Секция «LegalTech. Цифровая криминалистика»

Рекомендации по созданию тестовой среды для динамического анализа вредоносного программного обеспечения на базе QEMU

Научный руководитель – Молодцова Юлия Владимировна

Ульянова Мирослава Андреевна
Студент (специалист)
Московский государственный технический университет имени Н.Э. Баумана, Социально-гуманитарные науки, Кафедра Юриспруденции, интеллектуальной собственности и судебной экспертизы, Москва, Россия
E-mail: ulyanova.mira@yandex.ru

В силу непрерывного совершенствования техник и тактик злоумышленников, занимающихся созданием вредоносного программного обеспечения (далее - ВПО), необходимо постоянное корректирование методов и средств анализа подобных программ.

Анализ ВПО осуществляется двумя методами: статический и динамический [1].

Статический анализ - анализ ПО без запуска программы на выполнение (извлечение срок, исследование метаданных, дизассемблирование и анализ машинного кода).

Динамический анализ - анализ ПО посредством анализа поведения исследуемой программы в процессе её выполнения в реальной или виртуальной системе.

По причине применения злоумышленниками методов противодействия анализу ВПО обратная разработка, являющаяся основой статического анализа, значительно затрудняется. В то же время динамический анализ позволяет классифицировать ВПО, определить техники и тактики, а также сформировать список индикаторов компрометации с более эффективными временными затратами, что в условиях массовости рассылки ВПО имеет особую важность.

Однако атакующие все чаще внедряют в ВПО дополнительный функционал для детектирования виртуализации [2]. Так, в случае попадания в виртуальную среду вредоносного образца, имеюшего подобные возможности, происходит завершение всех его процессов и самоудаление.

Несмотря на то, что ВПО будет запускаться на выполнение в виртуальной среде, следует помнить о необходимости обеспечения комплексной безопасности проектируемой системы. Необходимо понимать, что существуют образцы ВПО, способные путем эксплуатации уязвимостей ПО для виртуализации осуществлять выход из виртуального окружения, что может привести к компрометации хост-системы. Так как превалирующее большинство ВПО предназначено для ОС Windows, в качестве хостовой системы рекомендуется выбирать UNIX-based системы: в указанном случае для выполнения программы потребуются дополнительные модули, реализация которых маловероятна.

Для реализации тестовой среды для динамического анализа ВПО как правило используются гипервизоры (ПО для виртуализации системных ресурсов). Существуют 2 типа гипервизоров:

1. работающие напрямую с обрудованием (QEMU, KVM, Xen);
2. работающие поверх операционной системы (Oracle Virtualbox, VMWare).

Наиболее подходящими для анализа ВПО в современных условиях являются гипервизоры 1-го типа, позволяющие осуществить максимально тонкую настройку виртуальной среды, что снижает риск обнаружения факта виртуализации вредоносной программой.

Нами выделяются 2 основные стратегии построения виртуального стенда:

1. в виде виртуальной машины с интегрированными средствами анализа;

1.1. с размещением инструментов мониторинга и отладки в тестовой виртуальной системе;

1.2. с размещением инструментов мониторинга и отладки за пределами тестовой машины, но в одной экосистеме с ней (интегрированные решения, состоящие из модуля гипервизора с дополнениями, размещенными поверх гостевой системы и связанными с ней при помощи скриптов);

2. в виде виртуальной локальной сети, где одна из машин является хостом, на котором происходит запуск ВПО, а другие предназначены для эмуляции штатной сетевой инфраструктуры и реализации на их основе средств анализа (размещение анализаторов трафика по модели прокси-сервисов, например).

В случае монолитного решения необходимо обеспечить изоляцию гостевой системы от сети хоста путем редактирования конфигурационных файлов либо при помощи графического интерфейса (если имеется). Для этого в гипервизорах второго типа необходимо выбрать в настройках режим host-only, а в гипервизорах первого типа указать в конфигурационных параметрах значение “none” для сетевого интерфейса.

Для реализации платформы для динамического анализа ВПО нами выбран гипервизор 1-го типа QEMU, так как на его основе можно выстраивать различные схемы интеграции дополнительных средств анализа (инструменты для отслеживания сетевого трафика, содержимого оперативной памяти). Управление конфигурацией виртуальных машин осуществляется путем редактирования конфигурационного файла формата XML через virt-manager (стандартное средство библиотеки libvirt). Для каждой машины создается свой XML-файл.

Перед разворачиванием виртуальной среды необходимо отредактировать файы QEMU - изменить стандартные имена оборудования (по умолчанию имена содержат строку “QEMU”). В стандартном каталоге QEMU целевыми будут файлы core.c, atapi.c и scsi-disk.c. При помощи утилиты sed возможно осуществить замену строк на имена реального оборудования, после чего пересобрать ядро гипервизора при помощи утилиты make.

Первым этапом построения тествой среды является создание виртуального диска. QEMU позволяет создавать диски в формате QCOW и IMG. Формат файла виртуального диска необходимо выбирать, учитывая следующие особенности:

1. QCOW-диск по умолчанию является динамическим диском (т.е. занимает место на физическом накопителе в соотвествии с фактическим размером содержимого), что увеличивает риск обнаружения виртуализации теми образцами ВПО, которые содержат функционал для определения физического размера диска. Проблема может решаться размещением в виртуальной системе балластирующих файлов.

2. QCOW является собственным форматом QEMU, следовательно он не может быть использован в других гипервизрах без конвертации. Переконвертация может осуществляться штатными средствами гипервизора.

3. QCOW позволяет создавать снимки состояния виртуальной машины (снапшоты), которые необходимы в процессе динамического анализа для возварщения виртуальной машины в изначальное состояние (до выполнения анализируемого образца ВПО).

2. Виртуальный диск в формате IMG является статическим (резервирует на физическом накопителе весь указанный при его создании объем памяти).

После установки системы необходимо отредактировать XML-файл, расположенный в подкаталоге QEMU основной директории менеджера libvirt. В параметрах разделов <bios> и <system> конфигурационного файла необходимо прописать значения, соответствующие реальному оборудованию (либо выбрать режим копирования настроек хост-устройства. Также следует дополнительно установить ПО seabios, эмулирующий BIOS реального устройства.

Прототипированный образец, созданный нами в соответствии с данными рекомендациями, успешно проходит проверки специализированными утилитами (например, pafhish). Произведенные манипуляции позволяют снизить риск детектирования виртуализации вредоносным ПО, что позволяет эффективнее осуществлять динамический анализ и проводить исследования в рамках производства по делам о создании и распространении распространении ВПО (ст. 273 УК РФ), а также по делам о неправомерном доступе к компьютерной информации (ст. 272 УК РФ) [3].

Источники и литература

1) Бутин А.А. Методические аспекты разработки систем защиты программного обеспечения // Вестник науки и образования. 2018. №16-1 (52). С. 39-43.
2) Дроботун Е.Б. Анализ активности и тенденций развития вредоносных программ типа «Блокиратор-шифровальщик файлов» // Программные продукты и системы. 2016. №2 (114). С. 23-28.
3) Уголовный кодекс РФ от 13 июня 1996 г. № 63-ФЗ // Собрание законодательства Российской Федерации - 17 июня 1996 г. № 25. Ст. 2954.