Polikarpov Vitaly Semionovich
Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.
E-mail: philosof@egf.tsure.ru.
44, Nekrasovskiy, Taganrog, 347928, Russia.
Phone: 8(8634)371-615.
Department of Philosophy.
Professor.

УДК 681.3.053:681.32

Е.П. Тумоян

РАЗРАБОТКА МЕТОДА МОДЕЛИРОВАНИЯ СЕТЕВЫХ АТАК НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ И ВЕРОЯТНОСТНЫХ ГРАФОВ

В статье предложена новая формальная модель представления сетевых атак на основе искусственных нейронных сетей и автоматов с вероятностными функциями перехода. Дано теоретическое обоснование предложенного метода с точки зрения теории графов. Предложенный метод позволяет проводить моделирование сетевых атак на реальные компьютерные системы.

Моделирование атак; стадия атаки; идеальная атакующая система; кластеры состояний.

E.P. Tumoyan

DEVELOPMENT OF NETWORK ATTACK MODELING METHOD BASED ON NEURAL NETWORKS AND PROBABILISTIC GRAPHS

In this paper, we proposed new formal model for network attack representation based on artificial neural networks and automata with probabilistic transition functions. A theoretical justification of the proposed method is given in terms of graph theory. The proposed method allows simulating network attacks in real computer systems.

Attack modeling; attack stage; theoretical attacking system; state clusters.

Введение

Одним из наиболее важных направлений обеспечения безопасности систем является разработка методов и средств, обнаружения недокументированных возможностей программного обеспечения, использование которых может привести к нарушению безопасности системы. Обнаружение уязвимостей является непрерывным процессом и включает в себя обнаружений уязвимостей на этапе разработки и эксплуатации системы. Поиску уязвимостей на этапе разработки систем уделяется значительное внимание. Существуют программные средства (Microsoft Prefast и др.), которые позволяют исключить ошибки использования языка или внешних интерфейсов, а также средства (Microsoft Application Verifier, PEACH и SPIKE), позволяющие устранить значительное

количество ошибок, связанных с логикой обработки данных и реализацией протоколов. Однако сложные информационные системы при эксплуатации проявляют существенно иные свойства, чем отдельные программные компоненты, из которых они состоят. К сожалению, в настоящее время не существует общепринятых автоматизированных методов оценки безопасности системы, кроме наиболее простых случаев. Наиболее часто используются методы penetration testing, которые предполагают экспертную оценку системы со стороны квалифицированных специалистов. Такие исследования занимают большое время и чрезвычайно дороги.

Целью данной работы является разработка модели, которая предоставляет возможности описания атак в условиях динамически меняющейся внешней среды и обеспечивает возможности автоматизированной разработки 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 уязвимостей.

1. Разрабатываемая модель

Разрабатываемая модель предназначена для исследования атаки, как последовательности этапов, которая приводит целевую систему в состояние, необходимое атакующему. Для моделирования системы будет использована модель на основе автоматов. Модели на основе различных типов автоматов применялись для решения сходных задач, например в работах [2-6]. Для упрощения начального описания модели предположим, что моделируемая *идеальная* система обладает следующими свойствами:

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. $M = \{Q, \Sigma, Y, \delta, \lambda, q_0, \Omega\}$, где Q – конечное *множество состояний* автомата, $Q = \{q_i\}$, Σ – допустимый *входной алфавит*, $\Sigma = \{a_i\}$, Y – допустимый *выходной алфавит*, δ *– функция переходов автомата*, λ *– функция выходов автомата*, q_0 *– начальное состояние* автомата, Ω – множество *заключительных* состояний, таких что $\Omega \subset Q$, при достижении одного из которых работа автомата прекращается. При этом выполняются следующие условия:

1) 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.

2) Выход автомата вычисляется после перехода автомата в некоторое состояние и остается неизменным до следующего перехода $y_i = \lambda(q_i)\ \forall i$.

В терминах предметной области состояние автомата - это этап проведения атаки, выраженный в общей форме. Следовательно, заданная последовательность состояний автомата представляет собой атакующее воздействие. Предложенная модель позволяет решить проблемы моделирования атаки, и подобна модели, предложенной, например, в работах [3, 4]. Однако модель обладает рядом недостатков:

1) Модель не учитывает неполноту информации о состоянии моделируемой системы. Неполнота возникает вследствие того, что атакующий не имеет возможности отслеживать внутреннее состояние атакуемой системы.
2) Модель сложна для анализа и расчета, поскольку содержит большое количество состояний.
3) Функции переходов данного автомата в общем случае может не иметь аналитического или табличного выражения.
4) Функции выходов автомата имеют чрезвычайно сложный вид, поскольку обеспечивают расчет выходных данных выраженных в различной форме (сетевые пакеты, строковые данные, операции над GUI атакуемой системы и т.д.)

2. Неполнота информации о моделируемой системе

Пусть множество доступных для контроля входных параметров a_i', $a' \subseteq a$. Тогда сложно однозначно определить следующее состояние автомата, т.е. функция переходов $\delta(q_i, a'_i)$ недетерминирована, 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*. Данный случай является более общим по отношению к приведенному в (1). Доказана теорема [7] о том, что детерминированные и недетерминированные автоматы являются эквивалентными, т.е. любой недетерминированный автомат можно представить как детерминированный с иным набором состояний. Однако в данном случае переход к детерминированному автомату приведет к усложнению модели, поэтому в дальнейшем будем использовать недетерминированный автомат.

3. Общая память и кластеры состояний

Для решения проблемы сложности модели предлагается использовать принцип общей памяти. Как показано в ряде работ, например, [8] состояние автомата можно представить моделью с общей памятью (shared memory) при определенных ограничениях. Пусть в общей памяти содержится подмножество данных $P_i \subseteq a_i$, которое влияет только на выход данного состояния. Такое

предположение не нарушает принципов автоматной модели, поскольку не затрагивает функций перехода между состояниями.

Кроме того введем понятие кластеров состояний. Кластер состояний X_n – это множество состояний, такое что $\bigcup_{\forall n} \mathrm{X}_n = Q$, при этом:

1) Все состояния кластера имеют одну функцию переходов, т.е. $\forall q_i, q_j \in \mathrm{X}_n : \lambda_i = \lambda_j$. Функцию переходов кластера состояний X_n назовем λ^n. Тогда $y_i = \lambda^n(P_i) \ \forall q_i \in \mathrm{X}_n$.
2) Для всех состояний кластера можно найти такие функции F и G, что $\delta'(a_i') \approx G(F(a'_i))$ и $\forall q_i, q_j \in X_n : F_i(a') = F_j(a')$.

Таким образом, кластер состояний – это множество состояний, в котором модель генерирует выходные данные и обрабатывает входные одинаковым образом.

3. Функции переходов автомата

Как было показано выше, используемая модель представлена недетерминированным автоматом. При этом функции переходов данного автомата в общем случае могут не иметь аналитического или табличного выражения. Для решения данной задачи предлагается использовать аппроксимацию функции переходов для каждого данного состояния q_i на основе конечного набора известных пар $\{a_i, Q_{i+1}\}$.

Как показано ранее, функция переходов может быть представлена в виде суперпозиции функций F и G, причем функция F одинакова для всех состояний кластера. Что бы гарантировать корректное вычисление общей функции δ, для представления F будут использованы сети функций радиального базиса (RBFN). Доказано [9, 10], что данный тип нейронной сети позволяет приблизить любую непрерывную функцию от входных аргументов. Функция G обеспечивает аппроксимацию разрывов функции δ. Для этого может быть использованы различные методы, но учитывая, что число возможных состояний конечно, можно использовать табличное преобразование.

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.

4. Функции выходов автомата

Другой проблемой является сложность расчета функции выходов автомата $\lambda(.)$. 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 выходов для каждого данного состояния. Построение функции выходов на основе различных видов правил может быть приемлемо только для частных случаев, поэтому необходимо использовать аппроксимацию функции выходов.

В данной работе предлагается приближение $\lambda(.)$ многослойными нейронными сетями прямого распространения с гладкими активационными функциями (Multilayer Perceptrons, MLP). Существует ряд теорем, доказывающих, что данный тип нейронной сети также является универсальным аппроксиматором. Сеть обучается на воспроизведение отображений вида $\{P_i, y_i\}$.

5. Функционирование модели

Рассмотрим функционирование полученной модели. Автомат начинает функционирование с некоторого начального состояния q_0. При переходе в данной состояние автомат генерирует выходные данные на основе общей памяти. 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.

При получении входных данных автомат изменяет переменные, находящиеся в общей памяти и переходит в новое множество кластеров состояния в соответствии с функцией переходов. В понятиях предметной области функция F носит характер оценки выполнения данного атакующего воздействия, не зависит от текущего этапа атаки и является таким же атрибутом атакующего воздействия, как и функция λ. Расчет данной функции также представляется очень трудоемким, однако выполняется один раз для каждого атакующего воздействия. Функция G 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 чрезвычайно быстро.

Функционирование автомата продолжается до тех пор, пока не будет достигнуто одно из терминальных состояний, соответствующее состоянию атакуемой системы, которое необходимо атакующему.

Заключение

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 разработаны методы, позволяющие производить над автоматами наборы действий, в том числе оптимизацию обхода графа пред-

ставляющего автомат. Разработка таких методов позволит разработать эффективные методы автоматического построения атак в целях тестирования безопасности систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Schneier B.* Attack Trees // Dr. Dobb's Journal. 1999. URL: http://www.schneier.com/paper-attacktrees-ddj-ft.html (дата обращения: 20.12.2009).
2. *Camtepe S.A.* A Formal Method for Attack Modeling and Detection / *S.A. Camtepe, B. Yener* // TR-06-01, Rensselaer Polytechnic Institute, Computer Science Department. 2006. URL: http://citeseer.ist.psu.edu/751069.html (дата обращения: 20.12.2009).
3. *Sheyner O.,* Automated Generation and Analysis of Attack Graphs / *O. Sheyner, J. Haines, S. Jha, R. Lippmann, J.M. Wing* // Proceedings of the IEEE Symposium on Security and Privacy. – Oakland, USA, 2002. – P. 273–284.
4. *Jha S., Sheyner O., Wing J.* Two Formal Analyses of Attack Graphs /// 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 (дата обращения: 20.12.2009).
6. *Von Ohiemb D., Lotz V., Gollmann D., Günter K., Waidner M.* Formal security analysis with Interacting state machines // Lectute Notes in Computer Science. – 2002. – № 2502. – P. 212-228.
7. *Хопкрофт Д., Мотивани Р., Ульман Д.* Введение в теорию автоматов, языков и вычислений. – М.: Издательский дом «Вильямс», 2002. – 528 с.
8. *Rieke R.* Tool based formal Modelling, Analysis and Visualisation of Enterprise Network Vulnerabilities utilizing Attack Graph Exploration [Электронный ресурс]// EICAR 2004 Conference CD-ROM: Best Paper Proceedings.
9. *Poggio T., Girosi F.*A theory of networks for approximation and learning [Электронный ресурс]. – Режим доступа: ftp://publications.ai.mit.edu/ai-publications/1000-1499/AIM-1140.ps.Z (дата обращения: 20.12.2009).
10. *Хайкин С.* Нейронные сети: полный курс. – М.: Издательский дом «Вильямс», 2006. – 1104 с.

Тумоян Евгений Петрович

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: ugenie@tumoyan.ru.

347928, г. Таганрог, ул. Чехова, 2.

Тел.: 8(8634)371-905.

Кафедра безопасности информационных технологий.

Доцент.

Tumoyan Evgeny Petrovich

Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: ugenie@tumoyan.ru.

2, Chechov street, Taganrog, 347928, Russia.

Phone: 8(8634)371-905.

Department of Security in Data Processing Technologies.

Associate professor.