V Висновки

Проаналізовані найчастіше використовувані у практиці і теоретично обґрунтовані моделі захисту інформації в комп'ютерних системах. Розглянуто особливості реалізованих засобів захисту даних у середовищі реляційних, розподілених, об'єктно-орієнтованих СКБД.

Подано формалізований підхід до забезпечення безпеки розподілених баз даних, заснований на інтеграції моделей і засобів взаємодії і безпеки об'єктів у комп'ютерних мережах із моделями захисту інформації. Даний підхід у сполученні з організаційними, режимними і системними засобами безпеки КМ забезпечує не тільки високий рівень захисту даних, але й достатній рівень захисту РБД від багатьох загроз, що виникають у КМ.

Література: 1. Russel D., Gangemi G. T. Sr. Computer Security Basics. – N. Y.: O`Reilly & Associates, Inc., 1992. 2. National Computer Security Center. Trusted Network Interpretation // NCSC-TG-005, 1987. 3. Castano S., Fugini M., Martella G., Samarati P. Database Security. – Addison-Wesley, 1995. – 407 p. 4. Мельников Г. В. Защита информации в компьютерных системах. - М.: Финансы и статистика, 1997. – 368 с. 5. Герасименко В. А. Защита информации в автоматизированных системах обработки данных: В 2-х т. – М.: Энергоатомиздат, 1994. 6. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу // НД ТЗІ 1.1–002–99, ДСТСЗІ СБ України, Київ, 1999. 7. Нетесин И. Е. Вопросы безопасности и пути их решения в современных компьютерных сетях // Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні. – Київ: НТУУ «КПІ» . – 2000. – С. 199 – 203. – (Труди конф. «Безпека інформації в інформаційно–телекомунікаційних системах», Київ, 11 – 14 квітня 2000 р.) 8. Нетесин И. Е. Модели безопасности и защиты в распределённых компьютерных средах / / Проблемы программирования. –2000. – № 3–4. – С. 148–158. 9. Нетесин И. Е. Определение основ взаимодействия объектов в компьютерных сетях // Проблемы программирования.–2000.–№1–2.– С. 191–203.- (Спец. вып. "Материалы Второй Междунар. науч.-практ. конф. по программированию УкПРОГ'2000", Киев, 23-26 мая 2000 г.).

УДК 681.513

МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ТОЧКИ ЗРЕНИЯ БЕЗОПАСНОСТИ

Алексей Новиков, Сергей Кащенко

Физико-технический институт НТУУ «КПИ»

Анотація: **Детектори вторгнень продемонстрували на практиці свою високу ефективність у посиленні безпеки розподілених комп'ютерних систем. Існує два підходи до виявлення атак – заснований на поведінці та заснований на знаннях. Більш широкого розповсюдження набули системи, що використовують другий підхід. Це пояснюється недостатньою пропрацьованністю теоретичних й практичних аспектів першого. У статті пропонується модель поведінки програмного забезпечення з точки зору безпеки у вигляді стохастичного автомату та аналізується застосовність моделі.**

Summary: **Intrusion detection systems have demonstrated on practice their high efficiency in strengthening of distributed computer system security. There are two approaches to attack detection – behaviour-based and knowledge-based. More widespread are systems based on latter one. It can be explained by insufficient development of theoretical and practical aspects of former one. Model of software behaviour based on stochastic automaton is proposed and its applicability is analysed in the article.**

Ключевые слова: **Безопасность, системы обнаружения атак, стохастический автомат.**

I Введение

Развитие Internet и intranet-сетей создало необходимость дополнения традиционных средств обеспечения безопасности компьютерных систем (КС), таких как средства контроля доступа операционных систем и средства защиты периметра распределенных КС (межсетевые экраны), средствами мониторинга безопасности. Как правило, в роли таких средств выступают системы обнаружения атак (СОА). Рост доли инцидентов, выявленных при помощи СОА, а также рост рынка коммерческих систем свидетельствуют о высокой эффективности и признании роли средств мониторинга в задаче обеспечения безопасности распределенных КС [1].

СОА осуществляют сбор информации о связанных с безопасностью событиях в КС, преобразование (сжатие) этой информации в удобную для анализа форму, анализ или собственно выявление признаков атаки,

построение отчетов, реагирование на выявленные атаки либо нарушения политики безопасности [2]. По источнику информации СОА условно делятся на системные (host-based), использующие журналы аудита операционной системы и/или приложений, и сетевые (network-based), использующие получаемые благодаря широковещательной природе Ethernet-сетей данные сетевого обмена. По методам выявления признаков нарушений СОА делят также на две группы – работающие в рамках подхода, основанного на поведении (обнаружение аномалий), и в рамках подхода, основанного на знаниях (обнаружение злоупотреблений).

В основе работы методов первой группы лежит предположение о том, что любые нарушения безопасности будут проявлять себя как аномалии в поведении наблюдаемой компьютерной системы. В процессе ввода в эксплуатацию такая СОА "обучается", осуществляя построение модели наблюдаемой системы и ее объектов в той или иной форме. В процессе эксплуатации СОА сравнивает реальное поведение и предсказанное при помощи модели, и, при обнаружении значительных расхождений, сигнализирует о возможном нарушении. К данной группе методов анализа относятся метод статистических профилей, метод прогнозирующего порождения шаблона и метод искусственных нейронных сетей.

В основе работы методов второй группы положено описание всех известных атак либо нарушений безопасности в виде как можно более общих сигнатур или шаблонов. В процессе эксплуатации такая СОА непрерывно осуществляет поиск этих сигнатур либо шаблонов во входных данных и сигнализирует о возможном нарушении при их нахождении. К данной группе методов анализа относятся метод экспертных (продукционных) систем, метод модельных систем вывода, метод анализа переходов состояний, метод сопоставления шаблонов [3].

Таким образом, методы, основанные на знаниях, имеют существенное и принципиальное ограничение. С их помощью могут быть выявлены только известные атаки либо нарушения, т. е. такие, описание которых содержится в базе сигнатур (шаблонов). Недостатками СОА, использующих данный подход, также являются необходимость постоянного и своевременного обновления баз сигнатур вследствие постоянного появления новых атак и уязвимостей, неспособность адаптироваться к конкретной КС, значительные вычислительные затраты на поиск в больших базах сигнатур в режиме реального времени. Методы, основанные на поведении, лишены всех вышеперечисленных недостатков, однако страдают от проблемы "ложных тревог" (false positives) и "пропусков цели" (false negatives). "Ложной тревогой" называется ситуация, когда аномальное, а, следовательно, фиксируемое СОА поведение наблюдаемой КС или её объектов не является нарушением. "Пропуском цели" называется противоположная ситуация, когда нарушение безопасности не проявляется в виде аномалии поведения, и, следовательно, не фиксируется СОА. На практике для методов обнаружения атак, основанных на знаниях, уровни "ложных тревог" и "пропусков цели" также не являются нулевыми вследствие обобщенности описаний нарушений. Увеличение же точности описаний ведет к невозможности обнаружения такой СОА даже незначительных вариаций известных ей атак. Однако значения этих показателей значительно ниже, чем у методов первой группы [1].

Рассматривая теорему Байеса

$$P(s \mid \overline{x}) = \frac{P(s)P(\overline{x} \mid s)}{\sum_i P(s_i)P(\overline{x} \mid s_i)}$$

$$P(I \mid A) = \frac{P(I)P(A \mid I)}{P(I)P(A \mid I) + P(\neg I)P(A \mid \neg I)} = \frac{P(I)P(A \mid I)}{P(I)(1 - P(\neg A \mid I)) + P(\neg I)P(A \mid \neg I)},$$

применительно к данной задаче

где P(I|A) – апостериорная вероятность нарушения при условии фиксации его СОА,
P(I), P(¬I) – априорная вероятность нарушения безопасности или его отсутствия,
P(A|I) – вероятность обнаружения нарушения СОА,
P(A|¬I) – вероятность "ложной тревоги",
P(¬A|I) – вероятность "пропуска цели";

можно видеть [4], что, поскольку нарушения безопасности довольно редки по сравнению со всеми событиями в наблюдаемой КС (т. е. P(¬I) >> P(I)), определяющим в знаменателе является второе слагаемое. Иными словами, уровень "ложных тревог" оказывает гораздо большее влияние на P(I|A), чем уровень "пропусков цели".

Именно это в сочетании с высоким уровнем "ложных тревог" при использовании методов, основанных на поведении, определило более широкое распространение СОА, использующих методы, основанные на знаниях, несмотря на недостатки последних. В частности, авторам неизвестно о существовании коммерческих СОА, использующих первый подход.

Таким образом, актуальной становится задача разработки основанных на поведении методов анализа данных мониторинга с целью выявления признаков атак либо нарушений безопасности, имеющих более низкий уровень “ложных тревог”, чем существующие на данный момент.

II Моделирование поведения программного обеспечения с точки зрения безопасности при помощи стохастического автомата

В существующих методах первой группы модель наблюдаемой КС или ее объектов, как правило, была представлена в неявном виде. Её параметры, в зависимости от конкретного метода, задавались либо в виде статистических “профилей” как наборов характеристик, либо в виде наборов вероятностей следования одних событий за другими, либо в виде матриц синаптических весов нейронных сетей [1]. При этом задача выбора и предобработки входных данных для СОА решалась её разработчиками неформально, на основе опыта и интуиции, а в качестве наблюдаемых объектов чаще всего выступали пользователи системы. Выбор оптимального подмножества из N доступных наблюдению характеристик сам по себе является задачей с экспоненциальной оценкой $O(2^N)$. Другой трудностью является невозможность адекватного моделирования поведения человека. Наконец, анализ наиболее совершенных способов описания сигнатур атак либо нарушений безопасности, применяемых в методах второй группы [5], показывает важность в данной задаче не столько самих характеристик, сколько структурных отношений между событиями в виде частичной упорядоченности во времени. Это также мало учитывалось при использовании существующих методов первой группы.

Поэтому предлагается использовать в качестве наблюдаемых объектов не пользователей КС, а процессы, выполняющиеся в ней. Поведение процесса может быть точно описано детерминированным автоматом, однако громоздкость такого описания приведет к его непригодности на практике. В силу этого предлагается использовать укрупненное (и упрощенное) описание поведения процесса в виде стохастического автомата:

$$\alpha_s = \{\ Q,\quad l,\quad \delta,\quad q_0,\quad F\ \}$$

В качестве множества состояний автомата Q при этом выступает множество операторов передачи управления в коде процесса – ветвлений, циклов, переключений и т.д. В качестве входного алфавита l может служить произвольное подмножество поступающих на вход процесса данных. Обсуждение способов формирования входного алфавита выходит за рамки данной статьи, однако, в качестве примера стоит упомянуть возможность включения в него имен внешних объектов данных, к которым обращается процесс. Обязательным элементом алфавита должен являться “пустой” символ как представление переходов без ввода-вывода и обмена данными. Таблица переходов δ состоит из элементов вида

$$\delta(q_i, l_k) = \{\ [q_n, p_{ik}^n],\ [q_{n-1}, p_{ik}^{n-1}],\ .....,\ [q_0, p_{ik}^0]\ \},$$

то есть каждой комбинации состояния q_i и входного символа l_k соответствует набор состояний q, в которые возможен переход из данного состояния по получении данного символа входного алфавита, причем с каждым из этих состояний ассоциирована вероятность p осуществления перехода в него для данной комбинации. Переходам автомата соответствуют участки кода с последовательным выполнением операторов, а вероятности переходов представляют собой вероятности срабатывания тех или иных операторов передачи управления, и, следовательно, вероятности выполнения этих участков. Начальное состояние q_0 соответствует точке входа процесса. Множество конечных состояний F состоит всего из одного состояния q_n, соответствующего завершению процесса. Для упрощения модели мы будем условно относить точку входа и состояние завершения к операторам передачи управления.

Поскольку любой процесс осуществляет операции ввода-вывода и обмена данными в участках кода с последовательным выполнением, в рамках данной модели возможно только наблюдение переходов и нет возможности непосредственно наблюдать состояния стохастического автомата.

При вводе в эксплуатацию СОА, использующей данную модель, встает задача определения состояний стохастического автомата и расчета вероятностей $p^j{}_{ik}$ по множеству наблюдаемых последовательностей переходов $\delta_1, \delta_2, \ldots, \delta_n$ и соответствующих им последовательностей входных символов $l_1, l_2, \ldots, l_n$. При доступности исходного кода наблюдаемого процесса определение состояний и переходов тривиально, а таблица переходов δ может быть рассчитана введением в код контрольных точек, соответствующих переходам, и подсчетом их срабатываний во время эксплуатации процесса в целевой среде. В случае же недоступности исходного кода состояния и переходы должны быть определены по набору последовательностей наблюдений $k_1, k_2, \ldots, k_n$ и входных символов $l_1, l_2, \ldots, l_n$, т. е. должны быть определены соответствия $f:K\rightarrow\delta$ и $g:KxK\rightarrow Q$. Метод решения данной задачи авторам на данный момент не известен и подлежит разработке.

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

При эксплуатации СОА, использующей стохастический автомат, основным способом выявления нарушений как аномалий в поведении является подсчет вероятности текущей последовательности переходов

$$P(S) = p(\delta_n)p(\delta_{n-1})p(\delta_{n-2}).... p(\delta_1)p(\delta_0)$$

где n – длина последовательности (должна быть ограничена). Обнаружение мало вероятной последовательности или даже одного перехода и должно фиксироваться как нарушение или атака. Иными словами, нарушением в данном случае будет считаться использование "необычных" передач управления в коде, не задействованных на этапе ввода СОА в эксплуатацию ("обучения"). Побочным способом выявления может быть фиксация нарушений в случае поступления на вход автомата новых входных символов, т. е. расширение входного алфавита.

Достоинствами использования стохастического автомата как модели наблюдаемого процесса являются простота включения в модель априорных сведений об атаках либо нарушениях и простота "дообучения" или уточнения модели в процессе эксплуатации. Первое может осуществляться прямым (не по результатам "обучения") присвоением низких вероятностей некоторым из переходов в δ и позволит использовать модель в методах обеих групп. Второе может осуществляться повышением вероятности для некоторых переходов по запросу администратора безопасности в случае возникновения "ложной тревоги" и может привести к существенному снижению уровня последних.

III Анализ применимости модели на примере современных уязвимостей

Можно использовать следующее определение понятия "уязвимость". Если рассматривать состояния КС как делящиеся на авторизованные и неавторизованные с точки зрения политики безопасности, то "уязвимым" называется такое авторизованное состояние, из которого можно попасть в неавторизованное ("скомпрометированное") через авторизованные переходы, последовательность которых и является атакой. "Уязвимостью" тогда называется такая характеристика уязвимого состояния, которая отличает его от всех состояний, не являющихся таковыми. Существует класс уязвимостей, которые инвариантны конкретной КС, поскольку позволяют производить действия в обход встроенных механизмов защиты и подразумеваемой при разработке конкретного ПО политики безопасности. Именно уязвимости данного класса изучаются при разработке СОА.

Для анализа качественного состава современных уязвимостей авторами использовалась база данных ICAT, содержащая ссылки на 2412 уязвимостей (по состоянию на 04. 05. 2001). Одним из способов классификации содержащихся в базе данных уязвимостей является классификация по типу или причине возникновения. Типов всего 8:

- ошибки проверки ввода (включают ошибки проверки границ и ошибки переполнения буфера);
- ошибки проверки доступа;
- ошибки (сбои) при обработке исключений;
- ошибки среды выполнения;
- ошибки конфигурации;
- ошибки гонок;
- ошибки дизайна;
- другие (неклассифицированные) ошибки.

Уязвимость вследствие ошибки проверки ввода возникает, когда неполная либо некорректная проверка входных данных в ПО приводит к возможности атаки (нарушения). При этом существует два подкласса – ошибки проверки границ, когда источником уязвимости служат неправильные предположения разработчиков ПО относительно области определения элементов входных данных, и ошибки переполнения буфера, когда неверны предположения относительно размера входных данных. Уязвимости вследствие ошибок проверки доступа вызваны погрешностями в реализации механизмов контроля доступа. Причиной возникновения уязвимостей-ошибок обработки исключений является некорректность или неполнота кода-обработчика исключительных ситуаций. Уязвимости вследствие ошибок среды возникают из-за непредвиденного взаимодействия ПО с операционной системой или другим ПО. Неверные настройки ПО также могут служить источником уязвимости, особенно, когда оно поставляется с такими настройками по умолчанию. Уязвимости вследствие гонок возникают при нарушении атомарности проверок, связанных с политикой безопасности. И, наконец, уязвимости вследствие ошибок дизайна возникают из-за неправильных предположений при проектировании ПО.

Как упоминалось выше, при использовании стохастического автомата в качестве модели наблюдаемого процесса возможны два способа выявления нарушений как аномалий поведения. При использовании первого способа признаком нарушения считается задействование таких передач управления или ветвей алгоритма,

которые не наблюдались при вводе СОА в эксплуатацию. При использовании второго способа признаком нарушения считается поступление на вход процесса ранее не использовавшихся данных, например, обращение его к “необычным” внешним объектам данных. Использование необычных передач управления является обязательным атрибутом при реализации таких уязвимостей как ошибки переполнения буфера и ошибки при обработке исключений. В свою очередь, доступ к “необычным” внешним объектам данных является обязательным для реализации таких уязвимостей как ошибки проверки доступа и характерным при реализации уязвимостей, вызванных общими ошибками проверки ввода, ошибками проверки границ, и, в несколько меньшей мере, ошибками среды. В то же время ни один из двух упомянутых способов не гарантирует выявления нарушений при реализации таких типов уязвимостей как ошибки конфигурации, ошибки гонок и ошибки дизайна. Авторам представляется сомнительной возможность выявления ошибок дизайна и конфигурации любыми методами, основанными на поведении. Также существует класс уязвимостей, реализация атаки на которые обнаруживается первым из способов (по использованию необычных передач управления), и частично перекрывающий все вышеперечисленные типы. Это уязвимости, приводящие к возможности отказа в обслуживании (Denial of Service, DoS). Реализация отказа в обслуживании всегда приводит процесс из текущего состояния в конечное.

Общее распределение уязвимостей по типам вместе с долей уязвимостей, приводящих к отказу в обслуживании для каждого типа, приведено на рис. 1.

Таким образом, предложенный подход дает возможность обнаружить атаки на как минимум 72.8 % существующих уязвимостей.

IV Заключение

В статье рассмотрены два подхода к обнаружению атак на КС – основанный на поведении и основанный на знаниях, их преимущества и недостатки. Предложена модель поведения программного обеспечения (процессов) с точки зрения безопасности в виде стохастического автомата для использования в рамках первого подхода. Проанализирована применимость данной модели путем рассмотрения качественного состава современных уязвимостей.

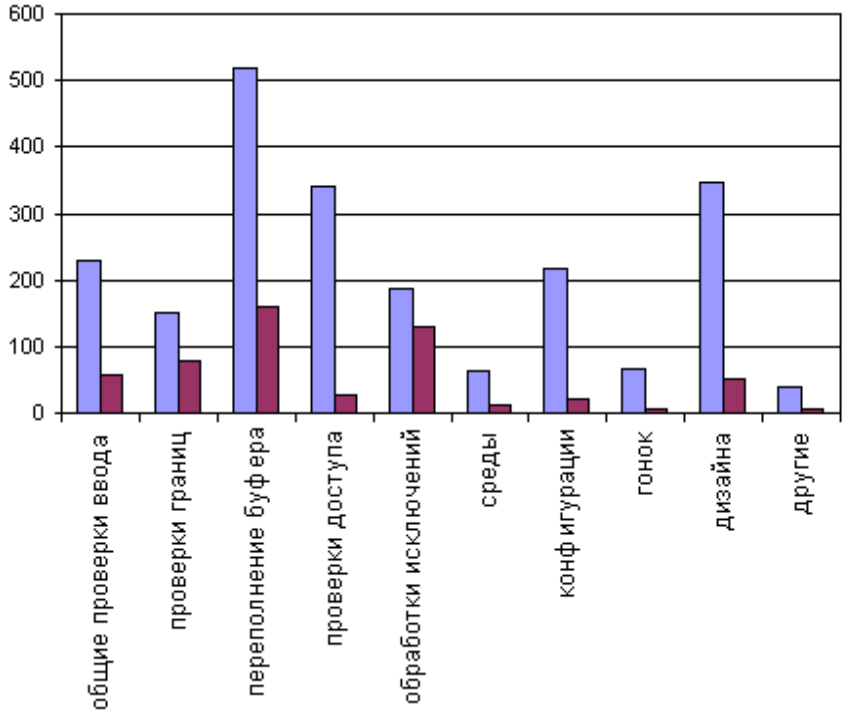

Рисунок 1 – Распределение уязвимостей по типам

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.