УДК 004:303.68

МОНИТОРИНГ РАБОТЫ КОРПОРАТИВНЫХ ПОЛЬЗОВАТЕЛЕЙ

С.В. Трошин

ГОУ ВПО «Московский государственный университет им. М.В. Ломоносова», г. Москва

Рецензент В.Е. Подольский

Ключевые слова и фразы: мониторинг работы; раннее обнаружение вторжений; статистический анализ.

Аннотация: Предложена технология (архитектура, модель, методы и алгоритмы) построения систем мониторинга работы корпоративных пользователей на основе сбора и анализа журналируемой информации. В качестве основных целей анализа рассмотрены задачи определения статистических параметров и характеристик работы пользователей и построения обобщенного профиля работы, а также задачи раннего обнаружения внутренних вторжений и анализ следов уже произошедших атак. В качестве основы для алгоритмов раннего обнаружения вторжений и следов атак взяты методы машинного обучения. Для поиска аномалий в работе пользователей предлагается строить модель нормального поведения, а затем классифицировать текущую активность пользователя как нормальную или аномальную на основе построенной модели. В работе описаны основные концепции сбора и обработки информации, основные модули систем, некоторые алгоритмы и методы, предложенные в рамках технологии, для реализации систем. Разработанная на основе технологии система мониторинга для ОС семейства Windows экспериментально протестирована в ряде государственных и коммерческих учреждений. Результаты экспериментов показали пригодность системы для решения первоочередных задач, стоящих перед службами администрации и IT-безопасности.

1. Введение

Развитие и повсеместное внедрение информационных систем, а также увеличение объемов и стоимости информации приводят к необходимости контроля использования информационных ресурсов предприятия, а также обеспечения информационной безопасности. Основную угрозу безопасности представляют вторжения в компьютерные системы. Под вторжением (или атакой) в компьютерную систему понимается любая деятельность пользователя или программы, нарушающая целостность, конфиденциальность и/или доступность данных. Наибольший ущерб причиняется внутренними вторжениями, поскольку такое вторжение осуществляется пользователем (инсайдером), который уже имеет доступ к компьютерной системе и даже, возможно, является легальным пользователем. Результатом внутренней атаки может являться разрушение или кража важной, конфиденциальной информации. Программные средства мониторинга работы пользователей широко применяются подразделениями служб информационной безопасности и администрации компаний, однако результаты исследования «National Survey on Managing the Insider Threat», опубликованного журналом ComputerWorld [4], а также совместные отчеты ФБР и Computer Security Institute [1] приводят следующие данные:

1) более 90 % компаний и организаций сталкивались и получали ущерб от внутренних вторжений, а 60 % сталкиваются с этим регулярно;

2) более 30 % времени работы отделов IT-безопасности уходит на обнаружение следов уже произошедших внутренних вторжений;

3) средняя Hi-Tech компания США сейчас тратит около 1 млн в год на защиту от внутренних вторжений и готова повышать эти расходы.

Согласно проведенным исследованиям первоочередными задачами мониторинга практически всех компаний являются:

Определение параметров работы пользователей. Оценка эффективности работы сотрудников с информационными ресурсами компании, выявление фактов нецелевого или непрофессионального использования корпоративных вычислительных средств. Сбор и комплексная обработка статистики по работе пользователей. Построение для выделенных пользователей и пользовательских групп моделей поведения – профилей использования информационных ресурсов корпоративной сети.

«Раннее обнаружение» внутренних вторжений. Выявление действий корпоративных пользователей, которые могут предшествовать внутренним вторжениям или попыткам похищения конфиденциальной информации. Механизм обнаружения аномалий может помочь не только обнаружить вторжение, но и предотвратить его.

Выявление следов внутренних вторжений. Интеллектуальный анализ больших объемов прецедентной информации с целью выявления попыток внутренних вторжений или похищения конфиденциальной информации. Как известно, традиционные системы обнаружения вторжений и предотвращения утечек конфиденциальной информации, основанные на экспертных знаниях (правилах или сигнатурах), не устойчивы к новым типам атак, данных о которых просто еще нет в базе знаний системы. Поэтому единственным решением в этой ситуации является использование интеллектуальных систем, основанных на анализе и моделировании поведения пользователей, и программ с целью обнаружения аномалий в этом поведении. Такой тип анализа также может использоваться для поиска в прецедентных данных следов попыток вторжений и кражи конфиденциальной информации, пропущенных традиционными системами защиты, с целью выявления причин и устранения последствий.

2. Существующие подходы

Традиционно в системах обнаружения вторжений используется сигнатурный подход [3, 7], основная идея которого заключается в сравнении записей о событиях в системе с определенными шаблонами – правилами, описывающими атаки. Так как набор правил фиксирован, такие системы зависят от эксперта и невосприимчивы к новым типам вторжений, пока эксперт не опишет новые правила. В подобных системах обычно используется техника периодического просмотра собранных системой журналов (лог-файлов) аудита и приложений, однако набор журналов, используемых такими системами, зачастую достаточно ограничен, отчасти из-за необходимости для каждого вида журналов разрабатывать свои правила. Сигнатурный подход имеет ряд серьезных недостатков.

1. Системы, построенные на основе сигнатурных методов, не обнаруживают новые типы вторжений, поскольку базы знаний для подобных систем формируются экспертом и обновляются вручную или через третьи организации, предоставляющие системы защиты.
2. Особенностью систем обнаружения внутренних вторжений, построенных на основе сигнатурных методов, является необходимость «ручной» настройки системы обнаружения администратором или экспертом под конкретную конфигурацию защищаемой компьютерной системы. Такая настройка требует как серьезных временных затрат, так и высокой квалификации администратора.
3. Существуют методы, позволяющие «замаскировать» вторжение так, что с помощью набора правил его невозможно будет определить как несанкционированное действие. Профессиональный злоумышленник, имея доступ к базе знаний системы защиты (он может всегда ее получить, в частности, приобретя ее, как легальный пользователь), почти всегда может «обмануть» традиционную систему обнаружения вторжений, основанную на сигнатурных методах.
4. Существует класс потенциальных потребителей систем обнаружения вторжений, для которых использование внешних баз знаний недопустимо по соображениям безопасности. Кроме того, качество функционирования систем, построенных на основе внешних баз знаний, существенно зависит от качества и оперативности работы компаний, поддерживающих наполнение этих баз. Таким образом, деятельность этих компаний становит-ся уязвимым звеном в системе обеспечения компьютерной безопасности.
5. Большинство сигнатурных систем работают в рамках одного источника информации об активности пользователя (например, осуществляют мониторинг работы с файловой системой или «просматривают» сетевой трафик и т.д.), используя при этом специфические для этого источника правила и политику безопасности. Но как показывают исследования, для внутренних вторжений одного источника информации, как правило, недостаточно, поэтому необходимо осуществлять мониторинг многих разнородных показателей одновременно, консолидировать полученную

информацию в единый поток событий и отслеживать зависимости и корреляции для разнотипных событий безопасности.

На сегодняшний день актуально создание программных технологий построения эффективных систем защиты от внутренних вторжений, основанных на несигнатурных методах [2, 5] и обладающих свойствами автономности, адаптируемости и самообучаемости.

Подобного рода системы защиты должны решать следующие задачи:

- сбор и анализ статистики работы пользователей и приложений;
- построение и визуализация моделей поведения пользователей;
- выявление аномалий в работе пользователей и ПО;
- выявление внутренних угроз.

3. Предлагаемое решение

В работе предлагается использовать централизованный сбор и анализ информации, так как данный подход обладает следующими преимуществами:

1) позволяет осуществлять сбор доказательной базы для расследования уже случившихся нарушений;

2) повышение качества анализа за счет корреляции данных из различных источников;

3) повышение наглядности анализа за счет отображения информации, сразу обо всей сети или ее подмножестве;

4) меньшая загруженность клиентских узлов и, как следствие, возможность создания «незаметных» для пользователя агентов сбора.

3.1. Архитектура системы и основные модули

Для решения поставленных задач предлагается распределенная мультиагентная архитектура (рис. 1).

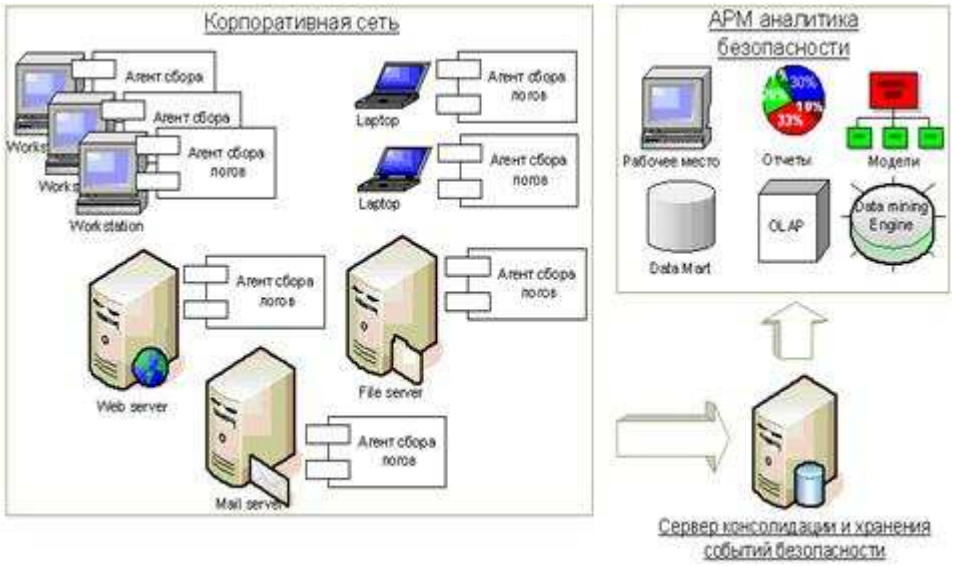

Рис. 1. Предложенная архитектура

Архитектура включает в себя агенты сбора данных, сервер консолидации данных, консоль управления сбором и рабочее место аналитика. Компоненты системы реализуют следующую основную функциональность:

1) *агент сбора* – считывает данные из журналов регистрации ОС, проводит предварительную обработку собранных данных (формирование фактов активности) и согласно политике планирования передает данные на сервер консолидации;

2) *сервер консолидации* – собирает данные от агентов сбора и помещает данные в общее хранилище;

3) *консоль управления сбором* – позволяет добавлять, удалять, настраивать агенты сбора, визуализирует статус работы агентов сбора и сервера консолидации;

4) *рабочее место аналитика безопасности* – позволяет запрашивать данные от сервера консолидации, фильтровать их, заполнять срезы данных (Data Mart), проводить различные виды анализа.

Разработанная архитектура позволяет функционировать агентам сбора в автономном режиме, агенты могут накапливать данные в локальном хранилище и передавать их на сервер консолидации позднее. Архитектура позволяет гибко настраивать параметры сбора исходных данных, а именно: набор журналов, набор событий, фильтрацию записей событий с использованием регулярных выражений. Архитектура позволяет настраивать политику передачи данных, а также просматривать статус работы компонентов системы и автоматически обнаруживать сбои в работе агентов.

Одним из важнейших требований к централизованной системе мониторинга работы пользователей является ее производительность. В современных корпоративных сетях, насчитывающих тысячи компьютеров, поток данных журналов регистрации может быть колоссален, особенно, если принять во внимание наличие журналов, запись в которые ведется со скоростью десятка событий в секунду. Консолидация, а главное – централизованный анализ такого объема данных являются вычислительно сложной, а иногда и невыполнимой задачей. Первым очевидным шагом к снижению потока данных является фильтрация данных во время сбора, однако, как показывает опыт, только фильтрации обычно недостаточно.

В работе предлагается подход распределения нагрузки анализа данных, заключающийся в формировании на агентах сбора фактов, описывающих активность пользователей. Факты активности формируются по указанным аналитиком правилам на основе собираемых событий. Такой подход позволяет распределить анализ между агентами сбора и рабочим местом аналитика, унифицировать анализ данных, так как анализируются уже факты, а не произвольные записи журналов, и, как будет показано ниже, обычно повышает наглядность и качество анализа. Также данный подход снижает потоки данных от агентов и нагрузку на сервер консолидации и хранилище данных.

3.2. Агенты сбора данных. Формирование фактов активности

Можно сформулировать следующие требования к агентам сбора данных:

1) «незаметное» для пользователей системы чтение событий произвольных системных и текстовых журналов регистрации;

2) сбор дополнительной активности, не журналируемой стандартными системными журналами;

3) реализация механизма фильтрации событий по типу событий и по значениям аргументов событий;

4) формирование на основе прочитанных событий фактов активности;

5) настройка политики передачи данных по сети;

6) обеспечение защиты себя и собранных, но еще не отправленных данных от порчи/кражи/подмены.

Журналируемое событие можно представить как пару – тип события и набор атрибутов:

$$\mathrm{event}_j = \left\langle \begin{matrix} (\mathrm{attr}_1, \ldots, \mathrm{attr}_m),\ m - \text{количество базовых атрибутов} \\ (\mathrm{attr}_{I(j)} \mid \mathrm{attr}_{I(j)} \in \mathrm{dom}(\mathrm{attr}),\ I(j) \in N_0),\ \mathrm{event\_type}_j \end{matrix} \right\rangle,$$

$$\mathrm{dom}(\mathrm{event\_type}) = (1, \ldots, k),\ k - \text{количество типов},$$

где атрибут определяется именем, типом и значением, а базовыми атрибутами являются атрибуты, присущие всем событиям: время события, имя пользователя, имя компьютера. Журнал регистрации можно представить как набор записей разного типа.

Набор дополнительных данных об активности пользователя также представлен в виде событий и зависит от конкретной операционный системы. Например, для сбора дополнительных событий в ОС Windows в рамках работы были реализованы модули, журналирующие следующую активность пользователей: активность работы с клавиатурой и мышью в каждом из приложений; активность работы пользователей в сети – основные параметры всех TCP-соединений; работа пользователя с файловой системой, включая съемные носители и удаленные компьютеры. Важно отметить, что набор модулей не ограничен. Для этого было предложено модули сбора оформлять в

виде отдельного приложения (процесса), работающего от имени системного пользователя. Модуль собирает и журналирует дополнительную активность, которая затем считывается агентом сбора в стандартном режиме.

Фильтрацию событий предлагается организовывать в два этапа: на первом этапе считываются события только требуемых типов; на втором – каждый возможный атрибут события проходит через два списка фильтров – для пропуска и для блокирования. Фильтрация значений атрибутов организована как проверка на соответствие или несоответствие регулярным выражениям. Если хотя бы один атрибут события не соответствует ни одному из требуемых регулярных выражений из списка пропуска, то всё событие целиком игнорируется. То же происходит, когда хотя бы один атрибут соответствует хотя бы одному регулярному выражению из списка блокирования. Предложенный подход обладает достаточной гибкостью для удовлетворения поставленным требованиям, а именно: минимизации потоков информации, так как позволяет описать как события, которые требуется собирать, так и события, которые собирать не требуется.

Одной из основных задач агента сбора является формирование на основе считанных событий фактов активности пользователей. Факт активности может быть представлен как новое мета-событие, содержащее в себе агрегированную информацию из некоторой группы событий. Группа – это связанная подпоследовательность последовательности событий $EV = \{event_1, event_2, ..., event_n\}$. События внутри группы связаны значениями атрибутов. Отдельно выделяется случай, когда дополнительно типы соседних событий совпадают. Таким образом, группа – это строго определенная последовательность событий определенных типов, некоторые из которых могут быть представлены во множественном числе. Например, <открытие соединения> <передача> … <передача> <закрытие соединения>. Связь по значениям атрибутов вводится, чтобы контролировать принадлежность всех событий одному факту, в данном примере, чтобы все передачи были в рамках открытого соединения. Такую группу можно описать, введя набор функций следования $func\_follow(\{event\}, event_{cur}) \rightarrow \{true, false\}$, то есть функций, показывающих может ли текущее событие следовать за множеством уже существующих в группе, и если да, то текущее событие может также быть добавлено к группе. Описание предложенных методов формирования групп событий, задания набора функций следования и методов построения фактов активности по группам довольно громоздки и не приводятся в данной статье.

Передача построенных фактов активности агентом данных может осуществляться по одной из следующих стратегий:

– *фиксированными объемами данных.* Агент накапливает определенный объем информации или фиксированное количество записей журналов и затем передает их на сервер консолидации;

– *через равные промежутки времени.* Агент через равные промежутки времени передает все имеющиеся у него в локальном хранилище данные независимо от их объема;

– *в режиме реального времени.* Агент немедленно передает данные о каждой вновь прочитанной записи в журнале регистрации.

3.3. Сервер консолидации

Получая данные от агента, сервер консолидации преобразует записи во внутренний формат и помещает их в хранилище. В рамках работы предлагается унифицированное представление анализируемых событий (фактов активности) в системе, позволяющее единообразно представлять факты активности для унифицированного хранения и анализа, не зависящего от типа исходного журнала.

Предложенное хранилище данных позволяет эффективно представлять собранные факты активности с использованием словарей реальных значений аргументов событий. Хранилище построено на основе файловой системы, что обеспечивает большую производительность, так как важнейшим параметром работы хранилища и системы консолидации в целом является именно производительность. Нагрузка на хранилище складывается из нагрузки на добавление вновь получаемых журналируемых данных и нагрузки на извлечение данных для анализа. Характер этих действий различный. Извлечение данных обычно подразумевает быстрое получение всех собранных записей за некоторый временной период, и эта проблема решается размещением записей в отсортированном по дате порядке. Сложнее дело обстоит с добавлением вновь полученных данных, поскольку один сервер консолидации должен объединять в себе данные от большого количества агентов.

Проблемы производительности удалось решить, разделив файлы хранилища на три информационные части:

– *основные параметры*, присутствующие практически во всех записях всех логов: тип события, время генерации и т.п.;

– *вспомогательные параметры*, описывающие дополнительную информацию о событии;

– *справочники* для хранения реальных значений как основных, так и вспомогательных параметров. Для доступа к значениям справочников используется механизм хэширования.

Запись о событии из лога разделяется между двумя файлами и файлами справочников: файл для хранения идентификаторов основных параметров, файл для хранения идентификаторов вспомогательных параметров и файлы справочников. Файлы справочников позволяют по идентификаторам параметров получать их реальные значения.

Сохранение данных в файлах основных и вспомогательных параметров и файлах справочников позволяет в некоторых случаях повысить производительность при обращении к хранилищу. Например, если для анализа собранных данных применять технологию OLAP [8], то для заполнения некоторых фактов в витрине данных (Data Mart) иногда можно обойтись без вспомогательных параметров. Даже если и приходится обращаться к вспомогательным параметрам, далеко не всегда существует необходимость получать их фактические значения, иногда достаточно просто их идентификатора.

Хранилище организованно в виде дерева, в корне которого находятся каталоги с именем домена, внутри каждого каталога с именем домена находятся каталоги с именем компьютеров этого домена. Внутри каталогов с именем компьютера лежат файлы с собранными записями лог-файлов, разделенные по датам, например, дням. Файлы-справочники могут храниться в произвольном месте. Иерархия файлов по каталогам (домен – компьютер – дата) не очень принципиальна, однако позволят более легко получать, например, все события из домена или с определенных компьютеров, поскольку обычно для анализа данных требуется получить из хранилища необходимый временной срез и дополнительно применить к нему фильтр (обычно по компьютерам, пользователям, типам логов). Сервер консолидации может работать только с зарегистрированными агентами, что обеспечивает защиту от «подмены» агента и загрузки на сервер консолидации некорректных данных.

3.4. Анализ данных

Предложенное решение не накладывает ограничений на возможные методы анализа и, как было показано выше, за счет формирования фактов активности позволяет проводить унифицированный анализ данных из различных источников. В рамках работы показана применимость двух типов анализа: статистического анализа на основе технологии OLAP и интеллектуального анализа данных на основе существующих алгоритмов Data Mining с целью раннего обнаружения вторжений и поиска аномалий в работе.

Статистический анализ. Технология комплексного многомерного анализа данных – OLAP допускает возможность осуществления любого логического и статистического анализа, а также многомерное концептуальное представление данных пользователю. Основными понятиями технологии OLAP являются меры, измерения и иерархии. Идея применения OLAP-технологии для проведения статистического анализа заключается в построении OLAP-куба по данным фактов активности. Для определения OLAP-куба необходимо задать отображение атрибутов фактов активности в измерения, меры и иерархии, так как OLAP-куб полностью ими определяется. Числовые атрибуты фактов активности используются в качестве мер, атрибуты временного и строкового типа используются в качестве измерений и иерархий. При таком распределении OLAP-куб может быть построен только по подмножеству однотипных фактов, то есть фактов, содержащих одинаковый набор атрибутов.

Сформированные меры, измерения и иерархии формируют OLAP-куб (рис. 2). Пользователю в виде отчетов визуализируются срезы куба.

Интеллектуальный анализ (Data Mining). Идея применения интеллектуального анализа основана на том, что на основе собранной активности пользователей системы может быть построена математическая модель, которая позволит обнаруживать аномалии в поведении пользователей. По свидетельству многих специалистов по компьютерной безопасности, такой подход является особенно перспективным в задачах обнаружения именно внутренних вторжений, поскольку он позволяет создавать так называемые системы «раннего обнаружения» (early warning). Опыт

эксплуатации систем обнаружения внутренних вторжений показывает, что в большинстве случаев непосредственно внутреннему вторжению предшествует некоторое аномальное (хотя возможно и разрешенное) поведениепользователя, то есть пользователь еще до атаки или кражи информации начинает совершать действия, не характерные для его предыдущей активности или активности пользователей той же группы.

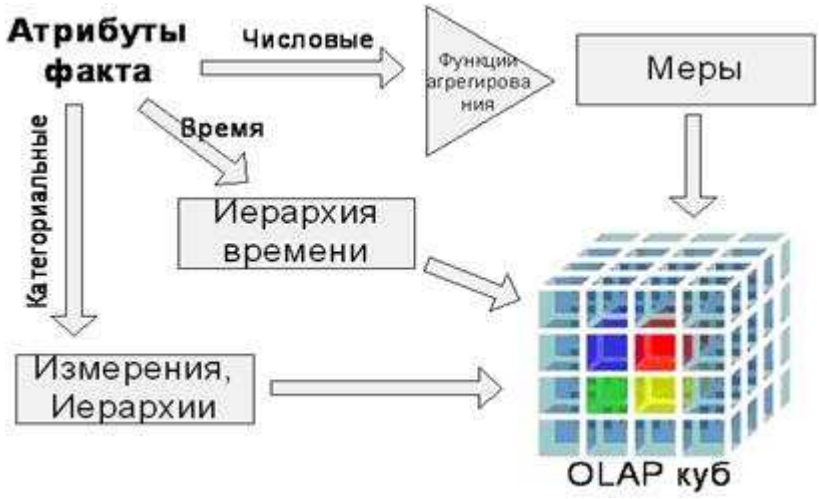

Рис. 2. Построение OLAP куба по фактам активности

Из хранилища выбирается тренировочный набор записей, отражающих поведение пользователей (обычно записи за временной период, в который, по мнению эксперта, не было вторжений). На выбранном тренировочном наборе «обучается» алгоритм выявления аномалий. Затем алгоритм применяется к вновь поступающим данным, классифицируя их как нормальные или как аномальные (рис. 3).

Предложенный в работе алгоритм моделирования поведения пользователей основан на использовании ассоциативных правил. При этом на этапе обучения для собранных данных строятся ассоциативные правила, а на этапе применения вновь собранные данные классифицируются по степени аномальности относительно уже построенных правил.

Для построения ассоциативных правил из собранных данных выделяются требуемые факты. Основная идея подхода заключатся в использовании алгоритмов поиска ассоциативных правил [6] для выявления корреляций между элементами в транзакции, что в нашем случае соответствует корреляциям между значениями атрибутов факта.

Пусть любой факт x описывается набором из n атрибутов (характеристик), результатом работы алгоритма является система из m ассоциативных правил $\{R_s(x)\}_{s=1}^{m}$ вида

$$R_s(x) = "A_{i1}(x), \ldots, A_{il}(x) \Rightarrow B_{j1}(x), \ldots, B_{jk}(x)", \quad [c, s],$$

где $A_i(x), B_i(x)$ – предикаты, задающие условия на значения l-го атрибута $x_l \in X_l$; s – поддержка (частота встречаемости) правила, определенная как число фактов, для которых выполнены все предикаты правила как в левой, так и в правой части

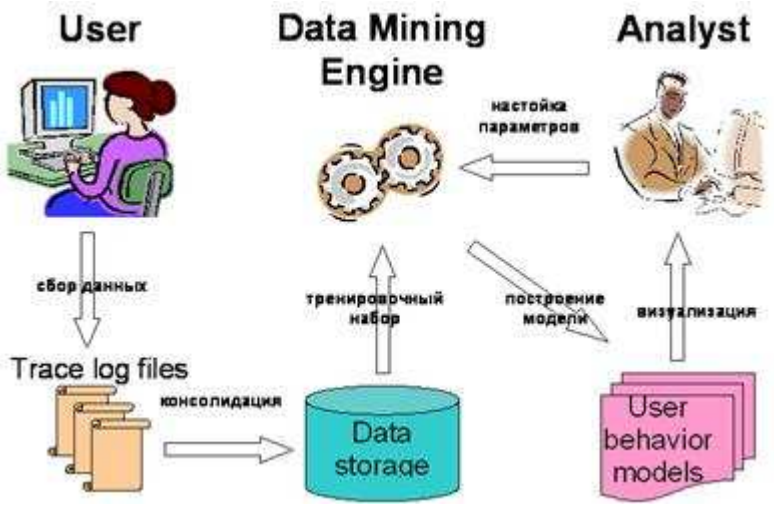

Рис. 3. Схема применения методов Data Mining
для построения моделей поведения пользователей

$$s = \text{support}\,(R(x)) = \left|\left\{x \in X \,|\, A_{i1}(x) \wedge \ldots \wedge A_{il}(x) \wedge B_{j1}(x) \wedge \ldots \wedge B_{jk}(x)\right\}\right|,$$

c – достоверность правила, определенная как число фактов, в которых из выполнения всех предикатов левой части правила следует выполнение всех предикатов правой:

$$c = \text{confidence}\,(R(x)) = \frac{\left|\left\{x \in X \,|\, A_{i1}(x) \wedge \ldots \wedge A_{il}(x) \wedge B_{j1}(x) \wedge \ldots \wedge B_{jk}(x)\right\}\right|}{\left|\left\{x \in X \,|\, A_{i1}(x) \wedge \ldots \wedge A_{il}(x)\right\}\right|}.$$

Ассоциативное правило $R(x)$ может быть проинтерпретировано так: «если атрибуты факта x удовлетворяют предикатам $A_{i1},\ldots,A_{il}$, то с вероятностью c данный факт будет удовлетворять предикатам $B_{j1},\ldots,B_{jk}$». Помимо простой и эффективной интерпретации самих правил, сильной стороной ассоциативного подхода является простое и эффективное определение «частого эпизода», то есть часто встречаемой устойчивой комбинации атрибутов. Любой «частый эпизод» также напрямую определяется ассоциативным правилом $R(x)$, где частота определяется как значение поддержки (support) правила.

Для поиска аномалий набор построенных правил применяется к новым данным о текущей активности пользователей, что позволяет оценить насколько новая активность отличается от предыдущей. Идея применения модели на основе ассоциативных правил для поиска аномалий базируется на том, что ассоциативные правила $\{R_i(x)\}_{i=1}^{m}$ можно использовать для прогнозирования значений одних атрибутов по другим. Для этого на основе системы правил $\{R_i(x)\}_{i=1}^{m}$ строится функция $P(x_i | x_1,\ldots,x_{i-1},\ x_{i+1},\ldots,x_n)$, которая вычисляет распределение условной вероятности значений i-го атрибута, в зависимости от остальных атрибутов.

В простейшем случае такая функция может быть определена следующим образом. Пусть область определения i-го атрибута $\text{dom}\,(x_i) = \{a_1,\ldots,a_s\}$ содержит s различных значений a_s, если i-й атрибут дискретный, или s числовых интервалов, если i-й атрибут непрерывный. Тогда вероятность того, что $P(x_i = a_s \,|x_1,\ldots,x_{i-1},\ x_{i+1},\ldots,x_n)$ определяется как максимальная достоверность правила

$$P(x_i = a \,|x_1,\ldots,x_{i-1},\ x_{i+1},\ldots,x_n) = \frac{\sum_j \text{confidence}\left(R_j(x_1,\ldots,x_{i-1},a,x_{i+1},\ldots,x_n)\right)}{\sum_{s,j} \text{confidence}\left(R_j(x_1,\ldots,x_{i-1},a_s,x_{i+1},\ldots,x_n)\right)}.$$

В этом случае уровень достоверности (нормальности) значения *i*-го атрибута как отношение условной вероятности реально наблюдаемого значения a_s к вероятности наиболее ожидаемого значения

$$\mathrm{Score}(x_i | x_i = a_s) = \frac{P(x_i = a_s | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)}{\max_l P(x_i = a_l | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)}.$$

Значение достоверности будет меняться от 0 до 1, чем меньше это значение, тем аномальнее значение атрибута. Достоверность всего события *x* можно определить, например, как произведение достоверностей его атрибутов.

Такой подход дает возможность не только обнаружить аномальные факты (события), но найти причину аномальности, то есть те атрибуты, которые являются не нормальными с точки зрения предыдущей активности пользователей.

Заключение

В работе предложена технология построения специализированных аналитических систем мониторинга корпоративных пользователей. Предложен уникальный подход обработки журналируемых событий, основанный на формировании фактов, описывающих активность пользователей. Предложенный подход позволяет существенно снизить потоки данных для анализа, что, в свою очередь, позволяет снизить нагрузку на сеть и анализ данных, без существенного снижения, а иногда и с улучшением, качества и наглядности анализа. Технология позволяет унифицировать сбор, консолидацию данных из различных источников, производить унифицированный статистический и интеллектуальный анализы с целью определения параметров работы пользователей и поиска аномалий в работе. Централизованный анализ выделенных фактов с помощью алгоритмов интеллектуального анализа данных позволяет обнаруживать новые и модифицированные известные типы вторжений, а также находить аномалии в работе пользователей. Применение интеллектуальных методов позволяет обнаруживать вторжения на раннем этапе, то есть в момент подготовительной активности. Расширяемость достигается путем создания новых агентов для требуемых платформ. Предложенное решение апробировано в виде мультиагентной системы консолидации и анализа данных журналов регистрации в ряде государственных и коммерческих организаций.

Работа поддерживается грантами РФФИ 05-01-00744, 06-01-00691 и 06-07-08035; грантом Президента РФ МК-4264.2007.9.

Список литературы

1. Computer Crime and Security Survey [Электронный ресурс] : обзор компьютерных нарушений и методов борьбы с ними / Computer Security Institute, 7 апреля 2002. – Режим доступа: http://www.gocsi.com/press/ 20020407.jhtml, свободный.
2. Cristina Abadyz, Jed Taylory, Cigdem Senguly, William Yurcik. Log Correlation for Intrusion Detection: A Proof of Concept. Department of Computer Science, University of Illinois at Urbana-Champaign, 2003. – P. 3–6.
3. Kathleen A. Jackson. Intrusion detection system (ids) product survey. Distributed Knowledge Systems Team Computer Research and Applications Group Computing, Information, and Communications Division Los Alamos National Laboratory Los Alamos, New Mexico USA, 1999. – P. 6–22.
4. National Survey on Managing the Insider Threat [Электронный ресурс] : обзор средств борьбы с внутренними вторжениями / Журн.ComputerWorld, 12 сентября 2006. – Режим доступа : http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9003211, свободный.
5. Theuns Verwoerd, Ray Hunt. Intrusion Detection Techniques and Approaches. Department of Computer Science University of Canterbury, New Zealand, 2002. – P. 2–14.
6. SQL Server 2005 Books Online. Microsoft Association Algorithm [Электронный ресурс] : SQL Server 2005 Books Online. Алгоритм построения ассоциативных правил от Microsoft. – Режим доступа : http://msdn2.microsoft.com/en-us/library/ms174916.aspx, свободный.
7. Лукацкий, А. Обнаружение атак / А. Лукацкий. – СПб. : БХВ–Петербург, 2001. – 624 с.

8. Алексей Федоров, Наталия Елманова. Введение в OLAP : часть 1 : Основы OLAP [Электронный ресурс]. – Режим доступа : http://olap.ru/basic/OLAP_intro1.asp, свободный.

Monitoring of Corporate Users' Work

S.V. Troshin

Moscow State University named after M.V. Lomonosov, Moscow

Key word and phrases: monitoring of work; earlier detection of attacks; statistic analysis.

Abstract: The paper proposes the technology (architecture, model, methods and algorithms) of designing monitoring systems of corporate users' performance on the basis of collection and analysis of journalized information. The main objectives of the analysis are the tasks of determining statistic parameters and characteristics of users'' work and construction of the generalized work profile as well as the tasks of earlier detection of in-process attacks and the analysis of the passed attacks traces. Methods of computer-aided instruction are used as the basis for early detection of attacks and the analysis of the passed attacks traces. In order to search anomalies of users it is proposed to design the model of normal behavior and then classify the user's current activity either as normal or abnormal on the basis of the designed model. The paper describes main principles of data collection and processing as well as some algorithms and techniques within the proposed technology for system implementation. The monitoring system designed for OS Windows has been tested in a number of state and commercial organizations. The results of testing proved the system applicability to solve the prior tasks which are to be solved by Management and IT Department.