Информатика, вычислительная техника и управление

УДК 004.457

СИСТЕМА АВТОМАТИЗИРОВАННОГО ДИНАМИЧЕСКОГО АНАЛИЗА ВРЕДОНОСНЫХ ПРОГРАММ НА ОСНОВЕ ПЛАТФОРМЫ ДИНАМИЧЕСКОЙ БИНАРНОЙ ИНСТРУМЕНТАЦИИ PIN

Ф.Ф. Буканов, Р.А. Сергеев

Самарский государственный технический университет
Россия, 443100, г. Самара, ул. Молодогвардейская, 244

Современные системы автоматизированного динамического анализа вредоносных файлов представляют собой особый класс инструментов в рамках динамического подхода к исследованию функциональности образцов. Целью данной статьи является разработка системы автоматизированного динамического анализа вредоносных программ с использованием платформы динамической бинарной инструментации Pin. Кратко охарактеризован принцип работы систем анализа. Описаны известные системы и сформулированы требования, которым должна удовлетворять разрабатываемая система. Проанализированы возможности Pin, используемые в разработанной системе. Приведено описание структуры предлагаемой системы, а также ее анализирующего компонента, реализованного в виде программы на языке C++.

Ключевые слова: *автоматизированный динамический анализ, вредоносные программы, платформа Pin.*

Вредоносные программы являются одной из наиболее серьезных угроз компьютерной безопасности в настоящее время. Для изучения функциональности таких программ их подвергают анализу. При статическом подходе к анализу функциональности образцов используются дизассемблеры, различные утилиты и фреймворки для исследования свойств файла. Особенностью данного подхода является то, что теоретически возможно исследование функциональности всех частей программы. На практике же статический анализ усложняется применением упаковщиков, обфускаторов, а также специальных техник, вызывающих неправильную работу алгоритмов дизассемблирования. Поэтому возникает необходимость в применении другого подхода – динамического, который предполагает исполнение программы во время анализа. Базовым инструментом при таком подходе является отладчик.

В рамках динамического подхода существует специальный класс инструментов – системы автоматизированного динамического анализа вредоносных программ, которые позволяют быстро получить информацию об операциях, прово-

Федор Федорович Буканов (к.т.н., доц.), профессор кафедры «Электронные системы и информационная безопасность».

Роман Алексеевич Сергеев, аспирант.

димых той или иной программой во время ее исполнения. Подобные системы призваны ускорить и автоматизировать задачу анализа растущего числа вредоносных программ. При этом у специалистов появляется время на исследование тех программ, которые нуждаются в более тщательном ручном анализе.

Современные системы автоматизированного динамического анализа представляют собой большие программные комплексы, в которых анализ производится путем запуска образца на исполнение в специальной среде и последующего автоматического мониторинга его активности. В результате работы системы получается отчет, содержащий в том или ином виде информацию об операциях, выполненных вредоносной программой. В общем виде отчет можно выразить при помощи формулы

$$O = (\mathbb{f}_1, \mathbb{f}_2, \mathbb{f}_3, \dots, \mathbb{f}_i, \dots, \mathbb{f}_N). \quad (1)$$

Здесь O – отчет, полученный по завершении работы системы автоматизированного динамического анализа; $\mathbb{f}_i$ – элемент отчета – функция, которая в зависимости от конкретной реализации системы может представлять собой функцию уровня библиотек API, функцию библиотеки ntdll.dll, функцию ядра операционной системы и т. п.; N – количество элементов (функций) в отчете.

От того, как реализована система автоматизированного динамического анализа, будут зависеть ее различные характеристики, в частности конкретный вид функций в отчете. В настоящий момент существует множество систем автоматизированного динамического анализа.

Обзор систем автоматизированного динамического анализа вредоносных программ

Система CWSandbox [1] состоит из приложения cwsandbox.exe и библиотеки cwmonitor.dll. Процесс анализа может производиться как на физическом оборудовании, так и в виртуальной машине. В основе CWSandbox лежит перехват API функций. Такой перехват осуществляется перезаписью первых байт функций инструкцией безусловного перехода на обработчик перехвата. Во время исполнения обработчики перехвата инспектируют параметры API функций, а также информируют основной процесс о вызове функции через объекты уведомлений. Cwmonitor.dll реализует некоторую функциональность руткита для сокрытия объектов, относящихся к системе CWSandbox, от вредоносных программ. Отчет по завершении анализа генерируется в виде XML и содержит информацию о взаимодействии образца с файловой системой, с реестром, с сетью, а также с сервисами операционной системы.

Система Cuckoo Sandbox [2] представляет собой модульную систему с открытым исходным кодом, состоящую из хост-машины и гостевых машин. В состав системы входит множество модулей: модули для управления виртуальными машинами, модули обработки результатов, модули для генерации отчетов в виде файлов разных типов (например, в виде HTML, JSON) и другие. Главный компонент процесса анализа – библиотека Cuckoo Monitor. Библиотека реализует логирование вызываемых функций и соответствующих аргументов через перехват путем перезаписи кода в памяти функций. При этом есть особенности такого перехвата по сравнению с перехватом, реализуемым CWSandbox. Библиотека способна рандомизировать записываемые на местах перехвата инструкции для затруднения детектирования перехватов, а также обработчик имеет в своем составе несколько специальных блоков-трамплинов, которые предотвращают логирование функций, вызываемых в обработчике, а также правильно обрабатывают зна-

чение LastError. Отметим расширение Cuckoo Sandbox zer0m0n, которое реализует перехваты через подмену входов таблицы SSDT и установку функций обратного вызова на события реестра, т. е. данное расширение работает на уровне ядра.

Система Ether [3], как и система DRAKVUF [4], задействует в процессе анализа вредоносных файлов возможности, предоставляемые расширением аппаратной виртуализации процессора. Обе системы построены на основе Xen. Ether реализует мониторинг системных вызовов путем указания заведомо не представленного в физической памяти виртуальной машины адреса в регистре MSR SYSENTER_EIP_MSR. В случае использования для системного вызова прерывания INT2Eh механизм мониторинга аналогичен. DRAKVUF осуществляет прямой перехват внутренних функций ядра через инъекцию точки останова (#BP). Скрытность систем, у которых анализ встроен в интроспекцию виртуальных машин, высока. В то же время отметим накладные расходы, возникающие при выходе из гостевой системы и передаче управления гипервизору (при возникновении событий, вызывающих VMExit).

Система TTAnalyze [5] явилась основой системы автоматизированного анализа Anubis. TTAnalyze основан на эмуляторе с открытым исходным кодом Qemu. Данный эмулятор модифицирован таким образом, чтобы вызывались процедуры обратного вызова прежде, чем случится трансляция очередного базового блока. TTAnalyze проводит анализ путем мониторинга вызовов API функций, а также вызовов системных сервисов. Мониторинг вызовов функций происходит путем сравнения текущего значения указателя инструкций виртуального процессора со стартовыми адресами всех интересующих функций. Такое сравнение использует тот факт, что стартовый адрес функции всегда соответствует первой инструкции в блоке трансляции. Таким образом, анализ происходит в функциях обратного вызова, находящихся за пределами виртуальной среды, что повышает скрытность системы.

Требования к разрабатываемой системе автоматизированного динамического анализа вредоносных программ

Сформулируем требования, которым должна удовлетворять разрабатываемая система автоматизированного динамического анализа вредоносных файлов.

Во-первых, компонент анализа должен работать в операционной системе на уровне режима пользователя. Причиной такого требования являются те задачи, которые ставятся перед системой анализа, – она должна в первую очередь анализировать активность вредоносных приложений на уровне пользователя. Системы типа DRAKVUF, работающие на уровне ядра, больше подходят, например, для анализа руткит-частей вредоносных программ. Более того, реализация и развертывание таких систем представляются достаточно трудными.

Во-вторых, система должна обеспечивать достаточный уровень скрытности. Системы типа CWSandox, реализующие перехваты на уровне пользователя, выдают свое присутствие наличием внедренных в код перехватываемых функций известных инструкций изменения потока управления.

В-третьих, система должна по результатам своей работы выдавать достаточно полный отчет. Идеальным случаем было бы попадание в отчет функций приложения уровня API. Однако такое решение не подходит для вредоносных программ, вызывающих напрямую сервисы системы через библиотеку ntdll.dll. Поэтому потребуем, чтобы в отчет попадали функции уровня системных вызовов

(уровень библиотеки ntdll.dll).

Наконец, система должна предусматривать возможность написания инструментов для низкоуровневого анализа. Система Cuckoo Sandbox, несмотря на широкие возможности по расширению функциональности, не приспособлена к комплексному анализу на уровне инструкций процессора.

Ни одна из рассмотренных систем в полной мере не удовлетворяет сформулированным требованиям.

Анализ возможностей платформы Pin

Pin – это платформа для создания инструментов анализа бинарного кода. Такие инструменты носят название pin tool и представляют собой библиотеки, которые загружаются в адресное пространство процесса анализируемой программы. В своей основе Pin реализует динамическую инструментацию бинарного кода [6]. Существует множество pin tools, которые используются для решения широкого круга задач по отладке памяти, анализу производительности, а также задач, связанных с безопасностью, например поиска потенциально уязвимых функций в бинарном коде.

Представляют интерес возможности платформы Pin по контролю за системными вызовами, производимыми программой, находящейся под анализом. Внутренне обработка системных вызовов при работе Pin происходит следующим образом. Pin детектирует инструкции системного вызова при генерации трасс в кэше кода. В случае архитектуры IA-32 такими инструкциями являются sysenter и int 2E, а в случае архитектуры Intel64 – syscall. Исполнение системных вызовов происходит в виртуальной машине, а не в кэше кода, для чего происходит вставка инструкции прыжка в виртуальную машину на место инструкции системного вызова. Специальная часть – эмулятор системных вызовов – выполняет все системные вызовы, которые могут повлиять на состояние виртуальной машины (например, отображение в память, завершение процесса, треда и т. д.), а остальные системные вызовы перенаправляются в другую специальную часть Pin – системный шлюз, в задачи которого входит прозрачное исполнение системных вызовов и обработка возможных прерываний прерываемых системных вызовов. Важным здесь является тот факт, что во время работы эмулятора системных вызовов происходит уведомление pin tool о событии системного вызова. Данное уведомление может поступить в pin tool как до исполнения системного вызова, так по завершении исполнения.

Pin предоставляет набор API функций для регистрации функций, получающих управление до/после системного вызова, получения номера системного вызова, аргументов системного вызова, возвращаемого значения, номера ошибки, а также установки аргументов системного вызова, его номера. С таким набором API появляется возможность создания эффективных инструментов анализа программ на уровне системных вызовов.

Предлагаемая система автоматизированного динамического анализа вредоносных программ

Архитектурно разработанная система представляет собой совокупность следующих взаимосвязанных компонентов: среда виртуализации, операционная система, платформа Pin и компонент анализа, представляющий собой pin tool.

В качестве среды для запуска операционной системы используется программный продукт по виртуализации VirtualBox. Данный программный продукт

позволяет делать снимки состояния системы. Вследствие этого появляется возможность быстро восстановить состояние системы до начала анализа, а значит, до возможного инфицирования системы.

Операционная система, на которой запускается образец для анализа, – Windows OS. Так как платформа динамической бинарной инструментации Pin способна работать на множестве версий данной операционной системы (XP, 7 и т. д.), то разработанный pin tool как главный компонент представляемой системы анализа должен также запускаться на разных версиях Windows OS.

Компонентом анализа, как уже упоминалось, является специальный pin tool. Данный pin tool реализован в виде программы на языке C++. Данный инструмент вначале производит разбор таблицы экспорта библиотеки ntdll с целью определить соответствие между номером системного вызова, закрепленного за функцией в ntdll, и ее экспортируемым символом. Далее pin tool регистрирует функции обратного вызова на события, связанные с системными вызовами анализируемой программы. При возникновении системного вызова управление передается на функцию-обработчик, где происходит анализ номера системного вызова, а по нему определяется имя вызываемой функции, которое далее записывается в файл отчета. Таким образом, элементом отчета $\mathbb{f}_i$ по формуле (1) является в данном случае имя системного вызова в виде, представленном библиотекой ntdll.dll.

На рисунке представлена структура разработанной системы автоматизированного динамического анализа вредоносных программ.

На рисунке Malware Process – процесс вредоносной программы; Pin.exe – исполнимый модуль платформы Pin; Pinvm.dll – библиотека виртуальной машины Pin; Pintool.dll – библиотека инструмента pin tool; Malware.exe и Malware.dll – основной модуль вредоносного процесса и библиотека вредоносного процесса (если имеется) соответственно; троеточием обозначены системные библиотеки и возможные дополнительные библиотеки вредоносного процесса; Ntdll.dll – библиотека системных вызовов операционной системы Windows.

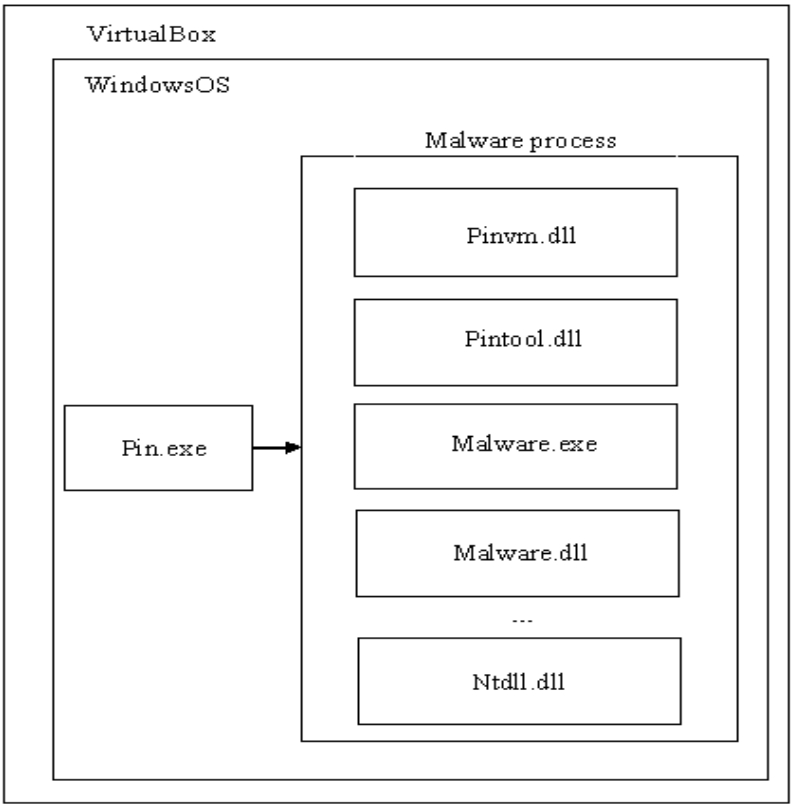
Обозначим основные особенности представляемой системы.

Скрытность системы. Компонентом анализа в предлагаемой системе является инструмент на основе платформы Pin. В настоящее время вероятность встретить образец вредоносного программного обеспечения, реализующего техники детектирования инструментирования, мала. Поэтому с этой точки зрения система имеет хороший уровень скрытности. Вопрос о скрытности среды виртуализации решается другими методами и выходит за рамки данной статьи.

Работа на уровне режима пользователя и простота реализации. Платформа Pin, как и инструменты, разработанные на ее основе, работают на уровне режима пользователя. Всю работу по перехвату системных вызовов, регистрации функций обратного вызова и прочее берет на себе платформа Pin. Здесь же можно отметить простоту развертывания в том смысле, что в предлагаемой системе не используются комплексные компоненты, работающие за пределами виртуальной среды, как, например, в Ether и DRAKVUF.

Уровень логирования системных вызовов. Решения по анализу, которые используют в своей работе перехваты функций уровня API, уязвимы к тем вредоносным программам, которые вызывают системные сервисы напрямую, с использованием библиотеки ntdll.dll. Разработанная система производит логирование системных вызовов анализируемых программ, тем самым не упуская моменты прямого обращения образцов к библиотеке ntdll.dll в обход API функций.

Здесь же отметим, что на уровне логирования системных вызовов получается достаточное представление о работе анализируемой программы.



Структура системы автоматизированного динамического анализа на основе Pin

Возможности для написания собственных инструментов низкоуровневого анализа. Платформа Pin, будучи фреймворком динамической бинарной инструментации, имеет широкие возможности для разработки средств по анализу бинарного кода. С точки зрения анализа вредоносных программ представляется актуальным внедрение в предлагаемую систему таких техник, как, например, динамическое символическое исполнение.

Заключение

В результате разработана система автоматизированного динамического анализа вредоносных программ на основе платформы Pin. Основной компонент анализа реализован в виде pin tool. Предлагаемая система производит логирование системных вызовов анализируемых программ, работает на уровне режима пользователя, отличается достаточным уровнем скрытности, а также поддерживает возможности по написанию дополнительных инструментов низкоуровневого анализа, тем самым удовлетворяя обозначенным выше требованиям.

В настоящее время использование автоматизированных динамических систем является одним из продвинутых способов быстрого изучения функциональности вредоносных программ. Разработанная система может использоваться для эффективного в смысле сформулированных требований динамического анализа вредоносных программ. Дальнейшим улучшением представленной системы является интегрирование низкоуровневых механизмов для обхода множества путей анализируемой программы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Toward Automated Dynamic Malware Analysis Using CWSandbox / Carsten Willems, Thorsten Holz, Felix Freiling // IEEE Security and Privacy. – 2007. – Volume 5 Issue 2. – P. 32-39.
2. Cuckoo Sandbox – open source automated malware analysis // Blackhat. URL: https://media.blackhat.com/us-13/US-13-Bremer-Mo-Malware-Mo-Problems-Cuckoo-Sandbox-WP.pdf/ (дата обращения: 10.01.2016).
3. Dinaburg Artem, Royal Paul, Sharif Monirul, Lee Wenke. Ether: malware analysis via hardware virtualization extensions // CCS '08 Proceedings of the 15th ACM conference on Computer and communication security. – ACM New York, NY, USA, 2008. – P. 51-62.
4. Lengyel Tamas K., Maresca Steve, Payne Brayan D., Webster George D., Vogl Sebastian, Kiayias Aggelos. Scalability, fidelity and stealth in the DRAKVUF dynamic malware analysis system // ACSAC '14 Proceedings of the 30th Annual Computer Security Applications Conference. – ACM New York, NY, USA, 2014. – P. 386-395.
5. TTAnalyze: A Tool for Analyzing Malware / Ulrich Bayer, Christopher Kruegel, Engin Kirda // Ucsb. URL: https://www.cs.ucsb.edu/~chris/research/doc/eicar06_ttanalyze.pdf/ (дата обращения: 13.01.2016).
6. Dynamic Instrumentation with Pin / Robert Cohn // Rice. URL: http://cscads.rice.edu/workshops /july2007/perf-slides-07/Cohn-Pin.pdf/ (дата обращения: 14.01.2016).

Статья поступила в редакцию 20 марта 2016 г.

AUTOMATED DYNAMIC MALWARE-ANALYSIS SYSTEM ON THE BASIS OF THE DYNAMIC BINARY INSTRUMENTATION PIN PLATFORM

F.F. Bukanov, R.A. Sergeev

Samara State Technical University
244, Molodogvardeiskaya st., Samara, 443100, Russian Federation

Modern automated dynamic malware-analysis systems are a special type of tools in the area of dynamic approach of the samples functionality study. The purpose of this paper is to develop the automated dynamic malware-analysis system using the dynamic binary instrumentation Pin platform. The principle of operation of analysis systems is characterized briefly. The known systems are described and requirements for the developed system are formulated. Useful capabilities of Pin for the developed system are analyzed. The structure of the proposed system and analysis component in the form of C++ program are shown.

Keywords: *automated dynamic analysis, malicious software, platform Pin.*

Fedor F. Bukanov (Ph.D. (Techn.)), Professor.
Roman A. Sergeev, Postgraduate Student.