* автентифікація відправника – дозволяє забезпечити захист від відмови від авторства і однозначно встановити належність певного об'єкта певному користувачу, тобто той факт, що об'єкт був створений або відправлений даним користувачем;

* автентифікація отримувача – послуга дає можливість забезпечити захист від відмови від одержання і дозволяє однозначно встановити факт одержання певного об'єкта певним користувачем.

Найбільш розповсюдженою і ефективною практикою реалізації спостереженості є реєстрація, яка включає протоколювання та аудит. Для АС з високим рівнем захищеності вони мають проводитися в реальному часі [3].

III Висновки

За допомогою поняття потоку інформації подано визначення доступу. Проведено аналіз множини доступу і визначені підмножини доступів, за допомогою яких уточнено формальне визначення основних загроз інформації. Отримані формальні визначення загроз інформації в подальшому дозволяють на різних класах політики безпеки аналізувати захищеність АС. Крім того, наведений формалізм дає можливість описати поняття профілю [3], а також проводити формальне дослідження інших політик (наприклад, рольової) безпеки в АС. Перелічуємо також основні послуги, за допомогою яких забезпечується захист від наведених загроз інформації.

Література: 1. Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу. – НД ТЗІ 1.1-002-98, ДСТСЗІ СБ України, Київ, 1998. 2. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. – НД ТЗІ 1.1-001-98, ДСТСЗІ СБ України, Київ, 1998. 3. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. – НД ТЗІ 2.2-001-98, ДСТСЗІ СБ України, Київ, 1998. 4. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. – НД ТЗІ 2.2.-002 -98, ДСТСЗІ СБ України, Київ, 1998. 5. Антонюк А. О., Жора В. В. Загрози інформації і канали витоку // Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні. – К.: НТУУ «КПІ». – К. 2001 – С. 42 – 46. 6. Грушо А. А., Тимонина Е. Е. Теоретические основы защиты информации. М.: «Яхтсмен», 1996. 7. Щербаков А. Ю. Введение в теорию и практику компьютерной безопасности. – М.: издатель Молгачева С. В., 2001, 352 с., ил.

УДК 681.513

ОБНАРУЖЕНИЕ АНОМАЛИЙ ПОВЕДЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Алексей Новиков, Сергей Кащенко
Физико-технический институт НТУУ «КПИ»

Анотація: **Розглядається модель поведінки програмного забезпечення з точки зору безпеки у вигляді стохастичного автомата, що дозволить обраховувати ймовірність появи поточних даних моніторингу на основі попередніх спостережень.**
Summary: **In the report the software security behavior model based on stochastic automaton is considered, which will allow to calculate probability of current monitoring data appearance on the basis of previous observations.**
Ключевые слова: **Безопасность, системы обнаружения атак, стохастический автомат.**

I Введение

В последнее десятилетие появились и приобретают все большее распространение средства обеспечения безопасности компьютерных систем нового класса – средства мониторинга или системы обнаружения атак (СОА). Появление и развитие такого рода средств обусловлено недостаточностью в современных условиях, характеризующихся экспоненциальным ростом глобальных и корпоративных сетей, традиционных средств обеспечения безопасности, таких как криптографические системы, средства контроля доступа операционных систем и средства защиты периметра распределенных компьютерных систем (межсетевые экраны). Ряд источников (например, [1]) указывают на постоянный рост рынка СОА и количества нарушений, выявленных с их помощью, что говорит о высокой эффективности и признании роли СОА в задаче обеспечения безопасности компьютерных систем.

СОА можно классифицировать двумя способами – по источнику данных мониторинга и по методам анализа этих данных с целью выявления нарушений. По источнику данных мониторинга СОА делятся на два больших класса – системные СОА, использующие данные журналов аудита операционных систем или приложений, и сетевые СОА, использующие перехваченные различными способами данные сетевого обмена (траффик). По методам анализа данных СОА также делятся на два класса – использующие методы, основанные на знаниях (knowledge-based) и основанные на поведении (behaviour-based). Методы первой группы также носят название "методы обнаружения злоупотреблений", а второй – "методы обнаружения аномалий".

При использовании методов, основанных на знаниях, СОА содержит в себе библиотеки описаний или моделей известных видов нарушений, например, представленные в виде набора правил экспертной системы или в виде сети Петри. В процессе эксплуатации такая СОА сравнивает текущие данные мониторинга со своими описаниями и фиксирует нарушения в случаях совпадений. Главный недостаток такого рода систем является прямым следствием основного принципа их работы – с их помощью возможно обнаружение только заранее известных нарушений (атак) или их вариаций. Учитывая постоянное появление новых уязвимостей и видов атак, это приводит к необходимости постоянного и своевременного обновления библиотек описаний в процессе эксплуатации СОА. Другими недостатками данного подхода являются неспособность адаптироваться к конкретной компьютерной системе и значительные затраты вычислительных ресурсов на поиск в больших наборах описаний в режиме реального времени. Тем не менее, в настоящий момент большее распространение получили именно такие СОА, прежде всего в силу своей более высокой точности.

Все методы, основанные на поведении, опираются на предположение о том, что любые нарушения безопасности будут проявлять себя как аномалии в поведении объектов мониторинга. В процессе ввода в эксплуатацию такая СОА "обучается", осуществляя построение и оценивание "моделей" наблюдаемых объектов или их составляющих. На данный момент в качестве таких "моделей" использовались статистические профили (векторы различных показателей работы объекта мониторинга, таких как, например, максимальное количество открытых файлов или средняя загрузка процессора), системы правил и искусственные нейронные сети. В процессе эксплуатации СОА, использующая методы данной группы, сравнивает реальное поведение объекта мониторинга с предсказанным при помощи модели, и фиксирует возможное нарушение при обнаружении значительных расхождений. Основным достоинством таких СОА является возможность обнаружения ранее неизвестных видов атак и нарушений. В то же самое время при их эксплуатации возникает проблема «ложных тревог» (false positives) и «пропусков цели» (false negatives). «Ложной тревогой» называется ситуация, когда аномальное, а, следовательно, фиксируемое СОА поведение наблюдаемого объекта не является нарушением. «Пропуском цели» называется противоположная ситуация, когда нарушение безопасности не проявляется в виде аномалии поведения, и, следовательно, не фиксируется СОА. Как показано в [2], вследствие достаточно низкой априорной вероятности атаки (нарушения) определяющее влияние на апостериорную вероятность атаки при условии фиксации её СОА оказывает именно уровень "ложных тревог". Иными словами, при большом количестве "ложных тревог" администратор безопасности вскоре перестает реагировать на сигналы СОА о выявлении атаки. Уровень же "пропусков цели" является меньшей проблемой, поскольку может быть существенно снижен путем использования дублирующих друг друга по объектам мониторинга, но разных по используемым методам анализа детекторов (модулей СОА).

Исходя из этого, в рамках данного подхода актуальной становится задача разработки новых методов анализа, имеющих более низкий уровень "ложных тревог". Такое повышение точности может быть достигнуто путем выбора адекватной математической модели наблюдаемого объекта с последующим "оцениванием" такой модели при обучении СОА.

II Модель поведения программного обеспечения

В то время как объектом мониторинга в ранних работах по методам анализа, основанным на поведении, является пользователь или группа пользователей, в данном докладе в качестве объекта мониторинга выступает процесс как модуль программного обеспечения (ПО). Такое решение имеет следующие преимущества:

- снимаются трудности, связанные с адекватным моделированием поведения человека;
- вследствие независимости от конкретных пользователей СОА становится масштабируемой;
- уменьшаются потребности СОА в вычислительных ресурсах, поскольку можно выделить для наблюдения относительно небольшое подмножество процессов (например, SUID-ные процессы в ОС Unix).

Данные мониторинга как поступающие из журналов аудита, так и получаемые в результате перехвата сетевого обмена, можно представлять как последовательности объектов различных типов. Каждый тип объекта при этом имеет свой набор параметров (или пространство параметров), значения которых меняются от объекта к объекту. Важной особенностью задачи анализа данных мониторинга, отмеченной рядом авторов [3, 4], является то, что какое-либо одно наблюдение очень редко является признаком нарушения. Принципиальное значение имеют последовательности наблюдений. Это позволяет утверждать о наличии структурных отношений между объектами в виде частичной упорядоченности во времени и приводит к необходимости учета таких отношений в математической модели.

Поведение процесса может быть точно описано детерминированным автоматом, однако громоздкость такого описания приводит к полной его непригодности на практике. Укрупнение же или обобщение описания неизбежно приводит к вероятностной (статистической) модели. Поведение процесса, с другой стороны, будет определяться только его текущим состоянием и не будет зависеть от предыдущих состояний. Таким образом, при фиксации любого из состояний вероятностная модель распадается на две независимые части. Это позволяет утверждать, что распределение вероятностей на последовательности состояний в данном случае является марковским, т. е. для любого $i = 1, ..., n$ справедливо:

$$p(k) = p(k_i) \cdot p\left(k_0^{i-1} / k_i\right) \cdot p\left(k_{i+1}^{n} / k_i\right)$$

где $p(k)$ – вероятность всей последовательности k,

$p(k_i)$ – вероятность того, что i-й элемент последовательности принял значение k_i ,

$p(k_0^{i-1}/k_i)$ – вероятность последовательности $k_0\ k_1\ ...k_{i-1}$ при условии что i-й элемент равен k_i ,

$p(k^n{}_{i+1}/k_i)$ – вероятность последовательности $k_{i+1}\ k_{i+2}\ ...k_n$ при условии что i-й элемент равен k_i ,

n – длина последовательности.

Исходя из этого, авторами в работе [5] было предложено в качестве модели поведения процесса использовать стохастический автомат, заданный:

- множеством состояний K;
- множеством выходных сигналов X;
- распределением вероятностей начального состояния $p(k_0)$;
- распределениями вероятностей переходов $p(k_i\ ,\ x_i / k_{i-1})$;

Множества состояний и выходных сигналов должны определяться при настройке СОА на конкретный модуль ПО, а распределения вероятностей – устанавливаться при “обучении” модели. Множество выходных сигналов при этом будет представлять собой отображение данных мониторинга, причем каждый тип объекта может отображаться в один или более сигналов.

Статистическая модель в данном случае представляет собой распределение вероятностей – функцию вида $X^n \text{ x } K^{n+1} \rightarrow R$, задающую совместную вероятность для каждой пары последовательностей x и k. Эту функцию можно однозначно определить через функции $p(k_0)$ и $p(k_i\ ,\ x_i / k_{i-1})$. Для данного класса моделей существует целый ряд методов распознавания [6], хорошо зарекомендовавших себя в таких задачах, как распознавание текстовых строк и речевых сигналов.

Так при использовании подобной модели вероятность появления последовательности выходных сигналов (наблюдений) x_1, x_2, ..., x_n можно рассчитать как многомерную сумму (по всем возможным последовательностям k)

$$p(x) = \sum_k p(k, x) = \sum_{k_0} \sum_{k_1} \text{K} \sum_{k_{n-1}} \sum_{k_n} p(k_0) \prod_{i=1}^{n} p(x_i, k_i / k_{i-1}), \tag{1}$$

вычисление которой сводится к вычислению матричного произведения

$$p(x) = \varphi \cdot \left(\prod_{i=1}^{n} P_i \right) \cdot f\ ,$$

где φ – $|K|$-мерный вектор-строка,

P_i – матрицы размера $|K|\text{x}|K|$, содержащие в (k_i)-строке и (k_{i-1})-столбце $p(k_i\ ,\ x_i / k_{i-1})$ для различных x_i ,

f – $|K|$-мерный вектор-столбец, все элементы которого равны 1.

Данный метод сохраняет свою работоспособность даже в случае неполных данных. Так рассчитать вероятность появления любого подмножества последовательности x можно по формуле:

$$p((x_i / i \in I), k) = \sum_k p(k_0) \prod_{i \in I} P_i(k_i / k_{i-1}),$$

где I – подмножество индексов i, для которого известны значения x_i ,

$P_i\ (k_i / k_{i-1})$ – матрицы размера $|K|\text{x}|K|$, содержащие вероятности $p(k_i,\ x_i / k_{i-1})$ при фиксированных $x_i\ \in I$.

III Оценивание модели поведения

Оценивание модели может быть выполнено как по совокупности пар последовательностей состояний и выходных сигналов (задача обучения), так и по совокупности последовательностей только выходных сигналов (задача самообучения). Эта совокупность в обоих случаях носит название “обучающая выборка”.

Первый способ достаточно просто реализуем при наличии исходных кодов наблюдаемого процесса. В этом случае в качестве множества состояний может выступать множество операторов передачи управления в коде процесса – ветвлений, циклов, переключений, вызовов собственных функций процесса и обращений к внешним, таким как, например, API операционной системы. При этом следует отметить, что выполнение одной и той же функции при вызове из разных мест кода будет представлять собой последовательности разных состояний. Переходам стохастического автомата будут соответствовать участки кода с последовательным выполнением операторов, а $p(k_i, x_i/k_{i-1})$ будут отображать вероятности результатов срабатывания операторов передачи управления, и, следовательно, вероятности выполнения этих участков. Для получения совокупности последовательностей состояний следует модифицировать исходный код процесса таким образом, чтобы обеспечить регистрацию состояний. Простота такой модификации позволяет выполнить её автоматически.

Алгоритм обучения зависит от способа получения обучающей выборки. В случае, когда каждая пара x и k была получена случайным выбором из множества всех возможных пар и можно предположить, что результаты экспериментов с процессом взаимно независимы, может использоваться простой алгоритм наиболее правдоподобного оценивания. Задача оценивания при этом формулируется как нахождение совокупности функций p_i, максимизирующей вероятность обучающей выборки, т. е.:

$$P^* = arg\max_{P^*} \prod_{j=1}^{l} p\left(x^j, k^j\right),$$

где P^* – совокупность функций p_i,

$j = 1, \ldots, l$ – индекс пары последовательностей,

l – общее число пар (экспериментов),

$p(x^j, k^j)$ – вероятность j-го эксперимента.

Ниже приведена более подробная формулировка:

$$P^* = \left(p_1^*, p_2^*, \ldots, p_n^*\right) = arg\max_{p_1} \ldots \max_{p_n} \prod_{j=1}^{l} p_1\left(k_0^j, x_1^j, k_1^j\right) \prod_{i=2}^{n} \frac{p_i\left(k_{i-1}^j, x_i^j, k_i^j\right)}{\sum_{k \in K} \sum_{x \in X} p_i\left(k_{i-1}^j, x, k\right)}.$$

Решением этой задачи являются числа $p(k_{i-1}, x_i, k_i)$, пропорциональные числу вхождений тройки k_{i-1}, x_i, k_i в обучающую выборку, для которых

$$\sum_{k' \in K} \sum_{x' \in X} \sum_{k'' \in K} p\left(k', x', k''\right) = 1.$$

В случае, когда нельзя обеспечить независимость и случайность экспериментов в обучающей выборке, следует использовать более сложный алгоритм минимаксного оценивания. Задача оценивания при этом формулируется как нахождение совокупности функций p_i, при которой вероятность каждого эксперимента не ниже определенной величины. Иными словами, искомая совокупность должна максимизировать минимальную по всем экспериментам в обучающей выборке вероятность пары, т. е.:

$$P^* = \left(p_1^*, p_2^*, \ldots, p_n^*\right) = \arg\max_{p_1} \ldots \max_{p_n} \min_{j} \left(p_1\left(k_0^j, x_1^j, k_1^j\right) \prod_{i=2}^{n} \frac{p_i\left(k_{i-1}^j, x_i^j, k_i^j\right)}{\sum_{k \in K} \sum_{x \in X} p_i\left(k_{i-1}^j, x, k\right)} \right).$$

Для решения данной задачи каждому эксперименту присваивается число – так называемый “вес”, и, последовательным применением алгоритма максимально правдоподобного оценивания осуществляется пошаговая оптимизация выпуклой функции над этими числами.

Второй способ оценивания, самообучение, применяется в случае недоступности для наблюдения состояний процесса. Данный способ может применяться в случае, когда нет возможности вносить изменения в исходный код наблюдаемого процесса, или же когда внесение описанных выше изменений представляется неоправданным с точки зрения потребления вычислительных ресурсов. Задача оценивания при этом, как и в случае обучения, формулируется как нахождение совокупности функций p_i, максимизирующей вероятность обучающей выборки:

$$P^* = \arg\max_{P} \prod_{j=1}^{l} \sum_{k \in K^{n+1}} p\left(x^j, k\right),$$

или, более подробно:

$$P^* = \left(p_1^*, p_2^*, \ldots, p_n^*\right) = \arg\max_{p_1} \ldots \max_{p_n} \prod_{j=1}^{l} \sum_{k_0} \sum_{k_1} \ldots \sum_{k_n} p_1\left(k_0, x_1^j, k_1\right) \prod_{i=2}^{n} \frac{p_i\left(k_{i-1}, x_i^j, k_i\right)}{\sum_{k \in K} \sum_{x \in X} p_i\left(k_{i-1}, x, k\right)}.$$

Мы не будем приводить здесь алгоритм решения данной задачи, поскольку он подробно описан в [6].

IV Обнаружение аномалий поведения программного обеспечения

После оценивания модели процесса СОА получает наборы $p(k_{i-1}, x_i, k_i)$ и, поскольку

$$p\left(k_0, x_1, k_1\right) = p\left(k_0\right) \cdot p\left(x_1, k_1 / k_0\right)$$

и

$$p\left(x_i, k_i / k_{i-1}\right) = \frac{p\left(k_{i-1}, x_i, k_i\right)}{p\left(k_{i-1}\right)} = \frac{p\left(k_{i-1}, x_i, k_i\right)}{\sum_{k \in K} \sum_{x \in X} p\left(k_{i-1}, x, k\right)},$$

может непосредственно вычислять вероятности текущих последовательностей наблюдений в соответствии с формулой (1). В качестве меры аномальности поведения процесса предлагается использовать величину, обратно пропорциональную вычисленной таким образом вероятности.

Собственно выявление нарушений может осуществляться следующими способами.

1. Вероятность текущего наблюдения может сравниваться с некоторым пороговым значением, и, в зависимости от результатов сравнения, указывать на возможность нарушения или атаки. Значение порога может задаваться администратором безопасности вручную или рассчитываться на основе статистики нарушений или данных анализа рисков для данной компьютерной системы.
2. Аналогично модели "нормального" поведения, СОА может содержать в себе группу моделей "аномального" поведения, т. е. таких моделей, оценивание которых было выполнено на парах последовательностей, полученных при реализации нарушений. Получив на основе последовательности x набор вероятностей этой последовательности $p_A(x), p_B(x), \ldots, p_M(x)$ в различных моделях, можно не только зафиксировать факт нарушения либо нормальной работы, но и определить конкретный вид нарушения. При использовании данного способа метод анализа данных мониторинга будет принадлежать как к основанным на поведении, так и к основанным на знаниях.

Одной из проблем при использовании методов распознавания стохастических автоматов (марковских последовательностей) является фиксированный размер n последовательности. Это число ("размер окна") должно быть выбрано заранее, поскольку для модели нет способа перейти от одного размера последовательности к другому. Выбор слишком маленького "размера окна" приведет к падению точности алгоритма вследствие недостатка информации для распознавания, в противоположном случае, при выборе слишком большого "размера окна" точность упадет вследствие того, что алгоритм будет обрабатывать много лишней информации.

V Заключение

Рассмотрена модель поведения программного обеспечения (процессов) с точки зрения безопасности в виде стохастического автомата. Модель предназначена для использования в рамках "основанного на поведении" подхода к анализу данных мониторинга в системах обнаружения атак. Рассмотрен способ оценивания для данного класса моделей. Предложено использовать для определения меры аномалии поведения ПО вероятность появления текущей последовательности наблюдений, вычисленную на основе совокупности предыдущих наблюдений.

Литература: 1. J. Allen, A. Christle, W. Fithen et al. State of the Practice of Intrusion Detection Technologies.– Tehnical Report CMU/SEI-99-TR-028.–2000.–220 p. 2. S. Axelsson. The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection // Proceedings of the 6th ACM Conference on Computer and Communications Security (Kent Ridge Digital Labs, Singapore) – 1999. 3. S. Kumar, E. Spafford. A Pattern Matching Model for Misuse Intrusion Detection// Proceedings of 17th National Computer Security Conference.– 1994.– p. 11–21. 4. H. Debar, B. Dorizzi. An Application of a Recurrent Network to an Intrusion Detection System // Proceedings of the International Joint Conference on Neural Networks. – 1992. – vol 2.– pp. 478–483. 5. А. Новіков, С. Кащенко.

Моделирование поведения программного обеспечения с точки зрения безопасности.– Науково-технічний збірник "Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні": Київ.– вип. 2–2001. – с. 124–129. 6. Шлезингер М. И. Три лекции по математическим методам в распознавании образов. – ***www.iasa.kiev.ua.***

УДК 681.3.06:519.248.681

ПРОТОКОЛЫ НАПРАВЛЕННОГО ШИФРОВАНИЯ В ГРУППАХ ТОЧЕК НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ И ИХ СВОЙСТВА

Иван Горбенко, Станислав Збитнев, Андрей Поляков, Дмитрий Балагура
Харьковский Национальный Университет Радиоэлектроники

Анотація: **Розглядаються протоколи направленого шифрування з використанням еліптичних кривих на скінченних полях. Здійснюється аналіз класичних протоколів направленого шифрування та протоколів на еліптичних кривих. Аналізуються розміри параметрів і ключів.**
Summary: **Considered protocols of Elliptic Curve Integrated Encryption Scheme. Produce analysis of classical protocols and protocols of Elliptic Curve Integrated Encryption Scheme. Analyzed size of parameters and keys.**
Ключевые слова: **Эллиптическая кривая, направленное шифрование, поле Галуа, криптографические ключи.**

Введение

В последнее время в мире в качестве математического аппарата несимметричных преобразований все шире используется математика преобразований в группах точек на эллиптических кривых (ЕС) над полем. Создано и реализовано уже большое количество алгоритмов, реализующих известные схемы цифровой подписи, различных вариантов протоколов генерации общих секретов, алгоритмов направленного шифрования. В данной статье мы опираемся на уже известные комбинированные алгоритмы направленного шифрования. Рассматриваются схемы реализации направленного шифрования в группах точек эллиптических кривых, приводится их полное описание, а также основные свойства. Кроме того, в статье представлены результаты оценки сложности и скорости направленного шифрования в группах точек эллиптических кривых [1].

1 Анализ применяемых алгоритмов направленного шифрования

Вначале рассмотрим реализацию направленного шифрования RSA [2]. При использовании этого алгоритма получатель должен знать общесистемные параметры P_j и Q_j, причем P_j и Q_j должны быть «сильными» простыми числами. Этим повышается стойкость алгоритма. Используя эти числа, получатель выполняет следующие действия:

- формирует открытый ключ E_k такой, что $1 \le E_k \le \varphi(N_j)$, где $N_j = P_j \cdot Q$, а $\varphi(N_j)$ – функция Эйлера, и $(E_k, \varphi(N_j)) = 1$;
- вычисляет личный ключ D_k как обратный элемент к E_k в кольце; D_k вычисляется из соотношения $E_k \cdot D_k \equiv 1 \pmod{\varphi(N)}$;
- передаёт отправителю открытый ключ получателя E_k и модуль преобразования N_j, обеспечивая их целостность и подлинность.

Зашифрование выполняется по следующей схеме: всё сообщение делится на блоки, длина которых равна длине модуля преобразований $M = M_1 \| M_2 \| \ldots \| M_n$, затем выполняется зашифрование каждого блока сообщения по формуле $C_i = M_i^{E_k} \pmod N$.

Расшифрование каждого блока выполняется по формуле $M_i' = C_i^{D_k} \pmod N$, а затем все блоки объединяются в сообщение.