Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Реферат

По дисциплине: “Современные методы защиты компьютерных систем”

Вариант 1

Выполнил:

Студент 4 курса

Группы ИИ-21

Богуш А. Д.

Проверил:

Хацкевич М. В.

Брест 2024

Вариант 1(1 mod 3)

1. Логирование. Изменение журнала логирования. PowerShell&Bash.
2. Cyber Kill Chain
3. MITRE ATT&CK
4. SIEM
5. **Логирование. Изменение журнала логирования. PowerShell&Bash.**

Логирование — это процесс записи информации о событиях, происходящих в системе или приложении, в структурированном виде, называемом журналом логов. Логирование является ключевым элементом обеспечения безопасности и мониторинга в компьютерных системах. Оно позволяет отслеживать действия пользователей, системные изменения, а также выявлять ошибки и инциденты безопасности.

**Основные цели логирования**

1. **Мониторинг и аудит**
   * Логирование позволяет контролировать действия пользователей и системных процессов, фиксируя все изменения в системе.
   * Используется для выполнения регулярного аудита системы, который помогает выявлять отклонения от нормальной работы.
2. **Обеспечение безопасности**
   * Логи служат источником информации для выявления попыток несанкционированного доступа, анализа кибератак и расследования инцидентов.
   * Позволяют отслеживать и предотвращать потенциальные угрозы.
3. **Отладка и диагностика**
   * Логи используются для анализа ошибок, возникающих в приложениях или операционных системах, что упрощает процесс их устранения.
   * Помогают выявить узкие места и улучшить производительность системы.
4. **Соблюдение нормативных требований**
   * Многие стандарты и законы (например, GDPR, PCI DSS, HIPAA) требуют хранения логов для обеспечения прозрачности операций.

**Типы логов**

1. **Системные логи**: содержат данные о работе операционной системы, например, запуск и завершение процессов, ошибки оборудования и событий.
2. **Приложенческие логи**: фиксируют работу отдельных приложений, включая их ошибки и взаимодействие с пользователями.
3. **Сетевые логи**: содержат информацию о сетевых соединениях, такие как исходные и конечные IP-адреса, протоколы, используемые порты.
4. **Безопасностные логи**: включают данные о попытках авторизации, доступе к конфиденциальным данным и другие события, влияющие на безопасность.

**Ключевые элементы логирования**

1. **Временные метки**: фиксация времени каждого события для точного анализа последовательности действий.
2. **Идентификаторы пользователей или процессов**: указание источника события.
3. **Код события**: уникальный идентификатор типа события для упрощения поиска.
4. **Описание события**: краткая информация о произошедшем событии.
5. **Уровни логирования**:
   * DEBUG: детальная информация для отладки.
   * INFO: информационные сообщения о штатной работе системы.
   * WARNING: предупреждения о потенциальных проблемах.
   * ERROR: ошибки, которые могут повлиять на работу системы.
   * CRITICAL: критические сбои, требующие немедленного вмешательства.

**Практика и технологии логирования**

Для реализации логирования применяются различные инструменты и технологии. Вот основные из них:

1. **Средства логирования в Windows и Linux**:
   * Windows Event Log — встроенная система логирования для ОС Windows.
   * Syslog — стандартный протокол для логирования в UNIX-подобных системах.
2. **Фреймворки для приложений**:
   * Log4j, Logback (для Java).
   * NLog, Serilog (для .NET).
   * Winston (для Node.js).
3. **Централизация логов**:
   * Для управления большими объемами логов используются системы централизации, такие как Elastic Stack (ELK), Splunk, Graylog.
   * Эти системы обеспечивают сбор, хранение, анализ и визуализацию логов.

**Важность корректной конфигурации логирования**

Правильная настройка логирования — это основа его эффективности. Она включает в себя:

1. **Определение необходимых уровней логирования**: логирование слишком большого объема данных может привести к перегрузке системы, а недостаточное — к потере важных сведений.
2. **Настройка ротации логов**: автоматическое удаление или архивирование устаревших логов для освобождения места.
3. **Обеспечение защиты логов**: предотвращение несанкционированного доступа и изменения журналов логирования.
4. **Соблюдение требований хранения**: логи должны храниться определённое время в соответствии с нормативными актами.

Изменение журнала логирования — это процесс модификации записей в логах, который может быть как легитимным, так и злонамеренным. Внесение изменений в журналы логов играет важную роль в следующих случаях:

1. **Корректировка ошибок**: исправление некорректно записанных данных, например, из-за ошибок в конфигурации.
2. **Обеспечение совместимости**: приведение формата логов в соответствие с требованиями системы анализа.
3. **Мошеннические действия**: злоумышленники могут изменять или удалять логи для сокрытия следов своей активности.

#### Риски, связанные с изменением логов

1. **Потеря достоверности**: любое изменение журнала может снизить доверие к его содержимому.
2. **Скрытие инцидентов**: удаление или модификация записей затрудняет расследование инцидентов безопасности.
3. **Нарушение нормативных требований**: изменения могут привести к несоблюдению стандартов хранения логов.

#### Средства защиты от несанкционированных изменений

1. **Контроль доступа**: ограничение прав доступа к журналам логирования только для авторизованных пользователей.
2. **Хэширование логов**: использование криптографических методов для проверки целостности записей.
3. **Резервное копирование**: регулярное создание резервных копий журналов для предотвращения их полной потери.
4. **Системы неизменяемого логирования**: использование решений, которые предотвращают модификацию записей (например, WORM-хранилища).

#### Процедуры легитимного изменения логов

1. **Журналирование изменений**: каждая модификация логов должна фиксироваться в отдельном журнале.
2. **Аутентификация и авторизация**: изменения могут вноситься только пользователями с соответствующими полномочиями.
3. **Согласование с политиками компании**: процесс изменения должен быть строго регламентирован внутренними стандартами.

#### PowerShell: обзор и функциональные возможности

PowerShell — это инструмент командной строки и язык сценариев, разработанный Microsoft, который используется для автоматизации задач управления и настройки систем. Он интегрируется с .NET Framework и .NET Core, что делает его мощным средством для работы с объектами и системными ресурсами.

Основные особенности PowerShell

1. **Объектно-ориентированная архитектура**:
   * Все данные в PowerShell представлены в виде объектов, что упрощает их обработку и передачу.
2. **Модули и скрипты**:
   * Поддержка модульной структуры позволяет расширять возможности PowerShell.
   * Скрипты пишутся в текстовых файлах с расширением .ps1.
3. **Интеграция с Windows**:
   * Возможность управления всеми аспектами системы: от реестра до служб и процессов.
4. **Кроссплатформенность**:
   * С PowerShell 6.0 доступна поддержка Linux и macOS.

Примеры задач, решаемых с помощью PowerShell

1. Управление пользователями и группами в Active Directory.
2. Автоматизация процессов резервного копирования и восстановления данных.
3. Мониторинг состояния системы и диагностика неисправностей.
4. Анализ и обработка логов.

#### Bash: обзор и функциональные возможности

Bash (Bourne Again Shell) — это Unix-оболочка и язык сценариев, широко используемый в Linux и других Unix-подобных системах. Bash предоставляет мощные инструменты для автоматизации и управления системами через командную строку.

Основные особенности Bash

1. **Простота и гибкость**:
   * Bash предоставляет простой синтаксис для выполнения системных операций и управления файлами.
2. **Скрипты**:
   * Скрипты пишутся в текстовых файлах с расширением .sh.
3. **Пайплайны и редиректы**:
   * Bash позволяет организовывать сложные цепочки операций с использованием пайпов (|) и перенаправления ввода/вывода.

Примеры задач, решаемых с помощью Bash

1. Написание автоматизированных сценариев для резервного копирования.
2. Мониторинг использования ресурсов с использованием встроенных утилит (например, top, df, ps).
3. Управление пакетами и обновлениями в Linux.
4. Сбор и обработка логов системных процессов.

#### Сравнение PowerShell и Bash

|  |  |  |
| --- | --- | --- |
| **Особенность** | **PowerShell** | **Bash** |
| Платформенность | Windows, Linux, macOS | Linux, macOS, Windows через WSL |
| Подход к данным | Объектно-ориентированный | Тексто-ориентированный |
| Интеграция с ОС | Глубокая интеграция с Windows | Глубокая интеграция с Linux |
| Простота использования | Более сложный синтаксис, но мощные возможности | Простой синтаксис, меньше возможностей |
| Область применения | Управление инфраструктурой, Windows | Администрирование Linux, автоматизация |

1. **Cyber Kill Chain**

Cyber Kill Chain — это концептуальная модель, разработанная компанией Lockheed Martin, которая описывает этапы атаки на информационные системы. Эта модель используется для выявления и предотвращения кибератак на разных стадиях их осуществления. Она предоставляет организациям структуру для анализа угроз, улучшения защитных механизмов и разработки эффективных стратегий реагирования.

#### Этапы модели Cyber Kill Chain

1. **Разведка (Reconnaissance)**
   * Цель: сбор информации о цели атаки.
   * Действия атакующих:
     + Использование открытых источников (OSINT) для сбора данных о сетевой инфраструктуре, доменах, IP-адресах.
     + Поиск уязвимостей и слабых мест в системе.
   * Методы обнаружения:
     + Мониторинг подозрительных запросов и сканирований сети.
     + Анализ логов систем безопасности.
2. **Вооружение (Weaponization)**
   * Цель: создание инструмента атаки.
   * Действия атакующих:
     + Разработка вредоносного ПО (вирусов, червей, троянов).
     + Создание фишинговых писем с вредоносными вложениями.
     + Формирование эксплойтов для известных уязвимостей.
   * Методы обнаружения:
     + Анализ вредоносного кода.
     + Использование песочниц (sandbox) для исследования поведения программ.
3. **Доставка (Delivery)**
   * Цель: доставка инструмента атаки к целевой системе.
   * Действия атакующих:
     + Использование электронной почты (фишинг).
     + Распространение через USB-носители.
     + Сканирование уязвимых портов.
   * Методы обнаружения:
     + Настройка фильтров для блокировки подозрительных вложений и ссылок.
     + Мониторинг активности сетевого трафика.
4. **Эксплуатация (Exploitation)**
   * Цель: использование уязвимости в системе для выполнения вредоносных действий.
   * Действия атакующих:
     + Запуск вредоносного кода.
     + Эксплуатация уязвимостей в программном обеспечении.
   * Методы обнаружения:
     + Использование систем обнаружения вторжений (IDS/IPS).
     + Регулярное обновление и патчинг систем.
5. **Установка (Installation)**
   * Цель: закрепление вредоносного ПО в системе для дальнейших действий.
   * Действия атакующих:
     + Установка бэкдоров.
     + Регистрация вредоносных программ в автозапуске системы.
   * Методы обнаружения:
     + Мониторинг изменений в реестре и конфигурациях.
     + Обнаружение аномалий в процессах системы.
6. **Управление и контроль (Command and Control, C2)**
   * Цель: установление связи между атакующим и вредоносным ПО.
   * Действия атакующих:
     + Использование скрытых каналов связи (DNS-туннелирование, зашифрованный трафик).
     + Управление заражённой системой.
   * Методы обнаружения:
     + Анализ сетевого трафика.
     + Блокировка доступа к известным серверам C2.
7. **Достижение цели (Actions on Objectives)**
   * Цель: выполнение конечной задачи атаки (например, кража данных, разрушение систем).
   * Действия атакующих:
     + Эксфильтрация данных.
     + Уничтожение или шифрование файлов.
   * Методы обнаружения:
     + Анализ аномального поведения пользователей.
     + Использование систем предотвращения утечек данных (DLP).

#### Применение Cyber Kill Chain

1. **Укрепление киберзащиты**
   * Идентификация слабых мест в существующих механизмах защиты.
   * Разработка мер противодействия для каждого этапа атаки.
2. **Анализ инцидентов безопасности**
   * Использование модели для анализа успешных атак.
   * Определение точки входа и путей распространения угроз.
3. **Обучение персонала**
   * Использование модели для тренировки сотрудников в области информационной безопасности.
   * Проведение симуляций атак для проверки готовности систем и персонала.
4. **Внедрение инструментов мониторинга**
   * Использование SIEM-систем для централизованного анализа событий безопасности.
   * Применение IDS/IPS для автоматического обнаружения и блокировки угроз.

#### Преимущества и ограничения модели Cyber Kill Chain

##### **Преимущества**:

1. **Структурированный подход**: чёткое представление этапов атаки упрощает анализ и разработку стратегий защиты.
2. **Проактивная защита**: возможность предотвратить угрозу до её реализации.
3. **Интеграция с современными инструментами**: модель может использоваться совместно с MITRE ATT&CK и другими фреймворками.

##### **Ограничения**:

1. **Сосредоточенность на внешних угрозах**: модель меньше подходит для анализа внутренних инцидентов.
2. **Фокус на линейных атаках**: современные угрозы могут включать комбинированные и многоступенчатые методы.
3. **Требования к ресурсам**: внедрение и поддержка модели могут быть затратными.

**3. MITRE ATT&CK**

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) — это фреймворк, разработанный компанией MITRE, который представляет собой глобальную базу знаний о тактиках и техниках кибератак. ATT&CK используется специалистами по кибербезопасности для анализа угроз, разработки защитных стратегий и повышения осведомлённости о поведении злоумышленников.

#### Основные элементы MITRE ATT&CK

1. **Тактики (Tactics)**
   * Тактики описывают основные цели злоумышленников на разных этапах атаки.
   * Примеры тактик:
     + Initial Access (Начальный доступ): методы, позволяющие злоумышленникам проникнуть в систему.
     + Persistence (Закрепление): обеспечение долгосрочного доступа к системе.
     + Exfiltration (Эксфильтрация): кража данных.
2. **Техники (Techniques)**
   * Техники детализируют, как именно злоумышленники достигают своих целей.
   * Пример: использование фишинга для получения учётных данных (Credential Phishing).
3. **Подтехники (Sub-Techniques)**
   * Подтехники углубляют описание техник, добавляя специфические детали.
   * Пример: Credential Phishing может включать использование вредоносных вложений в электронной почте.
4. **Примеры случаев (Use Cases)**
   * Реальные примеры атак, использующих определённые тактики и техники.
5. **Соответствие инструментам и программам**
   * База знаний содержит информацию о вредоносных программах, инструментах и группах злоумышленников.

#### Структура фреймворка

MITRE ATT&CK представлен в виде матрицы, которая организует тактики и техники в логической последовательности. Матрица делится на следующие типы:

1. **Enterprise ATT&CK**
   * Для корпоративных систем.
   * Включает платформы: Windows, Linux, macOS, Cloud.
2. **Mobile ATT&CK**
   * Для мобильных устройств.
   * Описывает угрозы для Android и iOS.
3. **ICS ATT&CK**
   * Для систем управления промышленными процессами (Industrial Control Systems).

#### Применение MITRE ATT&CK

1. **Анализ угроз**
   * Определение используемых злоумышленниками техник и тактик.
   * Построение профилей угроз на основе известных групп атакующих (APT).
2. **Тестирование безопасности**
   * Использование фреймворка для проверки эффективности защитных мер.
   * Проведение Red Team/Blue Team упражнений.
3. **Обучение и осведомлённость**
   * Тренировка специалистов по кибербезопасности.
   * Обучение пользователей распознаванию потенциальных атак.
4. **Разработка SIEM-правил**
   * Интеграция ATT&CK с системами мониторинга и анализа событий.

#### Преимущества MITRE ATT&CK

1. **Глобальная база знаний**
   * Фреймворк постоянно обновляется на основе реальных инцидентов.
2. **Прозрачность и доступность**
   * ATT&CK является открытым ресурсом, доступным для всех.
3. **Универсальность**
   * Фреймворк подходит для организаций любого масштаба.
4. **Интеграция с другими инструментами**
   * ATT&CK может быть интегрирован с SIEM, EDR и другими системами защиты.

#### Ограничения MITRE ATT&CK

1. **Фокус на известных методах**
   * Фреймворк описывает уже известные угрозы и может быть менее эффективным для защиты от новых техник.
2. **Сложность внедрения**
   * Требует ресурсов и времени для эффективного применения.
3. **Зависимость от данных**
   * Полезность ATT&CK напрямую связана с качеством данных, собираемых системой.
4. **SIEM**

SIEM (Security Information and Event Management) — это класс систем, предназначенных для сбора, анализа, корреляции и управления событиями информационной безопасности в реальном времени. SIEM объединяет функции управления журналами событий (Log Management) и систем обнаружения угроз, позволяя организациям оперативно реагировать на инциденты безопасности.

#### Основные функции SIEM

1. **Сбор данных**
   * SIEM собирает данные из различных источников:
     + Лог-файлы серверов и приложений.
     + Сетевые устройства (маршрутизаторы, межсетевые экраны).
     + Антивирусы и системы обнаружения вторжений (IDS/IPS).
2. **Корреляция событий**
   * Анализ собранных данных для выявления взаимосвязей между событиями.
   * Пример: аномальная активность на рабочей станции в сочетании с попытками удалённого доступа может указывать на взлом.
3. **Обнаружение угроз**
   * Использование предопределённых и настраиваемых правил для обнаружения подозрительных событий.
   * Применение методов машинного обучения для выявления аномалий.
4. **Реагирование на инциденты**
   * Автоматическое уведомление администраторов о выявленных угрозах.
   * Возможность автоматического принятия мер, например, блокировки IP-адресов злоумышленников.
5. **Отчёты и аналитика**
   * Создание отчётов о состоянии системы безопасности.
   * Помощь в соблюдении требований нормативного соответствия (например, GDPR, ISO 27001).

#### Архитектура SIEM

1. **Сбор данных**
   * Агенты или встроенные механизмы собирают события из различных источников.
2. **Хранилище данных**
   * Данные сохраняются для дальнейшего анализа и аудита.
   * Используются базы данных или системы больших данных (Big Data) для обработки больших объёмов информации.
3. **Модуль корреляции**
   * Система анализа событий в режиме реального времени и исторических данных.
4. **Интерфейс управления**
   * Панель мониторинга для визуализации событий и управления настройками.

#### Преимущества SIEM

1. **Централизованный мониторинг**
   * Все события безопасности собираются и анализируются в единой системе.
2. **Повышение эффективности реагирования**
   * Быстрое обнаружение и реакция на инциденты.
3. **Соответствие требованиям**
   * Автоматизация создания отчётов для соблюдения нормативных требований.
4. **Масштабируемость**
   * Возможность обработки больших объёмов данных и добавления новых источников.

#### Ограничения SIEM

1. **Высокая стоимость**
   * Внедрение и обслуживание SIEM-систем требуют значительных финансовых вложений.
2. **Сложность настройки**
   * Требуется время и квалификация для адаптации системы под нужды организации.
3. **Зависимость от качества данных**
   * Эффективность работы SIEM зависит от полноты и точности поступающей информации.
4. **Ложные срабатывания**
   * Возможны ошибки при интерпретации событий, что может приводить к избыточным предупреждениям.

#### Примеры SIEM-систем

1. **Коммерческие решения**
   * Splunk
   * IBM QRadar
   * ArcSight
   * SolarWinds
2. **Открытые решения**
   * ELK Stack (Elasticsearch, Logstash, Kibana)
   * Wazuh

#### Применение SIEM в реальной практике

1. **Обнаружение угроз**
   * Внедрение правил для выявления известных угроз.
   * Пример: обнаружение активности, связанной с ботнетами.
2. **Анализ инцидентов**
   * Использование исторических данных для расследования инцидентов безопасности.
3. **Оптимизация процессов**
   * Автоматизация управления событиями и предупреждений.
4. **Интеграция с другими системами**
   * SIEM может взаимодействовать с EDR, DLP и системами управления уязвимостями.