РЕАЛИЗАЦИЯ АЛГОРИТМА СРАВНЕНИЯ ПОРТРЕТОВ ПРОЦЕССОВ

П.А. Лозов

lozov.peter@gmail.com

Руководитель: М.В. Баклановский

Аннотация

В данной работе исследуется выявление схожести портретов процессов – наборов часто повторяющихся последовательностей системных вызовов, совершаемых этим процессом с целью обнаружения и объединения дубликатов – различных портретов одного и того же процесса.

Введение

Одной из важнейших задач информационной безопасности является борьба с вредоносными программами. Для ее решения разрабатываются различные методы обнаружения вредоносных программ.

Одним из возможных решений является анализ поведения процессов с целью обнаружения подозрительного поведения, не свойственного данной системе. И проект CODA[1, 2] является одним из таких решений.

Проект CODA

CODA – это система противодействия вредоносным программам, разрабатываемая с 2009 года на кафедре системного программирования СПбГУ.

Неформально говоря, CODA имеет описания поведений разрешенных процессов и в течение работы системы сравнивает поведение разрешенных поведений с поведениями текущих процессов. И чем меньше поведение какого-либо процесса похоже на разрешенные, тем меньше прав предоставляется данному процессу.

Запоминает разрешенные поведения CODA следующим образом: собирает **следы** процессов – последовательности номеров системных вызовов, совершаемых каким-либо потоком процесса; и выделяет из них **портреты** процессов – наборы достаточно длинных и часто повторяющихся последовательностей номеров системных вызовов.

Однако, при создании портретов возникает проблема **дубликатов** – различных портретов одного и того же процесса. Данные портреты необходимо обнаружить и объединить в один новый портрет. Для обнаружения дубликатов необходим алгоритм сравнения портретов.

Алгоритм сравнения портретов

Так как портрет, это набор последовательностей номеров системных вызовов, можно считать его набором строк над алфавитом номеров системных вызовов. Поэтому, хорошим критерием схожести является наличие большого количества достаточно длинных подстрок в обоих портретах.

Для определения схожести портретов был разработан алгоритм (рисунок 1), который строит суффиксные деревья[3, 4] для каждого портрета (шаги 1 и 2), вычисляет веса этих деревьев, не учитывая одинаковые подстроки (шаг 3), строит дерево объединённого портрета (шаг 4), вычисляет вес общих для обоих портретов частей дерева (шаг 5) и вычисляет отношение веса общих частей и минимума из весов деревьев отдельных портретов. Имперически было определено, что если это значение выше 0.25, то сравнивались портреты одного портрета, иначе различных.

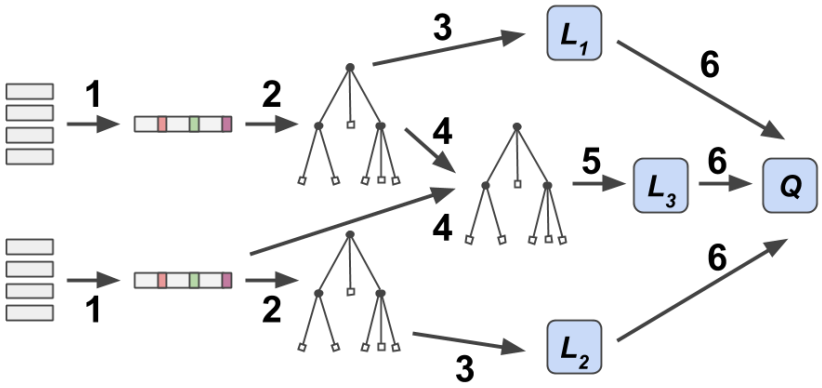

Рисунок 1: Схема алгоритма сравнения портретов

При тестировании на портретах реальных процессов данный алгоритм показал точность, равную 100%, и полноту, равную 61%.

Низкая полнота связана с тем, что различные следы одного процесса были собраны в различных эксплуатационных ситуациях.

Сравнение кластеризованных портретов

Проблему низкой полноты алгоритма сравнения портретов можно решить, разделив портреты на компоненты непохожие друг на друга. Это позволит сравнивать портреты покомпоненто, игнорируя особенности других компонент. Задачу разделения набора объектов на группы решают алгоритмы кластеризации. Для кластеризации портретов был выбран алгоритм FOREL[5], так как он показал наилучшие результаты при тестировании на портретах реальных процессов.

При сравнении двух портретов покомпонентно получается матрица схожести, которая состоит из коэффициентов схожести между кластерами портретов. Однако для сравнения портретов нужно свести данную матрицу к одному коэффициенту. Для этого было создано по 5 портретов для 30 различных процессов, все они были кластеризованы. Каждая пара портретов была сравнена между собой покомпонентно. В итоге получается набор из матриц схожести, причем для каждой известно, является ли она результатом сравнения портретов одного процесса или различных.

Далее, из этих матриц были эмпирически выбраны различные характеристики, такие как сумма всех значении, сумма всех значений с учетом размеров кластеров, максимальное, максимальное, минимальное значение и другие. С помощью алгоритма наименьших квадратов[6] были выявлены характеристики, которые наилучшим образом описывают схожесть портретов. Другими словами, характеристики, которые имеют большое значение, если это портреты одного портрета, и малое значение, если это портреты различных процессов. Из этих характеристик были исключены из рассмотрения те, которые хоть раз определили схожими портреты различных процессов. После отсеивания остались характеристики, для вычисления которых использовались только коэффициенты схожести нескольких наибольших кластеров. Итоговой коэффициентом похожести является максимум из всех отобранных характеристик, так как ни одна из них не дает большого значения, если портреты различных процессов (все такие характеристики были отсеяны).

Как и в случае с исходным алгоритмом, если этот коэффициент более 0,25, то сравнивались портреты одного портрета, иначе различных.

При тестировании на портретах реальных процессов данный алгоритм показал точность, равную 100%, и полноту, равную 89%.

Литература

1. Баклановский М.В., Ханов А.Р. CODA – новая система компьютерной

безопасности: обзор архитектуры системы // Материалы секции 22, XXXVIII Академические чтения по космонавтике. 2014. С. 649–650.
2. Баклановский М. В., Ханов А. Р. Поведенческая идентификация программ // Моделирование и анализ информационных систем. Ярославль, Том 21, Номер 6, 2014. С. 120–130.
3. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. СПб., 2003.
4. Ukkonen E. On-line construction of suffix-trees // Algorithmica. New York, 1995. P. 249–260.
5. Загоруйко, Н. Г. Прикладные методы анализа данных и знаний // Новосибирск: ИМ СО РАН. 2003. С. 38–39.
6. Линник Ю.В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений // Государственное издательство Физико-математической литературы. 1958.