Интеллектуальный анализ процессов по данным журналов событий информационных систем

А.М. Хасанова

Аннотация - Целью данной работы является исследование и реализация алгоритмов интеллектуального анализа процессов с целью оптимизации работы ОС, а также с целью выявления внештатных и вредоносных событий на примере журналов событий различных операционных систем (Windows, Linux). Журналы событий информационных систем в различных сферах деятельности человека (горнодобывающая промышленности, атомная индустрия при проектировании и эксплуатации АЭС, транспортная сфера городов, банковская сфера и т.д.) могут стать источником ценной информации о процессах, происходящих в системе. Поскольку почти все эти системы предназначены для круглосуточной работы, обслуживая тысячи компьютеров и пользователей одновременно, высокая их доступность, надежность и безопасность становятся обязательными.

В статье приведено исследование журналов событий разных операционных систем и описание разработанных методов для получения, обработки и анализа журналов событий с целью предупреждения, и прогнозирования сбоев, отказов или аномальных событий, а также с целью повышения оптимизации существующих процессов. В работе приводится моделирование вредоносных событий и их обнаружение, а также примеры кода для демонстрации всех перечисленных алгоритмов.

***Ключевые слова* — интеллектуальный анализ процессов, Process Mining, оптимизация, безопасность, журналы событий.**

I. Введение

Различные современные компании используют информационные системы для поддержки своих бизнес-процессов и для обеспечения своей деятельности. Будь то ERP-система для отслеживания производственных процессов или CRM-система для координации процессов продаж и обслуживания клиентов. В настоящее время практически нет бизнес-процесса, который не поддерживается хотя бы одной IT-системой. Эти системы постоянно собирают большие объемы данных, например, каждый раз, когда делается заказ на покупку или предоставляется услуга — цифровые следы бизнес-процесса.

К тому же постоянный рост использования информационных технологий как в обычной жизни, так и в деятельности различных организаций, а также появление новых компаний порождает геометрический приток данных, поступающих из различных разрозненных и разнородных источников. Увеличение масштаба таких систем определяет необходимость использования инструментов автоматизации мониторинга состояния информационных систем предприятия, обработки неструктурированных данных, позволяющих извлекать ценную информацию из журнальных файлов большого объема без необходимости вмешательства человека, с целью предупреждения и прогнозирования сбоев, отказов или вмешательств со стороны злоумышленников.

На сегодняшний IT технологии становятся неотъемлемой частью всех компаний, поддерживая широкий спектр онлайн-сервисов (таких как поисковые системы, социальные сети, различные типы ассистентов) и интеллектуальных приложений (таких как прогноз погоды, бизнес-аналитика, биомедицинская инженерия и т. д.). Большинство подобных систем обеспечивают работу сложного оборудования в различных отраслях: горнодобывающей промышленности, атомной индустрии при проектировании и эксплуатации АЭС, транспортной сфере городов и т.д. [1]. Поскольку почти все эти системы предназначены для круглосуточной работы, обслуживая тысячи компьютеров одновременно, высокая их доступность и надежность становятся обязательными. В следствии чего эффективность работы современных предприятий как с точки зрения безопасности, так и с точки зрения беспрерывного функционирования информационных систем, которые их обслуживают, становится одной из ключевых задач современного мира.

Одним из инструментов для эффективного анализа процессов, протекающих в IT-инфраструктуре компаний, стала такая наука как Интеллектуальный анализ процессов (Process mining). За последнее десятилетие интеллектуальный анализ процессов стал новой областью исследований, которая фокусируется на анализе процессов с использованием данных о событиях. Классические методы интеллектуального анализа данных, такие как классификация, кластеризация, регрессия, изучение правил ассоциации и анализ последовательности/эпизода, не фокусируются на моделях бизнес-процессов и часто используются

Kasanova A.M is with the National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Russia, Moscow (e-mail: AMKhasanova@mephi.ru).

только для анализа определенного шага в общем процессе [2]. Интеллектуальный анализ процессов сосредоточен на изучении и анализе процессов. Модели процессов используются для анализа (например, моделирования и проверки) безопасности и работоспособности, а также для оптимизации существующих процессов в системе. Все действия, выполняемые людьми, машинами и программами, оставляют следы в так называемых журналах событий. Методы интеллектуального анализа процессов используют такие журналы для обнаружения, анализа и улучшения бизнес-процессов.

Целью данной работы является исследование и реализация алгоритмов интеллектуального анализа процессов с целью оптимизации работы ОС, а также с целью выявления внештатных и вредоносных событий на примере журналов событий различных операционных систем (Windows, Linux).

II. Журналы событий операционных систем

Журнал событий – это специальные лог-файлы, в которые система и приложения записывают все значимые для ОС события, такие как установка нового устройства; ошибки в работе приложений; вход пользователей в систему; незапустившиеся службы и т.д. Анализ данных из журнала событий поможет системному администратору (и даже обычному пользователю) устранить неисправности в работе операционной системы, программного обеспечения и оборудования, отследить аномальные или вредоносные событий.

ОС Windows и Linux были выбраны для исследования как наиболее популярные операционные системы в современном мире.

В зависимости от ОС событий данной системы имеют разный формат и хранятся в различных местах. Сами события в журналах также различаются в зависимости от ОС. Список возможных событий в системах Windows превышает 10000 [3,4]. В системах Unix и Linux различают более 10 источников событий и большее количество уровней приоритета [3,4].

A. *Журналы событий ОС Windows*

В операционной системе Windows существует встроенная программа сбора журналов событий - «Журнал событий Windows» (Рис. 1).

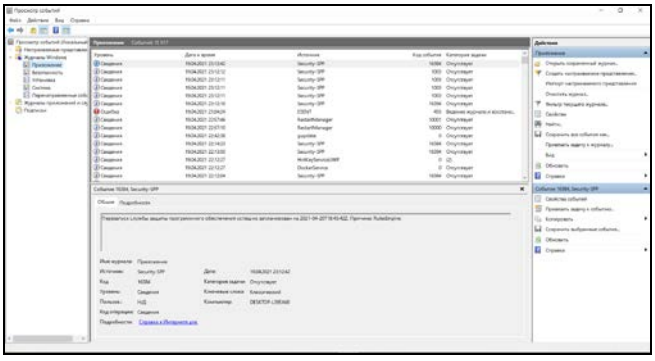

Рис. 1 – Интерфейс «Просмотр событий Windows».

Журналы событий Windows включает три основные и две дополнительные категории событий: основные – это Приложение, Система, Безопасность (рисунок 3); дополнительные – Установка и Перенаправленные события [5]. Рассмотрим их назначение:

- Приложение – хранит важные события, связанные с приложениями, установленными в системе. При этом записываются ошибки, возникающие в приложении, информационные события и предупреждения программных приложений. Эти данные помогут системному администратору установить причину отказа той или иной программы.
- Система – хранит события операционной системы или ее компонентов. Файлы системного журнала могут содержать данные об аппаратных изменениях, драйверах устройств, системных изменениях и всех действиях, связанных с самой системой.
- Безопасность – содержит сведения о входе/выходе из системы, управление учётными записями, изменение разрешений и прав доступа к файлам и папкам и других действиях, связанных с безопасностью Windows. Эти события определяются политикой аудита системы.

Журналы Windows

Файл	Тип	Число событий	Размер
Приложение	Административный	7 275	7,07 МБ
Безопасность	Административный	30 198	20,00 МБ
Установка	Работает	20	68 КБ
Система	Административный	7 445	6,07 МБ
Перенаправленные события	Работает	0	0 байт

Рис. 2 – Интерфейс «Просмотр событий Windows».

Для получения полного журнала событий из всех источников ОС Windows в данной работе была использована утилита Sysmon, которая отображает аггрегированные журналы событий в средстве просмотра событий ОС Windows. С помощью power-shell можно получить все события, которые агрегирует Sysmon в формате XML:

Get-WinEvent -LogName «Microsoft-Windows-Sysmon/Operational» | Export-Clixml «sysmon_logs.xml»

B. *Журналы событий ОС Linux*

В Linux есть специальный каталог для хранения журналов, который называется /var/log. Этот каталог содержит журналы самой ОС, служб и различных приложений, работающих в системе. На Рис. 3 показан типичный журнал событий для Ubuntu.

```
ubuntu@ip-172-31-11-241:~$
ubuntu@ip-172-31-11-241:~$ cd /var/log
ubuntu@ip-172-31-11-241:/var/log$ ls
alternatives.log      btmp.1          dpkg.log.8.gz   news
alternatives.log.1    cloud-init.log  dpkg.log.9.gz   puppet
alternatives.log.2.gz ConsoleKit      fontconfig.log  rsyslog-stats
alternatives.log.3.gz datasync        fsck            syslog
alternatives.log.4.gz dist-upgrade    kern.log        syslog.1
alternatives.log.5.gz dmesg           kern.log.1      syslog.2.gz
alternatives.log.6.gz dmesg.0         kern.log.2.gz   syslog.3.gz
alternatives.log.7.gz dmesg.1.gz      kern.log.3.gz   syslog.4.gz
alternatives.log.8.gz dmesg.2.gz      kern.log.4.gz   syslog.5.gz
apache2               dmesg.3.gz      landscape       syslog.6.gz
apport.log            dmesg.4.gz      lastlog         syslog.7.gz
apport.log.1          dpkg.log        mail.err        sysstat
apt                   dpkg.log.1      mail.err.1      tomcat6
auth.log              dpkg.log.10.gz  mail.err.2.gz   udev
auth.log.1            dpkg.log.2.gz   mail.err.3.gz   ufw.log
auth.log.2.gz         dpkg.log.3.gz   mail.log        unattended-upgrades
auth.log.3.gz         dpkg.log.4.gz   mail.log.1      upstart
auth.log.4.gz         dpkg.log.5.gz   mail.log.2.gz   wtmp
boot.log              dpkg.log.6.gz   mail.log.3.gz   wtmp.1
btmp                  dpkg.log.7.gz   mail.log.4.gz
```

Рис. 3 – Пример журналов событий ОС Linux

Кроме того, посмотреть журналы на Linux можно и с

помощью графических утилит. Программа «Журналы» может быть использована для удобного просмотра и отслеживания системных журналов на ноутбуке или персональном компьютере с Linux (Рис. 4).

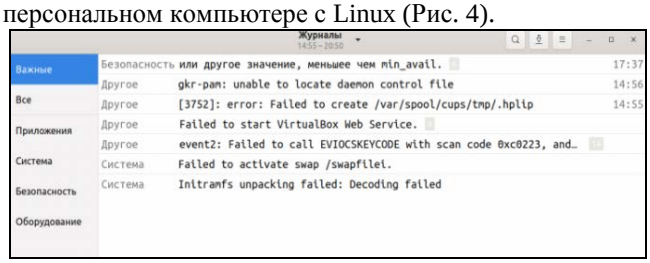

Рис. 4 – Графическая утилита Linux для просмотра журналов событий.

Файлы журналов, созданные в среде Linux, как и в ОС Windows, можно разделить на четыре категории [7]:

- Журналы приложений;
- Журналы событий;
- Сервисные журналы;
- Системные журналы.

Некоторые из наиболее важных системных журналов Linux включают:

- /var/log/syslog и /var/log/messages хранят все глобальные данные об активности системы, включая сообщения о запуске. Системы на основе Debian, такие как Ubuntu, хранят это в /var/log/syslog, а системы на основе Red Hat, такие как RHEL или CentOS, используют /var/log/messages.
- /var/log/auth.log и /var/log/secure хранят все события, связанные с безопасностью, такие как вход в систему, действия пользователя root и выходные данные подключаемых модулей аутентификации (PAM). Ubuntu и Debian используют /var/log/auth.log, а Red Hat и CentOS используют /var/log/secure.
- /var/log/kern.log хранит журналы событий ядра, ошибок и предупреждений, которые особенно полезны для устранения неполадок в пользовательских ядрах.
- /var/log/cron хранит информацию о запланированных задачах.

III. Общий ход работы

Процесс анализа журналов для обнаружения аномалий включает четыре основных этапа:

1. Сбор журналов.

В данной работе сбор журналов событий ОС Windows осуществлялся с помощью программного обеспечения Sysmon, командной строки PowerShell и встроенного в операционную систему Windows модуля просмотра событий. Сбор журналов событий ОС Linux осуществлялся с помощью командной строки и syslog. Сбор журналов событий подразумевает:

- извлечение из системы полного набора логов определенный период времени (объемом порядка 60 тысяч событий),
- экспорт полученных данных в формат XML.

2. Предварительная обработка журналов событий.

Этап предполагает парсинг журналов событий в необходимый формат для последующего интеллектуального анализа: CSV, DataFrame, формат для построения различных алгоритмов Process mining, а также построение иерархии процессов для восстановления соотношения «Событие - Процессы, относящиеся к этому событию». Целью синтаксического анализа журналов является извлечение группы шаблонов событий, с помощью которых можно структурировать необработанные журналы.

3. **Извлечение признаков и необходимых данных** - приведение типов данных, т.е. преобразование строковых данных в числовые, в формат времени и даты и т.д., а также извлечение шаблонных событий с помощью алгоритмов обнаружения процессов (Alpha алгоритм, эвристический алгоритм и индуктивный).

4. **Обнаружение аномалий** - разработка алгоритмов по обнаружению событий, не соответствующих нормальному ходу процесса (аномальные, вредоносное воздействие).

5. **Проведение эксперимента.**

IV. Предварительная обработка журналов событий

A. Преобразование формата журналов событий

Для начала работы с полученными журналами событий в формате XML, необходимо преобразовать данные в понятный для Process mining формат: CVS или XES.

В данной работе была разработана функция, которая парсит XML файл с помощью библиотеки lxml и преобразовывает данные в DataFrame и сохраняет данные в CSV формат с помощью библиотеки Pandas:

• xml_to_df(xml_file) – функция преобразования данных их XML формата в DataFrame и сохранение данных в CSV формате

• xml_to_df2(df) – функция получения дополнительной информации из параметров свойства и сообщение события

На вход функция xml_to_df(xml_file) принимает название XML файла и возвращает DataFrame. С помощью цикла осуществляется проход по всему файлу (Рис. 5) и по тэгам в XML файле выделяются необходимые поля. Количество и наименование полей для каждого события может отличаться в зависимости от типа данного события, а также в зависимости от источника данных (Windows/Linux), каждый тип событий обладает своим набором параметров. Тем не менее каждое событие обладает общими параметрами такими, как:

- Имя компьютера (MachineName)
- ID события (EventID)
- Категория события (CategoryNumber)
- Источник события (Source)
- Свойства события (Properties)
- Уникальный ID экземпляра (ProcessId)
- Сообщение события (Message)

```
for index, child in enumerate(root):
    event = []
    for child2 in child:
        if child2.tag == 'Props':
            for i in child2:
                if i.tag == 'S' and i.attrib['N'] == 'MachineName
                    MachineName.append(i.text)
```

Рис. 5 – Пример получения имени компьютера из XML файла.

Результаты преобразования данных в DatFrame с помощью функция представлены на Рис.6 и Рис.7.

Рис. 6 – Пример одного процесса ОС Windows до обработки.

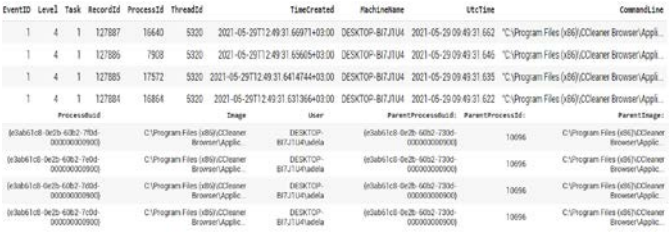

Рис. 7 – Преобразованные логи событий ОС Windows.

Таким образом, после преобразования каждый процесс имеет 16 параметров, которые характеризуют данный процесс.

B. *Преобразование типов и извлечение признаков*

После получения и преобразования журналов событий из системы Windows 10 необходимо преобразовать журнал событий по умолчанию в форму, удобную для использования алгоритмов Process mining с помощью библиотеки pm4py [8, 9, 10].

Минимальные требования для преобразования журнала событий в необходимый формат можно посмотреть Руководстве пользователя Disco [11]. В журнале событий должны быть как минимум три элемента для обеспечения анализа интеллектуального анализа процессов:

• отметка времени (Timestamp) – время создания процесса;

• идентификатор случая (Case ID) – уникальное Id события, которое объединяет в себе все процессы, относящие к данному событию;

• действие (Activity) – данные о самом событии, пусть к файлу, который запустил данный процесс.

Эти три элемента позволяют взглянуть на данные с точки зрения процесса. Также можно использовать другие элементы, такие как, например, ресурсы, состояние, приоритет и т. д.

Исходя из полученного журнала событий необходимо преобразовать существующие колонки для последующего применения алгоритмов интеллектуального анализа в соответствии с Таблицей I.

Таблица I. Переименование параметров (колонок) журнала событий

Название параметра для применения алгоритмов Process mining с помощью библиотеки Pm4py	Название параметра для использования журнала событий в программе Disco	Текущее название параметра
case:concept:name	Case ID	Case_id
concept:name	Activity	Image
time:timestamp	Timestamp	UtcTime

Используя уникальный Id процесса (ProcessId) и уникальный Id родительского процесса (ParentProcessId) можно восстановить полную иерархию всех событий, необходимую для дальнейшего анализа журналов событий. Для поиска цепочек событий и построения иерархии, описанной выше были разработаны следующие функции:

1. get_paths(df, find) – функция, которая принимает на вход два параметра df – DataFrame, содержащий все события, find – начальный процесс Id для события. Возвращает все цепочки событий, которые начинаются с find.

2. get_all_paths(df) – функция, которая принимает на вход df – DataFrame, содержащий все события и для каждого процесса составляет полную цепочку процессов, группируя их в события.

3. csv_to_logs(df) – функция, которая принимает на вход df – DataFrame, содержащий все события и преобразовывает существующие параметры в необходимый формат.

Пример данных, преобразованных с помощью указанных функций представлен на Рис. 8.

Рис. 8 - Преобразованные журналы событий ОС Windows.

V. Интеллектуальный анализ журналов событий

A. *Figures and Tables*

Согласно манифесту Process mining [12], журналы событий могут быть использованы для осуществления трех форм Process mining:

- Извлечение;
- Соответствие;
- Усовершенствование.

Первый тип интеллектуального анализа процессов -

На сегодняшний день разработано большое количество алгоритмов, предназначенных для извлечения процессов из журналов событий [13-19]. Среди алгоритмов извлечения обычно выделяют три основные, задающие целые семейства подходов:

1. Alpha-алгоритм и Alpha+ алгоритм [20].
2. Эвристические подходы [19].
3. Индуктивные алгоритмы [16].

Вторая форма Process mining – это проверка соответствия. На данном этапе проводится

сопоставление существующей модели процесса с журналом событий этого же процесса и оценка полученной модели. Проверка соответствия может быть использована для оценки того, насколько реальные данные журнала соответствуют модели, и наоборот. В библиотеке Pm4py для построения алгоритмов Process Mining представлены все вышеописанные алгоритмы, также библиотека предоставляет методы оценки полученной модели с помощью различных метрик:

- соответствие модели журналам событий (Fitness)
- точность (Precision)
- обобщенность (Generalization)

Третий тип интеллектуального анализа процессов - это улучшение. На данном этапе происходит расширение или улучшение существующей модель процесса, используя информацию о фактическом процессе, записанную в некотором журнале событий. В отличии от второго типа, который измеряет соответствие между моделью и реальностью, третий тип анализа процессов направлен на изменение или расширение априорной модели.

Каждый из алгоритмов обнаружения процессов был протестирован на полученных журналах событий ОС Windows и Linux и проведен сравнительный анализ данных алгоритмов по различным метрикам, соответсвующих второму типу Process Mining. Результат сравнения алгоритмов представлен в Таблице II.

Таблица II. Сравнение алгоритмов обнаружения процессов

Алгоритм	Соответствие модели журналам событий (Fitness)	Точность (Precision)	Обобщенность (Generalization)
Alpha Miner	0.491	0.197	0.422
Heuristic Miner	0.765	0.521	0.487
Inductive Miner	0.96	0.709	0.957

Исходя из полученных результатов сравнения, наилучшим алгоритмом для обнаружения процессов и построения модели, оказался индуктивный алгоритм, на основе которого модель была визуализирована с помощью DFG графа и сети Петри.

Для дальнейшего поиска аномалий или нарушений ОС Windows с помощью алгоритмов Process mining используется построенная модель для журналов событий.

VI. Обнаружение аномалий

Имея созданную модель по журналу событий, можно отслеживать события, которые выходят за рамки поведения ОС. Используя данную модель, есть возможность отслеживать аномальные или вредоносные события.

Для проверки разработанных алгоритмов была проведена симуляция вредоносного события в ОС Windows.

Пользователь ОС Windows в результате фишинговой атаки открывает вложение к письму — документ Microsoft Excel с завлекающим названием (например, «Данные по зарплате за май.xlsx»). Файл содержит вредоносный макрос, который, обманом получив у пользователя разрешение на запуск, выполняет следующую последовательность действий:

1. Подключается к серверу управления атакующего и скачивает файл viewpage.php, содержащий полезную нагрузку — meterpter reverse shell;
2. Переименовывает загруженный файл и сохраняет его в каталоге %TEMP% под именем sysprov32.dll;
3. Прописывает в ключ реестра HKCU\Software\Microsoft\Windows\CurrentVersion\Run значение userprep со следующим содержимым: rundll32 C:\Users\Adele\AppData\Local\Temp\sysprov32.dll;
4. Запускает полезную нагрузку, которая содержится в файле sysprov32.dll, командой rundll32 C:\Users\vadmin\AppData\Local\Temp\sysprov32.dll;
5. Прописывает полезную нагрузку в ключ реестра;
6. Выполняет дапм учетных данных и происходит кража учетных данных.

После того как запущен reverse shell, хакер получает удаленный доступ к компьютеру пользователя и при помощи PowerShell-скрипта Invoke-Mimikatz.ps1 выполняет загрузку учетных данных пользователей из локальной базы. Но для успешного завершения получения личных данных пользователей, необходима последняя версия утилиты Mimikatz. Поэтому атакующий загружает на компьютер пользователя последнюю версию утилиты Mimikatz в виде исполняемого PE-файла и сохраняет его в каталоге c:\Users\vadmin\Documents\ с именем m.exe. После запуска этого файла атакующий крадет учетные данные из оперативной памяти.

На первом этапе работы вредоносного макроса-скрипта из файла MS EXCEL атакующий создает сетевое подключение к удаленному серверу. Такие события в Sysmon будут иметь Event ID = 3.

Используя данные из Threat Intelligence, можно получить IP-адреса сервера, которые считаются вредоносными. Согласно данным Threat Intelligence IP-адрес 31.179.135.186 используется вредоносным программным обеспечением. Таким образом при открытии вредоносного файла в операционной системе появляется скомпрометированный хост или группа хостов, которые осуществляли или продолжают осуществлять подключения к вредоносному серверу управления с IP-адресом 31.179.135.186.

По специфичному event_id для сетевых подключений и IP- адрес на начальном этапе можно обнаружить аномальное событие.

Так как атака начинается с открытия MS Excel, то мы в журнале событий увидим новый процесс, связанный с открытием программы:

C:\Program Files (x86)\Microsoft Office\Office16\excel.exe

Даже если бы в базе TI-платформы не оказалось IP-адреса атакующего, мы бы заметили, что офисное приложение подключается к внешнему IP-адресу (Рис. 9).

User	Image	ProcessId	UtcTime	ip_src_ipv4	ip_dst_ipv4	dst_port
DESKTOP-L39EA60\Adele	C:\Program Files (x86)\Microsoft Office\Office16\EXCEL.EXE	13700	23.05.2021 17:49	192.168.8.194	31.179.135.186	443
DESKTOP-L39EA60\Adele	C:\Program Files (x86)\Microsoft Office\Office16\EXCEL.EXE	13712	23.05.2021 17:48			

Рис. 9 – Процессы открытия офисного приложения и подключения офисным приложением к внешнему IP-адресу.

Одни из самых эффективных атак, которые осуществляют атакующие, заключаются в том, что злоумышленники внедряют вредоносные макросы в виде легковесного и незаметного кода в различные документы. В рассматриваемом случае злоумышленник использует стандартные приложения Microsoft Office Excel (также могут быть использованы MS Word, MS PowerPoint и др.). В MS Excel в данном случае был внедрен код злоумышленника, который сохраняет в файловой системе файл с полезной нагрузкой для его последующего запуска. Если атакующий не замаскировал расширение файла под более безобидное, то можно использовать событие FileCreate Sysmon (Event ID = 1) для обнаружения подобной активности.

Результаты запроса показывают, что на хосте DESKTOP-L39EA60 процессом C:\Program Files (x86)\Microsoft Office\Office16\excel.exe был создан исполняемый файл C:\Users\vadmin\AppData\Local\Temp\sysprov32.dll (Рис. 10).

EventID	UtcTime	Image	ProcessId	ParentProcessId	ParentImage:
13	23.05.2021 17:54	HKU\S-1-5-21-3921924719-2751751025-4067464375-1003\Software\Microsoft\Windows\CurrentVersion\Run	1564	5048	C:\Users\vadmin\AppData\Local\Temp\sysprov32.dll
1	23.05.2021 17:52	C:\Users\vadmin\AppData\Local\Temp\sysprov32.dll	5048	10380	C:\Windows\SysWOW64\rundll32.exe
3	23.05.2021 17:52	C:\Windows\SysWOW64\rundll32.exe	10380	7308	C:\Windows\System32\cmd.exe
1	23.05.2021 17:51	C:\Windows\System32\cmd.exe	7308	13700	C:\Program Files (x86)\Microsoft Office\Office16\EXCEL.EXE
3	23.05.2021 17:49	C:\Program Files (x86)\Microsoft Office\Office16\EXCEL.EXE	13700	13712	C:\Program Files (x86)\Microsoft Office\Office16\EXCEL.EXE
1	23.05.2021 17:48	C:\Program Files (x86)\Microsoft Office\Office16\EXCEL.EXE	13712		

Рис. 10 – Пример вредоносного события.

Т.е. когда пользователь открыл скаченный Excel документ, то макрос получил разрешение на выполнение. Далее макрос загружает с сервера злоумышленника исполняемый файл с основной полезной нагрузкой (DLL-библиотека Reverse Shell-a) и сохраняет его в каталоге временных файлов под именем sysprov32.dll.

После того как файл с полезной нагрузкой скачен, он будет запущен через командную строку. Здесь как раз будет важным знание об иерархии процессов «родительский процесс — дочерний процесс», а с помощью построенной модели с использованием алгоритмов Process Mining есть знание о том, какие пары «родительский процесс — дочерний процесс» являются для ОС Windows нормальными. Запуск процессом офисного приложения командного интерпретатора cmd — аномальное событие. Оно может свидетельствовать об исполнении вредоносного кода, встроенного в документ, например, макроса или DDE.

Из журнала событий видно, что на хосте DESKTOP-L39EA60 процессом C:\Program Files (x86)\Microsoft Office\Office16\excel.exe был запущен командный интерпретатор cmd с командной строкой C:\Windows\System32\cmd.exe /c rundll32 C:\Users\vadmin\AppData\Local\Temp\sysprov32.dll

Таким образом, для исполнения полезной нагрузки, которая содержится в файле sysprov32.dll, злоумышленник использовал программу rundll32.

Rundll32.exe — легитимная программа ОС Windows, предназначенная для запуска программ, хранящихся в файлах DLL. DLL - это библиотека динамической компоновки, общий набор процедур, используемых рядом программ в Windows. Чтобы запустить одну из этих подпрограмм напрямую, программа rundll32.exe соответствует своему названию и запускает программный файл dll. Но у программы есть один недостаток, который заключается в том, что она может легко запустить процесс, который является скрытым вредоносным ПО.

Когда пользователь открыл вредоносный документ, атакующий получил удаленный доступ к компьютеру пользователя. Но при перезагрузке системы пользователем или ее отключения, удаленный доступ будет потерян. Для того чтобы сохранить постоянный доступ, могу быть использованы различные методы закрепления в системе (Persistence). Один из примеров такого метода – это прописать полезную нагрузку в ключи реестра, которые отвечают за автозагрузку программ и выполнения определенного кода в тот момент, когда пользователь осуществляет вход в систему.

В случае данного инцидента полезная нагрузка была сохранена в ключе реестра HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\ Run. Но событие, которое характеризуется тем, что стандартное офисное приложение вносит свое значение в ключи реестра, отвечающие за автозагрузку, является аномальным и может свидетельствовать о попытках вредоносного кода закрепиться в системе.

Из журнала событий видно, что процессом C:\Program Files (x86)\ Microsoft Office \Office16\excel.exe в ключе реестра HKU\ S-1-5-21-3921924719-2751751025-4067464375-1003\ Software\ Microsoft\ Windows\ CurrentVersion\ Run было создано значение userprep с содержимым rundll32 C:\Users\Adele\AppData\Local\Temp\sysprov32.dll.

В журналах событий Sysmon, код Event ID = 13 (идентификатор событий реестра, который определяет различие изменения в реестре), либо аудит событий безопасности ОС Windows (Event ID = 4657), могут быть использованы для отслеживания и обнаружения активности, связанной с внесение новых значений в реестр для обнаружения аномальных событий.

Для детектирования данного инцидента были использованы следующие поведенческие признаки, на базе которых были разработаны соответствующие правила:

- обращение процесса приложения Microsoft Office к адресу, который фигурирует в используемых источниках Threat Intelligence как вредоносный;
- взаимодействия процесса приложения Microsoft Office с внешними адресами;
- создание процессом приложения Microsoft Office файла с исполняемым расширением;
- прописывание полезной нагрузки в ключ реестра.

Все данные события и процессы были обнаружены из

анализа журнала событий ОС Windows и сделан вывод о том, что совершены вредоносные действия.

VII. Заключение

Данная работа посвящена применению технологии Process mining для выявления аномальных ситуаций в работе высокотехнологичного оборудования с использованием журналов событий. В качестве примера использовалась ОС Windows и Linux, как одна из самых популярных современных операционных систем с большим количеством журналов событий.

Разработанные алгоритмы позволяют автоматизировать процесс получения полных журналов системных событий, а также их обработку, ускорить режим просмотра журналов, автоматизировать процесс обнаружения аномальных событий в системе, что способствует повышению безопасность и эффективность системы.

В дальнейшем планируется расширить работу и использовать для анализа журналы событий других операционных систем, таких как Mac OS, Android. А, также оптимизировать уже разработанные алгоритмы по времени и используемой памяти.

Благодарности

Автор выражает благодарность НИЯУ МИФИ за помощь в возможности реализации данной работы и публикации полученных результатов.

Библиография

[1] Brzychczy E., Gackowiec P., Liebetrau M. Data Analytic Approaches for Mining Process Improvement—Machinery Utilization Use Case //Resources. – 2020. – Т. 9. – №. 2. – С. 17.

[2] Van Der Aalst W. Process mining: Overview and opportunities //ACM Transactions on Management Information Systems (TMIS). – 2012. – Т. 3. – №. 2. – С. 1-17.

[3] Bassil Y. Windows and Linux operating systems from a security perspective //arXiv preprint arXiv:1204.0197. – 2012.

[4] Sosnowski J., Gawkowski P., Cabaj K. Event and performance logs in system management and evaluation //Information Systems in Management XIV, Security and Effectiveness of ICT Systems. – 2011. – С. 83-93.

[5] Dolak R., Janakova M., Botlik J. Process Mining of Events Log from Windows //SIMPDA. – 2018. – С. 73-77.

[6] Zeng L. et al. Computer operating system logging and security issues: a survey //Security and communication networks. – 2016. – Т. 9. – №. 17. – С. 4804-4821.

[7] Choi J. et al. Live forensic analysis of a compromised linux system using LECT (Linux Evidence Collection Tool) //2008 International Conference on Information Security and Assurance (isa 2008). – IEEE, 2008. – С. 231-236.

[8] Šrol E. Process Mining usage for potential insider threat identification utilizing PM4Py.

[9] Berti A., van Zelst S. J., van der Aalst W. Process Mining for python (PM4Py): bridging the gap between process-and data science //arXiv preprint arXiv:1905.06169. – 2019.

[10] Pm4py documentation // Pm4py – URL: https://pm4py.fit.fraunhofer.de/ (дата обращения: 10.05.2021)

[11] 23. Fluxicon Disco User's Guide, https://fluxicon.com/disco/files/Disco-User-Guide.pdf McGrath, M., Price, M.: Windows 10 in easy steps - Special Edition: To venture further. In Easy Steps Limited, Warwickshire (2015)

[12] Van Der Aalst W. et al. Process Mining manifesto //International Conference on Business Process Management. – Springer, Berlin, Heidelberg, 2011. – С. 169-194.

[13] .Van der Aalst W. M. P. Process Mining: discovery, conformance and enhancement of business processes. Springer, 2011.

[14] Van der Aalst W.M.P., Weijters A.J.M.M., Maruster L. Workflow Mining: Discovering Process Models from Event Logs // IEEE Transactions on Knowledge and Data Engineering, 2004. Vol. 16(9). P. 1128–1142.

[15] Van der Aalst W.M.P., Adriansyah A., Van Dongen B.F. Replaying history on process models for conformance checking and performance analysis // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Vol. 2(2). Wiley Online Library. 2012. P. 182–192.

[16] Adriansyah A., Van Dongen B.F., Van der Aalst W.M.P. Conformance checking using costbased fitness analysis // 15th IEEE International Conference on Enterprise Distributed Object Computing Conference (EDOC). 2011. P. 55–64

[17] Leemans S. J. J., Fahland D., Van der Aalst W. M. P. Discovering Block-Structured Process Models from Incomplete Event Logs. Tech. Rep. BPM-14-05. Eindhoven University of Technology. March 2014.

[18] Van der Werf J. M. E. M. et al. Process discovery using integer linear programming // Applications and Theory of Petri Nets. Springer Berlin Heidelberg, 2008. P. 368–387.

[19] Weijters A., Van der Aalst W. M. P., De Medeiros A. K. A. Process Mining with the heuristics miner-algorithm // Technische Universiteit Eindhoven, Tech. Rep. WP. 2006. Vol. 166. P. 1–34.

Process mining methods to analyze event logs of information systems

Adelya Khasanova

***Abstract* - The purpose of this work is to study and implement algorithms for intelligent process analysis in order to optimize the operation of the OS, as well as to identify abnormal and malicious events using the example of event logs of various operating systems (Windows, Linux). Event logs of information systems in various fields of human activity (mining, the nuclear industry in the design and operation of nuclear power plants, the transport sector of cities, the banking sector, etc.) can become a source of valuable information about the processes occurring in the system. Since almost all of these systems are designed to operate around the clock, serving thousands of computers and users at the same time, their high availability, reliability and security become mandatory.**

The article provides a study of event logs of different operating systems and a description of the developed methods for obtaining, processing and analyzing event logs in order to prevent and predict failures, failures or abnormal events, as well as to improve the optimization of existing processes. The paper provides modeling of malicious events and their detection, as well as code examples to demonstrate all of the above algorithms.

***Keywords*: Process mining, Windows OS, Linux, anomalous situations, optimization, security, event logs.**

References

[1] Brzychczy E., Gackowiec P., Liebetrau M. Data Analytic Approaches for Mining Process Improvement—Machinery Utilization Use Case //Resources. – 2020. – Т. 9. – №. 2. – С. 17.

[2] Van Der Aalst W. Process mining: Overview and opportunities //ACM Transactions on Management Information Systems (TMIS). – 2012. – Т. 3. – №. 2. – С. 1-17.

[3] Bassil Y. Windows and Linux operating systems from a security perspective //arXiv preprint arXiv:1204.0197. – 2012.

[4] Sosnowski J., Gawkowski P., Cabaj K. Event and performance logs in system management and evaluation //Information Systems in Management XIV, Security and Effectiveness of ICT Systems. – 2011. – С. 83-93.

[5] Dolak R., Janakova M., Botlik J. Process Mining of Events Log from Windows //SIMPDA. – 2018. – С. 73-77.

[6] Zeng L. et al. Computer operating system logging and security issues: a survey //Security and communication networks. – 2016. – Т. 9. – №. 17. – С. 4804-4821.

[7] Choi J. et al. Live forensic analysis of a compromised linux system using LECT (Linux Evidence Collection Tool) //2008 International Conference on Information Security and Assurance (isa 2008). – IEEE, 2008. – С. 231-236.

[8] Šrol E. Process Mining usage for potential insider threat identification utilizing PM4Py.

[9] Berti A., van Zelst S. J., van der Aalst W. Process Mining for python (PM4Py): bridging the gap between process-and data science //arXiv preprint arXiv:1905.06169. – 2019.

[10] Pm4py documentation // Pm4py – URL: https://pm4py.fit.fraunhofer.de/ (дата обращения: 10.05.2021)

[11] Fluxicon Disco User's Guide, https://fluxicon.com/disco/files/Disco-User-Guide.pdf McGrath, M., Price, M.: Windows 10 in easy steps - Special Edition: To venture further. In Easy Steps Limited, Warwickshire (2015)

[12] Van Der Aalst W. et al. Process Mining manifesto //International Conference on Business Process Management. – Springer, Berlin, Heidelberg, 2011. – С. 169-194.

[13] Van der Aalst W. M. P. Process Mining: discovery, conformance and enhancement of business processes. Springer, 2011.

[14] Van der Aalst W.M.P., Weijters A.J.M.M., Maruster L. Workflow Mining: Discovering Process Models from Event Logs // IEEE Transactions on Knowledge and Data Engineering, 2004. Vol. 16(9). P. 1128–1142.

[15] Van der Aalst W.M.P., Adriansyah A., Van Dongen B.F. Replaying history on process models for conformance checking and performance analysis // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. Vol. 2(2). Wiley Online Library. 2012. P. 182–192.

[16] Adriansyah A., Van Dongen B.F., Van der Aalst W.M.P. Conformance checking using costbased fitness analysis // 15th IEEE International Conference on Enterprise Distributed Object Computing Conference (EDOC). 2011. P. 55–64

[17] Leemans S. J. J., Fahland D., Van der Aalst W. M. P. Discovering Block-Structured Process Models from Incomplete Event Logs. Tech. Rep. BPM-14-05. Eindhoven University of Technology. March 2014.

[18] Van der Werf J. M. E. M. et al. Process discovery using integer linear programming // Applications and Theory of Petri Nets. Springer Berlin Heidelberg, 2008. P. 368–387.

[19] Weijters A., Van der Aalst W. M. P., De Medeiros A. K. A. Process Mining with the heuristics miner-algorithm // Technische Universiteit Eindhoven, Tech. Rep. WP. 2006. Vol. 166. P. 1–34.

First A. Khasanova Adelya Marselevna. Date of birth: May 6, 1997. Place of birth: Russia, rep. Bashkortostan, Sterlitamak. Education: NRNU MEPhI, «Informatics and Computer Engineering», Bachelor's Degree (2015-2019); NRNU MEPhI, «Software Engineering», Master's Degree (2019-2021). PhD: NRNU MEPhI, «Informatics and Computer Engineering» (2021-2025)/