ТЕСТИРОВАНИЕ АЛГОРИТМА ОБНАРУЖЕНИЯ АНОМАЛИЙ В РАБОТЕ ПРОЦЕССОВ ПРОГРАММ С ЦЕЛЬЮ ПОИСКА ВРЕДОНОСНОЙ АКТИВНОСТИ

Баклановский М.В. Университет ИТМО, mb@cit.ifmo.ru,
Комаров К.М. Санкт-Петербургский государственный университет,

Лозов П.А. Санкт-Петербургский государственный университет,

Ханов А.Р. Университет ИТМО awengar@gmail.com

Аннотация

В данной работе рассмотрен алгоритм идентификации процесса программы на основе трасс системных вызовов. На основе списков номеров системных вызовов программ строится модель. Каждая траса неизвестной программы проверяется на соответствие модели. В случае если соответствия нет, то фиксируется аномалия. Данный алгоритм применяется для поиска трасс вредоносных программ. Был проведен тест на 1800 экземпляров компьютерных вирусов, измерены уровни ложных срабатываний и обнаружения.

Введение

Обнаружение вредоносных программ по их поведению в ОС является одной из самых востребованных задач в области борьбы с ними. Современные средства проактивной защиты основаны на четких сигнатурных правилах или эвристиках.

Описание алгоритма моделирования процессов

Каждая программа после старта запускает один или несколько потоков исполнения. Каждый поток – это последовательность инструкций целевого процессора с вызовами системных библиотек. Системные библиотеки предоставляют программе вспомогательные сервисы. Однако некоторые из них позволяют взаимодействовать с самой операционной системой и изменять ее состояние: работать с файлами, процессами, сокетами, вывести данные на принтер. Все эти действия может совершить и сама программа без всяких библиотечных вызовов, обращаясь непосредственно к сервисам ядра ОС. В ОС семейства Windows сервисы ядра на платформе x86 доступны благодаря командам sysenter и int 2Eh. Был написан драйвер для перехвата всех вызовов в ядро, совершаемых всеми процессами

системы. Для каждого потока каждого процесса благодаря этому драйверу была получена траса – последовательность номеров функций, которые вызывает поток в процессе своей работы. Были решены следующие задачи:

1. На основе трасс всех потоков процесса построить модель процесса
2. Для любого набора трасс потоков процесса среди набора моделей некоторых процессов должна быть найдена та модель, которая была построена по набору трасс потоков того же процесса (даже если он был запущен заново)
3. Если имеется процесс, который не участвовал при построении набора моделей процессов, то он должен быть обнаружен в качестве аномалии.

Ранее нами был разработан алгоритм построения моделей программ на основе трасс их потоков [1]. Модель состоит из набора цепочек различной длины – термов. Алгоритм получения термов по трассе исполнения потока состоит из следующих шагов:

1. По алгоритму Каркайнена–Сандерса [2] по трассе строится суффиксный массив и множество наибольших общих префиксов всех суффиксов.
2. Среди наибольших общих префиксов выбирается множество последовательностей, каждая из которых встречается не менее N раз без наложений на другие и на саму себя и имеет длину не менее L.

Модель каждого процесса – это множество термов всех его потоков. Пусть даны трассы потоков неизвестного процесса. Имея множество заранее построенных моделей процессов, нужно найти ту из них, которая в наибольшей степени соответствует данному процессу. Для самой длинной трассы неизвестного процесса вычислим взвешенную сумму двух характеристик: количество встретившихся термов и максимальная длина встретившегося терма. Модель, которая дала максимальную оценку, будем считать наиболее близкой к данному неизвестному процессу. В работе [1] было установлено, что таким образом можно идентифицировать процесс практически со 100% точностью.

Теперь нужно установить, насколько аномально повел себя процесс во время записи трассы в рамках найденной для него модели. Для этого были подобраны следующие характеристики:

1. LOfMTh – длина наиболее длинного потока процесса;
2. PatCntMTh – количество термов, найденных в наиболее длинном потоке процесса;

3. LestPatMTh – длина самого длинного терма, найденного в самом длинном потоке процесса;
4. MidPatMd – средняя длина терма в модели, найденной для самого длинного потока процесса;
5. AllMdMiss – число вызовов, которые не вошли ни в один терм ни одной модели из набора;
6. RecMd – количество моделей, чьи термы были найдены в самом длинном потоке;
7. ThMaxPatL – наибольшая длина терма модели, подобранной для самого длинного потока, найденная в остальных потоках процесса;
8. ThPatCnt – количество термов модели, подобранной для самого длинного потока, найденных в остальных потоках процесса;
9. ThMiss – количество вызовов всех потоков процесса, не вошедших ни в один терм модели, подобранной для процесса по самому длинному потоку.
10. Отношение AllMdMiss/LOfMT – доля всех вызовов всех потоков, не вошедших в термы ни одной модели;
11. Отношение ThMiss/TotalCount, где TotalCount – суммарное количество вызовов всех потоков процесса.

Данные характеристики выбраны путем ручного сравнения моделей процессов.

Описание процесса тестирования

В течение получаса были записаны трасы 98 процессов ОС Windows 7 при активном ручном взаимодействии с ними. По каждому процессу была составлена модель – множество термов каждого потока.

Было выбрано 1800 экземпляров вредоносных программ из открытой академической базы. Из них сумели корректно запуститься 430 штук. Каждая вредоносная программа была переименована в virus.exe, чтобы различать ее процесс среди других. Каждая программа запускалась по 1 минуте, в течение которой каждые 2 секунды записывались 11 характеристик работы каждого процесса. Также были записаны 11 характеристик каждого процесса без запуска вредоносных программ в течение 10 минут взаимодействия с программами. Полученные вектора были классифицированы с помощью нейронной сети типа персептрон.

В обучении нейронной сети участвовало 1000 векторов легитимных процессов и 700 векторов вредоносных процессов. При тестировании точности классификации нейронной сетью были использованы 1137 векторов легитимных процессов и 342 вектора вредоносных процессов.

Был построен график ROC-кривой, посчитаны показатели AUC –

площадь под графиком ROC-кривой – и ACC – точность распознавания.

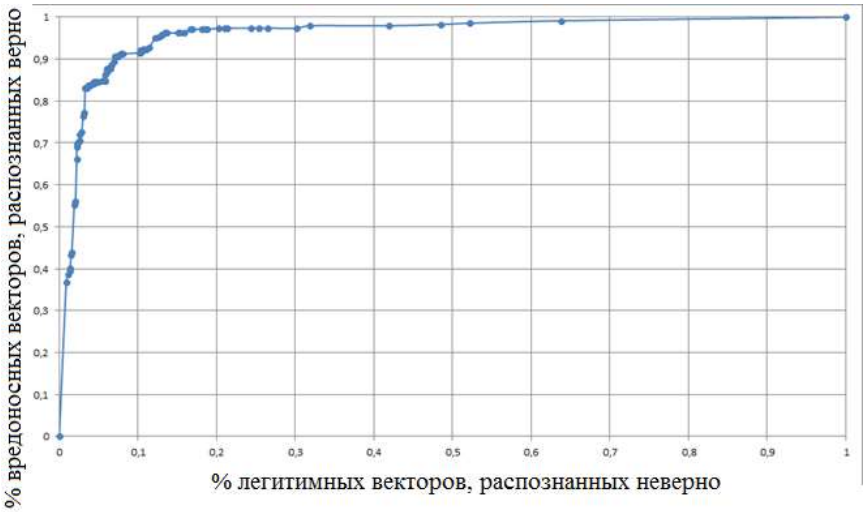

Рисунок 1. ROC-кривая нейронной сети, распознающей вредоносные и легитимные вектора.

AUC = 0.96, ACC=0,948

Около 5% всех векторов были классифицированы ошибочно. Нейронная сеть была выбрана как лучшая из нейронных сетей с произвольными параметрами по показателю ACC.

Заключение

В данной работе представлены результаты тестирования алгоритма моделирования процессов программ. Проверялась его способность отличать вредоносные программы от легитимных. Для этого было протестировано 430 вредоносных программ. В результате всего 5% периодов длиной 2 секунды были классифицированы ошибочно. Таким образом была доказана на практике эффективность алгоритма.

В дальнейшем будет расширена тестовая выборка, а также алгоритм будет доработан для того, чтобы лучше обрабатывать короткие потоки программ.

Литература

1. Баклановский М. В., Ханов А. Р. Поведенческая идентификация программ// Моделирование и анализ информационных систем. 2014. № 21:6 , С. 120–130 .
2. Karkkainen J., Sanders P., Burkhardt S. Linear work suffix array construction // Journal of the ACM, Volume 53 Issue 6, November 2006, p. 918-936