УДК 004.056.53
DOI: 10.17586/0021-3454- 2017-60-10-940-944

МЕТОД РЕАЛИЗАЦИИ „ПЕСОЧНИЦЫ“ ДЛЯ ПОТЕНЦИАЛЬНО ОПАСНЫХ ПРОГРАММ

К. А. Щеглов[1], А. Ю. Щеглов[2]

[1]*Научно-производственное предприятие „Информационные технологии в бизнесе“, 194044, Санкт-Петербург, Россия*
[2]*Университет ИТМО, 197101, Санкт-Петербург, Россия*
E-mail: info@npp-itb.spb.ru

Предложен метод реализации „песочницы“ — ограниченной среды в компьютерной системе, предназначенной для исполнения потенциально опасных программ, в том числе для запуска непротестированного кода, непроверенного кода из неизвестных источников, а также для запуска и обнаружения вирусов. Основу метода составляет реализация перенаправления запросов доступа, основанного на использовании запатентованного авторами технического решения. Излагаемый метод создания песочницы, характеризуемой минимальным влиянием на загрузку вычислительного ресурса и простотой администрирования, в работе проиллюстрирован примером практической реализации. Рассматриваемое техническое решение апробировано в коммерческой системе защиты информации.

Ключевые слова: *потенциально опасная программа, песочница, метод защиты, перенаправление запроса доступа, система защиты информации*

Под песочницей (англ. Sandbox) принято понимать ограниченную среду в компьютерной системе, предназначенную для исполнения потенциально опасных программ без их доступа к системным объектам операционной системы и иных приложений. Песочницы часто используют для запуска непротестированного кода, непроверенного кода из неизвестных источников, а также для запуска и обнаружения вирусов.

Именно реализация песочницы составляет основу современных средств антивирусной защиты, поскольку сигнатурный анализ, как известно, эффективной защиты априори обеспечить не может (его следует использовать в качестве дополнительного инструмента для анализа программ, в поведении которых уже выявлены какие-либо аномалии с точки зрения безопасности).

Однако в общем случае песочница должна использоваться и в отношении программ, легально полученных из известных источников, при условии, что высока вероятность наделения этих программ вредоносными свойствами. Наделить программы вредоносными свойствами возможно как в результате эксплуатации выявляемых в них уязвимостей реализации (ошибок программирования), так и в результате эксплуатации штатных возможностей некоторых программ, состоящей в чтении и исполнении командных файлов (скриптов и т.д.) [1].

Основные подходы к реализации песочниц рассмотрены, например, в [2, 3]. Основу механизма защиты составляет использование разграничительной политики доступа потенциально опасных программ к защищаемым системным объектам (файловым и объектам реестра операционной системы Microsoft Windows). Для предотвращения доступа таких программ (любой запрос доступа определяется двумя сущностями: пользователем — учетной записью — и процессом), реализовать песочницу можно, задав соответствующие права доступа к системным объектам либо для пользователей, либо для процессов. (Под *процессом* здесь и далее понимаем запущенную — исполняемую в системе — программу). Таким образом, можно ассоциировать песочницу с создаваемой для нее учетной записью, под которой должны

запускаться потенциально опасные программы. Альтернативой служит разграничение прав доступа к системным объектам для потенциально опасных процессов (песочница при этом ассоциируется с конкретными процессами).

У обоих подходов есть общие принципиальные недостатки. Во-первых, крайне сложно корректно настроить правила доступа к системным объектам. Дело в том, что ключевым требованием к построению безопасной системы является реализация разрешительной разграничительной политики доступа — все, что в явном виде не разрешено, то запрещено [1]. В отношении конкретного потенциально опасного процесса песочница будет реализована корректно, только если этому процессу предоставляются лишь необходимые для функционирования права доступа к системным объектам. Но чтобы эти права определить, сначала потенциально опасную программу (в общем случае не одну) надо запустить, и по журналу аудита ее действий в системе определить, какие системные объекты процессу необходимы.

При постройке подобной песочницы на практике разработчики антивирусных решений устанавливают „по умолчанию“ ограниченный набор запретов доступа к системным объектам, предлагая пользователю расширить этот набор самостоятельно, т.е. самим создать разграничительную политику доступа к системным объектам для потенциально опасных процессов.

Во-вторых, далеко не ко всем системным объектам можно разграничить права доступа. К таким объектам можно отнести каталоги коллективного доступа, не разделяемые между пользователями системой и приложениями (в том числе папки, предназначенные для временного хранения файлов), а запрет записи в некоторые системные объекты может привести к некорректной работе процессов.

Эти недостатки преодолеваются использованием для создания песочниц средств виртуализации. Одна, при необходимости несколько, гостевая система (одна из виртуальных машин) может использоваться в качестве песочницы. При этом, однако, появляются иные проблемы: на компьютере уже должны одновременно работать минимум три системы — гипервизор и две виртуальные, одна из которых используется как песочница. Нагрузка на вычислительный ресурс в этом случае катастрофически возрастает. На практике может потребоваться не одна песочница: например, одна для работы с сетью, вероятность осуществления атак из которой высока, другая — для установки непроверенного кода из неизвестных источников, а также для запуска и обнаружения вирусов.

Существует и иная проблема в использовании средств виртуализации для реализации песочниц. Применительно к ним появляются дополнительные угрозы безопасности, в том числе связанные с необходимостью защиты гипервизора. При этом песочница может служить источником сетевых атак на основную виртуальную машину.

Все сказанное обусловливает актуальность задачи разработки новых методов реализации песочницы.

Основу предлагаемого метода составляет контроль доступа к системным объектам, состоящий в перенаправлении запросов доступа [1, 4]. Создается копия системного объекта, доступ потенциально опасного процесса к которому требуется запретить. В разграничительной политике задаются условия перенаправления запроса доступа к копии объекта. Средство защиты перехватывает и анализирует запросы доступа к объекту, и если в правилах доступа для существующих условий указано перенаправление запроса доступа, диспетчер модифицирует исходный запрос, подставляя в него адрес соответствующей копии системного объекта. В результате анализируемый средством защиты запрос доступа относится не к исходному объекту, а к его копии.

Рассмотрим использование изложенного метода реализации песочницы. Пусть исходно на системном диске C установлены операционная система и приложения. Создадим две песочницы. Для этого заведем в системе две учетные записи User 1 и User 2.

На дисках D и E создадим копии системного диска C — в них копируются соответствующие файловые объекты (в том числе скрытые файлы). Таким образом, на диске C установлена базовая (или оригинальная) система, на D и E — виртуальные системы. В общем случае копии системного диска C можно помещать в отдельные каталоги. Число создаваемых песочниц определяется числом созданных учетных записей и копий системных объектов.

Теперь зададим правила перенаправления запросов доступа к соответствующим копиям системных объектов. Для User 1 это будет правило перенаправления запроса доступа к диску C на диск D, для User 2 — правило перенаправления запроса доступа к диску C на диск E. Реализуем соответствующую разграничительную политику доступа к файловым объектам — запретим пользователю User 1 доступ к диску E, а User 2 — к диску D. В этом случае реализуется виртуализация системы (не машин), проиллюстрированная на рис. 1: для каждой песочницы создается своя виртуальная система, при этом с базовой системой может взаимодействовать только системный пользователь.

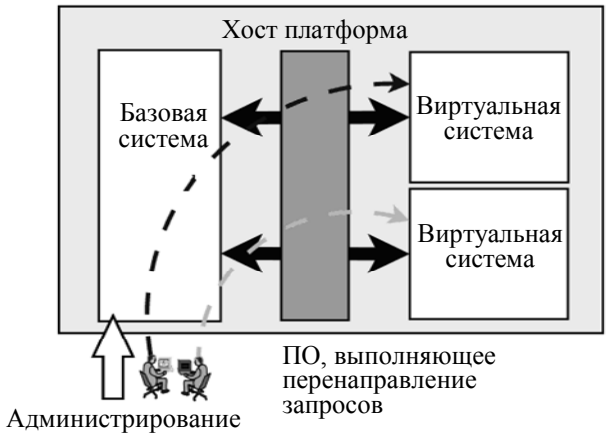

Рис. 1

Рассмотрим, как будет работать подобная система с реализованной виртуализацией. С базовой системой может взаимодействовать только системный пользователь, т.к. запросы к ней интерактивных пользователей User 1 и User 2 перенаправляются к созданным для них виртуальным системам. Если User 1 запускает с базовой системы какое-либо приложение, например браузер, реально оно будет запущено с диска D. При дальнейшей работе приложения любое обращение с правами пользователя User 1 к базовой системе будет перенаправляться к соответствующей виртуальной системе. При этом приложение, установленное на диск C, будет работать корректно, поскольку все перенаправления его запросов доступа „прозрачны“ для приложения, в частности, корректно будут сохраняться временные данные, cookies, конфигурационные файлы, кэшированная информация и т.д., но уже на диске D.

Если User 2 запустит с базовой системы какое-либо приложение, „физически“ оно будет запущено с диска E. При этом исходно установленное на диск C приложение будет работать корректно, поскольку все перенаправления его запросов доступа „прозрачны“ для приложения, все конфигурационные файлы сохраняются на диске E.

Таким образом обеспечивается полная изолированность системных средств для песочниц — атака потенциально опасного процесса, работающего в соответствующей песочнице, может быть успешно реализована только на виртуальную систему этой песочницы. Важно, что в результате виртуализации системы всегда есть доверенная операционная система (базовая), с которой осуществляется корректная загрузка, и с которой может быть восстановлена виртуальная система, поскольку к базовой системе доступ имеет исключительно системный пользователь. Как видим, сложность администрирования подобного средства защиты минимальна.

На рис. 1 представлены две песочницы: одна для работы с условно безопасными программами (под учетной записью User 1), другая — для работы с потенциально опасными

(User 2). При этом базовая система защищена, в том числе и от возможных атак со стороны условно безопасных программ. Сказанное справедливо и для реализации песочницы с целью защиты объектов реестра операционной системы.

Альтернативный способ построения — песочница ассоциируется с конкретными процессами — отличается тем, что запросы доступа в соответствующие созданные копии системных объектов перенаправляются с учетом процесса, запросившего доступ. При этом программы, обрабатываемые в песочнице, должны инсталлироваться в копию системных объектов (в виртуальной системе), созданных для этой песочницы, работа в соответствующих песочницах осуществляется автоматически для всех соответствующих процессов.

Изложенный метод реализован и апробирован в коммерческой системе защиты информации [5]. Интерфейс создания правил перенаправления запросов доступа, на примере файловых объектов, представлен на рис. 2.

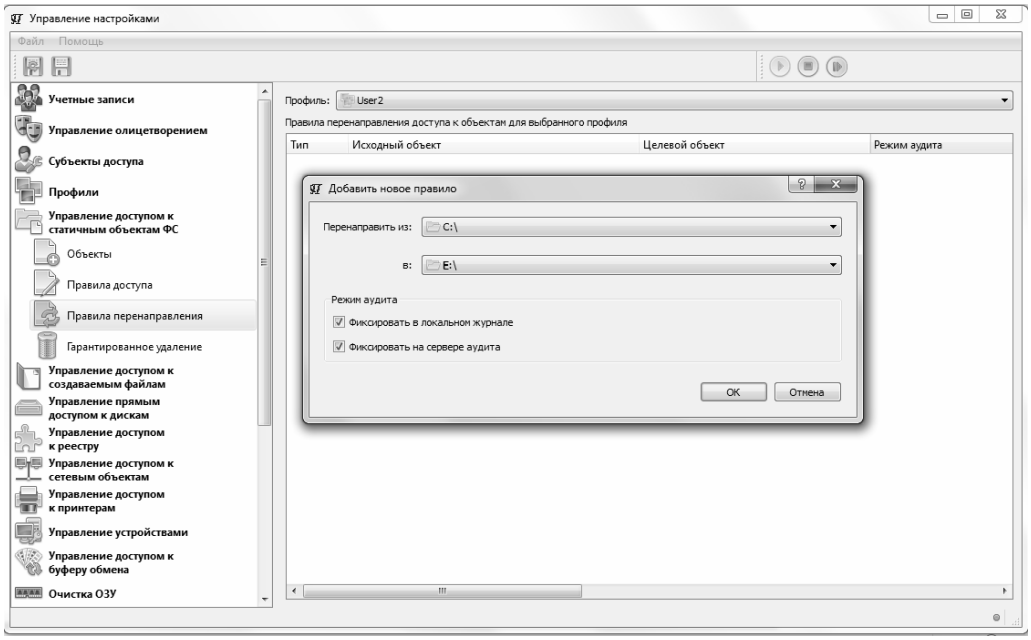

Рис.2

Субъектом доступа, для которого задаются правила перенаправления, является сущность „профиль“. В один и тот же профиль объединяются учетные записи либо полнопутевые имена исполняемых файлов программ, для которых назначением лишь одного соответствующего правила перенаправления запросов доступа создается песочница (см. рис. 2).

Отметим, что при реализации песочницы для эффективной защиты должен решаться еще ряд задач [1]: контроль прямого доступа к диску, контроль сервисов олицетворения, запрет запуска создаваемых файлов, что легко реализуется с использованием запатентованного авторами решения [6].

В заключение отметим, что рассмотренный метод кардинально упрощает проектирование системы защиты информации. За счет введения обоснованных допущений в отношении реализации угроз атак на защищенную информационную систему он позволяет решить две важнейшие задачи проектирования — определить оптимальный набор функций защиты, реализуемых системой защиты информации, и сформировать требования к эксплуатационным параметрам средств защиты, входящих в состав этой системы.

СПИСОК ЛИТЕРАТУРЫ

1. *Щеглов А. Ю., Щеглов К. А.* Анализ и проектирование защиты информационных систем. Контроль доступа к компьютерным ресурсам. Методы, модели, технические решения. СПб: Профессиональная Литература, 2017. 416 с.

2. Антивирусные песочницы. Введение [Электронный ресурс]: <http://www.securrity.ru/articles/1116-antivirusnye-pesochnicy-vvedenie.html>.

3. Виртуальные песочницы: обзор 5 решений для запуска любых программ в среде, изолированной от операционной системы [Электронный ресурс]: <http://it-news.complexdoc.ru/1071474.html>.

4. Пат. 2538918 РФ. Система переформирования объекта в запросе доступа / *А. Ю. Щеглов, К. А. Щеглов.* Опубл. 10.01.2015.

5. Свид. о рег. прогр. для ЭВМ № 2014660889. Комплексная система защиты информации „Панцирь+“ / *А. Ю. Щеглов, И. П. Павличенко, С. В. Корнетов, К. А. Щеглов*. Опубл. 20.11.2014.

6. Пат. 2524566 РФ. Система контроля доступа к файлам на основе их автоматической разметки / *А. Ю. Щеглов, К. А. Щеглов.* Опубл. 27.07.2014. Бюл. № 21.

Сведения об авторах

Константин Андреевич Щеглов — Научно-производственное предприятие „Информационные технологии в бизнесе“; исполнительный директор; E-mail: scheglov.konstantin@gmail.com

Андрей Юрьевич Щеглов — д-р техн. наук, профессор; Университет ИТМО; кафедра вычислительной техники; E-mail: info@npp-itb.spb.ru

Рекомендована кафедрой вычислительной техники

Поступила в редакцию 03.07.17 г.

Ссылка для цитирования: *Щеглов К. А., Щеглов А. Ю.* Метод реализации „песочницы“ для потенциально опасных программ // Изв. вузов. Приборостроение. 2017. Т. 60, № 10. С. 940—944.

IMPLEMENTATION OF SANDBOX METHOD FOR POTENTIALLY MALICIOUS APPLICATIONS

K. A. Shcheglov[1], A. Yu. Shcheglov[2]

[1]*Scientific and Production Enterprise "Information Technologies in Business", 194044, St. Petersburg, Russia*

[2]*ITMO University, 197101, St. Petersburg, Russia*
E-mail: info@npp-itb.spb.ru

A method is proposed for sandbox implementation for potentially malicious applications including those which received malicious characteristics after infection. The method is based on redirection of access requests based on patented technical solution. The sandbox implementation method produces minimal impact on end machine productivity and is easily administrated; an example of practical interest application is presented. The described technical solution is tested in a commercial information security system.

Keywords: malicious application, sandbox, security method, access request redirection, information security system

Data on authors

Konstantin A. Shcheglov — Scientific and Production Enterprise "Information Technologies in Business", Executive Director; E-mail: scheglov.konstantin@gmail.com

Andrey Yu. Shcheglov — Dr. Sci., Professor; ITMO University; Department of Computer Science, E-mail: info@npp-itb.spb.ru

For citation: Shcheglov K. A., Shcheglov A. Yu. Implementation of sandbox method for potentially malicious applications. *Journal of Instrument Engineering*. 2017. Vol. 60, N 10. P. 940—944 (in Russian).

DOI: 10.17586/0021-3454- 2017-60-10-940-944