МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

“САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

«РАЗРАБОТКА МЕТОДА ДИНАМИЧЕСКОГО АНАЛИЗА ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВИРТУАЛЬНЫХ МАШИН ДЛЯ ВЫЯВЛЕНИЯ ЗАГРУЗОЧНЫХ ЗАКЛАДОК»

Автор _____ _____

Направление подготовки (специальность) _____

Квалификация _____

Руководитель _____ _____

К защите допустить

Зав. кафедрой _____ _____

«___» _____ 2016г.

Санкт-Петербург, 2016 г.

Студент _____ Группа _____ Кафедра _____

Факультет _____

Направленность (профиль), специализация _____

Консультанты: а) _____ _____

ОГЛАВЛЕНИЕ

Стр.

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ

КС - компьютерная система ПО - программное обеспечение

ВСТУПЛЕНИЕ

Когда запускается персональный компьютер, одно из первых, что в нем происходит, - это загрузка встроенного программного обеспечения, которое находится в специальном модуле на материнской плате. Это первый шаг, который задаётся программным способом, а также задаёт последующее поведение персонального компьютера. Захватив управление этим модулем, появляется возможность вывода из строя всех программных механизмов защиты на компьютере, и достигается постоянное пребывание в системе, потому что этот компонент неразрывно связан с ней физически. Поэтому этот компонент является приоритетным для злоумышлеников.

Существуют методы, которые защищают встроенное программное обеспечение на аппаратном уровне. Они реализуют необходимый пласт защиты от большого множества угроз. А теперь представим, что система не представлена физически. Это виртуальные машины, которые часто используются для различных целей. На них ставят сервера, поднимают критические системы, тестируют огромное количество программных продуктов. Они обладают множеством полезных качеств для этого, но как можно понять - них нет аппаратных составляющих, а значит защита, которая реализовывалась для встроенного программного обеспечения, на виртуальных машинах не существует.

Сейчас возможности встроенного программного обеспечения растут, в некоторых даже пресутствует стек технологий TCP/IP, который позволяет выходить в сеть не используя механизмы операционных систем. Новая технология - UEFI/BIOS, которая только начинает появляться, как и все новые системы не обладает должной защитой. А уровень вхождения в неё намного ниже, чем в технологию прошлых лет - BIOS.

ГЛАВА 1

Цель работы

Основной целью работы является - повышение уровня безопасности встроенного программного обеспечения виртуальных машин.

Объект работы - виртуальная машина.

Предмет работы - встроенное программное обеспечение.

Метод достижения цели - динамический анализ для выявления загрузочных закладок.

Необходимость защиты встроенного программного обеспечения выражена тем, что максимальное число привилегий над компьютерной системой системой сосредоточено в данном модуле. Также данная тема требует глубокого рассмотрения, чего не присутствует в современной системе по обеспечению информационной безопасности. Автор считает, что следует уделять больше внимания выбранной области, а также привлекать в неё всё больше специалистов.

Объектом является виртуальная машина из-за простоты работы, а также большей доступностью для различных исследований. Предмет выделенный из объекта для изучения является встроенным программным обеспечением, что показывает намерения автора в дальнейшем перенести накопленные знания в область работы не только с виртуальными машинами.

Метод выбран исходя из текущего положения информационной безопасности объявленной области, существуют механизмы, которые позволяют пошагово выполнять работу встроенного программного обеспечения, на не существует механизмов для анализа выполняющихся шагов. В то же время существуют программные комплексы проводящие статический анализ образов встроенного программного обеспечения. Поэтому выбор был сделан в пользу перспективного направления, которое в близжайшем будущем может быть очень востребованным.

ГЛАВА 2

Общие сведения

2.1 Встроенное программное обеспечение

Термин «Встроенное программное обеспечение» не часто используется в повседневной жизни, чаще используется аналог, появившийся исторически - «Прошивка». Поэтому во многих источниках литературы чаще встречается подобное название. В законодателных актах используется понятие - «Встроенное программное обеспечение», но нет достаточного определения. Автор приводит собственное определение после того, как назовёт те, что приводятся в использованных источниках.

«Прошивкой (англ. Firmware, fw) называют содержимое энергонезависимой памяти компьютера или любого цифрового вычислительного устройства — микрокалькулятора, сотового телефона, GPS-навигатора и т. д., в которой содержится его микропрограмма.» [1]

Для отнесения ПО в различные классы Минкомсвязи создал классификатор: Приказ Минкомсвязи России от 31.12.2015 N 621 "Об утверждении классификатора программ для электронных вычислительных машин и баз данных".[2] Согласно нему существует отдельный раздел - «Встроенное программное обеспечение», в котором один единственный класс - «BIOS и иное встроенное ПО» - программы, хранящиеся в постоянной памяти.[2] Данный приказ позволит определить, что является целью рассмотрения, чтобы в последствии не путать различные классы ПО. В самой же работе будет рассмотрена только технология BIOS, а именно её расширение UEFI/BIOS.

По мнению автора, более подходящее для работы определение имеет следующий вид:

Встроенное программное обеспечение - программное обеспечение, исходный код которого хранится аппаратными средствами, не предназначенными для выполнения задач, поставленными перед устройством, а его деятельность направленна на начальную инициализацию аппаратного обеспе-

чения.

Как можно заметить в данном определении у встроенного программного обеспечения выделено две черты:

а) Оно хранится в независимой области;

б) Его основная функция - начальная инициализация.

2.1.1 История термина "прошивка"

Термин «прошивка» появился в 1960-х годах, когда в ЭВМ использовалась память на магнитных сердечниках. В постоянных запоминающих устройствах (ПЗУ) использовались Ш-образные и П-образные сердечники. Ш-образные сердечники имели зазор около 1 мм, через который и укладывался провод. Для записи двоичной «1» провод укладывался в одно окно сердечника, а для записи «0» — в другое. В сердечник высотой 14 мм укладывалось 1024 провода, что соответствовало 1К данных одного разряда. Работа выполнялась протягиванием провода вручную с помощью «карандаша», из кончика которого тянулся провод, и таблиц прошивки. При такой кропотливой и утомительной работе возникали ошибки, которые выявлялись на специальных стендах проверки. Исправление ошибок осуществлялось обрезанием ошибочного провода и прошивкой взамен него нового.

В начале 1970-х годов появились П-образные сердечники, которые позволяли использовать для прошивки автоматические станки. Прошивка выполнялась уже не в устройстве ПЗУ, а в жгутах по 64, 128 или 256 проводов. Прошиваемые данные вводились в станок с помощью перфокарт. На специальной оснастке жгуты снимались со станка, обвязывались нитками, и концы проводов распаивались на колодки. После этого жгуты укладывались в блок ПЗУ. Как при ручной прошивке, так и при работе на прошивочном станке требовалась аккуратность и хорошее зрение, поэтому на прошивке работали молодые девушки.

В 1980-х годах термин «прошивка» стал вытесняться понятием «прожиг», что было вызвано появлением микросхем ПЗУ с прожигаемыми пе-

ремычками из нихрома или кремния, однако при более новых технологиях «прожиг» вышел из употребления, а «прошивка» осталась.[1]

2.1.2 Лицензионное соглашение с потребителем

Обычно, заключая договор с производителем материнских плат, пользователь подписывает лицензионное соглашание, которое запрещяет извлекать встроенное программное обеспечение, а также изучать его различными способами. В юридическом плане не всегда существует возможность для изучения встроенного программного обеспечения. На аппаратуре до технологии UEFI и даже на некоторых моделях с поддержкой данной технологии не редки случаи, что существует всего несколько обновлений, создание которых датировано несколько летней давностью. Таким образом получается, что встроенное программное обеспечение не обновляется, в нём не закрываются различные уязвимости, бывает, что искусственно занижены аппаратные возможности. Все эти вопросы сложно решить с помощью легальных методов.

Чаще всего в изучении данного вопроса помогают исследования, которые были произведены против лицензионного соглашения, либо утечки исходного кода, одна из которых произошла в компании American Megatrends Incorporated[3].

Способы, которыми фирмы-производители следят за сохранностью встроенного программного обеспечения[1]:

а) Лицензионное соглашение с потребителем запрещает извлекать и изучать «прошивки» тем или иным способом;
б) Самовольная замена «прошивки» на другую («перепрошивка») обычно прекращает действие гарантийных обязательств фирмы;
в) Процедуры обслуживания и изменения режимов работы микропрограмм обычно не разглашаются и в лучшем случае известны только работникам фирменных сервисных центров.

Но не смотря на такую строгость с лицензией на использование встро-

енного программного обеспечения, существуют проекты, которые являются открытыми, распространяются под лицензиями - BSD, GNU GPL, а также исходный код которых находится в свободном доступе.

Список основных проектов:

а) Tianocore[4] - UEFI/BIOS, который рассматривается в работе;
б) OpenBIOS[5] - проект, нацеленый на замену проприетарного ПО;
в) SeaBIOS[6] - основной BIOS, используемый в QEMU (на котором производится работа)
г) и др.

Данная работа будет выполнена с использованием виртуального интерпретатора QEMU, с встроенным программным обеспечением от Tianocore.

2.1.3 SLIC

Перед компаниями производящими ПО стоял вопрос о подтверждении лицензии пользователя. Для этой цели было создано три компонента подтверждения лицензии, а именно таблица ACPI_SLIC table(SLIC), в которой расположены OEM SLP и OEM certificate.

OEM (от англ. original equipment manufacturer — «оригинальный производитель оборудования») - организация, продающая под своим именем и брендом оборудование, сделанное другими предприятиями.

Каждой организации выдаются ключ-лицензия (OEM SLP) и цифровой сертификат (OEM certificcate), а информация об этом хранится в таблице (ACPI_SLIC).

SLIC (от англ. software licensing description table) - таблица, в которой хранится информация о лицензировании ПО.

OEM SLP (от англ. system locked pre-installation - «код продукта OEM») - специальный 25 значный ключ-лицензия.

OEM certificate («Цифровой сертификат OEM») - файл в формате XML с расширением *.xrm-ms. Выдаётся фирмой Microsoft каждому крупному производителю ПК.

Таким образом получается, что на персональном компьютере получится запустить только ту систему, владелец которой обладает ключом, исключая возможность нелицензионного использования ПО.

2.2 BIOS, UEFI/BIOS

BIOS (от англ. basic input/output system - «базовая система ввода-вывода») - набор микропрограмм, реализующих API для работы с аппаратурой компьютера и подключёнными к нему устройствами.

UEFI (от англ. Unified Extensible Firmware Interface - "универсальный интерфейс расширяемой прошивки") создаётся для того, чтобы заменить технологию, которая уже считается устаревшей.

Разработка спецификации программного обеспечения UEFI, а также SDK (от англ. software development kit - «комплект средств разработки»), известного под названием edk2 (EFI developnment kit 2), производится компанией Unified Extensible Firmware Interface Forum. А до этого разработка была начата компанией Intel Corporation, которой были созданы первые редакции стандарта.

На момент написания работы спецификация расположена в свободном доступе на официальном сайте разработчика под версией 2.6, насчитывая 12 предшественников до версии 2.0. Помимо спецификации на саму систему UEFI возможно найти спецификации на ACPI, UEFI shell, UEFI Platform Initializatio, а также другие документы по данной технологии.

Основные производители UEFI/BIOS для персональных компьютеров:

а) Award Software International Inc.;
б) American Megatrends Incorporated;
в) Insyde Software.

2.3 Виртуальная машина

Виртуальная машина - программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы, или, виртуализиру-

ющая некоторую платформу.

Использование данной технолонии помогает исследовать критические элементы, которые сложно исследовать на реальных системах. Упрощает процесс воспроизведения ПО под различными платформами, а также существенно упрощает процесс отладки. Так как нет необходимости использовать два устройства, соединять их с помощью различных средств, а с использованием определённых решений в сфере виртуализации, так и вовсе отпадает необходимость настройки базовых составляющих.

На сегодняшний момент множество компаний заинтересовано в виртуальных технологиях. Всё чаще можно встрерить сервер на виртуальной машине или найти работника, который выполняет на ней основную свою деятельность, т.к. это позволяет избежать множества проблем с безопасностью, а также управлением системы.

Применение виртуальных машин:

а) Ограничение возможностей программ (Песочница);
б) Работа с различными архитектурами;
в) Разделение ресурсов сервера (запуск нескольких серверов на различных виртуальных машинах);
г) Тестирование и отладка систем.

Применение виртуальных машин постепенно становится чем-то обыденным. А если виртуальные машины становятся такими популярными, то всё острее стаёт вопрос их безопасности. Существуют направления в сфере информационной безопасности, в которых обсуждаются вопросы манипуляцией внутри виртуальной машины, возможность обойти её, способы определения типа виртуального интерпретатора, гипервизора. У виртуальных машин, как и у реальных, существует возможность исполнения кода, заменяющего встроенное программное обеспечение. По сути оно почти ничем не отличается от настоящего, кроме некоторых поправок, которые можно не рассматривать в первом приближении.

В работе используется эмулятор - QEMU[7].

2.4 Динамический анализ ПО

«Динамический анализ кода - анализ программного обеспечения, выполняемый при помощи выполнения программ на реальном или виртуальном процессоре (в отличие от статического анализа).» [8]

Динамический анализ применяется в тех областях, где главный критерий - надёжность программы. Данный подход к анализу позволяет выявить то, что сложно, либо невозможно понять с помощью статического подхода. Статический подход не всегда отвечает, какую функциональность несёт программа.

Этапы динамического анализа:

а) Подготовка исходных данных;
б) Проведение тестового запуска программы и сбор необходимых параметров (Динамическое тестирование);
в) Анализ полученных данных.

В данной работе будет выполнен запуск ПО, с последующим сбором данных, необходимых только для выполнения поставленной академической задаче для подствержднения концепта выполнения.

Принципы динамического тестирования:

а) Белый ящик - исследуются данные о программном коде;
б) Черный ящик - исследуются входные и выходные данные;
в) Серый ящик - подбор входных данных по известной структуре программы.

В данной работе не задаются входные данные, а также не смотрится то, что получается в итоге. Поэтому логично предположить, что работа проводится с белым ящиком.

ГЛАВА 3

Вопросы безопасности встроенного программного обеспечения виртуальных машин

3.1 Встроенное программное обеспечение в законодательстве РФ

Помимо указаного раньше классификатора ПО от минкомсвязи[2] термин BIOS в правовых актах используется только с тезисом, говорящем об установке пароля, для защиты от возможных изменений стандартных настроек. Очень часто, подобная защита обходилась простым сбросом всех настроек BIOS с последующей настройкой минимально необходимых.

Более детально настройка базовой системы ввода-вывода расписана в приказе Федеральной Службы по Интеллектуальной Собственности[9], регламентирующем инструкции по обеспечению режима секретности. В частности в пунктах 6.8-6.10 говорится, что необходимы настройки, которые исключают нестандартные виды загрузки ОС, опять же парольная защита, и в случае если встроенные тесты прошли неудачно, исключается возможность работы на компьютерной системе. На этом деятельность по настройке BIOS заканивается.

Более детальное рассмотрение с точки зрения безопастности BIOS преведено Федеральной Службой, ответсвенной в области технического регулирования. Как объект возможного достижения НСД BIOS рассматривается в выписке Федеральной Службы по Техническому и Экспортному Контролю[10], описывающей базовую модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных от 15.02.2008.

Согласно данной выписке, существует три группы угроз непосредственного доступа в операционную среду информационной системы персональных данных:

а) Угрозы, реализуемые в ходе загрузки ОС;

б) Угрозы, реализуемые после загрузки ОС;

в) Угрозы, реализация которых определяется тем, какая из прикладных

программ запускается пользователем.

Рассматриваемый в работе тип угроз относится к первой группе.

Так как данная модель угроз рассматривается Федеральной Службой, значит существуют методы и средства, возможно определённые подрядчики, занимающиеся анализом данной группы угроз. Согласно статье[11], преведённой в журнале, нацеленном на информационную безопасность, метод заключается в статическом поиске сигнатур, по которым делается вывод о состоянии безопасности. Сложно найти информацию, которая бы подтверждала бы выводы автора статьи или опровергала их.

В любом случае, защита необходима не только системам по обработке персональных данных, а это значит, что необходимо существование методов, которыми бы могли воспользоваться большии слои населения, а также определенные классы компаний.

3.2 Встроенное программное обеспечение виртуальных машин

Почему то считается, что угроза со стороны BIOS маловероятна, но существуют различные события в истории, которые показывают обратное. Например, публикации некоторых технологий Агенства Национальной Безопасноси США, от не безызвестного символа свободы, Сноудена, а также различные уязвимости, связанные с ноутбуками Mac от компании Apple, перепрошивка которых была возможна сразу после режима сна ОС[12]. А также существуют исследования людей, которые выкладывают в открытый доступ свою работу, в которой расписывается, как внедряется загрузочная закладка в встроенное программное обеспечение. Один из таких SmmBackDoor[13].

Также считается, что чтобы перезаписать встроенное программное обеспечение необходимо аппаратное подключение специального модуля, который бы выполнял данную деятельность. После истории с компанией Apple, ноутбуки которых позволяли перепрошивать встроенное программное обеспечение после режима сна, после истории с компанией AMD, криптографические ключи которой утекли в сеть вместе с исходным кодом их BIOS, а также

редким обновлением данной системы, сложно поверить, что всё так без облачно. Тем более, новая технология UEFI намерена регулярно обновляться из-под ОС, а это значит, что будут искаться пути обхода различных защитных механизмов и наверняка будут найдены, т.к. это сравнительно молодая технология, безопасностью которой ещё предстоит заниматься. При всём при этом, важно отметить и то, что данная работа производится с виртуальными машинами встроенное программное обеспечение, которые представляется в виде файла на жестком диске, то есть для перезаписи не нужно ничего, кроме устройства способного переносить файлы на жестком диске.

Также можно вспомнить, что BIOS не молодая технология и созданы различные аппаратные способы защиты, которые защищают его например от перезаписи. Эти технологии теоретически невозможно обойти, поэтому основная защита уже присутствует на материнской плате. Здесь опять же важно отметить тот факт, что данная работа проводится с виртуальными машинами, а следовательно вся защита может быть представлена только программным способом.

Но, по мнению автора, больше всего усугубляет проблему то, что количество специалистов, работающих в данной области ничтожно мало, в основном это сотрудники, которые непосредственно создают это встроенное программное обеспечение, и редкие энтузиасты.

В сети интернет очень сложно найти доступный материал по данной тематике. А то, что предлогается, не всегда удобно использовать, либо обладает недостатком - некомпетентность.

3.3 Загрузочный вирус, bootkit

Загрузочный вирус (англ. Boot virus) — компьютерный вирус, записывающийся в первый сектор гибкого или жёсткого диска и выполняющийся при загрузке компьютера. [14] Данная секция создана, чтобы показать различия между загрузочной закладкой во встроенном программном обеспечении и другим более известным - загрузочный вирус.

Основное отличие в том, что загрузочная закладка находится в той же области памяти, что и встроенное программное обеспечение, а также запускается вместе с ним. Загрузочный вирус - это более обширное понятие, которое включает в себя загрузочную закладку во встроенном программном обеспечении, но также может принимать и другие вариации.

Далее будут приведена информация по некоторым экземплярам загрузочных вирусов.

3.3.1 Stoned

Так сложилось в истории, что загрузочные вирусы появились практически сразу с появлением компьютера. После стали возникать средства защиты. И долгое время считалось, что всё, что необходимо для безопасности, уже создано. После основной зоной интересов были высокоуровневые структуры и языки программирования, поэтому на загрузочные вирусы обращали мало внимания.

Один из самых известных загрузочных вирусов, пришедших к нам из прошлого является Stoned[15]. Его создание датируется 1987 годом, он был создан студентом в Новой Зеландии. Тема загрузочных вирусов не нова, но занимаются ей не многочисленное общество. Обычно, загрузочные вирусы появляются из познавательного интереса, их авторы зачастую студенты либо люди, занимающиеся исследованиями. Данный тип вирусов очень сложно воссоздать, даже обладая пониманием всех протикающих процессов в системе.

3.3.2 Dreamboot

На данный момент, вопрос с изучением загрузочных вирусов может встать более остро. С введением технологии UEFI/BIOS, а также созданием проекта Tianocore, существует возможность ознакомиться с основными составляющими встроенного программного обеспечения. Также в открытом доступе необходимая документация по проекту. Неудивительно, что данная

область привлекает всё больше специалистов.

Компанией, занимающейся иновациями в сфере информационной безопасности, - QuarksLab создан загрузочный вирус под названием Dreamboot[16]. Информация о данном вирусе впервые прозвучала на конференции по информационной безопасности 2013 года под названием HITBSecConf. Он был создан для того, чтобы показать насколько стоит остро вопрос информационной безопасности в современных системах встроенного программного обеспечения.

В системе контроля версии github находится исходный код, а также исполняемый файл, который возможно проверить на собственной машине. Таким образом было дано подтверждение тому, что угроза со стороны загрузочных вирусов реальна.

3.4 Возможная модель нарушителя

Основной критерий отбора - это возможности потенциального нарушителя.

История говорит о том, что это могут быть различный спецслужбы, способные добавлять загрузочную закладку во встроенное программное обеспечение ещё на момент производства компьютерных систем, а также что это могут быть люди, способные использовать уязвимости, найденные в компьютерных системах.

Немногие способны загрузить свой модуль во встроенное программное обеспечение. Сегодня можно встретить в сети интернет инструкции по модификации и внесению своих исполняемых модулей, которые способны сделать все желающий, но будут необходимы определённые навики или аппаратура, которая удовлетворяла бы определённым требованиям.

Усугубляет ситуацию то, что в основном компьютерные системы производятся в иностранных государствах. Так например, существует информация о том, что основные компании производящие процессорные устройства вставляют аппаратную закладку в свои модули. Или потенциально возмож-

ные изменения программного кода на заводе изготовителе. Также государство США упрощает юридическую процедуру обыска компьютеров в любой стране. [17]

Таким образом складывается следующая модель нарушителя:

а) Специальные службы иностранных государств;
б) Криминальные группировки, обладающие повышенными возможностями;
в) Исследователи информационной безопасности;
г) Люди, обладающие инженерными навыками, а также ознакомленные с вопросом.

Очевидно, что у нарушителя должны быть большие возможности для осуществленния атаки на встроенное программное обеспечение, но это легко компенсировать полученным результатом - неудаляемая без специальных технических средств загрузочная закладка, полный контроль над системой.

3.5 Существующие методы защиты

Незадолго до написания работы компанией Google в сервисе под названием VirusTotal была добавлена функция, которая позволяет статически анализировать встроенное программное обеспечение. Основной показатель, который сейчас используется - наличие в образе исполняемых файлов операционной системы Windows, что не является доставерным индикатором. На данный момент технология только появилась в списке функционала сервиса, поэтому следует ожидать её расширения.

Аналогом статического анализа могут служить утилиты для выделения структурных элементов из образа, например UEFITool, PhoenixTool, Intel Flash Image Tool и др. С помощью них возможно не только просмотреть содержимое образа, а также присутствует возможность модификации отдельных составляющих.

Ещё один метод защиты от загрузочных вирусов предложен компанией Лаборатория Касперского совместно с компанией KraftWay - антивирус

для UEFI, который должен обеспечивать защиту от загрузочных вирусов.[18] Сложно предположить методику работы антивируса, но можно сказать, что с большой вероятностью код антивируса не будет записываться на носитель кода, составляющего встроенное программное обеспечение. Может возникнуть ситуация, что закладка во встроенном программном обеспеченим прекратит работу антивирусного ПО.

ГЛАВА 4

Реализация динамического анализа

ЗАКЛЮЧЕНИЕ

Согласно выполненной работе, получается ...

Литература

[1] Встроенное программное обеспечение. — 2016. — URL: https://ru.wikipedia.org/wiki/Встроенное_программное_обеспечение (дата обращения: 2.05.2016).

[2] Приказ Минкомсвязи России от 31.12.2015 N 621 "Об утверждении классификатора программ для электронных вычислительных машин и баз данных". — 2015.

[3] Исходный код AMI-BIOS и ключи подписи UEFI попали в открытый доступ. — 2016. — URL: https://xakep.ru/2013/04/06/60404/ (дата обращения: 12.04.2016).

[4] Tianocore. — 2016. — URL: www.tianocore.org (дата обращения: 12.04.2016).

[5] OpenBIOS. — 2016. — URL: http://www.openfirmware.info/OpenBIOS (дата обращения: 12.04.2016).

[6] SeaBIOS. — 2016. — URL: http://www.seabios.org/SeaBIOS (дата обращения: 12.04.2016).

[7] QEMU. — 2016. — URL: http://wiki.qemu.org/Main_Page (дата обращения: 12.04.2016).

[8] Динамический анализ кода. — 2016. — URL: https://ru.wikipedia.org/wiki/Динамический_анализ_кода (дата обращения: 2.05.2016).

[9] ПРИКАЗ от 5 июля 2013 г. N 82 ОБ УТВЕРЖДЕНИИ ИНСТРУКЦИЙ ПО ОБЕСПЕЧЕНИЮ РЕЖИМА СЕКРЕТНОСТИ ПРИ ОБРАБОТКЕ СЕКРЕТНОЙ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ КОМПЬЮТЕРНОЙ СИСТЕМЫ В РЕЖИМНО-СЕКРЕТНОМ ПОДРАЗДЕЛЕНИИ РОСПАТЕНТА. — 2013.

[10] "Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных" (Выписка) (утв. ФСТЭК РФ 15.02.2008). — 2008.

[11] Китайские закладки: непридуманная история о виртуализации, безопасности и шпионах. — 2011. — URL: https://xakep.ru/2011/12/26/58104/ (дата обращения: 12.04.2016).

[12] OS X после сна разрешает себя перепрошить кому угодно. — 2015. — URL: https://xakep.ru/2015/06/03/os-x-rootkit/ (дата обращения: 12.04.2016).

[13] SmmBackDoor. — 2015. — URL: https://github.com/Cr4sh/SmmBackdoor/ (дата обращения: 12.04.2016).

[14] Загрузочный вирус. — 2016. — URL: https://ru.wikipedia.org/wiki/Загрузочный_вирус (дата обращения: 2.05.2016).

[15] Загрузочный вирус Stoned. — 2016. — URL: http://stoned-vienna.com/ (дата обращения: 2.05.2016).

[16] Загрузочный вирус Dreamboot. — 2016. — URL: conference.hackinthebox.org/hitbsecconf2013ams/materials/D2T1-SebastienKaczmarek-DreambootUEFIBootkit.pdf (дата обращения: 2.05.2016).

[17] Новостной портал ИБ - threadpost. — 2016. — URL: https://threatpost.ru/fbr-mozhet-provodit-obyski-kompyuterov-v-ly15982/ (дата обращения: 12.04.2016).

[18] Антивирус Лаборатории Касперского для UEFI. — 2016. — URL: www.kaspersky.ru/about/news/business/2013/S_chistogo_lista_tehnologii_Laboratorii_Kasperskogo_obespechat_zaschitu_ot_virusov_na_etape_zagruzki (дата обращения: 2.05.2016).