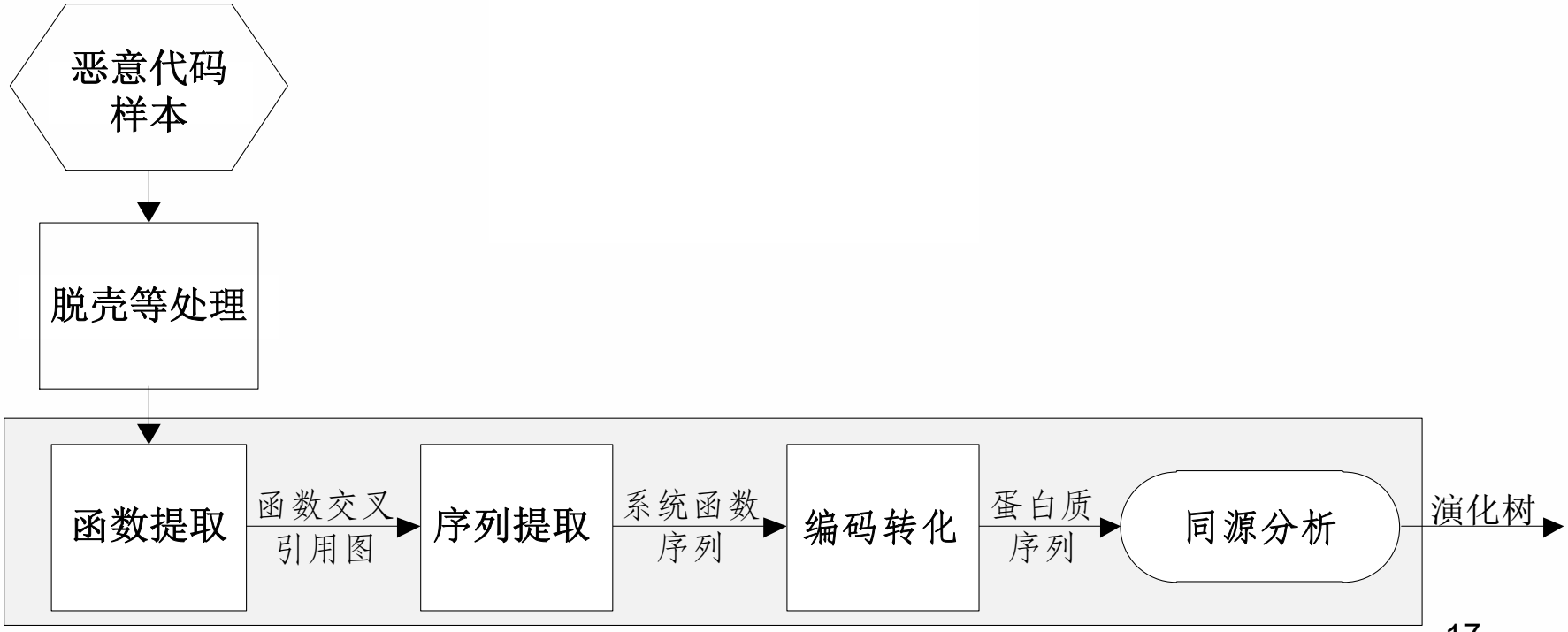
- 前言
- 产业界的工作
- 学术界的研究
- 反思：恶意代码同源分析方法
- 结束语

学术界的工作

- 借鉴生物学同源分析方法—利用相似性构建同源
  - A和B相比C更相似，意味着A和B可能有一个更近的祖先
- 基于相似性构造家族树的基本步骤
  - 第一，相似性计算函数
  - 第二，基于相似性构造家族树

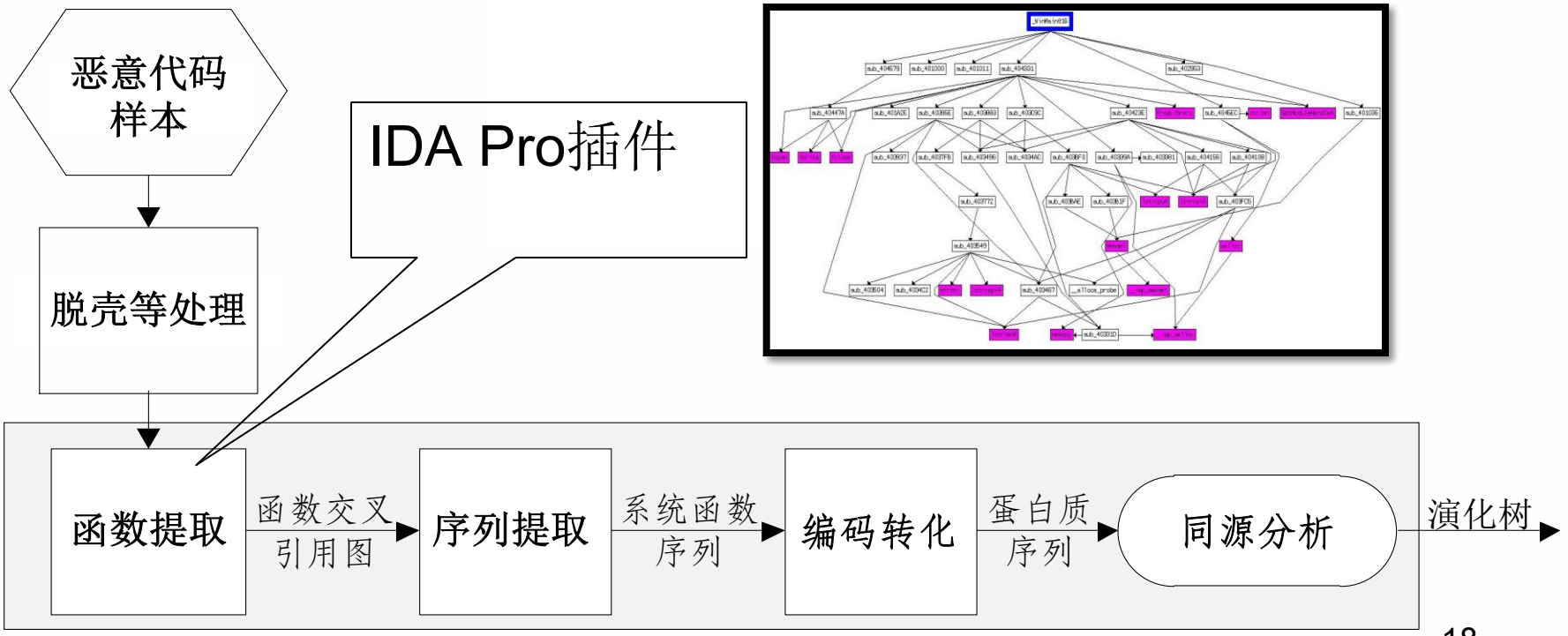
恶意代码家族树构造-例1：函数序列相似

- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年



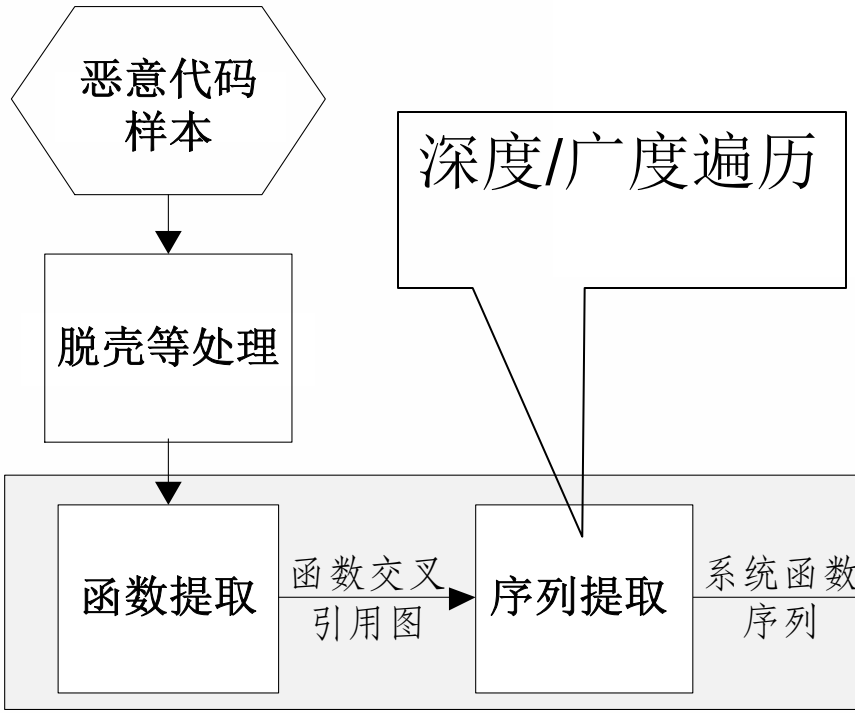
恶意代码家族树构造-例1：函数序列相似

- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年



恶意代码家族树构造-例1：函数序列相似

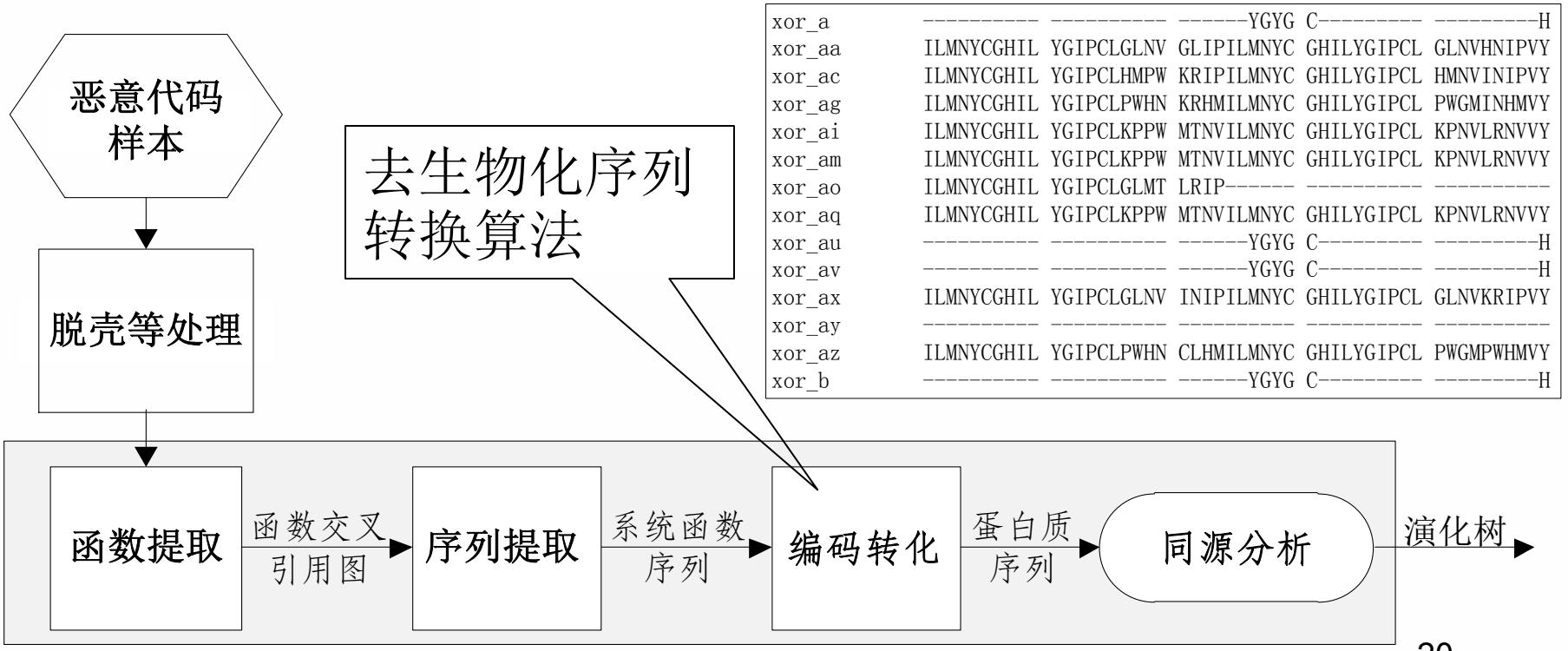
- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年




```
<?xml version="1.0" encoding="GB2312" ?>
<header>
 <LIST>
  <FILE name="Krap.ar_d5420f875f59313732ded111c750b9f9.dat">
   <FUN name="start">
    <CALL>SetFileTime sub_40119B</CALL>
   </FUN>
   <FUN name="sub_40119B">
    <CALL>SetConsoleCursorMode sub_40126C</CALL>
   </FUN>
   <FUN name="sub_40126C">
    <CALL>GetSystemDefaultUILanguage ExitThread _lcreat</CALL>
   </FUN>
  </FILE>
 </LIST>
</header>
```

恶意代码家族树构造-例1：函数序列相似

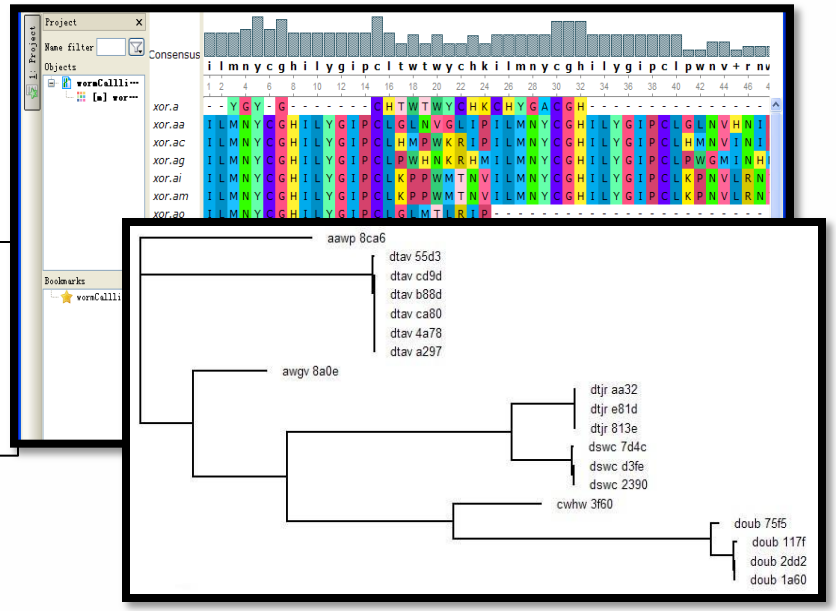
- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年




```
xor_a     ---------- ---------- ------YGYG C--------- ---------H
xor_aa    ILMNYCGHIL YGIPCLGLNV GLIPILMNYC GHILYGIPCL GLNVHNIPVY
xor_ac    ILMNYCGHIL YGIPCLHMPW KRIPILMNYC GHILYGIPCL HMNVINIPVY
xor_ag    ILMNYCGHIL YGIPCLPWHN KRHMILMNYC GHILYGIPCL PWGMINHMVY
xor_ai    ILMNYCGHIL YGIPCLKPPW MTNVILMNYC GHILYGIPCL KPNVLRNVVY
xor_am    ILMNYCGHIL YGIPCLKPPW MTNVILMNYC GHILYGIPCL KPNVLRNVVY
xor_ao    ILMNYCGHIL YGIPCLGLMT LRIP------ ---------- ----------
xor_aq    ILMNYCGHIL YGIPCLKPPW MTNVILMNYC GHILYGIPCL KPNVLRNVVY
xor_au    ---------- ---------- ------YGYG C--------- ---------H
xor_av    ---------- ---------- ------YGYG C--------- ---------H
xor_ax    ILMNYCGHIL YGIPCLGLNV INIPILMNYC GHILYGIPCL GLNVKRIPVY
xor_ay    ---------- ---------- ---------- ---------- ----------
xor_az    ILMNYCGHIL YGIPCLPWHN CLHMILMNYC GHILYGIPCL PWGMPWHMVY
xor_b     ---------- ---------- ------YGYG C--------- ---------H
```

恶意代码家族树构造-例1：函数序列相似

- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年



恶意代码样本

脱壳等处理

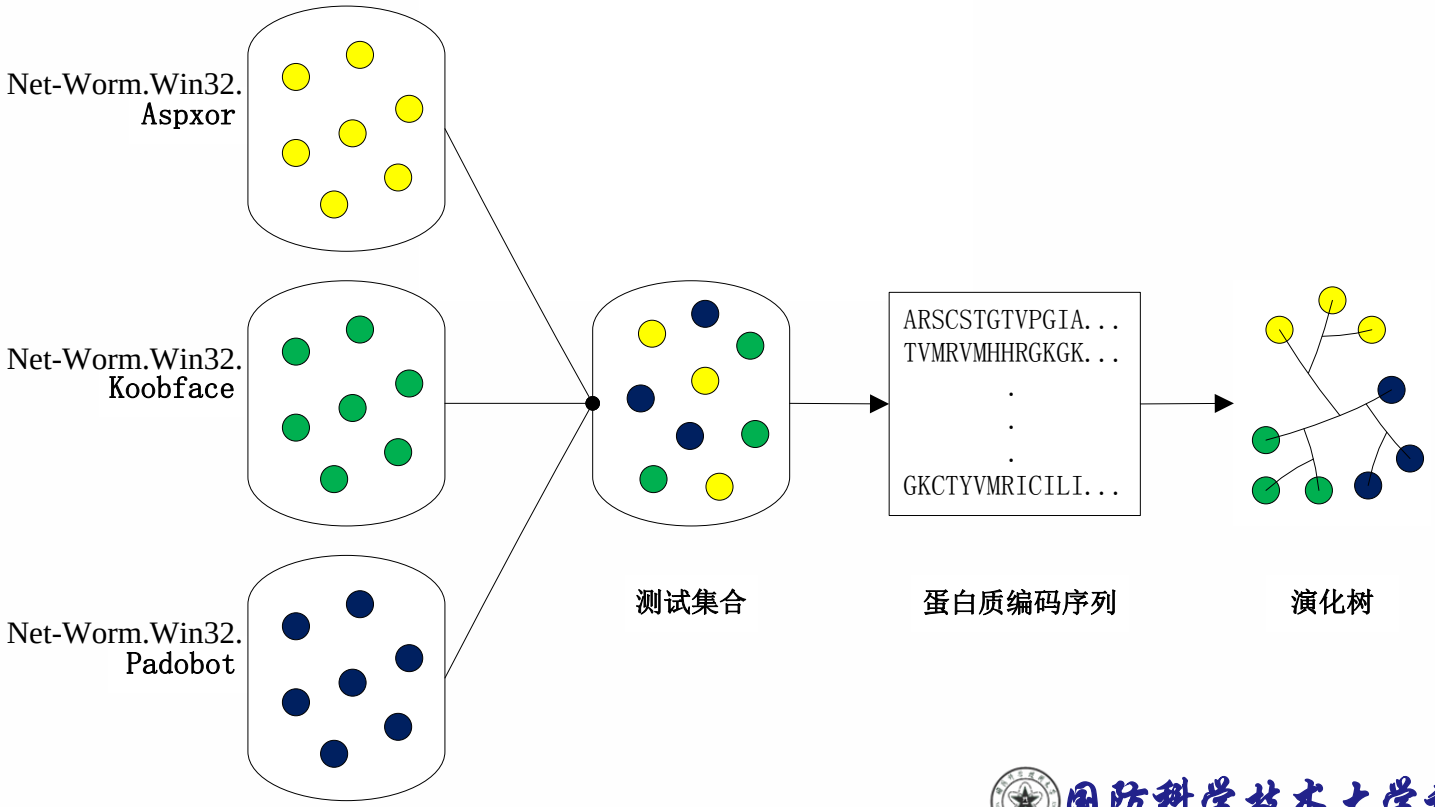
序列比对：Unipro UGENE

家族树生成：PHYLIP

函数提取 → 函数交叉引用图 → 序列提取 → 系统函数序列 → 编码转化 → 蛋白质序列 → 同源分析 → 演化树

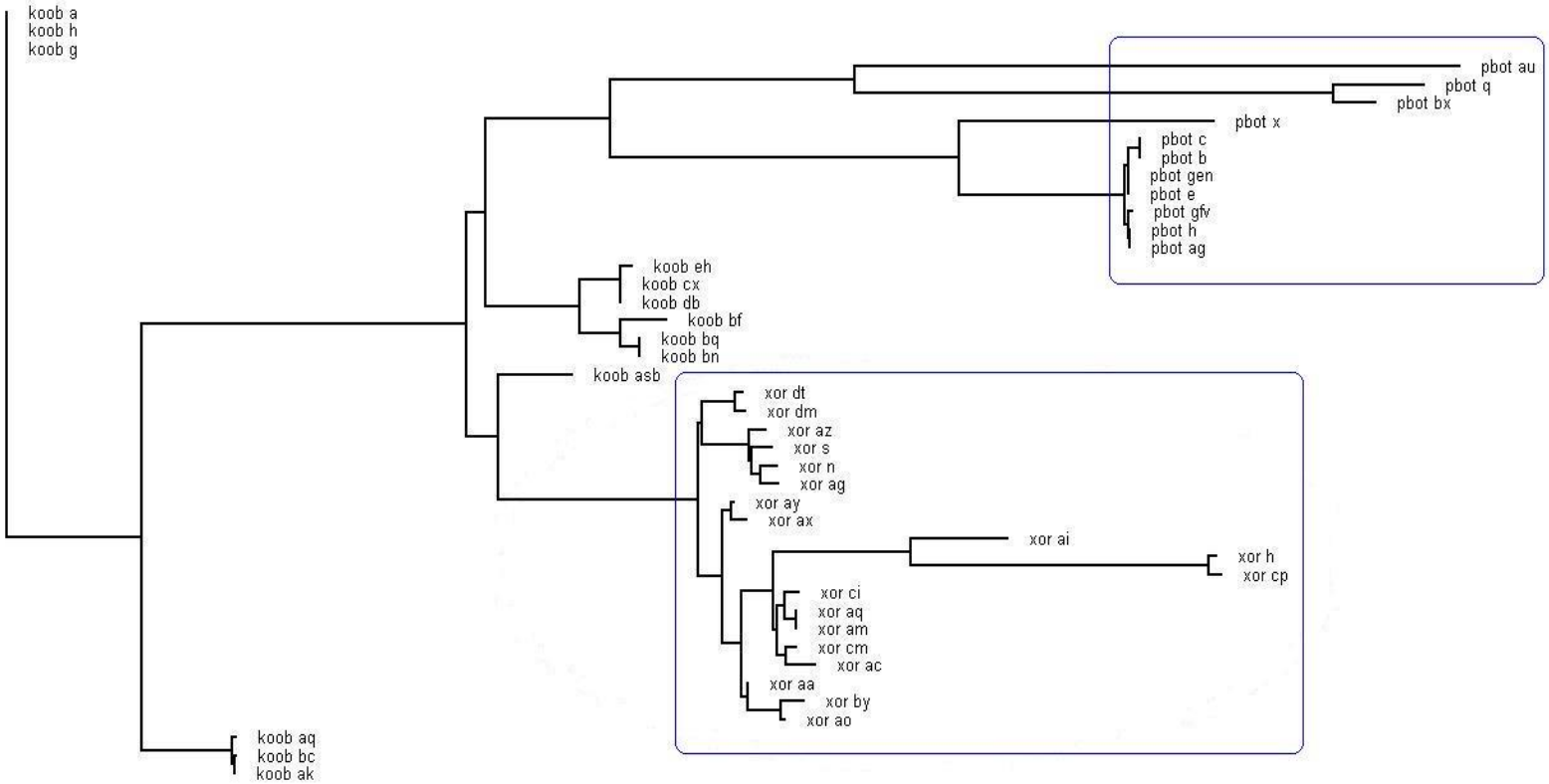
恶意代码家族树构造-例1：函数序列相似

- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年
- 实验验证

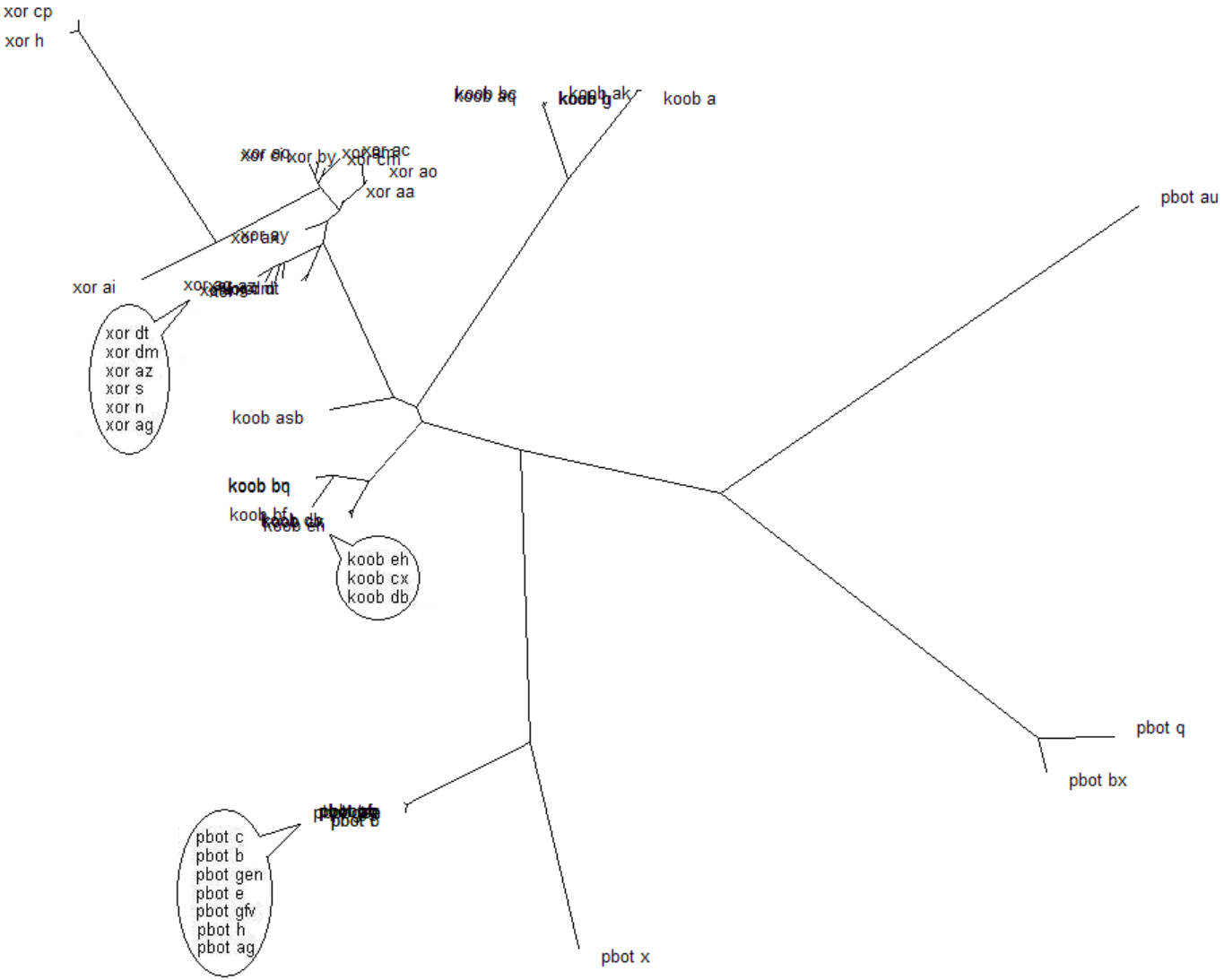


恶意代码家族树构造-例1：函数序列相似

- 《A Malware Homologous Analysis Method Based on Sequence of System Function》 2012年
- 实验验证



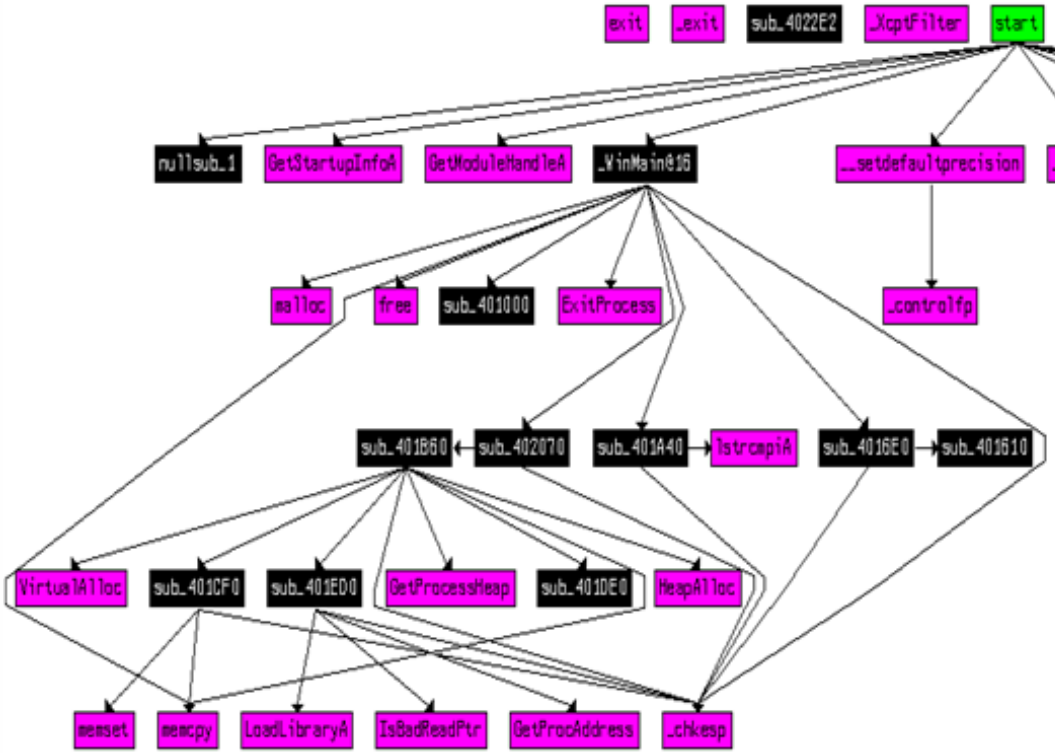
恶意代码家族树构造-例1：函数序列相似



Sequence of

恶意代码家族树构造-例2：图相似

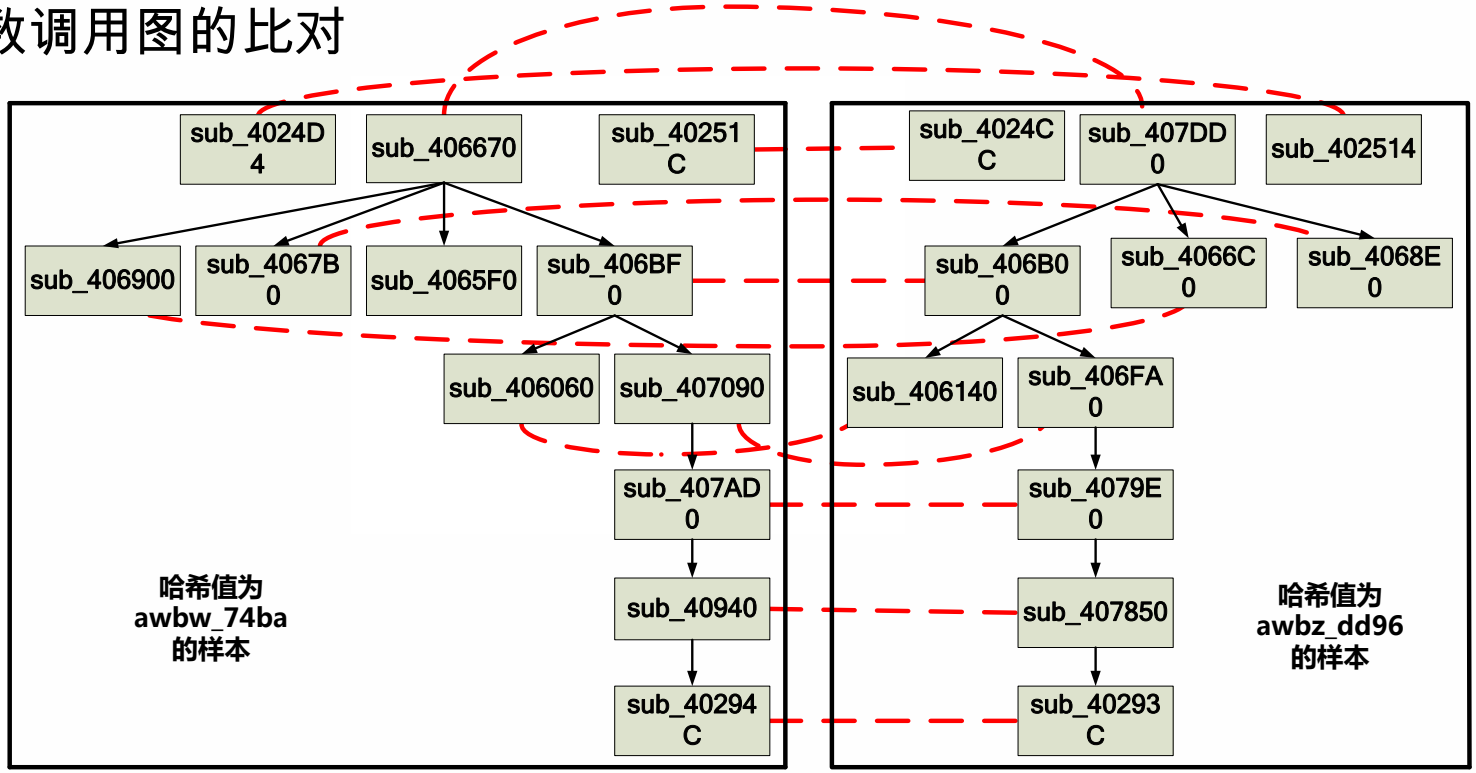
- 函数调用图



恶意代码样本Trojan-Downloader.Win32.Genome.cfzk_66Ac08ff710af7240c60fdee9f27f70e的函数调用图

恶意代码家族树构造-例2：图相似

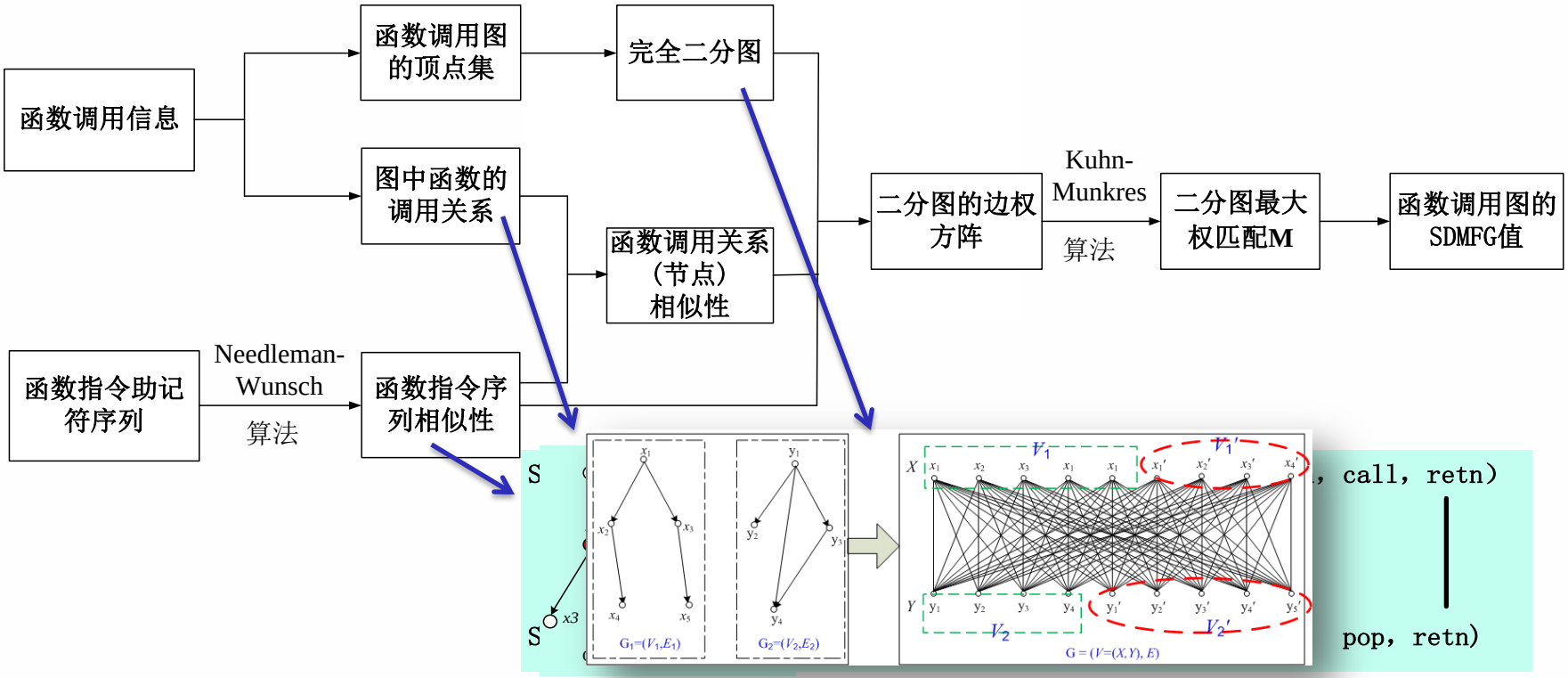
- 函数调用图的比对



- 比对的关键：
  - 图结构的相似性
  - 图内部节点（函数）相似性

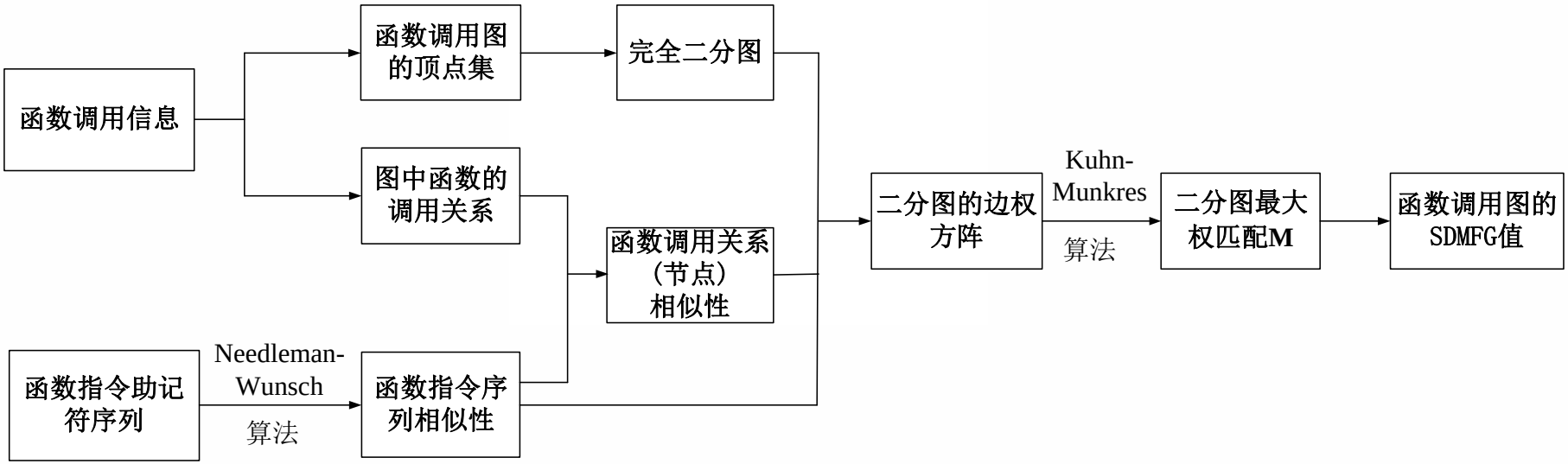
恶意代码家族树构造-例2：图相似

- 恶意代码函数调用图的相似性比对方法SDMFG



恶意代码家族树构造-例2：图相似

- 恶意代码函数调用图的相似性比对方法SDMFG



- 时间复杂度 $O(n^2l^2 + (2n)^3)$

恶意代码家族树构造-例2：图相似

- 恶意代码系统发生树构建算法
  - 得到了恶意代码样本的函数调用图相似性距离矩阵
  - 选择生物信息学中基于距离的系统发生树分析方法来分析恶意代码样本的同源关系
- 生物信息学中基于距离的系统发生分析方法主要有加权组平均法（PGMA）、非加权组平均法（UPGMA）和近邻法（neighbor-joining method，NJ）
  - PGMA是建立在沿着树的所有分支的突变率不等的假设之上
  - NJ法适用于进化距离不大，信息位点较少的短序列。
  - 由于不知道恶意代码样本的进化速率是否相同，假设是相同的，因此采用UPGMA法构建恶意代码样本的系统发生树

恶意代码家族树构造-例2：图相似

表　5个恶意代码样本的两两函数调用图的相似性距离

样本	A	B	C	D
B	2.1			
C	3.2	4.2		
D	3.3	3.2	3.6	
E	5.1	4.5	2.2	1.7

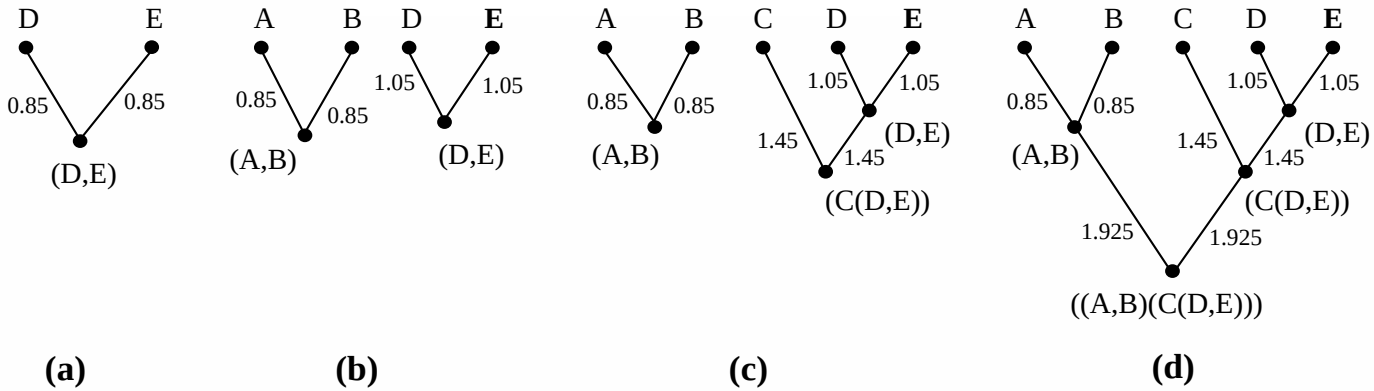
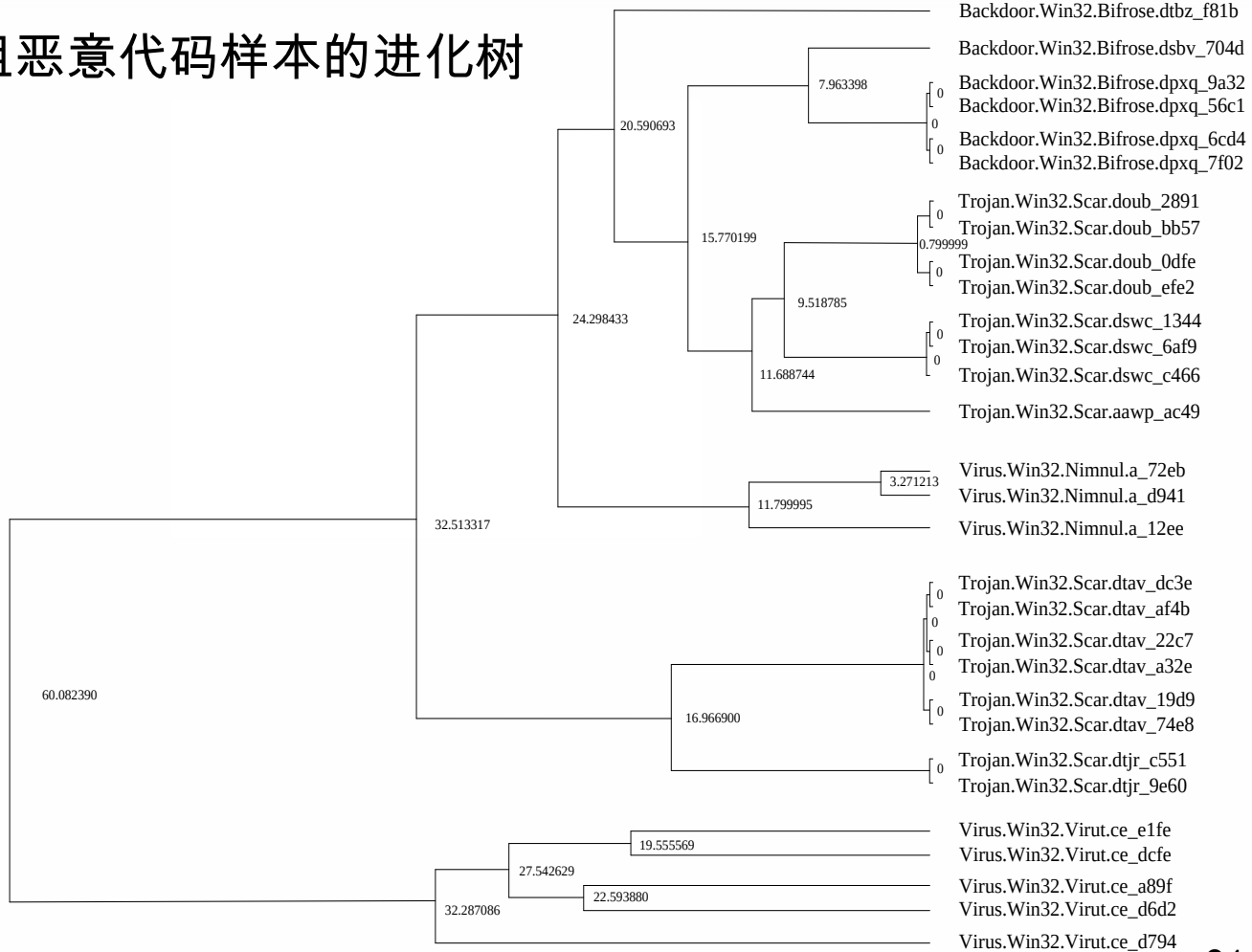


图5.1 用UPGMA法构建的5个恶意代码样本的系统发生树

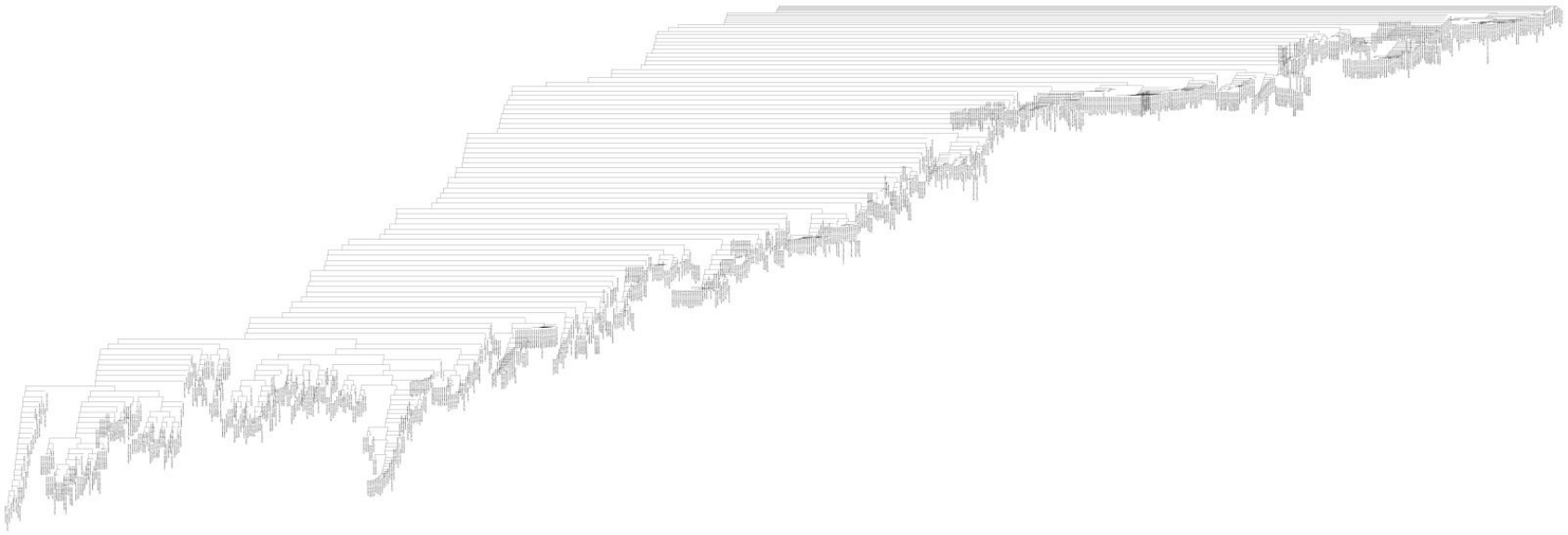
恶意代码家族树构造-例2：图相似

- 生成的一组恶意代码样本的进化树



恶意代码家族树构造-例3：图相似

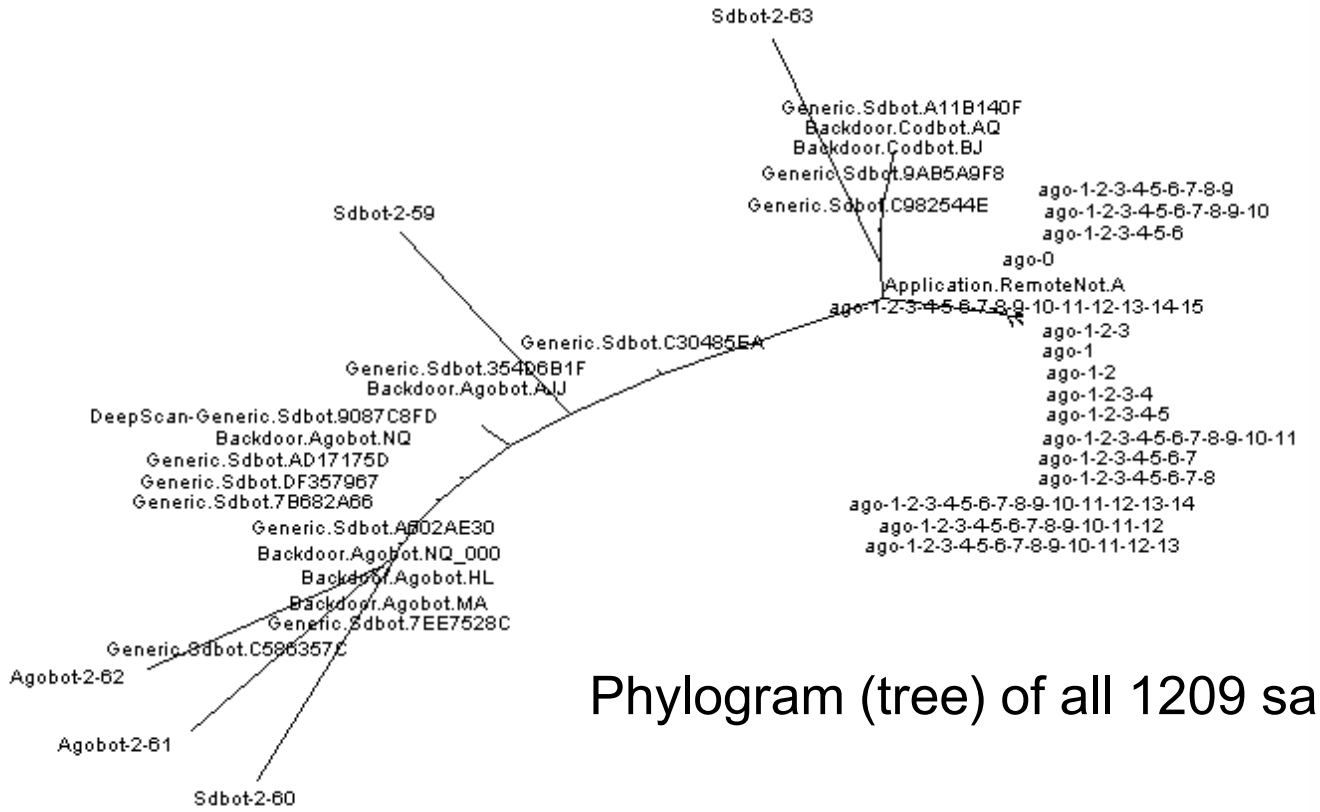
- 《Phylogenetic Comparisons of Malware》



Phylogram (tree) of all 1209 samples

恶意代码家族树构造-例3：图相似

- 《Phylogenetic Comparisons of Malware》



Phylogram (tree) of all 1209 samples

提纲

- 前言
- 学术界的工作
- 产业界的研究
- 反思：恶意代码同源分析方法
- 结束语

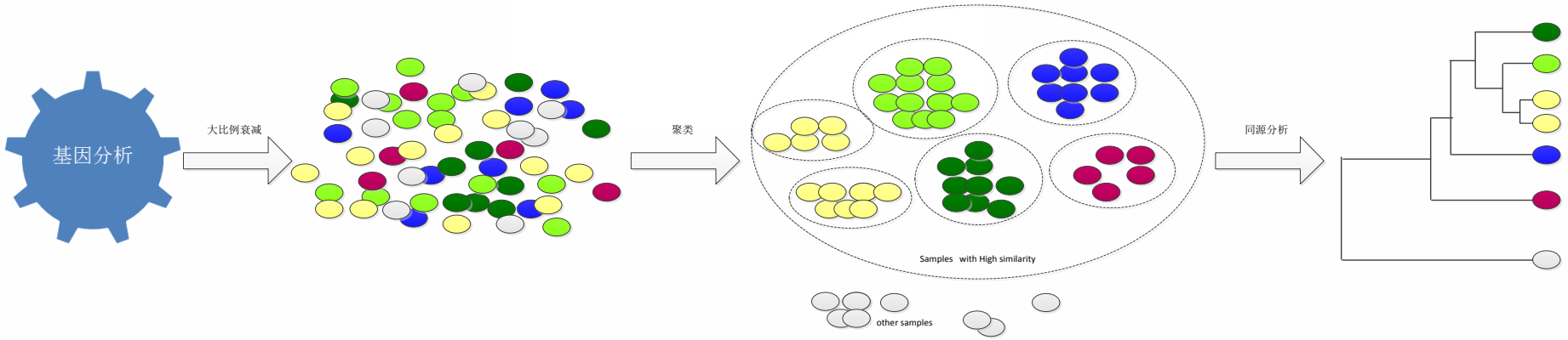
学术界和产业界的不足

- 学术界研究的不足
  - 仅依靠全局相似值构造家族树不准确
  - 树形结构不太适合描述恶意代码同源关系，特别是对于大量样本
  - 性能较差
  - 没有全面办法对抗加壳等恶意代码自保护措施
- 产业界的不足
  - 侧重检出率，命名正确与否不太重要
  - APT样本可能会被错误归类遗漏到汪洋大海中
  - 难以直面性能的挑战

相似性分析同源性的局限

- 同源分析对于反病毒厂商的意义
  - 只有不断识别新的（有价值）家族，才能确立领先地位
- 学术方法和产业界工程能否结合？

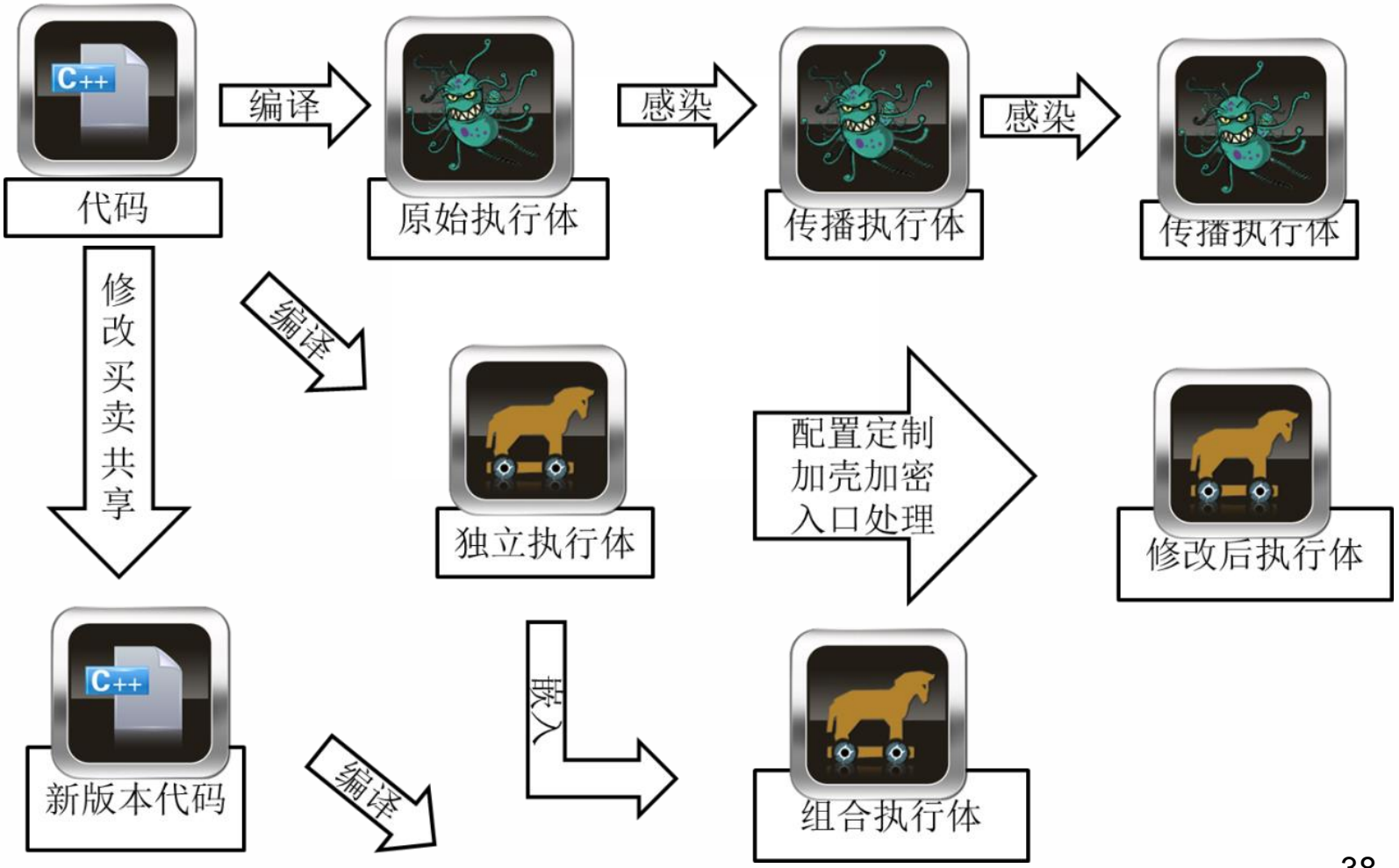
  构建新的流水线，思路：分类、聚类、同源分析



恶意代码聚类技术研究

- 活跃的研究领域
  - 《Large-Scale Malware Indexing Using Function-Call Graphs》Proceeding CCS '09
  - 《Malwise—An Effective and Efficient Classification System for Packed and Polymorphic Malware》IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 6, JUNE 2013
  - 《基于特征聚类的海量恶意代码在线自动分析模型》 通信学报2013.08
  - ……
- 聚类与同源分析的区别
  - 无时序 vs 有时序
  - 关注样本 vs 关注族群
  - 高性能 vs 不要求实时

恶意代码同源分析的模型是否合适？



恶意代码演化模型的需求

- 模型的不完善
  - 家族树只能表现基于全局相似的关联关系，无法刻画局部的关联
  - 不能刻画恶意代码完整的演化方式
- 恶意代码的各种演化关系
  - 一到多源代码演化：一个源代码演化出多个独立的版本后代
  - 交叉源代码共享：相互借鉴代码
  - 独立源代码升级：源代码更新，升级功能
  - 编译差异：不同的编译器或编译器选项
  - 二进制代码自我变形：二进制到二进制复制，代码混淆
  - 共享自保护组件：共用一个加壳器，有类似的反汇编、反调试功能
  - 独立自保护组件更新：采用独有加壳器，加壳器升级
- From 《A Transformation-based Model of Malware Derivation》

新的恶意代码演化模型

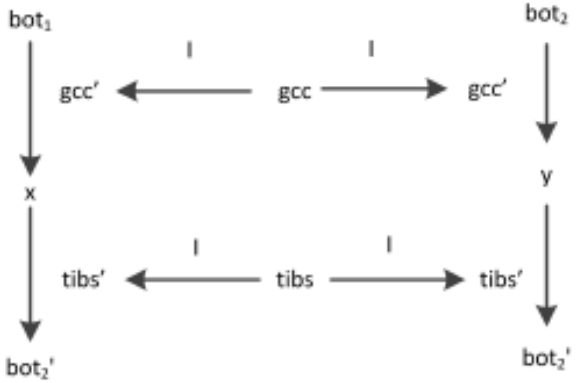
- 模型定义
  - B：二进制代码集合
  - H：高级语言代码（源代码）
  - C：代码集合C = B ∪ H
- 三类转换
  - Preserving（保留）：功能和语义不变
  - Mutating（变异）：功能变化
  - Combining（组合）：将两个代码组合

新的恶意代码演化模型

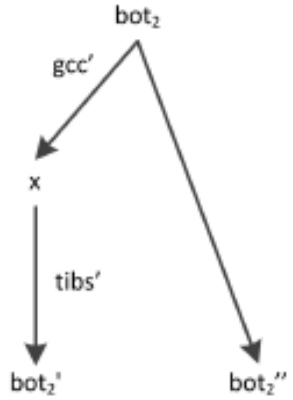
- H → H 转换
  - $Translators^{P}$：代码转换
  - $Generators^{P}$：代码框架生成
  - $Editors^{M}$：代码修改（包括人）
  - $Mutators^{M}$：代码自动（随机、计划地）变形
- H → B
  - 编译，例如gcc
- B → B
  - $Packers^{P}$：加壳
  - $Encrypters^{P}$：加密
- B × B → B
  - $Linkers^{C}$
  - $Loaders^{C}$：运行时加载
- I(C) → (C → C)

新的恶意代码演化模型

- 演化相关的概念定义
  - Derivation演化 ： $C \overset{*}{\rightarrow} C$ ， $C^n \rightarrow C$.
  - Production生产：演化但是非变异变化（Mutating）
  - Evolution进化：变异变化，产生新的恶意代码变种
- 例子



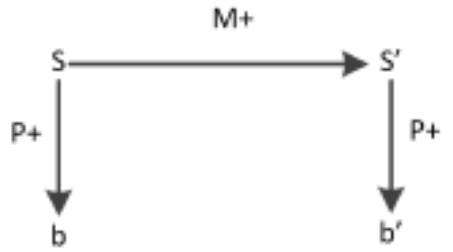
(a) common production code base



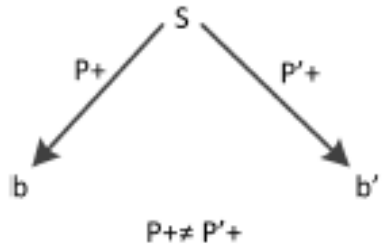
(b) common malcode base, different productions

新的恶意代码演化模型

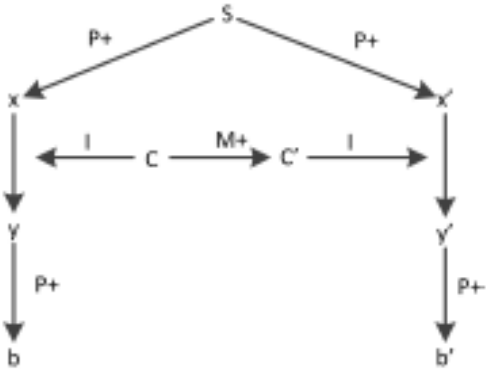
- 例子



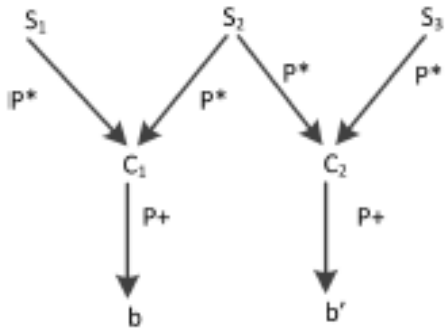
(a) descendant / ancestor



(b) polymorph



(c) e-polymorph



(d) sibling

结束语

- 同源分析研究的方向
  - 更好的模型：超越家族树（森林？），可实施
  - 更好的工程支撑：特征提取
  - 更好的分析体系：从侧重检出，到并重分类（大海捞针能力）
  - 更好的评测数据集：Open

感谢安天提供的素材和建设性的讨论！

谢谢！