УДК004.49
DOI 10.21685/2587-7704-2019-4-1-12

Open Access | RESEARCH ARTICLE

Исследование поведения вредоносных программ в защищенной среде исполнения

Д.А. Умаров
Пензенский государственный университет, Россия, 440026 г. Пенза, ул. Красная, 40

С. Н. Борисова
Пензенский государственный университет, Россия, 440026 г. Пенза, ул. Красная, 40

Аннотация. В работе рассматриваются альтернативные традиционным методы защиты от вредоносных программ и атак. Одним из таких методов является запуск подозрительного кода в изолированной среде исполнения. Принцип работы изолированной среды исполнения позволяет, в отличие от альтернативных решений по защите, не только обнаружить следы вредоносного кода или атаки в системе, но и заблокировать атаку еще на стадии доставки вредоносного кода на атакуемую машину.

Ключевые слова: вирус, вредоносная программа, виртуальная среда исполнения, руткит, троянская программа.

Investigating malware behavior in the protected runtime environment

D. A. Umarov
Penza State University, 40 Krasnaya Street, 440026, Penza, Russia

S. N. Borisova
Penza State University, 40 Krasnaya Street, 440026, Penza, Russia

Abstract. The article considers alternative methods for protection against malware and attacks. One of these methods is to run a suspicious code in an isolated runtime environment. The principle of an isolated runtime environment allows, in contrast to alternative protection solutions, not only to detect traces of malicious code or an attack in the system, but also to block the attack even at the stage of delivering malicious code to the machine under attack.

Keywords: virus, malware, virtual runtime environment, rootkit, Trojan program.

Известно большое множество различных типов вредоносных программ. Но каждый тип состоит из огромного количества подтипов, также отличающихся друг от друга в той или иной степени. Для борьбы со всеми ними нужно уметь однозначно классифицировать любую вредоносную программу и легко отличить ее от других вредоносных программ.

Эксперты на данный момент классифицируют все виды вредоносного программного обеспечения и потенциально нежелательных объектов в соответствии с их активностью на компьютерах пользователей. Такова классификация Лаборатории Касперского. Предложенная система классификации лежит в основе классификации многих производителей антивирусных программ.

Данная система четко описывает каждый обнаруженный объект и назначает конкретное местоположение в дереве классификации, показанном на рис. 1. Дерево классификации делится на:

– типы поведения, представляющие наименьшую опасность (показаны в нижней области диаграммы);

– типы поведения с максимальной опасностью (отображаются в верхней части диаграммы).

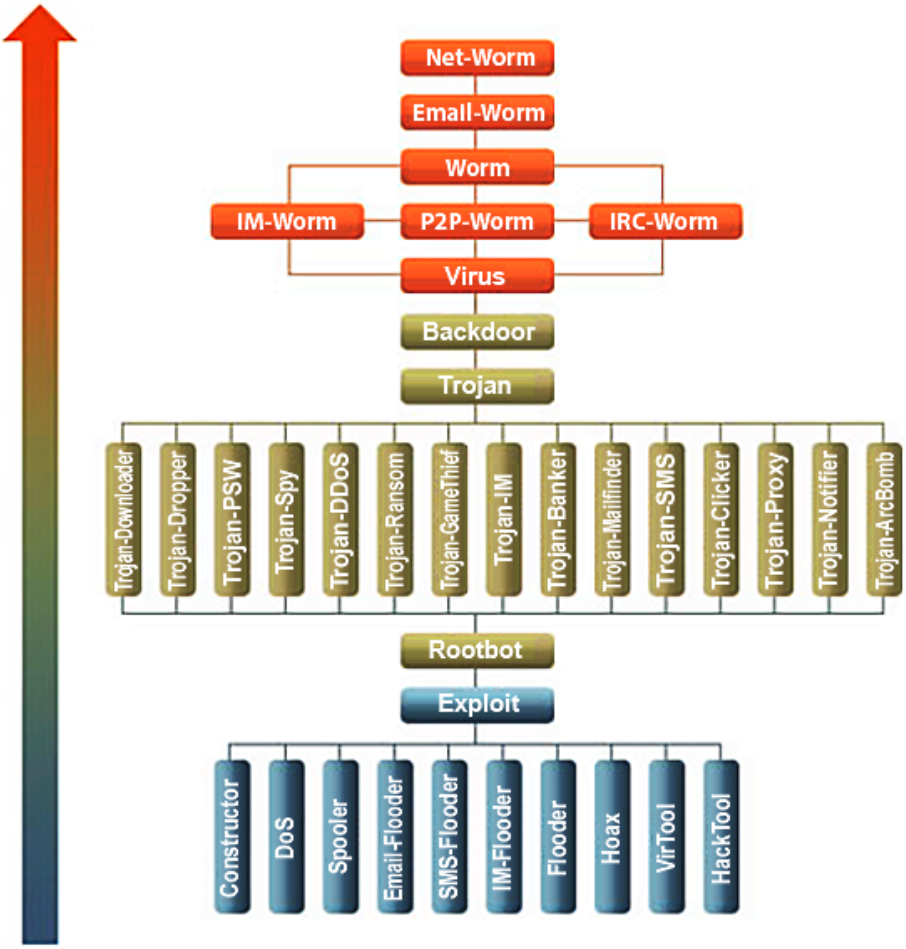

Рис. 1. Дерево классификации вредоносных программ

В свою очередь вредоносные программы подразделяются на следующие типы [2]:

1. **Вирус** – самовоспроизводящийся программный код, который внедряется в установленные программы без согласия пользователя. Вирусы можно разделить по типу объектов, которые они заражают, по методам заражения и выбора жертв.

2. **Червь**. Черви являются в некотором роде вирусами, так как созданы на основе саморазмножающихся программ. Однако черви не могут заражать существующие файлы. Вместо этого червь поселяется в компьютер отдельным файлом и ищет уязвимости в сети или системе для дальнейшего распространения себя.

3. **Троянские программы**. По своему действию являются противоположностью вирусам и червям. Их предлагают загрузить под видом законного приложения, однако вместо заявленной функциональности они делают то, что нужно злоумышленникам. Троянские программы не самовоспроизводятся и не распространяются сами по себе, заразить компьютер может только сам пользователь по недосмотру и неосторожности. Однако в настоящее время трояны эволюционировали до таких сложных форм, как, например, бэкдор (троян, пытающийся взять на себя администрирование компьютера) и троян-загрузчик (устанавливает на компьютер жертвы вредоносный код).

4. **Руткит**. В современном мире руткит представляет собой особую часть вредоносных программ, разработанных специально, чтобы скрыть присутствие вредоносного кода и его действия от пользователя и установленного защитного программного обеспечения. Это возможно благодаря тесной интеграции руткита с операционной системой. А некоторые руткиты могут начать свою работу, прежде чем загрузится операционная система.

5. **Бэкдор (средство удаленного администрирования).** Бэкдор, или RAT (remote administration tool), – это приложение, которое позволяет системному администратору или злоумышленнику управлять чужим компьютером на расстоянии. В зависимости от функциональных особенностей конкретного бэкдора, злоумышленник может установить и запустить на компьютере жертвы любое программное обеспечение, сохранять все нажатия клавиш, загружать и сохранять любые файлы, включать микрофон или камеру, т.е. брать на себя контроль за компьютером и информацией жертвы.

6. **Загрузчик**. Данный тип программ является небольшой частью кода, используемой для дальнейшей загрузки и установки полной версии вируса. После того как загрузчик попадает в систему путем сохранения вложения электронного письма или, например, при просмотре зараженной картинки, он соединяется с удаленным сервером и загружает весь вирус на компьютер жертвы.

Для защиты от вредоносных программ используются антивирусные программы, которые обычно при запуске приложений самим пользователем предупреждают об ответственности за запуск неизвестных файлов. Отдельные вредоносные программы часто выполняют несколько вредоносных функций и используют несколько способов распространения. Например, пользователь может получить по электронной почте фишинговое письмо от внушающего доверие отправителя. В письмо вложен файл с эксплойтом. Если пользователь предпринимает попытку открытия файла, происходит заражение компьютера вирусом, трояном (шифровальщиком) или другой вредоносной программой.

Поэтому неизвестные файлы необходимо проверять в изолированной среде, никак не взаимодействующей с основной системой. Подобными системами являются изолированные виртуальные среды исполнения программного кода (Sandbox). Sandbox («песочница») – специально выделенная среда для безопасного исполнения компьютерных программ. Обычно она представляет собой жестко контролируемый набор ресурсов для исполнения гостевой программы, например место на диске или в памяти. Доступ к сети, возможность сообщаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. «Песочницы» представляют собой пример виртуализации. Повышенная безопасность исполнения кода в «песочнице» зачастую связана с большой нагрузкой на систему, именно поэтому некоторые виды «песочниц» используют только для неотлаженного или подозрительного кода.

Переходя к антивирусным «песочницам», суть которых есть защита основной рабочей системы от потенциально опасного контента, можно выделить три базовые модели изоляции пространства «песочницы» от всей остальной системы [3]:

1. Изоляция на основе полной виртуализации. Использование любой виртуальной машины в качестве защитного слоя над гостевой операционной системой, где установлен браузер и иные потенциально опасные программы, через которые пользователь может заразиться, дает достаточно высокий уровень защиты основной рабочей системы. Недостатки подобного подхода, кроме огромного размера дистрибутива и сильного потребления ресурсов, кроются в неудобствах обмена данными между основной системой и «песочницей». Более того, нужно постоянно возвращать состояние файловой системы и реестра к исходным для удаления заражения из «песочницы». Если этого не делать, то, например, агенты спам-ботов будут продолжать свою работу внутри «песочницы». Блокировать их «песочнице» нечем.

2. Изоляция на основе частичной виртуализации файловой системы и реестра. Совсем необязательно таскать с собой движок виртуальной машины, можно постепенно перераспределять процессам в «песочнице» дубликаты объектов файловой системы и реестра, помещая в «песочницу» приложения на рабочей машине пользователя. Попытка модификации данных объектов приведет к изменению лишь их копий внутри «песочницы», реальные данные не пострадают. Контроль прав не дает возможности атаковать основную систему изнутри «песочницы» через интерфейсы операционной системы.

Недостатки подобного подхода также очевидны: обмен данными между виртуальным и реальным окружением затруднен, необходима постоянная очистка контейнеров виртуализации для возврата «песочницы» к изначальному, незараженному состоянию. Также возможны пробои либо обход такого вида «песочниц» и выход зловредных программных кодов в основную, незащищенную систему. Пример подхода – SandboxIE, BufferZone, ZoneAlarmComodo Internet Security sandbox, Avast Internet Security sandbox.

3. Изоляция на основе правил. Все попытки изменения объектов файловой системы и реестра не виртуализируются, но рассматриваются с точки зрения набора внутренних правил средства защиты. Чем полнее и точнее такой набор, тем большую защиту от заражения основной системы предоставляет программа. То есть этот подход представляет собой некий компромисс между удобством обмена данными между процессами внутри «песочницы» и реальной системой и уровнем защиты от зловредных модификаций. Контроль прав не дает возможности атаковать основную систему изнутри «песочницы» через интерфейсы операционной системы. К плюсам такого подхода относится также отсутствие необходимости постоянного отката файловой системы и реестра к изначальному состоянию.

Недостатки подобного подхода – программная сложность реализации максимально точного и полноценного набора правил, возможность лишь частичного отката изменений внутри «песочницы».

Так же, как и любая «песочница», работающая на базе рабочей системы, возможен пробой либо обход защищенной среды и выход зловредных кодов в основную, незащищенную среду исполнения. Пример подхода – DefenseWall, Windows Software Restriction Policy, Limited User Account + ACL.

Существуют и смешанные подходы к изоляции процессов «песочницы» от остальной системы, основанные как на правилах, так и на виртуализации. Они наследуют как достоинства обоих методов, так и недостатки.

Принцип работы «песочниц» позволяет, в отличие от альтернативных решений по защите от целевых атак, не только обнаружить следы целевой атаки в информационной системе, но и заблокировать атаку еще на стадии доставки зловредного кода пользователю. А независимость от сигнатурных методов детектирования позволяет обнаружить вредоносный код, специально разработанный/модифицированный для проведения целевой атаки. Тем не менее реализация описанных механизмов защиты требует значительных вычислительных ресурсов. Их реализация на уровне конечных рабочих станций невозможна либо неэффективна, поэтому производители предлагают специализированные программно-аппаратные комплексы, работающие на уровне сети.

Библиографический список

1. Антивирусные песочницы. Введение. – URL: https://habrahabr.ru/post/105581/ (дата обращения: 05.02.1018).
2. Борисова, С. Н. Вредоносные программы: классификация и особенности / С. Н. Борисова // Современные информационные технологии. – 2009. – № 10. – С. 170.
3. Петрунин, С. В. Защищенная среда запуска приложений (Sandbox) / С. В. Петрунин, Н. К. Теплов, С. Н. Борисова // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. Всерос. межвуз. науч.-практ. конф. (г. Пенза, 14 марта 2018 г.) . – Пенза : Изд-во ПГУ, 2018. – С. 216.

Образец цитирования:

Умаров, Д. А. Исследование поведения вредоносных программ в защищенной среде исполнения / Д. А. Умаров, С. Н. Борисова // Инжиниринг и технологии. – 2019. – Vol. 4(1). – С. 1–4. – DOI 10.21685/2587-7704-2019-4-1-12.