Министерство науки и высшего образования
Российской Федерации
ФГБОУ ВО
"Владимирский государственный университет имени Александра
Григорьевича и Николая Григорьевича Столетовых"

УТВЕРЖДАЮ

Заведующий кафедры ИЗИ

_____ М.Ю. Монахов

«_____» _____ 2023 г.

ОТЧЁТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

Кафедра информатики и защиты информации

по теме:

МЕТОДЫ ПРОТИВОДЕЙСТВИЯ РУТКИТАМ В ОС WINDOWS

(заключительный)

Руководитель темы к.т.н. _____ Ю.М. Монахов

Владимир 2023

СПИСОК ИСПОЛНИТЕЛЕЙ

Первый исполнитель _____ Силаков Дмитрий Алексеевич

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете о НИР применяют следующие термины с соответствующими определениями.

Атака-уклонение — это техника, при которой злоумышленник предпринимает манипуляции с входными данными или образцами, чтобы обмануть модель и сделать ее неправильные предсказания или дать ложные результаты..

Гипервизор (англ. hypervisor) — программное обеспечение или слой абстракции, который обеспечивает виртуализацию и управление виртуальными машинами на физическом аппаратном оборудовании.

Дроппер (dropper) — вредоносное программное обеспечение, предназначенное для доставки на компьютер жертвы другого вредоносного ПО.

Полезная нагрузка (payload) — описания действий на компьютере жертвы, которые должны выполнять вирусы.

Реестр — системная база данных с информацией, необходимой для загрузки и настройки конфигурации системы.

Руткит (англ. Rootkit) — разновидность вредоносных программ, позволяющих злоумышленнику незаметно возвращаться во взломанную систему. Это понятие появилось в мире UNIX, где оно означает набор утилит или специальный модуль ядра, которые атакующий устанавливает на взломанной им компьютерной системе сразу после получения прав суперпользователя (корневого пользователя).

Эксплойт (exploit) — подвид вредоносных программ, содержащих исполняемый код, использующий уязвимости в программном обеспечении.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

GDI — Graphic Device Interface — Интерфейс графических устройств Microsoft Windows.

ALPC — Advanced Local Procedure Call — Вызов расширенных локальных процедур.

DLL — dynamic link library — Динамическая подключаемая библиотека.

ОС — Операционная система.

ВПО — Вредоносное программное обеспечение.

ТДП — Таблица дескрипторов прерываний.

ТДС — Таблица дескрипторов служб.

ПО — Программное обеспечение.

ВВЕДЕНИЕ

Актуальность темы. В современном информационном обществе, где цифровые технологии играют все более важную роль, защита персональных компьютеров от вредоносных программ и кибератак становится критически важной задачей. Одними из наиболее опасных вредоносных программ являются программы семейства руткит, в силу своих следующих особенностей: способность скрывать своё присутствие на компьютере и обходить стандартные механизмы обнаружения и удаления вредоносного ПО.

С одной стороны руткиты используются для тестирования безопасности компьютерных систем, в целях повышения безопасности информационных систем, так же они используются для форензических исследований, позволяя получить скрытый доступ к системным данным и артефактам, которые могут быть полезны при выявлении преступлений или нарушений политики безопасности. Так же руткиты могут использоваться для своевременного реагирования на компроментацию, а именно использоваться для обнаружения и устранения вредоносных программ или злоумышленников, которые уже получили несанкционированный доступ к компьютерной системе.

С другой стороны руткиты могут быть использованы для различных криминальных целей, включая кражу личных данных, шпионаж и проведение сетевых атак. Так же руткиты скрывают другие вредоносные программы(вирусы, трояны, шпионские ПО и др.), не вызывая подозрений и избегая обнаружения антивирусными программами. Незаметное выполнение команд: руткит может обеспечить злоумышленникам незаметное выполнение команд на компьютере жертвы. Это может включать удаленное выполнение вредоносного кода, сбор и передачу конфиденциальной информации, запуск дополнительных атак и т. д.. Обход системных механизмов защиты: руткит может изменять системные компоненты, такие как ядро операционной системы или драйверы устройств, чтобы обойти механизмы обнаружения и защиты. Это делает его сложным для обнаружения и удаления. Привилегированный доступ:

руткит может использоваться для получения привилегированного доступа к компьютерной системе, такого как административные права или привилегированный-доступ (на системах UNIX-подобных). Это позволяет злоумышленникам иметь полный контроль над системой и выполнять действия, включая удаление, модификацию или кражу данных. Защита от удаления: руткит может создавать механизмы самозащиты, которые затрудняют его обнаружение и удаление. Это может включать скрытие файлов и процессов, маскировку своей активности или восстановление после удаления.

Несмотря на существующие методы и инструменты обнаружения руткитов, злоумышленники постоянно совершенствуют свои техники, чтобы оставаться незамеченными. Это требует постоянного развития и улучшения существующих методов защиты и создания новых, которые могут эффективно обнаруживать и предотвращать атаки с использованием руткитов.

Таким образом, актуальность исследования методов противодействия руткитам обуславливается необходимостью учета наиболее эффективного и надежного метода для обеспечения высокого уровня безопасности своего компьютера или системы.

Разработанность темы. Вопросам обнаружения и противодействия руткитам посвящены работы следующих российских и зарубежных ученых: Кирилловна Ксения Сергеевна, Цветков Александр Юрьевич, Волкогонов Владимир Никитич, Коркин И. Ю., Бурмистров А.В., Якунина Ю.Ю., Ömer Aslan, Samet Refik, Sutton Sara, Bond Benjamin, Tahiri Sementa, Rrushi Julian, Corregedor Manuel, Von Solms Sebastiaan, Nadim Mohammad, Lee Wonjun, Akopian David, Yan Guanglu, Luo Senlin, Feng Fan, Pan Limin, Safi Qamas Gul Khan, Hu Guangyuan, Zhang Tianwei, Lee Ruby B., Maris Vlad, Pham Duy-Phuc, Marion Damien, Heuser Annelie.

Объект исследования - персональный компьютер с операционной системой Windows, являющийся узлом телекоммуникационной сети.

Предмет исследования - методы противодействия руткитам.

Цель работы - повысить эффективность методов противодействия руткитам, чтобы уменьшить затраты ресурсов системы, за счёт использования эффективных методов обнаружения руткитов.

Задачи исследования - В связи с поставленной целью решались следующие задачи исследования: 1. Составить перечень типовых или наиболее распространенных реализаций руткитов. Провести лабораторные испытания штаммов руткитов для установления их фактической работоспособности. 2. Провести эксперимент с целью установления факта успешного обнаружения типовых руткитов различными методами. 3. Составить сравнительную таблицу методов обнаружения.

Научной новизны в данной исследовательской работе нет.

1 Уточнение задачи исследования

В данной главе приводятся объект и предмет исследования. Анализируются методы противодействия руткитам, уточняются задачи исследования.

1.1 Описание объекта исследования

Объектом исследовательской работы является персональный компьютер с операционной системой Windows, являющийся узлом телекоммуникационной сети. Персональный компьютер с операционной системой Windows, являющийся узлом телекоммуникационной сети, представляет собой компьютерное устройство, работающее под управлением операционной системы Windows и интегрированное в телекоммуникационную сеть. В качестве узла сети, компьютер выполняет функцию передачи, приема и обработки данных, обеспечивая связь между различными устройствами в сети. Такой компьютер может выполнять разнообразные задачи, включая обработку и передачу информации, обмен данными с другими узлами сети, а также обеспечение доступа к ресурсам и услугам сети.

Структурная схема объекта исследования[]:

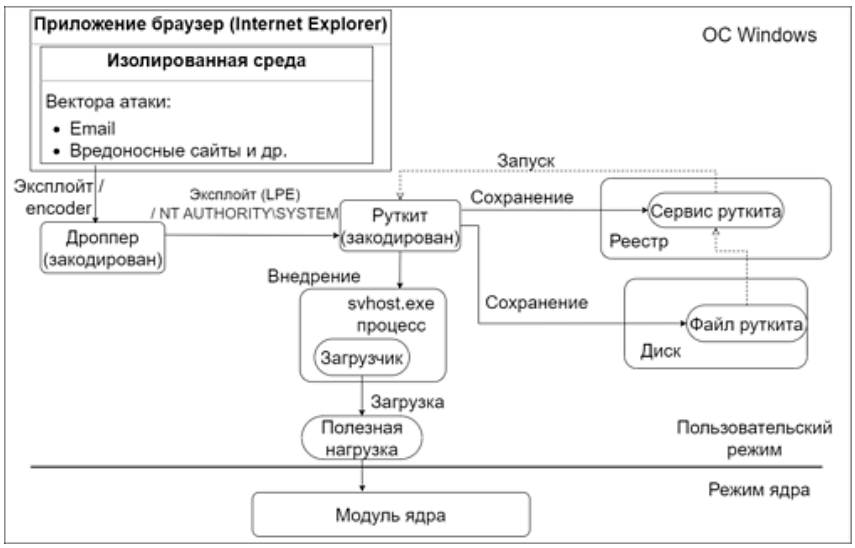

Рисунок 1.1 — Структурная схема объекта исследования

Описание схемы: Пользовательские приложения взаимодействуют с операционной системой на прикладном уровне. Когда компьютер подключается к телекоммуникационной сети через интернет, данные и команды могут быть переданы и получены через Сетевой драйвер. Как часть ядра операционной системы, сетевой драйвер может быть уязвимым местом для атак руткитов. Вредоносное ПО (руткит) может использовать уязвимости в сетевом драйвере для своего внедрения и скрытия от обнаружения. Руткит может модифицировать или заменить функции ядра, чтобы обеспечить свою невидимость и контроль над системой. Таким образом, методы противодействия руткитам должны быть направлены на обнаружение и удаление этих нежелательных изменений в ядре операционной системы, а также на повышение уровня безопасности сетевого драйвера для предотвращения атак из интернета.

Один из наиболее распространенных методов атаки, позволяющих руткиту проникнуть в систему,- использование электронной почты и вредоносных сайтов с целью эксплуатации уязвимостей браузера. Злоумышленники прибегают к социальной инженерии, отправляя электронные письма с опасными вложениями или ссылками на вредоносные веб-ресурсы, которые содержат в себе вредоносные программы, способные использовать уязвимости и браузере.

Таким образом, пользователи могут быть заражены не подозревая об этом, если перейдут по таким ссылкам или откроют вложения, что позволяет руткитам затаиться в системе и обеспечить злоумышленникам несанкционированный доступ. Дроппер содержит код руткита,после загрузки на систему он выполняет свою функцию и распаковывает или устанавливает руткит на диск, руткит может быть скрыт изашифрован для обхода обнаружения. Руткит устанавливает себя как сервис в ОС, что позволяет ему пережить перезагрузку системы.

Он добавляется в реестр ОС, чтобы запускаться при каждом старте, он например внедряется в svhost.exe, чтобы запускаться на ранней стадии загрузки ОС и остаться скрытым от стандартных механизмов обнаружении. Руткит активирует свою полезную нагрузку после загрузки и

внедряется в ядро ОС, где может получить привилегированный доступ и контролировать работу системы.После внедрения в ядро руткит может выполнять различные вредоносные действия, такие как изменение системных файлов, сокрытие своей активности, кража информации и т.п..

Абстрактная схема попадания руткита в систему:

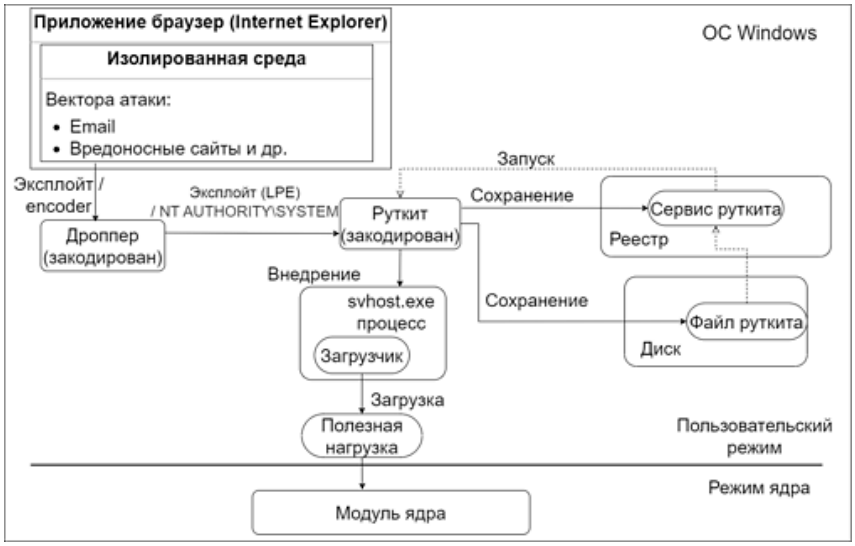

Рисунок 1.2 — Структурная схема объекта исследования

Таким образом, из-за растущих угроз информационной безопасности, необходимо разработать эффективные стратегии и средства для предотвращения внедрения и распространения руткитов. Это позволит обеспечить надежную защиту компьютерных систем и принимать соответствующие меры по укреплению их безопасности.

1.2 Анализ работ по предмету исследования

Вредоносное программное обеспечение (ВПО) семейства руткит представляет собой одну из наиболее угрожающих и хитрых форм кибератак. Определение и объяснение руткитов в различных источниках могут незначительно различаться. Важно отметить, что существует несколько типов руткитов, каждый из которых использует свои уникальные техники и механизмы скрытия. Некоторые руткиты могут проникать в операционную систему на уровне ядра (kernel-mode), изменяя ее функциональность

и манипулируя системными вызовами. Другие могут внедряться на уровне приложений (user-mode) и скрывать свои файлы и процессы от обычных методов обнаружения. Некоторые определяют руткиты как набор инструментов и программных модулей, созданных для обеспечения скрытого и непрерывного доступа к системе, а также подавления и изменения нормального функционирования ее компонентов. Другие определения выделяют руткит как комплекс программ и механизмов, позволяющих злоумышленникам получить привилегированный доступ к операционной системе, управлять ею и обойти средства обнаружения и защиты.

Так, например, в разных статьях приводятся следующие определения:

В статье "ПРОБЛЕМА ОБЕЗВРЕЖИВАНИЯ РУТКИТОВ УРОВНЯ ЯДРА В СИСТЕМАХ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ"[] Кирилловна Ксения Сергеевна, Цветков Александр Юрьевич, Волкогонов Владимир Никитич дают следующее определение руткиту:

Руткит (англ. Rootkit) — разновидность вредоносных программ, позволяющих злоумышленнику незаметно возвращаться во взломанную систему. Это понятие появилось в мире UNIX, где оно означает набор утилит или специальный модуль ядра, которые атакующий устанавливает на взломанной им компьютерной системе сразу после получения прав суперпользователя (корневого пользователя).

В статье "ТЕХНОЛОГИЯ РУТКИТОВ И ОСОБЕННОСТИ ИХ ОБНАРУЖЕНИЯ"[] Бурмистров А.В., Якунина Ю.Ю. предлагают свой вариант определения:

Руткиты - специальные механизмы, используемые вредоносным программным обеспечением (ВПО) для его стабильного и неопределяемого присутствия в компьютерной системе, как для пользователя, так и для средств защиты.

В статье "Implementing Rootkits to Address Operating System Vulnerabilities"[5] Corregedor Manuel, Von Solms Sebastiaa приводят следующее определение:

Руткит - это вредоносная программа или набор программ, которые пытаются скрыть свое существование на зараженном компьютере, атакуя операционную систему (OS) с помощью одного или комбинации следующих действий: изменение бинарных файлов программы, Перехват таблиц вызовов, таких как Системная таблица дескрипторов служб (SSDT) и Таблица дескрипторов прерываний (IDT) для перехвата потока управления ядром, модифицируя легитимный код для принудительного вызова руткит-кода или используя DKOM (прямое манипулирование объектами ядра).

В статье "Position Paper: Consider Hardware-enhanced Defenses for Rootkit Attacks"][12] авторы - Hu Guangyuan, Zhang Tianwei, Lee Ruby B. дали следующее определение:

Руткиты - это вредоносное программное обеспечение, которое пытается скомпрометировать функциональность системы, скрывая своё существование. Было предложено множество различных руткитов, а также различные программные защиты, но лишь очень немногие аппаратные защиты.

В статье "Rootkits and Malicious Code Injection | Journal of Mobile, Embedded and Distributed Systems"[18] Maris Vlad пишет следующее: Руткит можно рассматривать как набор программ, которые изменяют и заменяют существующие пути выполнения в системе. Этот процесс нарушает целостность системы. Википедия определяет руткит как "тип программного обеспечения, предназначенный для получения контроля на уровне администратора над компьютерной системой без обнаружения". Прежде чем переходить к методам противодействия руткитам стоит разобраться с тем, чему мы противостоим, сами механизмы работы руткитов делятся на несколько типов. В своей статье "Технологии сокрытия вредоносных программ и новые способы противодействия им"[8] Коркин И. Ю. представляет хорошую схему классификации ВПО, выглядит она следующим образом:

- Механизмы сокрытия ВПО
  - Стенографические механизмы

- Рутkит-механизмы
  - Механизмы, работающие "внутри"ОС
    - Изменение пути выполнения
    - Изменение структуры памяти
  - Механизмы, работающие "вне"ОС
    - Механизмы, основанные на режиме VT
    - Механизмы, основанные на режиме SMM

Стенографические механизмы скрывают настоящее предназначение программы путём маскировки их под легитимные. Конкретно для моей исследовательской работы интерес вызывают только руткит-механизмы. Руткиты, работающие "внутри"ОС - основываются на изменении пути выполнения или на изменении структур памяти. Руткиты, работающие "вне"ОС - Они представляют собой вредоносные программы, которые заражают компьютер или другое устройство еще до загрузки операционной системы. Это позволяет руткитам оперировать на уровне аппаратного обеспечения, что делает их очень труднодоступными для обнаружения и удаления. В своей статье "ТЕХНОЛОГИЯ РУТКИТОВ И ОСОБЕННОСТИ ИХ ОБНАРУЖЕНИЯ"[] Бурмистров А.В. и Якунина Ю.Ю. так же разделяют руткиты на две основных группы - работающие "внутри"ОС и работающие "вне"ОС.

Для каждой из этих групп руткит-механизмов есть свои методы противодействия. Основная проблема, вызываемая руткитами - это их способность скрываться, из чего следует, что обнаружение руткита - это первый и, можно сказать, основной шаг в противодействии ему. Методы противодействия руткитам нацелены на быстрое обнаружение и эффективное устранение этих скрытых угроз. Целью методов противодействия руткитам является не только обнаружение и удаление самого руткита, но и предотвращение его повторного внедрения, а также минимизация возможных повреждений и ущерба, наносимого компьютерной системе. При разработке эффективных методов противодействия следует учи-

тывать разнообразие и сложность руткитов, а также их способность адаптироваться к изменяющимся условиям и защитным механизмам.

Существует несколько подходов к обнаружению вредоносных программ. В статье "A Comprehensive Review on Malware Detection Approaches"[27] Ömer Aslan, Samet Refik дают общее представление о существующих подходах к обнаружению руткитов. Они выделили 8 основных подходов:

- Основанный на сигнатурах
- Основанный на поведении
- Основанный на эвристике
- Основанный на проверке модели
- Построенный на основе глубокого изучения
- Основанный на облачных вычислениях
- Основанный на мобильных устройствах
- Основанный на интернете вещей

Основанный на сигнатурах: Этот подход использует базы данных сигнатур (уникальных характеристик) известных ВПО для обнаружения подобных угроз. Если обнаруживается сигнатура, совпадающая с определенным типом ВПО, антивирусное программное обеспечение или система считает, что обнаружено ВПО.

Основанный на поведении: Этот подход анализирует поведение программ или системы в реальном времени и обнаруживает аномальные или подозрительные действия, которые могут указывать на наличие ВПО. Он фокусируется на действиях, а не на конкретных сигнатурах.

Основанный на эвристике: Этот подход использует эвристические алгоритмы для поиска необычных или характерных характеристик, которые могут свидетельствовать о наличии ВПО. Он ищет паттерны и поведение, которые могут не быть определенными сигнатурами.

Основанный на проверке модели: Этот подход использует методы формальной верификации для проверки системы или программного обеспечения на соответствие заданным спецификациям или свойствам. Он позволяет автоматически проверить систему на наличие ошибок или уязвимостей на основе формальной модели ее поведения.

Построенный на основе глубокого обучения: Этот подход использует нейронные сети и методы глубокого обучения для обнаружения ВПО. Он основан на анализе больших объемов данных и обучении модели для распознавания характерных признаков ВПО.

Основанный на облачных вычислениях: Этот подход использует облачные ресурсы и сервисы для обнаружения ВПО. Облачные вычисления позволяют эффективно обрабатывать и анализировать большие объемы данных для выявления потенциальных угроз.

Основанный на мобильных устройствах: Этот подход специализируется на обнаружении ВПО на мобильных устройствах, таких как смартфоны и планшеты. Он учитывает уникальные характеристики и уязвимости, связанные с мобильными платформами.

Основанный на интернете вещей: Этот подход направлен на обнаружение ВПО в системах и устройствах, связанных с Интернетом вещей. Он учитывает особенности и уязвимости, связанные с интернетом вещей и обеспечивает безопасность таких устройств и систем.

Авторы статьи выделяют следующие преимущества и недостатки для каждого из подходов:

а) Основанный на сигнатурах

- Плюсы
  - Быстрый и эффективный для известных ВПО
  - Используется в течении многих лет
- Минусы
  - Недостаточно сильный, чтобы обнаруживать ВПО нового поколения

— Чувствителен к ложным срабатываниям

— Извлечение сигнатур тратит много времени

— Уязвим к запутыванию и полиморфным методам

б) Основанный на поведении

- Плюсы

  — Определяет функциональность ВПО

  — Эффективен для обнаружения новых вредоносных программ

  — Эффективен для обнаружения различных вариантов одной и той же вредоносной программы

  — Эффективен против запутывания и полиморфных методов

- Минусы

  — Чувствителен к ложным срабатываниям

  — Сложность в групировке поведения как злонамеренное или нормальное, поскольку некоторые действия или операции, которые проявляются вредоносным программным обеспечением, могут также проявляться в обычных, не вредоносных образцах.

  — Невозможно узнать все варианты поведения

в) Основанный на эвристике

- Плюсы

  — Может обнаружить ранее неизвестные вредоносные программы

  — Может использовать как статические, так и динамические функции

- Минусы

— Много правил и этапов обучения

— Уязвим к метаморфическим подходам

г) Основанный на проверке модели

- Плюсы

  — Эффективен для обнаружения ВПО, принадлежащих к одному семейству

  — Эффективен против запутывания и полиморфных методов

- Минусы

  — Сложная и ресурсоемкая техника

  — Получает ограниченное представление о вредоносном ПО

  — Не удается обнаружить всё ВПО нового поколения

д) Построенный на основе глубокого обучения

- Плюсы

  — Мощный и эффективный

  — Значительно сокращает пространство для объектов

- Минусы

  — Уязвим к атакам-уклонениям

  — Создание и обучение скрытых слоев нейронной сети может занимать значительное кол-во ресурсов и времени

е) Основанный на облачных вычислениях

- Плюсы

  — Повышает эффективность обнаружения на ПК и мобильных устройствах

  — Большие базы данных вредоносных программ и интенсивные вычислительные ресурсы

— Легко доступный, управляемый и регулярно обновляемый

- Минусы

— Отсутствует мониторинг в режиме реального времени

— Может раскрыть некоторые конфиденциальные данные, такие как пароль и местоположение

— Взаимодействие между клиентом и сервером

ж) Основанный на мобильных устройствах

- Плюсы

— Эффективен для обнаружения традиционных и новых вредоносных программ

— Может использовать как статические, так и динамические функции

- Минусы

— Не удаётся обнаружить сложное ВПО

— Не удается масштабировать большой набор приложений

и) Основанный на IoT

- Плюсы

— Может использовать как статические, так и динамические функции

- Минусы

— Не удаётся обнаружить сложное ВПО

В статье "Kernel-level Rootkit Detection, Prevention and Behavior Profiling: A Taxonomy and Survey"[20] Nadim Mohammad, Lee Wonjun, Akopian David приводят таксономию обнаружения руткитов уровня ядра, в добавок к вышеперечисленным методам обнаружения руткитов, можно указать на обнаружение на основе целостности и обнаружение на основе перекрестного просмотра.

- Обнаружение на основе целостности:
  - Целостность статической области
    - Попытка доступа к разделу только для чтения в памяти
    - Хеширование известного региона памяти
    - Политика контроля доступа
    - Динамическое трассирование на уровне страниц
  - Динамическая целостность региона
    - Проверка указателей на функции
    - Разделение распределения данных ядра
    - Безопасное отображение страниц
    - Модель поведения, основанная на событиях

В статье "Countering Malware Via Decoy Processes with Improved Resource Utilization Consistency"[6], исследователи Sutton Sara, Bond Benjamin, Tahiri Sementa и Rrushi Julian проводят работу по применению приманок для борьбы с вредоносными программами. Основной целью исследования является повышение эффективности приманок в обнаружении и противодействии руткитам и другим видам вредоносных программ. В работе предлагается использовать машинное обучение и механизм обучения с использованием карты теплового изображения для улучшения согласованности использования ресурсов приманок. Подход исследователей заключается в создании ложных процессов (приманок), которые имитируют поведение реальных процессов, и обучении их на основе данных о реальных процессах. Таким образом, приманки становятся более реалистичными и чувствительными к изменениям в системе, что позволяет выявлять скрытые вредоносные программы. Эксперименты исследования проводились на машинах в производственной среде, включая процессы OPC-клиента. Результаты показали, что улучшенные приманки демонстрируют динамику использования ресурсов, которая практически неразличима от реальных процессов. Это свидетельствует о повышении

эффективности метода обнаружения руткитов и других вредоносных программ с использованием приманок.

В статье "MOSKG: countering kernel rootkits with a secure paging mechanism"[16] ученые Yan Guanglu, Luo Senlin, Feng Fan, Pan Limin, Safi Qamas Gul Khan разработали "MOSKG"(сделать сноску - "Безопасный механизм пейджинга"относится к системе управления виртуальной памятью в компьютерных операционных системах. Основная идея "безопасного механизма пейджинга"состоит в том, чтобы обеспечить безопасность и конфиденциальность данных, которые хранятся в виртуальной памяти, особенно когда происходит обмен данными между виртуальной и физической памятью.) , представляющий собой прозрачную архитектуру для противодействия руткитам. MOSKG (Multiple Operating Systems Kernel Guard) - это система защиты ядра для борьбы с руткитами, предназначенная для множества операционных систем. Её характеристики включают:

— Прозрачная архитектура: Все компоненты MOSKG находятся в гипервизоре и не видимы для вредоносных программ, что обеспечивает их изоляцию и безопасность.

— Отсутствие ограничений на целевую ОС: MOSKG не ограничивает тип операционной системы, что позволяет применять её на раз личных платформах и архитектурах.

— Абсолютная защита для защитника: Система обеспечивает надежную защиту и изоляцию для своих компонентов, что предотвращает их компрометацию злоумышленниками.

— Приемлемое влияние на производительность: Введение MOSKG не вносит существенной перегрузки, что обеспечивает приемлемую производительность системы.

— Фокус на защите во время выполнения: MOSKG ориентирована на обеспечение защиты виртуальных машин во время их работы, что повышает безопасность системы.

— Доверенная вычислительная база: Аппаратное обеспечение и гипервизор считаются частями доверенной вычислительной базы, что повышает общую надежность системы.

— Безопасная загрузка: MOSKG предполагает, что машина может выполнить безопасную загрузку, что обеспечивает дополнительный уровень защиты.

— Активация до экспозиции целевой ОС зловредному коду: MOSKG активируется до того, как целевая операционная система сталкивается с вредоносным кодом, что увеличивает эффективность её защиты.

MOSKG представляет собой эффективный и безопасный механизм обнаружения и противодействия руткитам, работая на различных операционных системах и обеспечивая надежную защиту во время выполнения. В статье "ULTRA: Ultimate Rootkit Detection over the Air"[21] учёными Pham Duy-Phuc, Marion Damien, Heuser Annelie предлагается фреймворк ULTRA.

ULTRA (ULTimate Rootkit classification and detection over the Air) - это система классификации и обнаружения руткитов, которая использует электромагнитные отклонения (EM trace) и приманки для анализа скрытых (неактивных) руткитов на устройствах. Система работает в черном ящике, где аналитик не имеет предварительных знаний о наличии руткита. ULTRA использует приманки на устройстве для вызова поведения руткитов и обнаружения их характеристик. После предварительной обработки данных система обучает нейронные сети и алгоритмы машинного обучения для обнаружения и классификации руткитов в различных практических сценариях. В статье говорится, что тестирование происходило на ОС Linux, но использование метода ULTRA (ULTimate Rootkit classification and detection over the Air) возможно не только в среде Linux, но и в ОС Windows. Ключевой особенностью ULTRA является то, что она использует электромагнитные отклонения (EM trace) и приманки для анализа руткитов, а не зависит от конкретной операционной системы. Электромагнитные отклонения и поведение руткитов являются характеристиками аппаратного и программного взаимодействия и, следовательно,

не зависят от операционной системы. Приманки также могут быть разработаны таким образом, чтобы вызвать поведение руткитов на устройстве, независимо от того, работает оно на Linux или Windows. Таким образом, метод ULTRA может быть адаптирован и применен для обнаружения руткитов в различных операционных системах, включая Windows.

1.3 Формализация задачи исследования

Для улучшения эффективности противодействия руткитам с помощью методов обнаружения с низким затратом ресурсов на выполнение методов обнаружения можно использовать следующий подход: Представим, что у нас есть следующие данные: Множество обнаруженных и устранённых руткитов: A DR1, DR2, DR3, ... Средние затраты ресурсов на выполнение методов противодействия (например, в процентах от доступных ресурсов): B B данной работе я стремлюсь улучшить эффективность методов противодействия руткитам на персональном компьютере с операционной системой Windows, за счёт методов обнаружения руткитов и затрат ресурсов на выполнение этих методов. Основной метрикой будет "Эффективность обнаружения на единицу ресурса которая измеряется как отношение количества обнаруженных и устраненных руткитов к средним затратам ресурсов. Я хочу оценить, как методы противодействия справляются с обнаружением руткитов с учетом затрат ресурсов. Для этого мы можем использовать следующую метрику: Эффективность обнаружения на единицу ресурса = Коэффициент обнаржуения руткитов(A) / Средние затраты ресурсов методом(B)

Методы с низким временем обнаружения:

— Основанный на целостности: Этот метод обнаружения руткитов обычно имеет наименьшие затраты ресурсов системы, так как он сравнивает хеш-суммы системных файлов и не требует сложных вычислений. Он часто используется для проверки целостности важных системных файлов.

— Основанный на сигнатурах: Следующим по уровню затрат ресурсов идет метод сигнатурного обнаружения. Он обычно быстрый и легок в использовании, но менее эффективен при обнаружении новых или измененных руткитов.

— Основанный на эвристике: Эвристическое обнаружение руткитов может требовать некоторых ресурсов, так как оно использует эвристические алгоритмы для поиска аномалий. Однако оно может быть эффективным при обнаружении неизвестных руткитов.

— Основанный на поведении: Методы, основанные на анализе поведения, могут потреблять больше ресурсов, так как они мониторят активность процессов и служб в реальном времени. Они могут быть более мощными при обнаружении скрытых и новых руткитов.

Комбинация методов с низким затратом ресурсов может быть эффективным для достижения поставленой цели.

Ограничения методов обнаружения с низким временем обнаружения:

— Ложноположительные срабатывания;

— Обход защиты: Руткиты могут быть разработаны так, чтобы обойти методы обнаружения. Они могут использовать различные техники для скрытия своего присутствия и изменения своего поведения, что делает их сложными для обнаружения. Таким образом, возможно, что методы обнаружения не всегда смогут эффективно определить наличие руткита.

— Обновления: Разработчики руткитов постоянно совершенствуют свои программы, чтобы избегать обнаружения.

— Время и ресурсы: При выборе методов противодействия следует учитывать ограничения времени и ресурсов, чтобы они были практичными и эффективными в реальных условиях эксплуатации.

Целевая функция - Минимизировать затраты ресурсов системы при одновременном максимизировании эффективности методов противодействия руткитам.

Целевая функция сводит в себе два основных аспекта цели исследования: уменьшение затрат ресурсов и повышение эффективности методов противодействия руткитам. Для оптимизации этой функции, необходимо находить баланс между использованием ресурсов и способностью обнаруживать и бороться с руткитами.

Ограничения:

Принимать во внимание ограничения времени и ресурсов при выборе методов обнаружения. Использование методов, которые потребуют слишком много времени или ресурсов, может быть непрактичным в реальных условиях. Постоянно обновлять и совершенствовать методы обнаружения, учитывая динамику развития руткитов и сопутствующих угроз.

ЗАКЛЮЧЕНИЕ

В ходе выполнения НИР планируется предоставить:

- Сравнительную таблицу эффективности методов противодействия руткитам, основанную на методах обнаружения руткитов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Baliga Arati. Automated containment of rootkits attacks // Computers & Security. — 2008. — Dec. — Vol. 27, no. 7-8. — P. 323–334. — Publisher: Elsevier Advanced Technology. online; accessed: https://www.sciencedirect.com/science/article/abs/pii/S0167404808000382 (online; accessed: 2023-07-05).

2. Baliga Arati, Iftode Liviu. Automated Defense from Rootkit Attacks // Future Work.

3. Cloaker: Hardware Supported Rootkit Concealment. — online; accessed: https://ieeexplore.ieee.org/abstract/document/4531160/ (online; accessed: 2023-07-05).

4. Cordova Jose, Eaton Virginia, Taylor Kimberly. Experiences in Computer Science Wonderland: A Success Story with Alice // Journal of Computing Sciences in Colleges. — 2011. — May. — Vol. 26. — P. 16–22.

5. Corregedor Manuel, Von Solms Sebastiaan. Implementing rootkits to address operating system vulnerabilities. — online; accessed: https://ieeexplore.ieee.org/abstract/document/6027521/ (online; accessed: 2023-07-05).

6. Countering Malware Via Decoy Processes with Improved Resource Utilization Consistency. — online; accessed: https://ieeexplore.ieee.org/abstract/document/9014383/ (online; accessed: 2023-07-05).

7. Countering Persistent Kernel Rootkits through Systematic Hook Discovery / Wang Zhi, Jiang Xuxian, Cui Weidong and Wang Xinyuan // Recent Advances in Intrusion Detection. — Springer, Berlin, Heidelberg. — 2008. — P. 21–38. — online; accessed: https://link.springer.com/chapter/10.1007/978-3-540-87403-4_2 (online; accessed: 2023-07-05).

8. Countering kernel rootkits with lightweight hook protection | Proceedings of the 16th ACM conference on Computer and communications security. — Archive Location: world. online; accessed: https://dl.acm.org/doi/10.1145/1653662.1653728 (online; accessed: 2023-07-05).

9. Countering unauthorized code execution on commodity kernels: A survey of common interfaces allowing kernel code modification // Computers & Security. — 2011. — Nov. — Vol. 30, no. 8. — P. 571–579. — Publisher: Elsevier Advanced Technology. online; accessed: https://www.sciencedirect.com/science/article/abs/pii/S0167404811001143 (online; accessed: 2023-07-05).

10. Detecting Kernel-Level Rootkits Using Data Structure Invariants. — online; accessed: https://ieeexplore.ieee.org/abstract/document/5551160/ (online; accessed: 2023-07-05).

11. Evaluation of open source anti-rootkit tools. — online; accessed: https://ieeexplore.ieee.org/abstract/document/6707876/ (online; accessed: 2023-07-05).

12. Hu Guangyuan, Zhang Tianwei, B. Lee Ruby. Position Paper: Consider Hardware-enhanced Defenses for Rootkit Attacks | Hardware and Architectural Support for Security and Privacy. — Archive Location: world. online; accessed: https://dl.acm.org/doi/10.1145/3458903.3458909 (online; accessed: 2023-07-05).

13. Information Security Management Handbook, Volume 2. — Access mode: https://books.google.com/books/about/Information_Security_Management_Handbook.html?hl=ru&id=EqpjYH_Z6MQC (online; accessed: 2023-07-05).

14. List Anti Rootkit & AntiVirus for Ubuntu, Linux & BSD (Edition 2018) | Guide books. — Access mode: https://dl.acm.org/doi/abs/10.5555/3285238 (online; accessed: 2023-07-05).

15. Luckett Patrick, McDonald J. Todd, Dawson Joel. Neural Network Analysis of System Call Timing for Rootkit Detection. — on-

line; accessed: https://ieeexplore.ieee.org/abstract/document/7942417/ (online; accessed: 2023-07-05).

16. MOSKG: countering kernel rootkits with a secure paging mechanism / Yan Guanglu, Luo Senlin, Feng Fan, Pan Limin and Safi Qamas Gul Khan // Security and Communication Networks. — 2015. — Dec. — Vol. 8, no. 18. — P. 3580–3591. — Publisher: John Wiley & Sons, Ltd. online; accessed: https://onlinelibrary.wiley.com/doi/10.1002/sec.1282 (online; accessed: 2023-07-05).

17. Malware Guard Extension: abusing Intel SGX to conceal cache attacks / Schwarz Michael, Weiser Samuel, Gruss Daniel, Maurice Clémentine and Mangard Stefan // Cybersecurity. — 2020. — Dec. — Vol. 3, no. 1. — P. 1–20. — Number: 1 Publisher: SpringerOpen. online; accessed: https://link.springer.com/article/10.1186/s42400-019-0042-y (online; accessed: 2023-07-05).

18. Maris Vlad. Rootkits and Malicious Code Injection | Journal of Mobile, Embedded and Distributed Systems. — 2012. — Jan. — online; accessed: http://jmeds.eu/index.php/jmeds/article/view/Rootkits-and-Malicious-Code-Injection (online; accessed: 2023-07-05).

19. Matrosov Alex, Rodionov Eugene, Bratus Sergey. Rootkits and Bootkits. — Access mode: https://books.google.com/books/about/Rootkits_and_Bootkits.html?hl=ru&id=NVv6DwAAQBAJ (online; accessed: 2023-07-05).

20. Nadim Mohammad, Lee Wonjun, Akopian David. Kernel-level Rootkit Detection, Prevention and Behavior Profiling: A Taxonomy and Survey. — 2023. — Apr. — online; accessed: https://arxiv.org/abs/2304.00473v1 (online; accessed: 2023-07-05).

21. Pham Duy-Phuc, Marion Damien, Heuser Annelie. ULTRA: Ultimate Rootkit Detection over the Air // Proceedings of the 25th International Symposium on Research in Attacks, Intrusions

and Defenses. — New York, NY, USA : Association for Computing Machinery. — 2022. — . — RAID '22. — P. 232–251. — Access mode: https://doi.org/10.1145/3545948.3545962 (online; accessed: 2023-07-05).

22. RootkitDet: Practical End-to-End Defense against Kernel Rootkits in a Cloud Environment / Zhang Lingchen, Shetty Sachin, Liu Peng and Jing Jiwu // Computer Security - ESORICS 2014. — Springer, Cham. — 2014. — P. 475–493. — online; accessed: https://link.springer.com/chapter/10.1007/978-3-319-11212-1_27 (online; accessed: 2023-07-05).

23. Rootkits, Spyware/Adware, Keyloggers and Backdoors: Detection and Neutralization. — Access mode: https://books.google.com/books/about/Rootkits_Spyware_Adware_Keyloggers_and_B.html?hl=ru&id=faDVAwAAQBAJ (online; accessed: 2023-07-05).

24. SMM rootkits | Proceedings of the 4th international conference on Security and privacy in communication netowrks. — Archive Location: world. online; accessed: https://dl.acm.org/doi/10.1145/1460877.1460892 (online; accessed: 2023-07-05).

25. Wang Xueyang. NumChecker: A System Approach for Kernel Rootkit Detection and Identification.

26. Zhou Liwei, Makris Yiorgos. Hardware-assisted rootkit detection via on-line statistical fingerprinting of process execution // 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). — 2018. — Mar. — P. 1580–1585. — ISSN: 1558-1101.

27. Ömer Aslan, Samet Refik. A Comprehensive Review on Malware Detection Approaches // IEEE Access. — 2020. — Vol. 8. — P. 6249–6271. — Conference Name: IEEE Access. online; accessed: https://ieeexplore.ieee.org/document/8949524.

ПРИЛОЖЕНИЕ А

Список использованных источников: [1] [2] [4] [5] [12] [15] [18] [19] [20] [3] [9] [8] [10] [13] [14] [23] [24] [27] [21] [11] [17] [6] [7] [25] [16] [22] [26]