МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

РЕФЕРАТ

по дисциплине
«Современные методы защиты компьютерных систем»

Выполнил:
студент 4-го курса,
ФЭИС,
группы ИИ-22
Борейша О.С.

Брест 2024

1 ЛОГИРОВАНИЕ WINDOWS LINUX

Логирование — это процесс записи системных событий, ошибок, действий пользователей и приложений, используемый для мониторинга, анализа и устранения неисправностей. Операционные системы Windows и Linux имеют свои особенности в подходах к логированию.

Логирование в Windows

Windows использует Event Viewer (Просмотр событий) для управления логами. Основной механизм сбора и хранения данных — это служба Windows Event Log. События разделяются на несколько журналов:

1. Системный журнал — записи о работе операционной системы, драйверов и служб.
2. Журнал безопасности — информация об успешных и неудачных попытках входа в систему, а также изменениях в безопасности.
3. Журнал приложений — сообщения от установленных программ.
4. Журналы приложений и служб — данные о работе отдельных приложений и служб, включая специальные логи.

Журналы хранятся в формате .evtx и могут быть просмотрены через Event Viewer или экспортированы для анализа. Для автоматизации работы с логами используется PowerShell и инструменты, такие как Windows Management Instrumentation (WMI).

Логирование в Linux

В Linux основным инструментом является система журнала syslog (или её модернизированная версия systemd-journald в современных дистрибутивах). Она позволяет собирать, фильтровать и отправлять логи в центральное хранилище.

Основные журналы:

1. /var/log/syslog — системные события и сообщения от большинства приложений.
2. /var/log/auth.log — записи о входах, выходах и изменениях прав.
3. /var/log/kern.log — сообщения ядра системы.
4. /var/log/dmesg — информация о загрузке системы и устройстве оборудования.

Для анализа логов часто используются команды:

- tail и less для просмотра логов в режиме реального времени.
- grep для поиска определённых сообщений.
- logrotate для управления размерами файлов логов и их архивации.

Сравнение

Windows предоставляет удобный графический интерфейс для анализа логов,

что делает её более доступной для администраторов без глубоких знаний. Linux, напротив, требует работы с текстовыми файлами и консольными командами, что обеспечивает большую гибкость и автоматизацию.

Логирование — ключевой инструмент для обеспечения безопасности и стабильности системы, вне зависимости от используемой операционной системы. Однако выбор подходов и инструментов зависит от специфики задач и уровня подготовки администратора.

2 POWERSHELL BASH

Windows и Linux являются очень производительными операционными системами, и у каждой из них есть множество плюсов и минусов, которые мы можем обсудить. Но мы часто не задумываемся о потенциале сценариев и автоматизации двух операционных систем.

Немного истории: PowerShell

PowerShell - это среда автоматизации и задач Microsoft, удобная для управления конфигурацией. PowerShell использует компоненты, называемые командлетами, которые встроены в PowerShell. Дополнительные функции доступны через модули. Они устанавливаются из галереи PowerShell непосредственно из командной строки.

PowerShell отличается от Bash, потому что он предназначен для взаимодействия со структурами .NET изначально в Windows. Это означает, что он может передавать объекты и данные между сценариями, приложениями и сеансами. Каждый объект имеет свой собственный набор свойств, что делает обработку данных в PowerShell еще более детальной. Данные могут быть указаны как числа (целые числа), слова (строки), логические (истина и ложь) и многие другие типы. Это означает, что вы можете по-настоящему определиться с тем, как ваши скрипты обрабатывают ввод и вывод данных.

Немного истории: Bash

Системы Linux и Unix всегда выигрывали от структурирования многопользовательской терминальной среды. Это означает, что вы можете запускать дополнительные сессии в той же системе и запускать сценарии и приложения, не влияя на основные сессии, в которые вошли другие пользователи. Это сильно отличалось от ранних систем Windows и DOS, которые были однопользовательскими средами с одной сессией, до появления Windows NT в середине 90-х годов.

Первоначальная оболочка, поставляемая с Unix, была известна как оболочка Bourne, названная в честь ее создателя Стивена Борна. Bash (Bourne again Shell) является преемником оболочки Bourne с открытым исходным кодом. Bash получил широкое распространение, когда Linux был создан в начале 90-х годов, поэтому он используется до сих пор.

Существует множество функций, которые делают Bash очень популярным, главными из которых являются стабильность системы и то, что это открытый

исходный код. Из-за этого он встречается практически в каждом дистрибутиве Linux. Все эти факторы делают его одной из наиболее часто используемых сред сценариев для ИТ-специалистов.

Когда использовать PowerShell

Администрирование Windows стало намного проще с тех пор, как разработка PowerShell стала частью среды Microsoft. Вместо того, чтобы бороться с неудобными пакетными файлами и планировщиком Windows, системные администраторы получают доступ к новому набору инструментов с впечатляющими приложениями и функциями.

PowerShell может уточнять детали для создания эффективных скриптов, а также некоторых коммерчески доступных приложений. PowerShell может извлекать данные прямо из подсистемы WMI, предоставляя вам в режиме реального времени глубокую информацию обо всем, от идентификаторов процессов и счетчиков обработчиков.

PowerShell включен в платформу .NET, поэтому вы можете создавать великолепно выглядящие меню и формы winform. Вы можете использовать PowerShell, чтобы делать что угодно - от запросов к базам данных SQL до захвата ваших любимых RSS-каналов прямо в сеанс PowerShell для дальнейших манипуляций. Это настоящий швейцарский нож для системного администрирования в среде Windows.

Когда использовать Bash

Если вы используете системы Linux, значит, вы знаете о необходимости автоматизации задач. Ранние ленточные накопители использовались для резервного копирования с архивированием tar. Эти операции могут быть написаны в Bash, а затем запущены через расписание cron. Сегодня мы воспринимаем подобные вещи как должное, но многие задачи приходилось выполнять вручную до создания таких сред, как Bash. Все, что связано с манипуляциями с файлами, такими как архивирование, копирование, перемещение, переименование и удаление файлов, подходит Bash.

Также возможны более сложные манипуляции с файлами. Вы можете найти файлы, созданные в определенные даты, и для каких файлов были изменены права доступа CHMOD и владельца. Bash также отлично подходит для создания интерактивных меню для запуска скриптов и выполнения системных функций. Они выполняются в неграфической среде, но работают очень хорошо. Это отлично подходит для обмена вашими библиотеками скриптов с другими.

Отличия

PowerShell и Bash в чем-то похожи, но также очень разные. Вот основные отличия:

PowerShell отличается от Bash способом обработки данных. PowerShell - это язык сценариев, но он может передавать данные в разных форматах таким образом, чтобы он выглядел как язык программирования. PowerShell также имеет дело с областями действия в своих скриптах.

Использование переменных с $session, $script и $cache дает вашим сценариям дополнительную гибкость, позволяя передавать переменные другим командам в том же сценарии или сеансе PowerShell.

Bash - это CLI

Bash - это CLI (Command Language Interpreter), что означает интерпретатор командного языка. Как и PowerShell, Bash может передавать данные между командами по каналам. Однако эти данные отправляются в виде строк. Это ограничивает некоторые вещи, которые вы можете делать с выводом ваших скриптов, например математические функции.

PowerShell - это и CLI, и язык

Интегрированная среда сценариев PowerShell по умолчанию (ISE - Integrated Scripting Environment), поставляемая с Windows, показывает, как можно быстро и легко создавать сценарии, не жертвуя прямым доступом к командной строке. По умолчанию верхний раздел позволяет набирать строки кода сценария и быстро его тестировать.

Окно ниже представляет собой командную строку PowerShell, которая дает вам быстрый доступ для выполнения отдельных команд. Это дает вам лучшее из обоих миров между языком сценариев и оболочкой командной строки. ISE - отличный инструмент для быстрого создания прототипов решений.

PowerShell и Bash - мощные инструменты

Среда, в которой вы работаете, определит, какой инструмент вы выберете. Системные администраторы Linux, пишущие сценарии в Bash, считают, что освоить сценарии PowerShell относительно легко. Навыки написания сценариев PowerShell также в определенной степени переносятся на сценарии Bash.

Основными различиями между этими двумя языками сценариев являются синтаксис и обработка данных. Если вы понимаете такие концепции, как переменные и функции, тогда изучение любого из этих языков становится проще.

3 CYBER KILL CHAIN

Cyber Kill Chain — это концепция, разработанная компанией Lockheed Martin, которая описывает этапы кибератаки от первоначального рекогносцирования до достижения конечных целей злоумышленника. Эта модель используется для анализа кибератак, выработки стратегии защиты и построения системы безопасности.

Этапы Cyber Kill Chain

Модель включает 7 ключевых этапов, каждый из которых охватывает важные шаги в процессе атаки:

1. Рекогносцирование (Reconnaissance)
   Злоумышленники собирают информацию о цели: изучают инфраструктуру, уязвимости, персонал и связанные с ним данные.
   Примеры: анализ публичных ресурсов, поиск данных в социальных сетях, сканирование сети.
2. Вооружение (Weaponization)
   На этом этапе создаётся эксплойт или вредоносное ПО, которое будет использовано для атаки. Оно может быть настроено для доставки в целевую систему.
   Примеры: создание троянов, внедрение эксплойтов в документы или ссылки.
3. Доставка (Delivery)
   Вредоносное ПО передаётся в целевую систему. Этот этап считается одним из самых критичных, поскольку требует взаимодействия с жертвой.
   Методы доставки: фишинговые письма, заражённые веб-сайты, USB-устройства.
4. Эксплуатация (Exploitation)
   На данном этапе злоумышленник использует уязвимости для выполнения вредоносного кода. Это может быть пробел в безопасности приложения или операционной системы.
   Примеры: запуск вредоносного кода через уязвимость в браузере или программе.
5. Установка (Installation)
   После успешной эксплуатации вредоносное ПО устанавливается в системе. Это позволяет злоумышленнику закрепить своё присутствие.
   Примеры: установка руткитов, создание бэкдоров для повторного доступа.
6. Управление (Command and Control, C2)
   Вредоносное ПО связывается с сервером управления, позволяя злоумышленнику контролировать систему удалённо.
   Используемые техники: зашифрованные каналы связи, использование стеганографии для передачи данных.
7. Действия на цели (Actions on Objectives)
   Финальный этап, где злоумышленник достигает своей цели: кража данных, уничтожение информации, нарушение работы систем.
   Примеры: кража интеллектуальной собственности, внедрение программ-вымогателей.

Применение Cyber Kill Chain

1. Выявление угроз: Модель позволяет определить этапы, на которых атака может быть прервана.
2. Разработка защитных мер: Организации могут усилить защиту на каждом этапе цепочки.
3. Анализ атак: Пост-атаковый разбор помогает понять, как злоумышленники достигли своих целей и что нужно улучшить.

Противодействие на этапах цепочки

- Рекогносцирование: мониторинг сетевой активности, ограничение доступа к внутренней информации.
- Доставка: обучение сотрудников распознаванию фишинга, использование фильтров электронной почты.
- Установка и управление: использование антивирусного ПО, сегментация сети, контроль за передачей данных.

Вывод

Модель Cyber Kill Chain помогает организациям систематизировать защиту от кибератак, понимая действия злоумышленников и блокируя их на ранних этапах. Это важный инструмент для построения стратегий кибербезопасности.

4 MITRE ATT&CK

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) — это глобальная база данных тактик и техник, используемых злоумышленниками для атак на информационные системы. Она разработана компанией MITRE с целью помочь организациям выявлять, предотвращать и реагировать на киберугрозы.

Основные цели MITRE ATT&CK

1. Углублённое понимание атак: база данных помогает понять методы, которые применяют злоумышленники.
2. Укрепление защиты: позволяет строить эффективные стратегии обнаружения и противодействия атакам.
3. Оценка текущих систем безопасности: организации могут использовать фреймворк для анализа пробелов в защите.

Структура MITRE ATT&CK

MITRE ATT&CK включает тактики, техники и подтипы атак, которые описывают действия злоумышленников:

1. Тактики (Tactics) — это цели, которых пытается достичь злоумышленник (например, выполнение вредоносного кода, повышение привилегий, эксфильтрация данных).
2. Техники (Techniques) — конкретные методы, используемые для достижения тактик (например, использование PowerShell, инъекции DLL, фишинг).
3. Процедуры (Procedures) — описания того, как конкретные угрозы реализуют техники.

Основные домены ATT&CK

1. Enterprise: охватывает атаки на корпоративные IT-системы.
2. Mobile: описывает методы атак на мобильные устройства.
3. ICS (Industrial Control Systems): включает угрозы для промышленных систем управления.

Пример тактик и техник

1. Тактика**:** Initial Access (Первоначальный доступ) Техника: фишинг (Phishing).
2. Тактика**:** Privilege Escalation (Повышение привилегий) Техника: использование уязвимости в драйверах.
3. Тактика**:** Exfiltration (Эксфильтрация данных) Техника: сжатие данных перед передачей (Data Compressed).

Применение MITRE ATT&CK

1. Обнаружение атак: с помощью базы данных организации могут настроить системы мониторинга (SIEM) для выявления подозрительных действий.
2. Обучение сотрудников: ATT&CK помогает специалистам лучше понять методы злоумышленников.
3. Красные и синие команды: используется для имитации атак (красные команды) и анализа защиты (синие команды).
4. Оценка текущей безопасности: сравнение действий злоумышленников с текущими мерами защиты.

Преимущества

- Детализация: описывает как общие тактики, так и редкие техники.
- Обновляемость: база данных регулярно обновляется, чтобы включать новые угрозы.
- Интеграция: ATT&CK используется многими инструментами кибербезопасности.

Вывод

MITRE ATT&CK — это универсальный инструмент для построения надёжной защиты, анализа угроз и оценки мер безопасности. Благодаря своей структуре и актуальности, он стал стандартом в области кибербезопасности и широко используется организациями по всему миру.

5 SIEM (Security Information and Event Management)

SIEM (Управление информацией и событиями безопасности) — это технология, которая объединяет функции сбора, анализа и мониторинга событий безопасности. SIEM-системы позволяют организациям выявлять угрозы, реагировать на инциденты и обеспечивать соответствие требованиям регуляторов.

Основные функции SIEM

1. Сбор данных: SIEM собирает журналы (логи) событий из различных источников: серверов, сетевых устройств, приложений, баз данных и систем безопасности.
2. Корреляция событий: сопоставляет данные из разных источников, чтобы выявить подозрительную активность или аномалии.
3. Реагирование на инциденты: отправляет оповещения и активирует автоматические действия при обнаружении угроз.
4. Хранение и анализ: обеспечивает длительное хранение логов для расследования инцидентов и выполнения требований нормативных актов.
5. Отчётность: предоставляет отчёты для анализа безопасности и аудита соответствия.

Преимущества SIEM

1. Централизованное управление: позволяет анализировать безопасность всей системы из одной точки.
2. Обнаружение сложных угроз: благодаря корреляции событий SIEM выявляет атаки, которые не видны при анализе отдельных логов.
3. Автоматизация реагирования: экономит время команды безопасности за счёт автоматизации оповещений и устранения угроз.
4. Соответствие нормативным требованиям: SIEM упрощает подготовку отчётов для таких стандартов, как PCI DSS, GDPR и ISO 27001.

Компоненты SIEM

1. Агрегатор логов: собирает данные из различных систем.
2. Движок корреляции: анализирует события, сопоставляя их с заранее заданными правилами и сценариями угроз.
3. Система оповещений: уведомляет о подозрительных действиях в режиме реального времени.
4. Интерфейс анализа: предоставляет инструменты для визуализации, построения графиков и проведения расследований.

Пример работы SIEM

1. Система обнаруживает необычную активность — несколько неудачных попыток входа на сервер.
2. Корреляционный движок анализирует логи и определяет, что эти попытки связаны с IP-адресом, ранее замеченным в попытках атак.
3. SIEM отправляет оповещение администратору и автоматически блокирует IP.
4. Администратор использует SIEM для детального анализа инцидента и предотвращения подобных атак в будущем.

Примеры SIEM-систем

1. Splunk: мощная и гибкая SIEM-система с развитой аналитикой.
2. IBM QRadar: ориентирована на крупные организации, с глубокими возможностями анализа угроз.
3. ArcSight: популярное решение для крупных корпоративных сетей.
4. Graylog: опенсорсная система, подходящая для небольших организаций.

Проблемы и ограничения SIEM

1. Высокая стоимость: внедрение и обслуживание SIEM-систем требуют значительных ресурсов.
2. Большое количество ложных срабатываний: требует настройки правил и алгоритмов для снижения шума.
3. Сложность внедрения: интеграция со всеми системами требует времени и квалификации.
4. Необходимость в обучении: специалисты должны обладать навыками работы с SIEM для эффективного использования.

Вывод

SIEM — это ключевой инструмент современной кибербезопасности. Он обеспечивает не только обнаружение угроз, но и аналитические возможности для улучшения защиты. Несмотря на сложности внедрения, SIEM остаётся необходимым для организаций, стремящихся защитить свою инфраструктуру и данные от современных угроз.