МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

КАФЕДРА
БЕЗОПАСНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

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

Научный руководитель:
_____ Сорокин И В
«___» _____ 2016г.

Студент группы Р3450
_____ Манеев А О
«___» _____ 2016г.

Санкт-Петербург
2016г.

Оглавление

Глава 1

Введение

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 злоумышленников.

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 присутствует 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 2

Общие сведения

2.1. Встроенное программное обеспечение

«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]

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 перемычками из нихрома или кремния, однако при более новых технологиях «прожиг» вышел из употребления, а «прошивка» осталась.

2.1.2. Лицензионное соглашение с потребителем

Обычно, заключая договор с производителем материнских плат, пользователь подписывает лицензионное соглашание, которое запрещает извлекать встроенное программное обеспечение, а также изучать его различными способами.

- лицензионное соглашение с потребителем запрещает извлекать и изучать «прошивки» тем или иным способом;

- самовольная замена «прошивки» на другую («перепрошивка») обычно прекращает действие гарантийных обязательств фирмы;

- процедуры обслуживания и изменения режимов работы микропрограмм обычно не разглашаются и в лучшем случае известны только работникам фирменных сервисных центров.

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:

1. Award Software International Inc.;
2. American Megatrends Incorporated;
3. Insyde Software.

2.3. Виртуальная машина

Виртуальная машина - программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы, или, виртуализирующая некоторую платформу.

Применение:

1. Ограничение возможностей программ (Песочница);
2. Работа с различными архитектурами;
3. Разделение ресурсов сервера (запуск нескольких серверов на различных виртуальных машинах);
4. Тестирование и отладка систем.

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).» [1]

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:

1. Подготовка исходных данных;
2. Проведение тестового запуска программы и сбор необходимых параметров (Динамическое тестирование);
3. Анализ полученных данных.

Принципы динамического тестирования:

1. Белый ящик - исследуются данные о программном коде;
2. Черный ящик - исследуются входные и выходные данные;
3. Серый ящик - подбор входных данных по известной структуре программы.

Глава 3

Вопросы безопасности встроенного программного обеспечения виртуальных машин

3.1. Встроенное программное обеспечение виртуальных машин

Представлены они файлом, который легко заменить. Отсутствуют методы аппаратной защиты. Нет специалистов.

3.2. Загрузочный вирус, bootkit

Загрузочный вирус (англ. Boot virus) — компьютерный вирус, записывающийся в первый сектор гибкого или жёсткого диска и выполняющийся при загрузке компьютера. [1]

3.2.1. Dreamboot

Информация о нём

3.2.2. Stoned

Информация о нём

3.3. Возможная модель нарушителя

Кто, почему, зачем будет пользоваться данными методами?

3.4. Возможная модель угроз

А как в общем-то будут аттаковать?

3.5. Существующие методы защиты

Рассказать о статическом анализе от Google(VirusTotal) Антивирус от Касперского Что-то ещё было на уровне антивируса

Глава 4

Реализация динамического анализа

4.1. Цель работы

Что я пытаюсь реализовать, что должно быть в конце, зачем это.

4.2. Метод динамического анализа встроенного программного обеспечения виртуальных машин

Что я непосредственно сделал?

Глава 5

Заключение

Согласно выполненной работе, получается ...

Литература

[1] asdf. asdfasdf. — 2016.