вать целостность системы, а также реализовать асинхронный ввод-вывод, что немаловажно для драйверов, поддерживающих только синхронные операции.

Список использованных источников:

1. Медицинский аппаратный комплекс «Акутест». Справочное руководство. - Москва, 2014 - 14 с.
2. Jeffrey Richter. CLR via C# (4th edition). - Microsoft Press, 2012 - 896 с.

ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ СОВРЕМЕННЫХ ПРОГРАММНЫХ ПЛАТФОРМ JAVA И .NET

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Марчук М. С., Саскевич А. В., Цыркунов Д. А.

Стройникова Е. Д. – ассистент кафедры информатики

В данной работе исследованы особенности платформ Java и .NET в сфере безопасности и защищенности по следующим критериям: 1) обеспеченность платформ собственными библиотеками цифровой подписи и шифрования данных; 2) защищенность скомпилированного кода от декомпиляции и дизассемблерирования, деобфускация кода и обратное восстановление; 3) безопасность кода во время выполнения; 4) возможность изменения скомпилированного кода; 5) внедрение в процесс, запущенный на платформе Java или .NET, и возможность изменения данных приложения во время работы,

6) цифровая подпись готовых сборок и приложений; 7) безопасность Android приложений; 8) наличие устранимых уязвимостей в платформах Java и .NET и возможные уязвимости в современных версиях Java 8 и .NET Framework 4.5. Исследование проводилось посредством реализации программных проектов на основе данных платформ.

1. Обеспеченность платформ собственными библиотеками цифровой подписи и шифрования данных

В языке Java существуют различные стандартные библиотеки шифрования и защиты данных, например, java.security.*. Библиотека предоставляет широкий набор интерфейсов и классов, предназначенных для настройки безопасности своего проекта. Данный пакет содержит такие классы, как, например: *AccessControlContext* – используется для принятия решения о предоставлении доступа к ресурсам системы; *KeyStore* – этот класс представляет хранилище для криптографических ключей; различные криптографические классы и алгоритмы шифрования. Также этот пакет поддерживает ряд классов для операций шифрования данных. Многие классы позволяют на своей основе реализовать собственные программные компоненты шифрования и защиты.

Платформа .NET располагает многими полезными классами и службами, которые облегчают написание защищенного программного кода и позволяют системным администраторам настраивать разрешения кода для доступа к защищенным ресурсам. Кроме того, среда выполнения и .NET Framework располагают полезными классами и службами, которые облегчают применение криптографии и системы безопасности, основанной на ролях. Платформа .NET предоставляет реализации многих стандартных криптографических алгоритмов. Эти алгоритмы просты в использовании и по умолчанию имеют наиболее безопасные из возможных значений свойств. Так, как и в Java, предоставляются интерфейсы классов, поэтому разработчик может написать свою реализацию методов.

2. Защищенность скомпилированного кода от декомпиляции и дизассемблерирования, деобфускация кода и обратное восстановление

Скомпилированный Java-код представляет собой набор класс-файлов. При запуске скомпилированный файл проходит так называемый процесс верификации. Скомпилированный байт-код доступен для декомпиляции, в результате чего может быть получен исходный код, аналогичный изначальному, однако из-за несовершенства современных инструментов в некоторых случаях может быть получен код со вставками байт-кода. Для защиты исходного кода могут быть применены обфускаторы – специальные инструменты, изменяющие названия методов, переменных, а также изменяющие сам код таким образом, что его практически невозможно расшифровать для последующей модификации. Так, например, инструмент *Zelix KlassMaster* изменяет байт-код таким образом, что впоследствии он не будет поддаваться декомпиляции.

Так как код платформы .NET, аналогично Java, работает в исполняемой среде, действующей в виртуальной машине, платформа .NET имеет некоторое общее с Java устройство. Однако .NET не производит верификации скомпилированного кода, все необходимые проверки осуществляются в процессе загрузки и исполнения. Скомпилированный код может быть преобразован в исходный код специальным инструментом *IL Disassambler*, что приводит к тому, что IL–код легко восстановить и модифицировать. В то же время существуют инструменты обфускации, изменяющие код и его структуру, при этом сохраняя его работоспособность.

3. Безопасность кода во время выполнения

В обоих случаях, для Java и .NET, исполняемый код проходит различные проверки, исключающие ошибочные или некорректные операции. Каждая из платформ предупредит пользователя о таких вещах, как выход за пределы разрядной сетки при выполнении арифметических операций, исключение при рабо-

те с IO, переполнение памяти и различных буферов, приведение недопустимых типов. Java не предоставляет пользователю низкоуровневых операций с ОС или системой, в то время как .NET предлагает специальные кодовые конструкции, позволяющие внедрять небезопасный unsafe–код. Однако Java позволяет работать с ОС посредством интерфейса JNI *(Java Native Interface)*, в случае которого пользователь может реализовать на языках C/C++ все необходимые операции с системой.

4. Возможность изменения скомпилированного кода

Непосредственное изменение скомпилированного кода может привести к его неработоспособности. В Java класс-файлы, измененные без использования исходного кода, должны повторно пройти процедуру верификации, иначе система не сможет их загрузить. Также платформа Java имеет различные сторонние инструменты кодогенерации и изменения кода, например библиотека ASM, позволяющая манипулировать байт–кодом. Встроенных инструментов платформа не имеет по соображениям безопасности. Платформа .NET имеет аналогичный встроенный инструмент кодогенерации, предоставляющий схожий функционал.

5. Внедрение в процесс, запущенный на платформе Java или .NET, и возможность изменения данных приложения во время работы

Платформа Java не позволяет получить доступ к данным и байт–коду, находящимся в процессе работы, при условии, что код не запущен в режиме отладки. Если он находится в указанном данном режиме, то посредством стандарта JPDA пользователь может получить доступ к данным приложения, его классам, потокам, полям и методам. Для этого существует программный интерфейс JDI, упрощающий отладку работающих приложений.

Платформа .NET предлагает отладчик командной строки MDbg, который также позволяет отлаживать работающие .NET-приложения (только управляемый IL–код), а значит, получать непосредственный доступ к данным и коду. Кроме того, IL–код, скомпилированный в нативный код, можно отлаживать посредством инструмента gdb, получая полный доступ к коду и данным из командной строки.

6. Цифровая подпись готовых сборок и приложений

Java предлагает возможность подписи готовых пакетов, приложений и апплетов на основе SSL–сертификатов. Сертификат генерируется на основе файлов (это означает, что изменение файлов ведет к тому, что сертификат перестает им соответствовать и среда не позволяет запустить код), а затем размещается внутри сборки. С одной стороны, это позволяет защитить код от подмены, с другой стороны, отключение проверки сертификатов в среде Java позволит злоумышленникам совершать незаконные операции с кодом. Аналогичная возможность есть в платформе .NET, где готовые сборки также могут быть подписаны сертификатом. Однако процесс сертификации не является обязательным, потому очень часто встречается несертифицированный код, который можно модифицировать или подменить.

7. Безопасность Android–приложений

На основе сертификатов устроена работа приложений Android. Сертификат исключает возможность непосредственного изменения содержимого пакета, однако существуют инструменты, позволяющие переподписать приложение после изменения. Весь код и ресурсы приложения собираются в единый apk-файл, причем ресурсы, код и манифест приложения шифруются. Весь байт-код хранится внутри одного dex-файла, в отличие от обычного приложения Java, где код размещен по отдельным классам. Dex-файл имеет структуру, аналогичную структуре класс–файлов. В итоге пользователь видит только интерфейс, доступ к коду для него закрыт. Кроме того ОС Android предлагает систему разрешений - некоторые небезопасные функции, например, отправка СМС, требуют подтверждения пользователя. Доступ к некоторым возможностям преднамеренно ограничен отсутствием API, однако доступ к этим возможностям можно получить посредством системы, аналогичной JNI, присутствующей в Java.

8. Наличие устранимых уязвимостей в платформах Java и .NET и возможные уязвимости в современных версиях Java 8 и .NET Framework 4.5

За последние годы наблюдается рост атак с использованием различных уязвимостей, например, широко известной уязвимости «нулевого дня». Однако компания Oracle регулярно выпускает обновления, направленные на улучшение безопасности платформы, также по причине того, что платформа является открытой, сообщество разработчиков с завидным постоянством находит и исправляет ошибки.

Платформа .NET также регулярно пополняется обновлениями, исправляющими критические ошибки, а т. к. с недавних пор она также стала открытой, в ближайшее время может наблюдаться рост исправления ошибок и уязвимостей. В общей сложности уязвимости, присутствие которых возможно в современных платформах, могут быть основаны либо на несовершенстве платформ, либо на несовершенстве программных библиотек. Также большая часть уязвимостей может быть найдена в части веб–инфраструктуры ввиду ее популярности.

Список использованных источников:

1. Шилдт, Г. Java. Полное руководство. Издательство 8-е / Г. Шилдт – М.: «Издательский дом «ВИЛЬЯМС», 2012. – 1102 с.
2. Шилдт Г. C# 4.0. Полное руководство. / Г. Шилдт – М.: «Издательский дом «ВИЛЬЯМС», 2011. – 1056 с.
3. Gosling, J. Java Language and Virtual Machine Specifications / J. Gosling // The Java Language specification [Electronic resource]. – 2015. – Mode of access: http://docs.oracle.com/javase/specs/
4. Стандарт ECMA-335 от 06.2012 [Электронный ресурс]. – 2012. – Режим доступа: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-335.pdf
5. Журнал «Лаборатории Касперского» [Электронный ресурс]. – 2015. – Режим доступа: http://business.kaspersky.ru/