комендуется заблокировать вывод в заголовках ответа HTTP сервера информацию об используемом программном обеспечении.

Данная система с отработанными и автоматизированными механизмами конфигурации системы, заведения новых доменов и пользователей, единым интерфейсом администрирования, конфигурацией квотирования места на жестком диске, а также системы квотирования системных ресурсов вполне может претендовать на статус коммерческого продукта.

УДК 681.3.053:681.32

Е.П. Тумоян

МЕТОД МОДЕЛИРОВАНИЯ КОМПЬЮТЕРНЫХ АТАК НА ОСНОВЕ ВЕРОЯТНОСТЫХ АВТОМАТОВ*

В последние годы вопросы обеспечения безопасности информационных систем приобретают чрезвычайно большое значение. Одним из наиболее важных направлений данной деятельности является разработка методов и средств, которые позволяют обнаружить факт наличия в программном обеспечении ошибок или недокументированных возможностей, использование которых может привести к нарушению безопасности системы. Поиску уязвимостей на этапе разработки систем уделяется весьма значительное внимание. Существуют методики и программные средства, которые позволяют исключить ошибки использования языка или внешних интерфейсов (Microsoft Prefast, Intel PGO) и устранить значительное количество ошибок связанных с логикой обработки данных, реализацией протоколов и т.д. (PEACH и SPIKE). Однако в настоящее время не существует общепринятых формальных или автоматизированных методов оценки безопасности системы, кроме наиболее простых случаев. Для оценки безопасности систем используются методы penetration testing, которые предполагают экспертную оценку системы со стороны квалифицированных специалистов. Такие исследования обычно занимают значительное время и чрезвычайно дороги. Целью данной работы является разработка модели, которая предоставляет возможности описания атак в условиях динамически меняющейся внешней среды и обеспечивает возможности автоматизированного выполнения атак на целевые системы с целью оценки их безопасности. Решение данной задачи позволит не только снизить время и стоимость оценки, но и повысить точность обнаружения.

Разрабатываемая модель предназначена для исследования атаки как последовательности этапов атаки, которая приводит целевую систему в состояние, необходимо атакующему. Кроме того, исходя из условий, которые являются типичными для компьютерных атак, определим следующие ограничения разрабатываемой модели:

1. Атакующий имеет заданный набор этапов атаки. Этап атаки представляет собой законченное воздействие на атакуемую систему, результаты которого могут быть каким-либо образом оценены атакующим. Воздействие может быть как злонамеренным, например эксплуатацией некоторой уязвимости целевой системы, так и нейтральным, например передачей некоторых допустимых данных.

* Работа выполнена при поддержке гранта РФФИ №07-07-00138а.

2. В общем случае атакующий не имеет возможности отслеживать внутреннее состояние атакуемой системы.
3. Атакующий может воздействовать на систему, а также получать ответы системы на воздействия через каналы коммуникаций данной системы.

Приведенное определение и ограничения позволяют предположить, что для представления атаки приемлемой будет модель, на основе автоматов. Модели на основе различных типов автоматов являются наиболее распространенными, например они были использованы в работах [2-6].

Для упрощения начального описания модели предположим, что моделируемая идеальная система обладает следующими свойствами (далее описание модели будет расширено, чтобы соответствовать реальным системам :

1. Система является конечной, поскольку содержит конечный набор состояний.
2. Система является детерминированной, поскольку при некоторых заданных входных данных система может перейти только в одно заданное состояние.

Для описания предлагаемой модели обратимся к понятиям теории автоматов. Учитывая приведенные выше свойства системы опишем модель системы как детерминированный конечный автомат, т.е. $M = \{Q, \Sigma, Y, \delta, \lambda, q_0, F\}$, где Q – конечное *множество состояний* автомата, $Q = \{q_i\}$ $\|Q\| = N$, N – количество состояний автомата, Σ – допустимый *входной алфавит,* $\Sigma = \{a_i\}$ $\|\Sigma\| = L$, L – мощность алфавита, Y – допустимый *выходной алфавит,* $Y = \{y_i\}$ $\|Y\| = W$, W – мощность алфавита, δ – *функция переходов автомата,* т.е. отображение множества $Q \times \Sigma$ во множество подмножеств *P(Q)*, при этом $\delta : Q \times \Sigma \to Q$, λ – *функция выходов автомата,* т.е. отображение множества $Q \times \Sigma$ во множество подмножеств *P(Y)*, при этом $\lambda : Q \times \Sigma \to Y$, q_0 — *начальное состояние* автомата $q_0 \in Q$, F — множество *заключительных* состояний, таких что $F \subset Q$, при достижении одного из которых работа автомата прекращается.

Для полученной абстрактной автоматной модели выдвинем следующие условия.

Переходы межу состояниями автомата зависят от состояния автомата и входа автомата в каждом состоянии, т.е. переход между состояниями q_i и q_j представляется как:

$$q_{i+1} = \delta(q_i, a_i) \; \forall i \,, \tag{1}$$

где a_i – входной символ автомата. Таким образом, рассматриваемый автомат является автоматом с памятью. Это условие является корректным следствием Свойства 2.

Выход автомата вычисляется после перехода автомата в некоторое состояние и остается неизменным до следующего перехода, т.е. рассматриваемый автомат является автоматом Мура. Тогда:

$$y_i = \lambda(q_i) \ \forall i \ . \tag{2}$$

Данное условие является корректным следствием теоремы о эквивалентности автоматов Мили и Мура.

Возвращаясь к предметной области можно сформулировать следующее определение состояния моделирующего автомата: состояние автомата - это этап проведения атаки выраженный в общей форме, при этом множество состояний автомата представляет собой множество этапов атаки. Следовательно, заданная последовательность состояний автомата представляет собой атакующее воздействие. Данная модель предоставляет возможность моделировать атакующие воздействия, а также выполнять действия над множествами атак, например, выбирать оптимальные по некоторому критерию атаки, выполнять распознавание атак и т.д.

Однако прямая реализация подобной модели для большинства реальных систем невозможна вследствие того, что:

1. Информация о состоянии системы неполна.
2. Расчета функций переходов автомата сложен.
3. Аналитическое представление функций выходов автомата в общем случае невозможно.

Рассмотрим фактор неполноты информации о системе. Пусть множество доступных для контроля входных параметров – $a_i^{'}$, а множество неизвестных параметров – $a_i^{''}$. Тогда $a''= a \setminus a'$. Если $a_i^{''} = \varnothing$, то мы имеем всю информацию о входных данных некоторого состояния, однако в большинстве реальных случаев это не так. Следовательно, можем сформулировать следующее утверждение: существует множество $a_i^{'} \subseteq a$ такое, при котором сложно однозначно определить следующее состояние автомата, т.е. функция переходов $\delta(q_i, a'_i)$ становится недетерминированной, т.е.:

$$Q^{i+1} = \bigcup \delta'(Q^i, a'_i) , \tag{3}$$

где $Q^i = \{q_k\}$ – множество состояний автомата в момент времени *i*, $Q^{i+1} = \{q_l\}$ – множество состояний автомата в момент времени *i+1*. Необходимо отметить, что данный случай является более общим по отношению к приведенному в (1), поскольку если $a_i' = a_i$, то $q_{i+1} = Q^{i+1}$.

Доказана теорема о том, что детерминированные и недетерминированные автоматы являются эквивалентными, т.е. любой недетерминированный автомат можно представить как детерминированный с иным набором состояний [7]. Однако в данном случае переход к детерминированному автомату приведет к усложнению модели, поэтому в дальнейшем будем использовать недетерминированный автомат.

Как было показано выше, используемая модель представлена недетерминированным автоматом. При этоманалитическое или табличное представление функций перехода может быть невозможным. Причинами этого являются:

1. Большая мощность алфавита Σ. Алфавит входных состояний для моделируемой системы выражается в данных различного типа, в частности, сетевых пакетах, строковых данных, реакции системы в режиме графического интерфейса и т.д.

2. Сложность взаимосвязи между входами и состояниями автомата. Входные данные системы могут быть связаны с переходами состояний сложным образом, через применение нескольких табличных или аналитических преобразований.

Для решения данной задачи предлагается использовать аппроксимацию функции переходов, т.е. на основе конечного набора известных пар $\{a_j, q_{i+1}\}$ для каждого данного состояния q_i построить функцию $Q^{i+1} = \delta'(a_j)\ j = 1..M, M << L$, где $Q^{i+1} = \{q_k\}$ – поскольку автомат является недетерминированным. При этом функция $\delta'(\cdot)$ как видно, не биективна, но должна быть всюду определенной. Вопросы реализации данного метода рассмотрены далее.

Другой проблемой является сложность аналитического представления функции выходов автомата $\lambda(\cdot)$. Для рассматриваемого случая количество отображений входов автомата в выходы является чрезвычайно большим, а связи между входами и выходами зачастую неявными. Как и в предыдущем случае перейдем к приближению функции переходов для каждого данного состояния.

Предложенная модель позволяет решить проблемы моделирования атаки, однако обладает одним недостатком — модель является чрезвычайно сложной, поскольку каждое состояние характеризуется собственной функцией переходов и выходов, при этом изменение входных данных порождает свой набор состояний.

Для решения данной проблемы предлагается использовать модель общей памяти. Как показано в ряде работ, например, [8] состояние автомата можно представить моделью с общей памятью (shared memory) при определенных ограничениях. Модель общей памяти позволяет отражать и накапливать данные о текущем состоянии системы, на основе которых будут рассчитываться функции выходов автомата.

Пусть в общей памяти содержится подмножество данных, которое влияет только на выход данного состояния $P_i \subseteq a_i$. Такое предположение не нарушает принципов автоматной модели поскольку не затрагивает функций перехода между состояниями.

Кроме того, введем понятие кластеров состояний. Кластер состояний Q_n – это множество состояний, такое, что $\bigcup_{\forall n} Q_n = Q$. Выдвинем следующие требования к элементам кластера состояний:

1. Все состояния кластера отличаются только содержимым общей памяти и не отличающихся функциями переходов, т.е. $\forall q_i, q_j \in Q_n : \lambda(P_i) = \lambda(P_j)$.
2. Пусть для всех состояний кластера можно найти такие функции $F(a'_i)$ и $G(a'_i)$, что их суперпозиция будет приближать функции перехода с необходимой точностью, т.е. $\delta'(a_i') \approx F(a'_i) \circ G(a'_i)$. Тогда потребуем, чтобы $\forall q_i, q_j \in Q_n : F(a'_i) = F(a'_j)$.

Возвращаясь к предметной области можно сформулировать понятие кластера состояний следующим образом. Это набор состояний системы, в котором она реагирует на входные данные одинаковым образом. Учитывая специфику системы определим, что кластер состояний — это процесс эксплуатации некоторой одной уязвимости.

Как показано в предыдущем разделе функция переходов автомата представляется в виде суперпозиции функций $F(a_i^{'})$ и $G(a_i^{'})$, причем функция $F(a_i^{'})$ одинакова для всех состояний кластера.

Чтобы гарантировать корректное вычисление общей функции $\delta'(\cdot)$ используем для расчета сети функций радиального базиса (RBFN). Доказано, что [9] данный тип нейронной сети позволят приблизить любую гладкую функцию от входных аргументов. Кроме того, для данного вида нейронной сети результат может быть интерпретирован как вероятность возникновения некоторого выхода, что как будет показано далее, может использоваться при различных операциях с полученной моделью, в том числе, при оптимизации.

Учитывая вышеприведенное для расчета $G(a_i^{'})$, можно использовать простые правила на основе табличных расчетов. Таким образом, функция перехода автомата представляется в виде функции, реализуемой RBFN, которая одинакова для всех состояний данного кластера и простой табличной функции, которая может быть различна для каждого состояния.

Построение функции выходов на основе различных видов правил может быть приемлемо только для частных случаев, поэтому необходимо использовать аппроксимацию функции выходов.

С учетом приведенных выше соображений в данной работе предлагается вычисление $\lambda(\cdot)$ на основе многослойных нейронных сетей прямого распространения с гладкими активационными функциями (Multilayer Perceptrons, MLP). Доказано, что [10] данный тип нейронной сети позволяет приблизить любую гладкую функцию от входных аргументов. Недостатком данной сети является большое время обучения.

Рассмотрим функционирование полученной модели. При поступлении входных данных автомат переходит в новый кластер состояний, при этом изменяются переменные общей памяти.

Далее генерируется выход от данных общей памяти. Необходимо отметить, что кластер состояний представляет собой модель одного атакующего воздействия, например, эксплуатации уязвимости или передачи некоторых данных и т.д. Функция выходов кластера определяет как данное атакующее воздействие влияет на атакуемую систему. Таким образом, для каждого этапа атаки достаточно обучить одну нейронную сеть, обеспечивающую преобразованием данных общей памяти в выходы. Данная функция соответствует *описанию действия этапа атаки* на систему. Функция инвариантна к сценарию атаки, в котором используется этап. Создание данного преобразования на основе многослойных персептронов является наиболее сложной и ресурсоемкой задачей, однако выполняется один раз для каждого этапа атаки.

Функция переходов обеспечивает переход в множество кластеров состояния. В понятиях предметной области функция $F(a_i^{'})$ носит характер оценки выполнения данного атакующего воздействия, а функция $G(a_i^{'})$ определяет в какое состояние будет осуществляться переход при данной оценке атакующего воздействия. Функция $F(a_i^{'})$ также не зависит от текущего этапа атаки и является таким же атрибутом атакующего воздействия, как и функция $\lambda(\cdot)$. Расчет данной функции также представляется очень трудоемким, однако выполняется один раз для каждого атакующего воздействия.

Поскольку нейронная сеть обеспечивает детерминированный выход, то функция $G(a_i^{'})$ должна быть недетерминированной. В данной работе $G(a_i^{'})$ выбрана вероятностной функцией от выходов нейронной сети, но нет принципиальных причин, чтобы не использовать другую недетерминированную функцию. Эта функция учитывает *изменения в атаке*, и меняется на основе проверки локальных условий, например, «удалось ли получить управление?», «удалось ли расширить привилегии?» и т.д. Данная функция должна быть рассчитана индивидуально для каждого этапа атаки при создании каждого сценария, однако создание табличных соответствий может быть реализовано достаточно быстро. На рис. 1 показано графовое представление разработанного автомата. На рис. 2 показана подробная DFD при переходе из состояния в состояние.

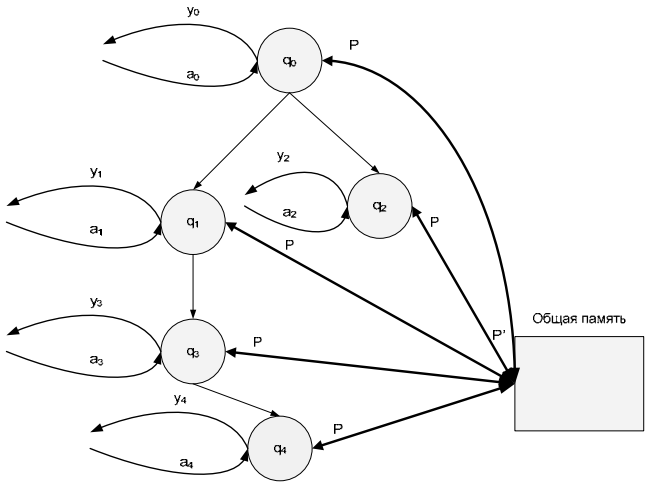

Рис. 1. Пример предлагаемой модели

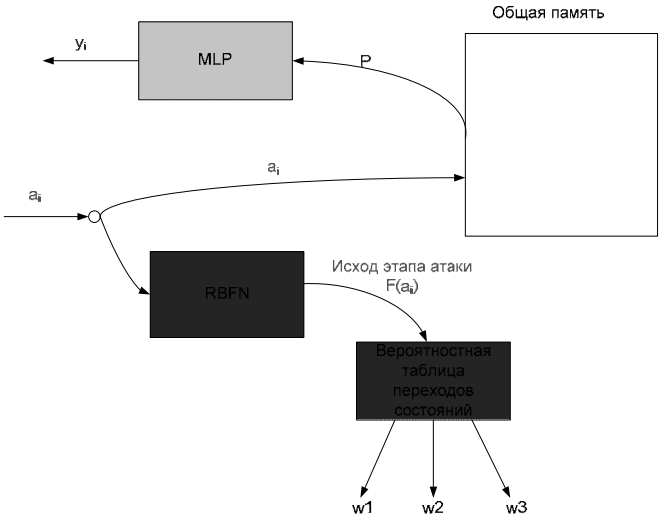

Рис. 2. DFD при переходе состояний автомата

Предложенная в работе модель позволяет выполнять поэтапное моделирование атаки для различных компьютерных систем. Одним из достоинств данной модели является то, что использование общей памяти не нарушает концепции автоматного моделирования, поскольку как уже отмечалось общая память является аргументом только для формирования функции выходов. Приведенные соображения позволяют предположить, что для данного типа автоматов может быть разработана алгебра, позволяющая производить над автоматами наборы действий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Schneier B.* Attack Trees [Электронный ресурс] /Brus Schneier // Dr. Dobb's Journal, 1999. Режим доступа: http://www.schneier.com/paper-attacktrees-ddj-ft.html.

2. *Camtepe, S.A.* A Formal Method for Attack Modeling and Detection [Электронный ресурс] /Seyit Ahmet Camtepe, Bulent Yener // TR-06-01, Rensselaer Polytechnic Institute, Computer Science Department. 2006. Режим доступа: http://citeseer.ist.psu.edu/751069.html.

3. *Sheyner, O.* Automated Generation and Analysis of Attack Graphs /Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, Jeannette M. Wing // Proceedings of the IEEE Symposium on Security and Privacy. Oakland, CA, USA, 2002. P. 273 – 284.

4. *Jha, S.* Two Formal Analyses of Attack Graphs /S. Jha , O. Sheyner, J. Wing// Proceedings of the 15th IEEE Computer Security Foundations Workshop. Nova Scotia, Canada, June 2002. P. 49-63.

5. *Sheyner, O.* AttackGraph Tool 0.5 [Электронный ресурс]. Режим доступа: http://www.cs.cmu.edu/~odobzins/scenariograph/as_files/AttackGraph-0.5.tar.gz

6. *Von Ohiemb, D.* Formal security analysis with Interacting state machines /David Von Ohiemb, Volkmar Lotz, Dieter Gollmann, Karjoth Günter, Michael Waidner// Lectute Neotes in Computer Science, 2002, № 2502. P. 212-228.

7. Хопкрофт Д. Введение в теорию автоматов, языков и вычислений / Д. Хопкрофт, Р. Мотивани, Д. Ульман // 2-е издание. – М.: Издательский дом «Вильямс», 2002. – 528 с.

8 *Rieke R.* Tool based formal Modelling, Analysis and Visualisation of Enterprise Network Vulnerabilities utilising Attack Graph Exploration [Электронный ресурс]/ *Roland Rieke//* In U.E. Gattiker (Ed.), EICAR 2004 Conference CD-rom: Best Paper Proceedings (ISBN:87-987271-6-8). 31 pages. Copenhagen: EICAR e.V.

9. Poggio, T. A theory of networks for approximation and learning /Poggio, T. Girosi, F.// Technical Report A.I. Memo 1140, Massachusetts Institute of Technology, Artificial Intelligence Laboratory and Center for Biological Information Processing, Whitaker College.

11. *M. H. Stone* (1937). Applications of the Theory of Boolean Rings to General Topology. Transactions of the American Mathematical Society 41 (3), 375–481.

УДК 681.3.06

Е.С. Абрамов, Д.В. Мордвин

ПРИМЕНЕНИЕ МОДЕЛИРОВАНИЯ ОБРАБОТКИ СЕТЕВОГО ТРАФИКА ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ ЛВС*

Современные средства противодействия атакам в локальных и глобальных сетях включают в себя межсетевые экраны (МЭ), средства обнаружения вторжений (IDS, intrusion detection system) и средства предотвращения вторжений (IPS, intrusion prevention system). На данном этапе развития данных технологий можно говорить об их взаимной интеграции и возможно окончательном объединении в рамках комплексных решений в будущем. Данные средства действительно позво-

* Работа выполнена при поддержке гранта РФФИ №07-07-00138а.