УДК 004.41

МОНИТОРИНГ И АНАЛИЗ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ КОМПЬЮТЕРНЫХ СИСТЕМ[1]

И.В. Машечкин, М.И. Петровский, С.В. Трошин

Московский государственный университет им. Ломоносова,
119899, Россия, Москва, Воробъевы горы, строение 2,
e-mail:mash@cs.msu.su, michael@cs.msu.su, troshin@mlab.cs.msu.su

Рассматриваются вопросы построения эффективных программных систем защиты от внутренних вторжений, основанных на несигнатурных методах и обладающих свойствами автономности, адаптируемости и самообучаемости. Отдельно рассмотрены проблемы консолидации исходных данных из журналов регистрации и протоколов ОС, методы промежуточного представления, передачи данных и хранения собранных данных. Предложена архитектура системы консолидации и рабочего места аналитика безопасности. Предложены методы применения технологии OLAP для анализа собранных данных об активности пользователей, а также алгоритмы интеллектуального анализа данных (Data Mining) для построения модели поведения пользователя на основе ассоциативных правил. Построенная модель поведения может быть использована для визуального представления аналитику безопасности в виде сети зависимостей, а также для автоматического поиска аномалий в поведении пользователей и оценки степени потенциальной угрозы, исходящего от каждого пользователя. Реализована экспериментальная пилотная версия такой системы, которая была верифицирована по методике DARPA Intrusion Detection Evaluation Program, с использованием эталонных наборов данных. Результаты экспериментальной верификации приведены в работе.

Рассматриваются вопросы построения эффективных программных систем защиты от внутренних вторжений, основанных на несигнатурных методах и обладающих свойствами автономности, адаптируемости и самообучаемости. Отдельно рассмотрены проблемы консолидации исходных данных из журналов регистрации и протоколов ОС, методы промежуточного представления, передачи данных и хранения собранных данных. Предложена архитектура системы консолидации и рабочего места аналитика безопасности. Предложены методы применения технологии OLAP для анализа собранных данных об активности пользователей, а также алгоритмы интеллектуального анализа данных (Data Mining) для построения модели поведения пользователя на основе ассоциативных правил. Построенная модель поведения может быть использована для визуального представления аналитику безопасности в виде сети зависимостей, а также для автоматического поиска аномалий в поведении пользователей и оценки степени потенциальной угрозы, исходящего от каждого пользователя. Реализована экспериментальная пилотная версия такой системы, которая была верифицирована по методике DARPA Intrusion Detection Evaluation Program, с использованием эталонных наборов данных. Результаты экспериментальной верификации приведены в работе.

1. Актуальность

Развитие и повсеместное внедрение информационных технологий, увеличение объемов и ценности хранимой информации приводит к необходимости ее защиты. Основную угрозу безопасности представляют вторжения в компьютерные системы. Под вторжением (или атакой) в компьютерную систему понимается любая деятельность человека 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.ю

Традиционно в системах обнаружения вторжений используется сигнатурный метод [1, 2], 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.

На сегодня актуально создание программных технологий построения эффективных систем защиты от внутренних вторжений, основанных на несигнатурных методах [3, 4] и обладающих свойствами автономности, адаптируемости и самообучаемости.

Подобного рода системы защиты должны решать такие 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 (со стороны инсайдеров).

[1] Работа поддерживается грантом Президента РФ № МК-2111.2005.9 и грантами РФФИ № 06-07-08035-офи, № 05-0100744

2. Архитектура

Система мониторинга и анализа поведения пользователей обеспечивает сбор данных, формирование статистических отчетов и интеллектуальный анализ (Data Mining) собранных данных, а именно: построение ассоциаций и моделей поведения, поиск аномалий работы как в рамках отдельной вычислительной системы, так и в рамках сети в целом. Система является мультиагентной. Источником данных являются журналы регистрации ОС и ПО.

Система состоит из сервера консолидации, агентов сбора, рабочего места аналитика (рис. 1). Исходные данные читаются агентом из журналов вычислительных систем и консолидируются на сервере. По собранным данным вычисляются статистические значения, производится анализ данных. Статистические отчеты, найденные ассоциации, построенные модели и выявленные аномалии передаются аналитику в виде интерактивных динамических отчетов.

Рабочее место аналитика — программный модуль, обеспечивающий работу пользователя с системой. Предоставляются следующие возможности: управление консолидацией (установка и настройка агентов), управление анализом данных (фильтрация данных, обучение алгоритмов анализа и т.п.), просмотр отчетов по заданным параметрам.

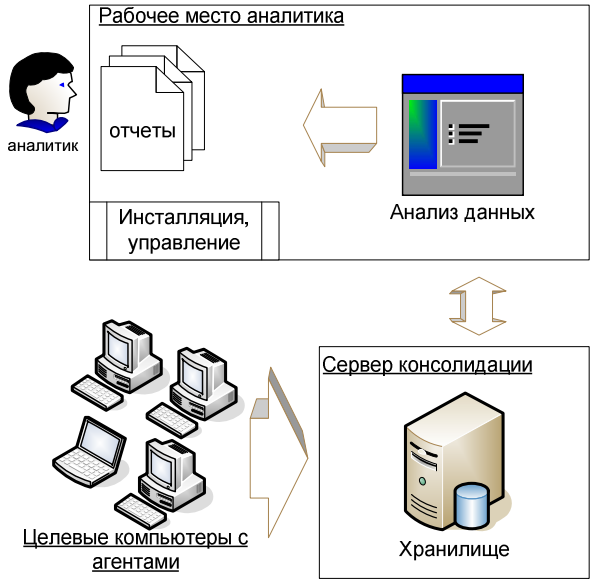

Рис. 1. Система мониторинга и анализа поведения пользователей

Задачу объединения всех записей журналов в хранилище с предоставлением унифицированного интерфейса доступа, не зависящего от структуры консолидируемых данных, решает подсистема консолидации записей журналов регистрации (логов) — система сбора и предобработки информации из журналов ОС и ПО. Архитектура подсистемы консолидации логов представлена на рис. 2.

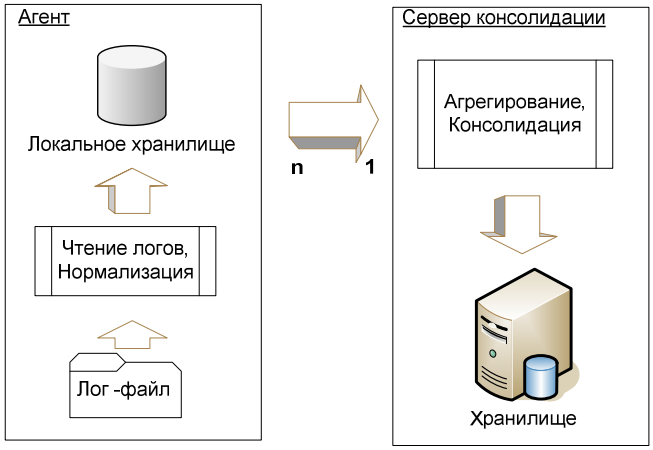

Рис. 2. Системы консолидации логов

Подсистема консолидации состоит из двух частей: серверная часть (сервер консолидации) и агенты, работающие на компьютерах, с которых собираются записи лог-файлов. Агентов может быть произвольное количество. Агенты могут работать на разных платформах и собирать информацию из разных журналов регистрации. Система позволяет добавлять новые агенты или модифицировать существующие «на лету» — без перезапуска сервера консолидации. Агент содержит локальное хранилище для промежуточной буферизации собранных данных перед отправкой, если это подразумевается настройкой передачи данных.

2.1. Агенты

Агент обеспечивает сбор информации о событиях из различных журналов ОС и ПО, установленного на целевом компьютере. Преобразует собранные записи в универсальный формат, разработанный на основе XML [5, 6, 7] и согласно расписанию передает данные на сервер консолидации. Передача собранных агентом данных может осуществляться по одной из следующих стратегий.

1. **Фиксированными объемами данных.** Агент накапливает определенный объем информации или фиксированное количество записей журналов и затем передает их на сервер консолидации.
2. **Через равные промежутки времени.** Агент через равные промежутки времени передает все имеющиеся у него в локальном хранилище данные независимо от их объема.
3. **По расписанию.** Агент передает данные только в указанное время.
4. **В режиме реального времени.** Агент немедленно передает данные о каждой вновь прочитанной записи в журнале регистрации. Данная стратегия наиболее требовательная к ресурсам.

Для обеспечения распределения возникающих при работе агента накладных расходов на чтение и первичную обработку записей журналов регистрации может использоваться механизм разделения нагрузки между целевым компьютером и дополнительным, выделенным для преобразования данных в универсальный формат. Механизм основан на возможности удаленного чтения некоторых журналов регистрации.

2.2. Сервер консолидации

Получая данные от агента, сервер консолидации преобразует записи во внутренний формат и помещает их в хранилище. Важным параметром работы хранилища и системы консолидации в целом является производительность. Нагрузка на хранилище складывается из нагрузки на добавление вновь получаемых данных лог-файлов 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 и еще от большего количества лог-файлов.

Среднее количество записей, помещаемых в журнал регистрации ОС Windows за единицу времени можно оценить по тесту DARPA99 [8]. Примерно 400000 событий в день, что эквивалентно приблизительно 10 событиям в секунду. Например, для корпоративной сети из 100 компьютеров сервер консолидации должен обрабатывать и добавлять в хранилище порядка 1000 событий в секунду. Каждая запись журнала безопасности ОС Windows состоит приблизительно из 20 параметров (содержательных признаков). Что приводит к 20000 параметрам в секунду. 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 (рис. 3): 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.

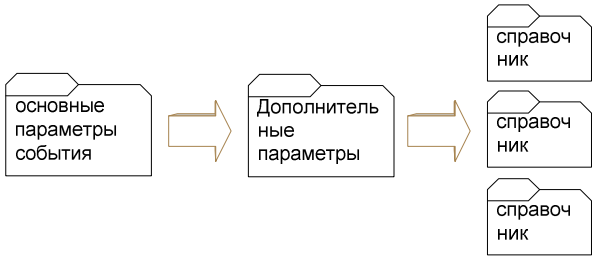

Рис. 3. Иерархия файлов

Сохранение данных в файлах основных и вспомогательных параметров, 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.

Структура специализированного хранилища представлена на рис. 4. 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.

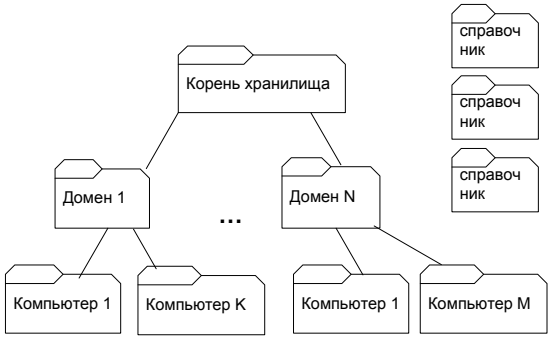

Рис. 4. Размещение объектов в файловой системе

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.

В случае недостаточной производительности сервера консолидации имеется возможность использования промежуточных серверов консолидации.

3. Анализ данных

Сервер консолидации и хранилище не накладывают никаких ограничений на подмножество методов и механизмов анализа данных, которые могут быть применены к данным журналов регистрации. Имеется возможность подключать различные аналитические средства: построение статистики, выявление ассоциативных правил, выявление аномалий и исключений и т.д. В рамках построенной системы реализуются две технологии анализа данных – OLAP и Data Mining.

3.1. Статистика на основе OLAP

Технологии комплексного многомерного анализа данных – OLAP (рис. 5), используемая для анализа собранных данных, допускает возможность осуществления любого логического и статистического анализа, а так же многомерное концептуальное представление данных пользователю.

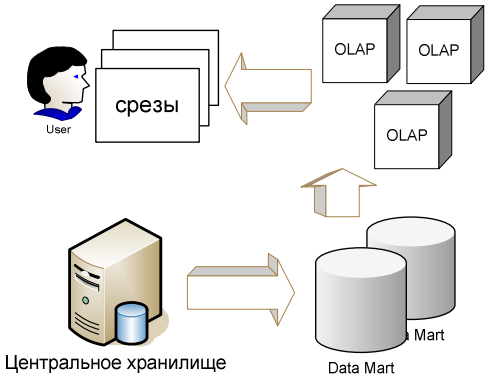

Рис. 5. Применение технологии OLAP

Для иллюстрации возможностей статистического анализа журнализированной информации с использованием технологии OLAP рассмотрим модельный пример: построение OLAP кубов по данным журнала регистрации ОС Windows.

Из логов ОС Windows выделены факты:

входа в систему;

запуска процессов;

обращения к файлам, папкам и реестру.

По фактам построены OLAP кубы – многомерные наборы значений, отражающие статистику работы пользователей.

Факты легко определить, зная коды событий журнала регистрации и форматы представления записей нужных типов событий. Для построения OLAP куба [9] следует заполнить Data Mart – реляционная база данных определенной (типа «звезда» или «снежинка») структуры, которая состоит из таблицы фактов и таблиц -

справочников. Фактами в строящейся Data Mart являются выявленные факты активности в системе. А доступный набор параметров у событий [8] определяет максимальный набор справочников (OLAP измерения).

Так, например, из записей лога безопасности для фактов запуска процессов (куба) выделены измерения: имя запущенного процесса, пользователь, компьютер, родительский процесс и т.д. Введена иерархия компьютеров: домен – компьютер; иерархия пользователей, иерархия времени. Иерархия в технологии OLAP позволяет получать интересующие нас значения – OLAP меры как для отдельного объекта или явления, так и для их группы, которую допускает иерархия.

Для всех типов событий определены следующие параметры – OLAP измерения и иерархии:

компьютеры (Иерархия: Домен > Компьютер);

время (Иерархия: Год > Месяц > Неделя > День > Час > …);

процессы (Иерархия: Компьютер > Имя файла);

родительские процессы (Иерархия: Компьютер > Имя файла);

ресурсы (Иерархия: Компьютер > Путь > Файл);

типы ресурсов (без иерархии: расширения файлов и т.д.);

типы входов в систему (без иерархии);

операции (без иерархии, например, открытие/закрытие дескриптора, стоп/старт процессов, изменение прав и т.д.);

пользователи (Иерархия: Домен > Пользователь);

статус завершения (без иерархии).

OLAP мерой являются те количественные показатели, которые требуется получить для определенных фактов, выбрав нужные измерения (параметры записи лога) и уровень иерархии. В качестве мер выбрано – количество запусков процессов, продолжительность работы процесса, средняя продолжительность работы процессов.

Обобщение на все рассматриваемые факты приводит к следующим OLAP мерам:

число операций (счетчик);

время работы (сумма) – между «парами операций» типа старт/стоп, открытие-закрытие и т.д.;

число операций в единицу времени (вычислимое поле).

Например, срез куба для фактов запуска процессов с мерой количества входов, измерениями – пользователи, типы входов в систему и процессы входа, будет выглядеть как на рис. 6.

	Login Types ▾	Processes ▾	
	⊞ Console logon	⊟ Network logon	
		KSecDD	Total
Users ▾	Login Facts Count	Login Facts Count	Login Facts Count
Administrator	17	7	7
alie	2		
ANONYMOUS LOGON		5	5
huws	1		
IUSR_HUME	7		
orionc	3		
triav	3		
Grand Total	33	12	12

Рис. 6. Пример среза для куба входов в систему

Технология OLAP может быть применена практически к любым типам журналов регистрации ОС и ПО. Например, из лога антивирусной программы можно выделить факты и параметры. Построить и заполнить по ним Data Mart, построить кубы.

Но технология OLAP позволяет строить только статистические отчеты и никак не решает задачи выявления изменений и их характера в поведении системы, приложений и пользователей.

3.2. Data Mining.

Интеллектуальный анализ данных (Data Mining) [10], примененный к собранным записям лог-файлов, позволяет находить скрытые, содержательные и потенциально полезные закономерности, строить ассоциативные правила, находить исключения и аномалии.

В системе используются следующие «Модели поведения».

Поиск корреляции в атрибутах фактов, которые описывают типичное поведение. Для этого применяется алгоритм на основе ассоциативных правил [11].

Анализ последовательности действий, выявление часто выполнимых сценариев с целью предсказания и поиска аномалий (на настоящий момент еще не реализовано, планируется использовать методы кластеризации).

Поиск статистических аномалий (в OLAP срезах) – отклонение значений мер от их усредненных показателей (мат. ожидания, медианы и т.д.).

3.2.1. Ассоциативные правила

Для построения ассоциативных правил, из собранных данных так же как при заполнении Data Mart для OLAP выделяются факты за требуемый временной период. Основная идея подхода заключатся в использовании алгоритмов поиска ассоциативных правил 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), где область определения атрибутов задана как $x = (x_1,...,x_n) \in X = dom(x_1) \times ... \times dom(x_n)$ Результатом работы алгоритма является система из m ассоциативных правил $\{R_s(x)\}_{s=1}^{m}$ вида:

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

где $A_l(x), B_l(x)$ – предикаты, задающие условия на значения *l-го* атрибута $x_l \in X_l$, в частности, это может быть диапазон значений для числовых атрибутов (например, “Duarion = 77–1384”), и конкретные значения для дискретных атрибутов (например, “Process = cmd.exe”);

s – поддержка (частота встречаемости) правила, определенная как (число фактов, для которых выполнены все предикаты правила, как в левой так и в правой части):

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

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

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

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

Построенные ассоциации можно визуализировать либо в виде правил, либо в виде сети зависимостей. Каждое правило представимо в виде «если …, то…» При этом для каждого правила определяется его вероятность. Имеется возможность фильтровать правила по количеству «предпосылок» и по вероятности, а также задавать фильтры по вероятности. Сеть зависимости позволяет наглядно отобразить вероятные следствия и предшествия пары «атрибут = значение».

3.2.2. Поиск аномалий

Основная идея применения методов обнаружения аномалий состоит в предположении о том, что активность пользователя системы может быть отслежена и построена ее математическая модель, которая позволит обнаруживать нарушения политики безопасности и 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) поведение пользователя, т.е. пользователь еще до атаки или кражи информации начинает совершать действия не характерные для его предыдущей активности или активности пользователей той же группы (рис. 7).

Из хранилища выбирается тренировочный набор записей журналов регистрации (обычно записи за временной период, в который по мнению аналитика не было вторжений). На выбранном тренировочном наборе «обучается» алгоритм выявления аномалий. Затем алгоритм применяется к вновь поступающим записям, классифицируя их как нормальные или как аномальные. При необходимости аналитик может «дообучать» алгоритм.

Для поиска аномалий можно использовать построенные ассоциативные правила. Для этого сначала алгоритм поиска ассоциативных правил применяется к данным о предыдущей активности пользователей, строится модель на основе ассоциативных правил $\{R_i(x)\}_{i=1}^{m}$, описывающая данную активность. Эта модель применяется к новым данным о текущей активности пользователей и позволяет оценить насколько новая активность отличается от предыдущей, оценить аномальность отдельных событий и их атрибутов. 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 $P(x_i \mid x_1,...,x_{i-1},x_{i+1},...,,x_n)$, которая вычисляет распределение условной вероятности значений *i-го* атрибута, в зависимости от остальных атрибутов.

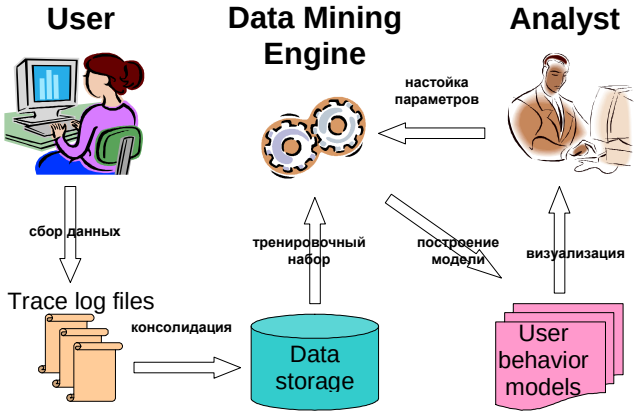

Рис. 7. Схема применения методов Data Mining для построения моделей поведения пользователей

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

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

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

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

Очевидно, что если значение *i-го* атрибута совпадает с наиболее ожидаемым, т.е. тем, которое прогнозируется на основе найденной ранее ассоциативной модели, то уровень достоверности такого атрибута равен 1, т.е. абсолютно «ожидаемое» значение. Если же такое значение ранее вообще не встречалось, то его условная вероятность будет равна нулю и уровень достоверности атрибута также будет равен нулю, что соответствует абсолютно «аномальному» значению. В остальных случаях значение достоверности будет меняться от 0 до 1, чем меньше это значение, тем аномальнее значение атрибута. Достоверность всего события *x* в этом случае можно определить как произведение достоверностей его атрибутов:

$$Score(x) = \prod_i Score(x_i).$$

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

4. Тестирование системы

В 1998 году MIT Lincoln Lab совместно с Defense Advanced Research Projects Agency (DARPA ITO) выпустила методику для оценки алгоритмов обнаружения вторжений под названием DARPA 1998 Intrusion Detection Evaluation Program. Методика содержит набор данных, симулирующих вторжения в локальную сеть. Для получения этих данных была смоделирована типичная локальная сеть, используемая в военно-воздушных силах США, кроме того, данная сеть была подвергнута множеству различных атак.

Система мониторинга и анализа поведения пользователей была протестирована на эталонном тестовом наборе данных DARPA 1999. Тестовый набор состоит из логов работы операционной системы ОС Windows NT и описания воздействия на систему, результатом которых стали данные логи. Что позволяет оценить вероятность обнаружения атак и вероятность ложного срабатывания систем обнаружения вторжений на данном тесте. При этом определяется и список атак, которые выявляются системой и атаки, которые не выявляются.

4.1. Обнаружение аномалий

Используемый для тестирования набор включает запись журнала регистрации (security log) за 5 недель. При этом были недели без атак, что может использоваться для обучения интеллектуальных систем. В другие же были совершены тестовые атаки. Набор DARPA 99 не содержит записей по работе с файлами и реестром, поэтому при обнаружении вторжений (аномалий) анализировались только факты запуска процессов.

Среди тестовых NT атак в используемом тесте были обнаружены:

– crashIIS – одиночный специально сформированный некорректный HTTP запрос, обработка которого приводит к аварийному завершению работы Web сервера;

– yaga – атака, направленная на создание нового пользователя в группе администраторов, путем взлома реестра;

– CaseSen – атака, направленная на получение администраторских прав пользователем. Атака использует чувствительность к регистру каталога объектов NT;

– netbus – атака заключается в установке на машине жертвы NetBus сервера. В последствии атакующий может подключаться к серверу удаленно и выполнять практически любые действия от имени работающего в данный момент пользователя;

– netcat – удаленная атака. Атакующий устанавливает программу netcat на 53 порт, которая затем используется как «черный ход», позволяющий заходить на компьютер-жертву, не вводя пароль.

Для решения задачи обнаружения аномалий в результате использования алгоритма поиска ассоциаций были построены ассоциативные правила для периода пользовательской активности, в которой не было атак, после чего построенная ассоциативная модель была применена для обнаружения аномалий в периоды пользовательской активности и вторжений.

Для оценки качества работы алгоритма требовалось определить два показателя: *коэффициента обнаружения (detection rate)* и *коэффициента ложных положительных ошибок (false positive rate)*. Обычно сравнение качества работы IDS производится с помощью ROC-кривых (Receiver Operating Characteristic curves). По оси абсцисс графика кривой откладываются значения коэффициента ложных положительных ошибок, по оси ординат – значения коэффициента обнаружения. Каждому результату работы алгоритма на таком графике будет соответствовать одна точка. При этом в зависимости от параметров алгоритма получается множество точек.

При тестировании системы на тестовом наборе DARPA были получены следующие результаты – рис. 8.

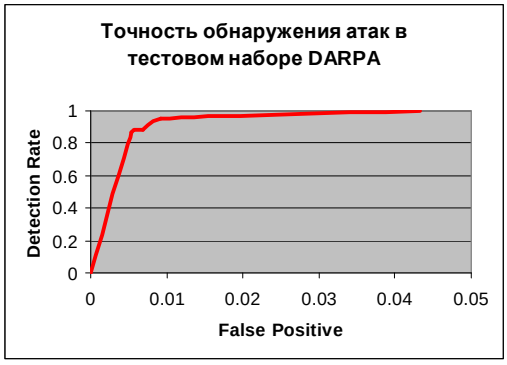

Рис. 8. Точность обнаружения атак методом выявления аномалий в эталонном наборе данных DARPA 1999

Оптимальная настройка соответствует detection rate 87 %, false positive 0.5 %, экстремальная (все атаки блокированы) – detection rate 100 %, false positive 4 %.

Обнаруженные/пропущенные вторжения и ложные срабатывания при оптимальной настройке представлены в таблице.

Таблица. Обнаруженные и пропущенные вторжения

Количество фактов	Атака/Нормальное поведение	Обнаружение/пропуск
12	Casesen	detected
7	Dos	detected
6	Netbus	detected
9	Netcat	detected
4	Sechole	detected
1	Secret	detected
14	Yaga	detected
3	Casesen	missed
5	Dos	missed
2046	Normal	allowed
11	Normal	blocked

Из таблици видно сколько фактов атак (аномалий) и каких было обнаружено (detected) или пропущено (missed) за время тестирования, сколько было нормальных фактов (allowed) и сколько нормальных фактов было принято за аномалии (blocked) – false positive. Под фактом подразумевался запуск процесса или несколько запусков, если они в совокупности образуют вторжение.

1. *Лукацкий А.* Обнаружение атак. – СПб.: БХВ-Петербург, 2001. – С. 50 – 200.
2. *Theuns* Verwoerd, Ray Hunt. Intrusion Detection Techniques and Approaches // Department of Computer Science University of Canterbury, New Zealand, – 2002. – С. 2 – 14.
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. – С. 6 –22.
4. *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. – С. 3–6.
5. *Ли Доддз.* XML и базы данных? Доверьтесь своей интуиции. [HTML] (http://www.iso.ru/journal/articles/206.html)
6. *Грейвс М.* Проектирование баз данных на основе XML. – М.: Вильямс, 2002. – С. 12 – 70.
7. *Фертф Д.* Где хранить надежду электронной коммерции? // Сетевой. – 2001. – №1. HTML (http://www.setevoi.ru/cgi-bin/text.pl/magazines/2001/1/58)
8. *Lincoln* Laboratory Massachusetts Institute of Technology. The Detectability of Attacks in NT Audit Logs [HTML] (http://www.ll.mit.edu/IST/ideval/docs/2000/ntaudit-table.html)
9. *Федоров А., Елманова Н.* Введение в OLAP. [HTML] (http://olap.ru/basic/OLAP_intro1.asp)
10. *Han J., Kamber M.* – Data mining: concepts and techniques., – 2000. – С. 279 – 310.
11. *SQL* Server 2005 Books Online. Microsoft Association Algorithm [HTML](http://msdn2.microsoft.com/en-us/library/ms174916.aspx)