DOI 10.37882/2223-2966.2024.10-2.15

# СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЛУЖБ МОНИТОРИНГА АКТИВНОСТИ В ОПЕРАЦИОННЫХ СИСТЕМАХ WINDOWS И GNU LINUX

## COMPARATIVE ANALYSIS OF ACTIVITY MONITORING SERVICES IN WINDOWS AND GNU LINUX OPERATING SYSTEMS

***D. Mokhorev***

*Summary*. This article explores the use of built-in logging services in operating systems as data sources for behavioral analysis. The aim of the research is to identify optimal data sources for creating a universal method for predicting the behavior of entities within an information system. To achieve this, the most common operating systems in Russia were identified, and a comparative analysis of the characteristics of their built-in logging tools was conducted. The results of the analysis allowed for the identification of the most suitable data sources for developing a universal method.

*Keywords*: behavior analytics, monitoring services, operating systems, Windows, Linux, logging, information security, behavior prediction.

***Мохорев Дмитрий Евгеньевич***
*Аспирант, ФГБОУ ВО «РЭУ им. Г.В. Плеханова»*
*mohorevde@gmail.com*

*Аннотация*. В данной статье рассматривается использование встроенных служб логирования в операционных системах в качестве источников данных для поведенческого анализа. Целью исследования является определение оптимальных источников данных для создания универсального метода прогнозирования поведения объектов в информационной системе. Для этого были выявлены наиболее распространенные в России операционные системы и проведен сравнительный анализ характеристик встроенных в них средств логирования. Результаты анализа позволили определить наиболее подходящие источники данных для разработки универсального метода.

*Ключевые слова*: поведенческий анализ, службы мониторинга, операционные системы, Windows, Linux, логирование, информационная безопасность, прогнозирование поведения.

## Введение

Анализ и прогнозирование поведения объектов в информационной системе является крайне актуальным методом мониторинга информационной безопасности. В связи ростом активности киберпреступников традиционные системы выявления угроз, основанные на сигнатурных методах, уступают в эффективности и возможностях обеспечения долгосрочной защищенности поведенческой аналитике. Это вызвано тем, что для поведенческого анализа, в отличие от сигнатурного, не требуется образец вредоносной программы, а выявление угроз выполняется с помощью обнаружения аномалий в активности. Поэтому данный метод позволяет детектировать ранее неизвестные вирусы и атаки. Это является важным перспективным преимуществом поведенческой аналитики, так как количество новых тактик злоумышленников растет [12, 13].

Поведенческий анализ основан на формировании базовой модели поведения для каждого наблюдаемого объекта информационной системы. Этим объектом могут быть любые элементы информационной системы: пользователь, рабочий компьютер, сервер, сетевое устройство, а также сегмент сети, группа пользователей или серверов. Выявление угроз осуществляется путем регистрации отклонений от базовой модели поведения. Таким образом возможно обнаружение большинства типов киберугроз: от инсайдерской деятельности до целевой атаки на инфраструктуру предприятия [11].

Базовая модель поведения составляется на основании сведений о состоянии наблюдаемого объекта. Источниками данных сведений являются записи журналов операционных систем и специальных средств мониторинга. От полноты и актуальности собранных сведений зависит качество и точность базовой модели поведения объекта. Поэтому сбор данных является одним из ключевых аспектов в построении системы поведенческой аналитики [13]. В рамках исследования, посвященного созданию универсальной модели прогнозирования поведения объектов в информационной системе, необходимо, в частности, определить состав данных, которые можно использовать для поведенческого анализа. Для этого в данной работе будут рассмотрены возможности мониторинга активности в различных операционных системах.

В настоящий момент нет опубликованных материалов, посвященных сравнению в контексте поведенческого анализа возможностей штатных средств мониторинга активности в различных операционных системах. Поэтому данное исследование является актуальным, а его результат может использоваться в рамках дальнейших работ по построению системы поведенческого анализа.

## Методика исследования

Универсальный метод прогнозирования поведения объектов в информационной системе предполагает использование независимых от типа операционной системы данных. Для того, чтобы результат работы данного метода с различными операционными системами был однозначным, необходимо выбрать такой набор данных, который по содержанию и значению будет общим для всех систем. Таким образом, для множеств из всех сведений об активности, которые предоставляют операционные системы и средства мониторинга A, B, C, необходимо выбрать только те значения, которые удовлетворяют условию $A \cap B \cap C$.

Одним из основных источников сведений о состоянии наблюдаемых объектов является логирование в операционной системе. Логирование — это процесс записи информации о происходящих в системе событиях в специальный файл, называемый лог-файлом. Лог-файлы содержат информацию о работе приложений и системных компонентов, обнаруженных ошибках, а также данные о пользовательской активности [14]. Для определения подходящего набора данных необходимо изучить особенности логирования в разных операционных системах.

Таким образом, целью данного исследования является изучение характеристик встроенных средств логирования наиболее распространенных в российских информационных инфраструктурах операционных систем. Исходя из поставленной цели сформулированы следующие задачи:

- определение наиболее распространенных в российских информационных инфраструктурах операционных систем;
- анализ возможностей встроенных средств логирования в выбранных операционных системах;
- составление сравнительной характеристики возможностей встроенных средств логирования в выбранных операционных системах.

Сравнительный анализ как метод исследования является мощным инструментом для выявления сходства и различия между выбранными объектами. В данном исследовании сравнительный анализ будет применен для выявления различий в функциональности средств журналирования событий разных операционных систем. Составленная сравнительная характеристика возможностей мониторинга активности в операционных системах будет применена для выбора оптимального набора данных для работы универсального метода прогнозирования поведения объектов в информационной инфраструктуре.

В соответствии с целью и задачами исследования выделены четыре направления, по которым будет выполняться сравнение служб мониторинга в операционных системах: содержание собираемых данных, возможности настройки мониторинга, возможности обработки данных и производительность. Сравнительный анализ операционных систем по данным направлениям позволит изучить характеристики встроенных средств логирования в контексте мониторинга активности и создания универсального метода прогнозирования поведения пользователей в информационной системе.

Базовая модель поведения формируется на основании данных, предоставленных системой мониторинга в событиях. От полноты и сложности информации из событий зависит качество модели поведения и точность поведенческой аналитики. Больший объем сведений с более сложной структурой позволит создать более релевантную модель поведения. Поэтому содержание событий безопасности является важным аспектом построения системы поведенческой аналитики. В рамках данного направления определены следующие критерии: структура данных, поддерживаемые типы собираемых данных, уровень детализации событий, поддержка метаданных.

Не менее важными для эффективного анализа логов являются возможности обработки собранных данных. Предполагает удобство как ручного анализа логов, так и использование программных средств для получения необходимого результата. В данном направлении определены следующие критерии: методы фильтрации и поиска, читаемость событий и удобство для ручного анализа, инструменты и библиотеки для обработки.

Построение системы поведенческой аналитики является сложной и комплексной задачей, которая требует широкие возможности по настройке средств мониторинга. Важно рассмотреть аспекты удаленного логирования, способы расширения собираемых данных и особенности интеграции с другими системами. В направлении «возможности настройки мониторинга» определены следующие критерии: гибкость и расширяемость, возможности интеграции с другими системами и поддержка стандартов, поддержка удаленного логирования.

Заключительным этапом сравнение служб мониторинга активности в операционных системах является оценка их эффективности и производительности. Работа комплексной системы анализа предполагает сбор, обработку и хранение данных с большого количества устройств. Выделение большого объема ресурсов на обеспечение функционирования данной системы может повлиять на скорость работы всей инфраструктуры в целом. В рамках данного направления обозначены следующие критерии: производительность; объем данных.

Таким образом, сравнительный анализ возможностей служб мониторинга активности в операционных системах по обозначенным критериям позволит изучить характеристики встроенных средств логирования. С помощью полученных результатов будет возможно определить набор данных для создания и работы универсального метода прогнозирования поведения объектов в информационной системе.

## Определение целевых операционных систем для сравнительного анализа

Универсальный метод анализа поведения объектов должен учитывать особенности российских информационных систем. Для этого необходимо определить наиболее распространенные в российских информационных инфраструктурах операционные системы. Согласно ежегодному исследованию РУССОФТ, посвященному индустрии программного обеспечения в России, в последние годы наблюдается снижение доли разрабатываемого ПО для операционных систем семейства MS Windows и рост популярности GNU Linux [1]. В 2023 году большинство опрошенных в рамках исследования организаций сообщили о том, что используют в работе и разрабатывают ПО для операционных систем семейства GNU Linux. Среди российских компаний, занимающихся разработкой ПО, в качестве наиболее популярных отмечены следующие операционные системы для стационарных компьютеров [1]:

— GNU Linux, 73 % опрошенных компаний;
— MS Windows, 68 % опрошенных компаний;
— Mac OS, 17 % опрошенных компаний.

Данная статистика отражает востребованность разработки ПО для каждой операционной системы на российском рынке. Поэтому она может использоваться для составления списка наиболее распространенных в российских компаниях операционных системах.

Для определения конкретных дистрибутивов Linux, находящихся в эксплуатации, можно использовать статистику пользователей сервиса Yandex Cloud [2]. Распределение дистрибутивов GNU Linux по частоте использования на данном сервисе можно отобразить следующим образом:

— Ubuntu, около 60 % пользователей;
— CentOS, около 19 % пользователей;
— Debian, 10 % пользователей.

Несмотря на то, что представленные сведения охватывают только облачный сегмент, можно предположить, что схожее распределение операционных систем сохраняется и в других сферах. Совместив данные из двух рассмотренных отчетов, можем подсчитать долю использования каждой из наиболее распространенных операционных систем в российских компаниях. Результаты представлены на рисунке 1.

Таким образом, с целью определения данных, подходящих для работы универсального метода прогнозирования поведения объектов в информационной системе, необходимо рассмотреть следующие операционные системы: Microsoft Windows, Ubuntu, CentOS, Debian. Для изучения возможностей стандартных средств мо-

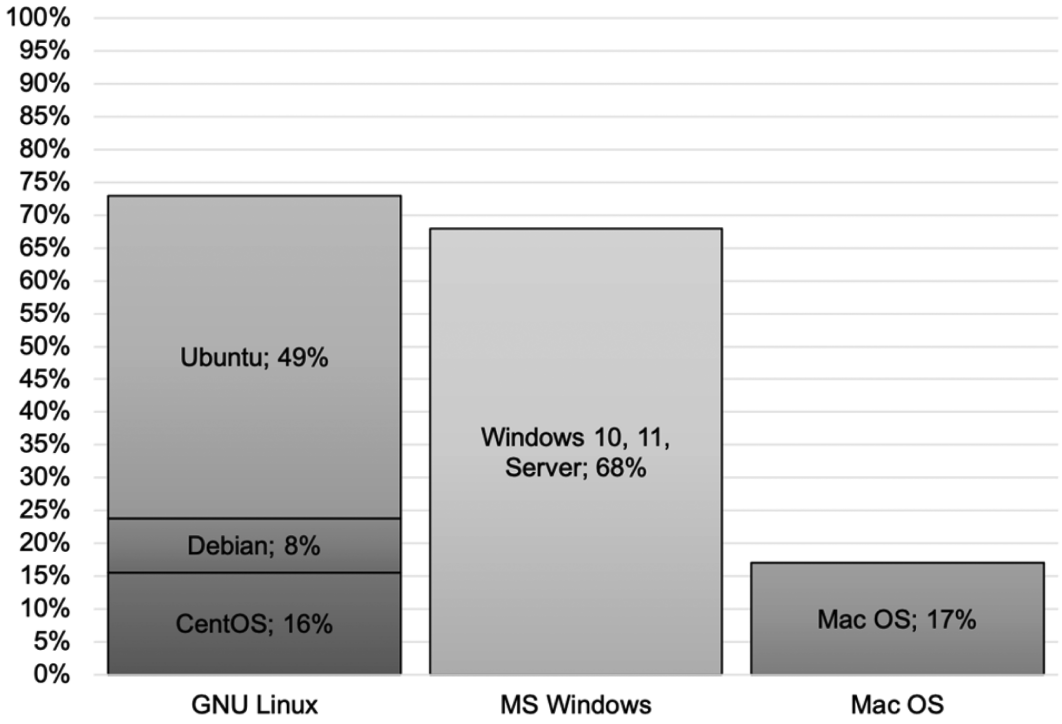


Рис. 1. Использование операционных систем в российских компаниях, источники: Ежегодное исследование РУССОФТ, Рейтинг популярности операционных систем Yandex Cloud [1, 2]

ниторинга активности в исследовании использовались следующие версии операционных систем: Microsoft Windows 10, Ubuntu 22.04.4, CentOS 7-2009, Debian 12.5.

При рассмотрении актуальных разработок в области поведенческой аналитики и математических теорий в работе «Разработка универсального метода прогнозирования поведения объектов в информационной системе: анализ актуальных разработок в области поведенческой аналитики и математических теорий» было отмечено, что большинство систем поведенческой аналитики направлены на решение узконаправленных задач в рамках одной операционной системе. Согласно полученным в данном исследовании результатам, в настоящий момент отсутствует явно лидирующая по частоте использования система, а большинство российских организаций используют в работе несколько операционных систем. Анализ популярности операционных систем в российских компания продемонстрировал актуальность разработки универсального метода поведенческого анализа в информационных системах.

## Анализ службы журналирования событий Windows

Несмотря на снижение популярности в течение последних нескольких лет, операционные системы семейства MS Windows по-прежнему являются одним из основных инструментов организации работы в российских компаниях. В данном разделе будет проведен анализ возможностей стандартных средств мониторинга активности по обозначенным критериям.

В основе мониторинга активности в Microsoft Windows лежит Windows Event Log (служба журналирования событий Windows). Данная служба осуществляет регистрацию всех происходящих в операционной системе событий в ключе реестра «System32\Winevt\Logs\System.evtx» [14]. События регистрируются в специальных журнальных файлах, в которых содержится подробная информация о произошедшем событии, его времени и месте возникновения. Для просмотра данных журналов используется интерфейс Windows Event Viewer.

### Содержание собираемых данных — структура данных

Windows Event Log хранит события в файлах журнала .evtx, расположенных каталоге «System32\Winevt\Logs\». События разнесены в разные журналы в зависимости от источника и предназначения. Основными источниками сведений для поведенческой аналитики могут быть журналы Security, System, Application и Windows PowerShell. Логически события разделены на следующие категории:

1. Account Logon
2. Account Management
3. Directory Service
4. Logon/Logoff
5. Non Audit (Event Log)
6. Object Access
7. Policy Change
8. Privilege Use
9. Process Tracking
10. System
11. Uncategorized

События имеют сложную структуру. Она определена тегами, каждый из которых выделяет значение поля события, например имя компьютера обозначено тегом <Computer>, имя процесса — <ProcessName>.

Основным форматом экспорта логов является XML (eXtensible Markup Language) — разметочный формат, предназначенный для хранения и транспортировки данных. Таким образом логи Windows имеют сложную структуру, которая позволяет осуществлять тонкую настройку фильтров.

### Содержание собираемых данных — поддерживаемые типы собираемых данных

Windows Even Log как и Journald имеет поддержку различных типов данных. С помощью атрибутов можно строковые, числовые, булевые значения, определить массивы и двоичные числа.

### Содержание собираемых данных — уровень детализации событий

Уровень детализации в Windows Event Log варьируется в зависимости от типа события, но обычно включает время, источник и описание события. Сбор дополнительной информации требует настройки. Таким образом, сравнительно с Journald в штатной конфигурации Windows Evenrt Log предоставляет меньше информации о системных событиях.

### Содержание собираемых данных — поддержка метаданных

Как уже отмечено выше, события Windows Event Log имеют сложную структуру, поэтому поддерживают метаданные. Использование тегов позволяет более тонко настроить фильтры и выборку событий.

### Возможности обработки — методы фильтрации и поиска

Windows Event Log имеет графический интерфейс с обширным функционалом фильтрации. В Event Viewer

возможна фильтрация по таким параметрам как время, уровень события, журналу, источнику событий, коду событий, категории задачи, ключевым словам пользователю и компьютеру. Также Event Viewer позволяет создавать поисковый запрос с помощью синтаксиса XML. Таким образом Windows Event Log предоставляет инструменты для фильтрации событий по различным критериям через Event Viewer.

### Возможности обработки — читаемость событий и удобство для ручного анализа

Формат XML можно охарактеризовать как громоздкий и сложный для чтения, особенно с большим количеством вложенных элементов. Однако, отличительной особенностью Windows Event Log является его графический интерфейс Event Viewer. Это удобный инструмент для просмотра и управления логами. С помощью Event Viewer легко определить суть события, для этого не требуется знание синтаксиса и XML структуры файла. Windows Event Log в совокупности с Event Viewer является наиболее простой для ручного анализа логов службой.

### Возможности обработки — инструменты и библиотеки для обработки

События Windows Event Log хорошо совместимы с программным обеспечением семейства Windows. Однако, для комбинированной работы с логами других форматов, например Linux или маршрутизаторов Cisco, Windows Event Log требуют затраты дополнительных ресурсов на парсинг.

### Возможности обработки — гибкость и расширяемость

Windows Event Log, как и Journald, имеет высокую степень расширяемости благодаря сложной структуры логов и поддержке метаданных. Поэтому возможно добавление новых элементов и атрибутов без разрушения существующей структуры. Службы-клиенты могут настраивать собственные поля для передачи в службу сбора событий.

### Возможности обработки — возможности интеграции с другими системами и поддержка стандартов

Windows Event Log соответствует стандартам Microsoft и широко используется в корпоративной среде. Благодаря широкому распространению операционных систем Windows большая часть программного обеспечения имеет поддержку логов Windows Event Log «по умолчанию». Для интеграции с другими системами существуют различные коннекторы с прописанными правилами обработки событий Windows. Формат экспорта логов XML используется в различных системах обмена данными, но менее популярен в современных веб-приложениях.

### Возможности обработки — поддержка удаленного логирования

Для обеспечения удаленного логирования в Windows имеется специальная служба Windows Event Forwarding. Она позволяет настроить отправку событий на удаленные серверы.

### Производительность

Несмотря на глубокую интеграцию Windows Event Log в операционную систему служба может расходовать большое количество вычислительной мощности. При большом количестве событий и в больших средах работа службы может нагружать систему и оказывать сильное влияние на производительность.

### Производительность — объем данных

Обычно файлы формата XML занимают больше места, чем JSON и CSV. Это связано с большим количеством тегов и атрибутов, для обозначения которых требуются дополнительные символы. Пример события Windows Event Log представлен на рисунке 2.

Данный файл в формате экспорта XML занимает 536Б.

## Службы мониторинга активности дистрибутивов GNU Linux

В настоящий момент в большинстве популярных дистрибутивов GNU Linux, в том числе в рассматриваемых в данной работе, используются две службы логирования: journald и rsyslog. Journald — это служба логирования, входящая в состав init-системы systemd. Systemd является системным менеджером и инициализатором для большинства дистрибутивов Linux. Systemd и Journald используются во всех актуальных версиях рассматриваемых в данном исследовании операционных систем семейства GNU Linux: Ubuntu, CentOS, Debian [5, 8, 10]. Rsyslog — системная служба UNIX-систем, предназначенная для сбора, хранения и перенаправления логов, основанная на протоколе syslog. Rsyslog ранее использовалась как основная служба логирования и остается на сегодняшний день актуальным инструментом для централизованного сбора логов с большого количества компьютеров. Это связано с тем, что в отличии от Journald, Rsyslog поддерживается во всех UNIX-системах, а не только в совместимых с Systemd [15].

Во всех рассматриваемых в исследовании операционных системах семейства GNU Linux используются

```
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4991-
AF32-52823B30307B}" />
    <EventID>4624</EventID>
    <Version>1</Version>
    <Level>2</Level>
    <Task>12544</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8020000000000000</Keywords>
    <TimeCreated SystemTime="2023-10-26T12:34:56.123456789Z" />
    <EventRecordID>1234567890abcdef</EventRecordID>
    <CorrelationActivityID>1234567890abcdef</CorrelationActivityID>
    <ExecutionProcessID>1234</ExecutionProcessID>
    <Channel>Security</Channel>
    <Computer>example-host</Computer>
    <Security />
  </System>
  <EventData>
    <Data Name="TargetUserName">user1</Data>
    <Data Name="TargetDomainName">EXAMPLE</Data>
    <Data Name="LogonType">2</Data>
    <Data Name="LogonProcessName">C:\Windows\System32\lsass.exe</Data>
    <Data Name="AuthenticationPackageName">Negotiate</Data>
    <Data Name="WorkstationName">example-host</Data>
    <Data Name="LogonGuid">1234567890abcdef</Data>
    <Data Name="TransmittedServices">0</Data>
    <Data Name="LmPackageName">NTLM</Data>
    <Data Name="KeyLength">0</Data>
    <Data Name="ProcessTokenSid">S-1-5-21-1234567890abcdef</Data>
    <Data Name="AuthenticationProcess">C:\Windows\System32\svchost.exe</Data>
    <Data Name="NetworkLogonId">1234567890abcdef</Data>
  </EventData>
</Event>
```

Рис. 2. Пример события Windows Event Log

идентичные системы логирования, а особенности дистрибутивов не связаны с процессом сбора и обработки логов [4, 6, 9]. В связи с этим анализ возможностей мониторинга активности в данных системах будет сосредоточен на изучении функционала служб Journald и Rsyslog. В последующих разделах будет проведен анализ возможностей данных средств мониторинга активности по обозначенным критериям.

### Анализ службы Journald

Journald является частью наиболее распространенной в современных дистрибутивах GNU Linux init-системы Systemd [17]. Эта служа используется для логирования событий «по умолчанию» — в штатных настройках операционных систем. В результате функционирования данной службы формируется база данных, состоящая из всех логов, собираемых в операционной системе. В отличии от Rsyslog, которая формирует логи в текстовый файл, Journald обогащает события специальными ключами и индексирует их. Такой подход позволяет детализировать поисковые запросы и быстрее находить нужные события. Однако, минусом формирования из логов базы данных и хранения их в таком виде является отсутствие совместимости с текстовыми форматами файлов, так как Journald записывает и хранит

данные в двоичном формате [16, с. 480]. В следствии этого, централизованный сбор логов с помощью Journald возможен только в случае, если все компьютеры в информационной системе поддерживают init-систему Systemd. Это существенно усложняет настройку мониторинга в сложных информационных системах с большим набором разных дистрибутивов Linuix [17].

Все логи, собираемые в Journald в штатной конфигурации, хранятся в каталоге «/var/log/journal». Для просмотра логов используется служба Journalctl. Все события в данной базе разделены на уровни критичности и источники — службы, которые их зарегистрировали.

### Содержание собираемых данных — структура данных

Особенностью Journald является организация логов в базу данных со сложной структурой. Записи в Journald имеют индексацию полей, которая позволяет создавать взаимосвязи и вложенные структуры, а также применять гибкие фильтры к данным. Также можно выделить два типа полей:

1. Trusted journal fields — недоступные доступные для редактирования клиенту, который передает записи в журнал. Название таких полей начинается с символа подчеркивания, например, «_HOSTNAME» — имя хоста, на котором было записано сообщение, | «_SYSTEMD_UNIT» — юнит systemd, связанный с процессом.
2. User fields — остальные поля, которые может заполнять клиент, передающий запись в журнал. Название таких полей не выделяется символом подчеркивания. Примерами таких полей являются MESSAGE_ID — уникальный идентификатор сообщения, MESSAGE — основное сообщение журнала.

Все записи Journald хранит в бинарном виде, поэтому для ее чтения необходимо использовать функционал Journalctl. Основным форматом экспорта журнала является JSON (JavaScript Object Notation), поддерживающий структуру данных с помощью пары «ключ-значение».

### Содержание собираемых данных — поддерживаемые типы собираемых данных

Journald обладает широкими возможностями по обработке различных типов данных. Служба поддерживает строковые значения, числовые значения, булевые значения, массивы, объекты. Таким образом Journald является гибкой для представления данных различного содержания.

### Содержание собираемых данных — уровень детализации событий

Journald предоставляет высокую степень детализации событий. В штатной конфигурации в службе имеется более ста полей различного формата, включая идентификаторы процессов, системные вызовы, временные метки и контекст выполнения. Таким образом, Journald предоставляет широкую картину активности системы для выявления аномалий и глубокого анализа поведения пользователей

### Содержание собираемых данных — поддержка метаданных

Метаданные в Journald включены в структуру и проиндексированы таким образом, что через них возможно получать связанные события, например поиск по полю PID — Идентификатор процесса, к которому относится запись, позволит найти события одной активности.

### Возможности обработки — методы фильтрации и поиска

Благодаря формированию поступающих событий в базу данных с индексированными полями Journald обладает широкими возможностями фильтрации. С помощью различных ключей поиска в инструменте анализа логов Journalctl возможна гибкая настройка поиска. Например, для выборки по времени поступления события в журнал можно использовать такие конструкции как «начиная с...», «до...», «в определенном промежутке между» и т.д. Широкий набор полей также способствует гибкости фильтрации, позволяя осуществлять поиск по контексту, например все события в сеансе пользователя с идентификатором сеанса _SYSTEMD_SESSION=session-2.scope: journalctl -u session-2.scope. Таким образом Journald поддерживает мощные команды для фильтрации и поиска по логам.

### Возможности обработки — читаемость событий и удобство для ручного анализа

Несмотря на наличие такого мощного инструмента для анализа, как journalctl, логи Journald нельзя назвать легкими для чтения и интерпретации. Для понимания событий Journald необходимо знание синтаксиса journalctl и структуры самих данных. Настройка службы осуществляется с помощью редактирования конфигурационного файла. Также интерфейс командной строки уступает в удобстве графическому интерфейсу Windows Event Viewer.

### Возможности обработки — инструменты и библиотеки для обработки

Основным форматом экспорта логов Journald является JSON, поэтому служба имеет поддержку большого количества совместимых с JSON инструментов и библиотек. Однако, в связи со сложной структурой самого файла, данный вариант экспорта журнала является ме-

нее универсальным, чем csv, который содержит простые строковые значения. Таким образом, для обработки данных, полученных из Journald, требуется больше ресурсов, чем для csv-файлов, полученных из Rsyslog.

### Возможности обработки — гибкость и расширяемость

Journald обладает легко расширяемым форматом данных. Благодаря поддержке пары «ключ-знвчение» добавление полей в Journald не требует изменения структуры логов. Поэтому клиенты, которые направляют события в журнал, могут определять новые поля.

### Возможности обработки — возможности интеграции с другими системами и поддержка стандартов

Journald обладает широкими возможностями интеграции с другими системами. Данная служба соответствует стандартам systemd, что делает ее совместимым со всеми современными Linux-дистрибутивами. Формат экспорта логов JSON также способствует универсальности данного решения, так как он хорошо совместим с современными веб-приложениями и API.

### Возможности обработки — поддержка удаленного логирования

Journald поддерживает отправку логов на удаленные серверы через протоколы, такие как syslog (например, через Rsyslog или другие совместимые системы). Это позволяет интегрировать его с существующими системами логирования.

### Производительность

Являясь частью init-системы Systemd, Journald хорошо оптимизирован для работы в системах Linux. Хранение логов в бинарном формате также снижает нагрузку на диск при записи и обработке логов, что позволяет эффективно индексировать запись и быстро извлекать данные. Таким образом благодаря своей архитектуре Journald может обрабатывать большое количество логов без значительного влияния на производительность.

### Производительность — объем данных

С целью сравнения объемов записей, которые создаются Journald, Rsyslog и Windowes Event Log, были сформированы события успеха входа пользователя в систему


```
{
  "_UID": 1000,
  "_GID": 1000,
  "_COMM": "sshd",
  "_PID": 12345,
  "_SYSTEMD_UNIT": "sshd.service",
  "MESSAGE": "User 'user1' logged in from 192.168.1.10",
  "PRIORITY": 6,
  "FACILITY": 4,
  "SYSLOG_IDENTIFIER": "sshd",
  "SOURCE_REALTIME_TIMESTAMP": "1690680779.123456789",
  "SOURCE_MONOTONIC_TIMESTAMP": "1690680779.123456789",
  "_BOOT_ID": "12345678-9abc-def0-1234-56789abcdef0",
  "MACHINE_ID": "12345678-9abc-def0-1234-56789abcdef0",
  "SYSTEMD_USER_UNIT": "user-1000.slice",
  "_CAP_EFFECTIVE": "0xffffffff",
  "LOGINUID": 1000,
  "USERNAME": "user1",
  "SOURCE_IP": "192.168.1.10",
  "CODE_FILE": "/usr/bin/sshd",
  "CODE_LINE": 1234,
  "CODE_FUNCTION": "auth_password"
}
```


Рис. 3. Пример события Journald

в форматах экспорта каждой службы. Все события содержат идентичную информацию, имеют структуру рассматриваемых журналов и записаны в форматах экспорта каждой службы JSON, CSV и XML для Journald, Rsyslog и Windows Event Log соответственно. Пример события Journald представлен на рисунке 3.

Размер данного файла JSON составляет 685Б. В связи с наличием сложной структуры лога JSON обычно занимает больше места по сравнению с CSV из-за использования дополнительных символов (например, фигурных скобок).

### Анализ службы Rsyslog

Rsyslog в настоящий момент является одной из самых распространенных служб мониторинга активности. Важной особенностью Rsyslog является то, что он совместим со всеми дистрибутивами Linux. Это связано с тем, что данная служба основана на работе протокола syslog, поддержка которого включена в стандартной конфигурации всех операционных систем семейства Linux [15].

Rsyslog представляет набор правил, на основании которых поступающие на обработку логи распределяются в различные каталоги. Данные правила позволяют определить объекты логирования — службы, являющиеся источниками логов. Стандартным расположением лог-файлов Rsyslog является «/var/log». Логи, создаваемые Rsyslog хранятся в текстовом виде, поэтому для их просмотра не требуется специальная служба. Данная особенность позволяет использовать Rsyslog для централизованного сбора логов в любых информационных системах, в отличии от Journald.

Rsyslog имеет идентичный с Journald список уровней критичности.

### Содержание собираемых данных — структура данных

В штатной конфигурации Rsyslog записывает логи в текстовом формате в раздельных на каждый источник файлах, расположенных в каталоге «/var/log». Основным форматом экскорта логов Rsyslog является CSV (Comma-Separated Values) — простой текстовый формат, который используется для хранения табличных данных, где каждая строка представляет собой запись, а значения в строке разделены запятыми (или другим разделителем, например, точкой с запятой). Поэтому структура лог файла имеет простой табличный формат: каждая новая строка — отдельное событие в журнале, а поля обозначены символами-разделителями. Такой формат хорошо подходит для плоских данных, но не поддерживает вложенные структуры.

### Содержание собираемых данных — поддерживаемые типы собираемых данных

В Rsyslog нет возможности задать тип записанных данных. Все логи Rsyslog хранятся в виде строк, поэтому для преобразования их в нужный тип требуется дополнительная обработка.

### Содержание собираемых данных — уровень детализации событий

Уровень детализации логов Rsyslog зависит от конфигурации. Возможна гибкая настройка сбора и фильтрации событий, позволяющая выделить только необходимую информацию. Состав полей, которые формирует Rsyslog не стандартизирован, как например в Journald или Windows Event Log, что может привести к проблемам при парсинге логов и анализе.

### Содержание собираемых данных — поддержка метаданных

Служба Rsyslog в штатной конфигурации не поддерживает работу с метаданными. Это связано с тем, что структура логов Rsyslog определяется в конфигурационном файле, который хранится отдельной от записей журнала.

### Возможности обработки — методы фильтрации и поиска

Rsyslog позволяет настраивать фильтры на уровне конфигурации для выбора необходимых логов. В основе фильтрации записей в Rsyslog лежит текстовый поиск. Имеется поддержка регулярных выражений. Таким образом возможно создавать фильтры по имени хоста, времени, имени программы, содержанию сообщения и т.д.

### Возможности обработки — читаемость событий и удобство для ручного анализа

Для понимания событий Rsyslog также как, как для событий Journald, необходимо знание структуры данных. Препятствием для чтения логов Rsyslog также является то, что структура хранится отдельно от самих записей. Поэтому, при наличии сложных данных, логи Rsyslog могут быть трудным для восприятия.

### Инструменты и библиотеки для обработки

Текстовый формат данных является универсальным во всех информационных системах, поэтому логи Rsyslog обладают широкой совместимостью с различным программным обеспечением. Основным форматом экспорта логов Rsyslog являтеся CSV, который поддерживается

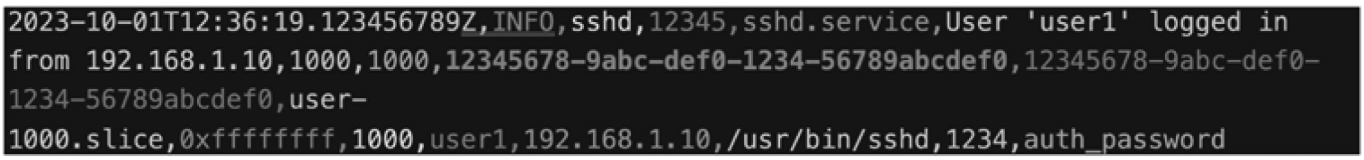


Рис. 4. Пример события Rsyslog

Таблица 1.

Сравнительная характеристика служб мониторинга в операционных системах

| Направление сравнения | Критерий | Microsoft Windows, Windows Event Log | Linux, Journald | Linux, Rsyslog |
|---|---|---|---|---|
| Содержание собираемых данных | Структура данных | Логи имеют сложную структуру, которая позволяет осуществлять тонкую настройку фильтров. | Логи имеют сложную структуру с помощью пары «ключ-значение», которая позволяет осуществлять тонкую настройку фильтров. | Имеет простой табличный формат, который хорошо подходит для плоских данных, но не поддерживает вложенные структуры. |
| | Поддерживаемые типы собираемых данных | Имеет поддержку различных типов данных через специальные теги | Имеет поддержку различных типов данных через ключи | Отсутствует возможность указать тип данных |
| | Уровень детализации событий | Зависит от типа события, обычно включает время, источник и описание события | Предоставляет высокую степень детализации событий | Зависит от конфигурации, набор полей не стандартизирован |
| | Поддержка метаданных | Поддерживает метаданные | Поддерживает метаданные | Поддержка метаданных отсутствует |
| Возможности обработки | Методы фильтрации и поиска | Имеет графический интерфейс для фильтрации событий, но есть ограничения по полям поиска | Поддерживает мощные команды для фильтрации и поиска по логам | Поддерживает текстовый поиск по полям |
| | Читаемость событий и удобство для ручного анализа | События понятны и легко читаются, имеет графический интерфейс для анализа | Для понимания событий необходимо знание синтаксиса journalctl и структуры данных | Для понимания событий необходимо знание синтаксиса и структуры данных |
| | Инструменты и библиотеки для обработки | Хорошо совместимы с программным обеспечением семейства Windows | Требует затраты дополнительных ресурсов для обработки | Поддерживается в большинстве языков программирования и инструментов для работы с данными |
| Возможности настройки мониторинга | Гибкость и расширяемость | Возможно добавление новых полей без внесения изменений в структуру данных | Возможно добавление новых полей без внесения изменений в структуру данных | Добавление новых полей требует изменение всей структуры данных |
| | Возможности интеграции с другими системами и поддержка стандартов | Соответствует стандартам Microsoft и широко используется в корпоративной среде | Соответствует стандартам systemd, что делает ее совместимым со всеми современными Linux-дистрибутивами | Поддерживает стандарты syslog, является универсальным и широко используется в различных системах |
| | Поддержка удаленного логирования | Поддерживается через специальную службу Windows Event Forwarding | Поддерживается через Syslog | Имеет встроенные функции удаленного логирования |
| Производительность | Производительность | При большом количестве событий может нагружать систему | Хранение логов в бинарном формате позволяет эффективно индексировать запись и быстро извлекать данные | При большом объеме данных требует существенно больших ресурсов, чем другие службы |
| | Объем данных | Занимает больше места, чем CSV из-за большого количества тегов и атрибутов. | Занимает больше места, чем CSV из-за большого количества символов, формирующих структуру | Имеет наименее объемный формат логов, так как хранит только данные и символы-разделители без метаданных |

в большинстве языков программирования и инструментов для работы с данными (например, Excel).

### Возможности обработки — гибкость и расширяемость

Структура данных Rsyslog не поддерживает пару «ключ-значение» и определяется только с помощью переноса строк и символов-разделителей, поэтому является менее гибкой, чем, например, структура Journald. Строгая последовательность полей при добавлении новых столбцов требует изменение всей структуры журнала. Также, расширение данных может привести к проблемам совместимости. Таким образом Rsyslog является менее гибким форматом сбора логов.

### Возможности обработки — возможности интеграции с другими системами и поддержка стандартов

Syslog является общепризнанным стандартом ведения логов и во многих системах имеется его поддержка. Он используется при логировании на сетевом оборудовании, в базах данных и других системах. Rsyslog поддерживает стандарты syslog и может быть настроен для работы с различными форматами. Также, текстовый формат логов делает способствует упрощению интеграции Rsyslog с другими системами.

### Возможности обработки — поддержка удаленного логирования

Rsyslog имеет встроенные функции для отправки и получения логов по сети, включая поддержку различных протоколов, таких как TCP и UDP. Служба может как отправлять, так и принимать логи от других систем. Таким образом Rsyslog является мощным инструментом для централизованного логирования и широко используется в информационных инфраструктурах.

### Производительность

Обработка данных журналов Rsyslog требуется существенно больших ресурсов, чем, например для Journald. Это связано с тем, что Rsyslog использует текстовые файлы для хранения логов, что может привести к большему объему данных и увеличению нагрузки на диск по сравнению с бинарным форматом Journald.

### Производительность — объем данных

Rsyslog имеет наименее объемный формат экспортируемых логов, так как хранит только данные и символы-

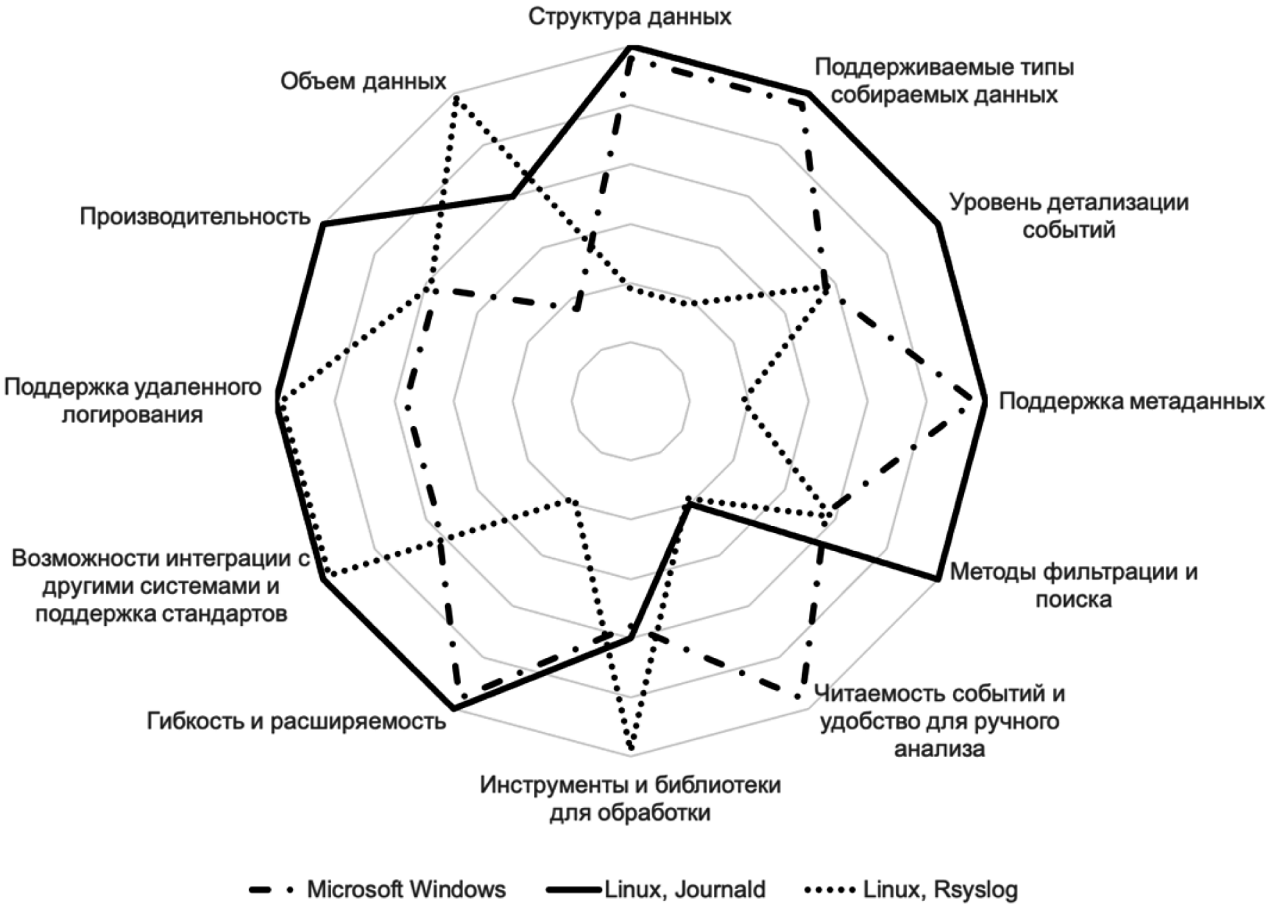


Рис. 5. Результаты сравнительного анализа служб мониторинга в операционных системах

разделители и не использует метаданные. Пример события Rsyslog представлен на рисунке 4.

Размер данного файла CSV составляет 268Б. Таким образом можно заключить, что экспортируемые логи Rsyslog имеют наименьший объем.

На основании обозначенных критериев проведен сравнительный анализ функциональности встроенных средств логирования наиболее распространенных в российских информационных инфраструктурах операционных систем. Рассмотрены возможности использования данных мониторинга в поведенческой аналитике. Составлена сравнительная характеристика, результаты анализа представлены в таблице 1.

Для наглядного представления результатов анализа в виде лепестковой диаграммы определены количественные значения каждого критерия со следующим весом:

1 — система мониторинга не поддерживает данный функционал / обладает базовым функционалом в данном направлении;

2 — система мониторинга обладает расширенным функционалом в данном направлении;

3 — система мониторинга имеет глубокий функционал в данном направлении.

Результаты анализа представлены в виде лепестковой диаграммы на рисунке 5.

## Заключение

В результате исследования Microsoft Windows 10, Ubuntu, CentOS и Debian были определены как наиболее распространённые в российских информационных инфраструктурах операционные системы. В контексте создания универсального метода прогнозирования поведения объектов в информационной системе проведен анализ служб мониторинга в данных операционных системах и составлена их сравнительная характеристика.

Рассмотренные системы мониторинга имеют различные преимущества и недостатки между собой. Так, служба журналирования событий Windows представляет наиболее удобный для анализа формат логов и графический интерфейс, но имеет больший объем хранимых данных, в сравнении с Rsyslog. Также результаты анализа демонстрируют, что информация, предоставляемая службами мониторинга активности в операционных системах, существенно различается по структуре, детализации и формату хранения. Качество базовой модели поведения зависит от подробности и точности информации, получаемой логов. Поэтому глубина детализации и возможности фильтрации являются ключевыми аспектами при выборе источника данных для анализа.

Для создания универсального метода анализа для множеств из всех сведений об активности, которые предоставляют операционные системы A, B, C, необходимо выбрать только те значения, которые удовлетворяют условию $A \cap B \cap C$. Наиболее точно сформировать набор данных из логов позволяет функционал служб Windows Event Log и Journald. Поэтому именно эти службы являются наиболее подходящими источниками данных для модели прогнозирования поведения объектов в информационной системе.

ЛИТЕРАТУРА


1. Индустрия программного обеспечения в России // Ежегодное исследование РУССОФТ. — 2023. — №20
2. Рейтинг популярности операционных систем // Yandex Cloud URL: https://yandex.cloud/ru/blog/posts/2022/07/os-rating-january-june (дата обращения: 25.02.2024).
3. Техническая документация по Windows // Microsoft URL: https://learn.microsoft.com/ru-ru/windows/ (дата обращения: 25.02.2024).
4. CentOS Wiki URL: https://wiki.centos.org/FrontPage.html (дата обращения: 20.02.2024).
5. Debian — Универсальная операционная система URL: https://www.debian.org/index.ru.html (дата обращения: 02.03.2024).
6. Debian Wiki URL: https://wiki.debian.org/ru/DebianRussian (дата обращения: 02.03.2024).
7. Q3 2022 Threat Landscape: Insider Threat, The Trojan Horse of 2022
8. The CentOS Project URL: https://www.centos.org (дата обращения: 20.02.2024).
9. Ubuntu Wiki URL: https://wiki.ubuntu.com (дата обращения: 01.03.2024).
10. Ubuntu: Enterprise Open Source and Linux URL: https://ubuntu.com (дата обращения: 01.03.2024).
11. UEBA-системы: что это, принципы работы, обзор рынка // Инсайдер.рф URL: https://инсайдер.рф/news/ueba_sistemy/ (дата обращения: 31.01.2024).
12. 2023 Cyberthreat Defense Report // CyberEDGE URL: https://cyber-edge.com/cdr/ (дата обращения: 15.02.2024).
13. 2023 Market Guide for Insider Risk Management Published 13 November 2023 // Gartner URL: https://www.gartner.com/en/documents/4931631 (дата обращения: 14.02.2024).
14. Event Logging (Event Logging) // Microsoft Learn URL: https://learn.microsoft.com/en-us/windows/win32/eventlog/event-logging (дата обращения: 19.02.2024).
15. Rsyslog Documentation // rsyslog.com URL: https://www.rsyslog.com/doc/index.html (дата обращения: 26.02.2024).
16. Уорд Б. Внутреннее устройство Linux. — 3-е изд. изд. — СПб.: Питер, 2022. — 480 с.
17. Systemd Documentation // systemd.io URL: https://systemd.io (дата обращения: 25.02.2024).



Журнал «Современная наука: актуальные проблемы теории и практики»