Секция безопасности информационных технологий

УДК 681.3.053:681.32

Е.П. Тумоян

МЕТОДЫ ФОРМАЛЬНОГО МОДЕЛИРОВАНИЯ СЕТЕВЫХ АТАК

Введение

В последние годы вопросы защиты информационных систем от атак приобретают чрезвычайно большое значение. Одним из наиболее важных направлений обеспечения такой защиты является разработка методов и средств, которые позволяют обнаружить факт начала или проведения атаки. К этому числу подобных средств относятся сетевые и хостовые системы обнаружения атак.

Существующие системы обнаружения атак основаны на сигнатурных методах или статистических методах распознавания вредоносных данных. Перспективные разработки предполагают использование интеллектуальных механизмов для обнаружения атак, например, нейронных сетей, скрытых марковских моделей и нечетких логических правил. Однако при разработке подобных методов мало внимания уделяется исследованию последовательности самих атакующих воздействий, в том числе их анализу и моделированию.

Целью данной работы является разработка модели, которая предоставляет возможности описания атак в динамически меняющихся условий внешней среды, а также метода распознавания атак на основе построенной модели. Построение адекватной и стабильной модели некоторого класса атак позволит существенно повысить точность обнаружения атак, особенно в условиях, когда атакующее воздействие маскируется преднамеренно.

1. Существующие модели атак

В настоящее время существует ряд моделей, позволяющих с разной степенью детализации описать процесс сетевой атаки. Модели используют разную математическую базу, но большинство из них основаны на конечных автоматах и представляют атаку как последовательность состояний автомата.

Этапная модель. Этапная модель предполагает, что атака состоит из нескольких изолированных этапов: этапа исследования, этапа получения управления, этапа расширения привилегий, этапа осуществления вредоносных действий, этап создания потайного хода, этапа заметания следов.

Между этапами могут проходить значительные промежутки времени. Часть этапов могут быть опущены. Данная модель является наиболее распространенной, однако, обладает существенными недостатками:

1. Модель не учитывает и не обеспечивает средства по моделированию взаимосвязи между этапами.
2. Модель не предоставляет средств для оценки успешности этапа.
3. Модель не предоставляет средств для уточнения наличия или отсутствия некоторых этапов.

Деревья атак. Модель предложена Б. Шнайером в работе [1] в 1999 году. Деревья атаки представляют собой концептуальные диаграммы, которые описывают угрозы системе и возможные атаки, направленные на их реализацию. Модель пригодна для описания атак на любые системы информационного или физического характера. Модель предоставляет возможность для введения оценок каждого шага по некоторым критериям, например, по времени выполнения, числу операций, оценочной стоимости и т.д. При этом последовательность шагов может быть оценена на основании критериев для каждого шага. В работе [1] приведены примеры оценки деревьев на основе двух критериев, однако, нет принципиальных препятствий для разработки алгоритма многокритериальной оценки.

Данная модель предоставляет большую степень детализации и точности оценок, чем предыдущая. Однако модель не может быть использована для моделирования атак, поскольку не обеспечивает средств динамического моделирования, а также средств для включения в модель условий внешней среды.

Улучшенные деревья атак. Модель описана в работе [2] в 2006 году. Модель опирается на использование деревьев и механизма конечных автоматов для моделирования уязвимостей протоколов и систем. Таким образом, модель представляет собой расширение и уточнение модели деревьев атак. Авторами приводится пример анализа безопасности протокола IEEE 802.11. Данная модель имеет те же недостатки, что и предыдущая.

Графы атак. Модель предложена коллективом исследователей в работе [3] в 2002 году. Видимо, модель графов атак основана на расширении модели деревьев атак. Однако графы атак:

1. Являются специализированным средством для описания сетевых атак.
2. Для описания используется граф, а не дерево.
3. Узлы графа представляют не концептуальные действия, а узлы сети, процессы программы, конфигурационные файлы, участки кода, переменные и т.д.

Моделирование систем на основе графов атак основано на конечных автоматах. Переходы между узлами осуществляются на основе применения детерминированных правил. При этом может учитываться текущее значение некоторых параметров системы, переменных и т.д. Целью является достижение необходимой вершины. Модель может предусматривать анализ условий, необходимых для достижения цели атаки, поэтому модель может использоваться для оценки безопасности программной системы или компьютерной сети. Модель получила широкое распространение [3,4], поскольку основана на простой и хорошо исследованной математической базе (конечные автоматы), сама достаточно проста и очевидна. Данная модель была реализована в нескольких инструментальных средствах, например в Lockheed Martin ANGI, а также средстве AttackGraph Tool 0.5 [5].

Недостатком данной модели является то, что, как и в предыдущем случае, она является средством для оценки сложности нарушения безопасности информационной системы, а не моделирования и исследования атак. Другим существенным недостатком данной модели является применение аппарата конечных автоматов. Модель реальной информационной системы в большинстве случаев не ограничивается моделью автоматов, поскольку включает плохоконтролируемые или скрытые факторы.

Interacting State Machines. Модель предложена коллективом исследователей в работе [6] в 2002 году. Модель основана на применении специального типа высокоуровневых конечных автоматов Interacting State Machines (ISM) для моделирования сложных систем. В данной работе ISM применяются для моделирования

атакуемого протокола с целью обнаружения ошибок, приводящих к уязвимостям системы, т.е. решаются задачи, существенно отличные от моделирования атак.

2. Предлагаемая модель

Опишем предлагаемую модель следующим образом. Пусть существует множество состояний системы $Q = \{q_i\}, \quad i = 0..N$. Тогда представим переход между q_i и q_j как $q_i \to q_j : \tau(q_i, q_j) = f(P(q_i), \overline{Q}(q_j)), \quad \forall i, j$, где $P(q_i)$ – параметры состояния q_i, $\overline{Q}(q_j)$ – в общем случае упорядоченное множество состояний, предшествовавших q_j . Далее сделаем следующие предположения:

1. Пусть всю информацию о состоянии q_i можно представить в виде $P(q_i)$. Тогда $\overline{Q}(q_j)$ содержит информацию только о процессе перехода из состояния q_0 в состояние q_j .

2. Пусть можно найти такие функции $g(.)$ и $h(.)$, что $f(P(q_i), \overline{Q}(q_j)) = g(P(q_i)) + h(\overline{Q}(q_j)), \quad \forall i, j$. Таким образом, $g(.)$ можно рассматривать как модель, характеризующую переходы из состояния в состояние, а $h(.)$ – модель, характеризующую историю системы.

3. Для того, чтобы учесть изменения, вносимые переходом системы в некоторое состояние q_i , введем понятие *влияние состояния*. Влияние состояния $A(q_i)$ – это изменение, вносимое в систему переходом системы в данное состояние, в процессе пребывания системы в данном состоянии или в результате выхода из данного состояния. При этом $P'(q_i) = P(q_i) + A(q_i)$ и $P(q_{i+1}) = P'(q_i) + H'(q_i)$, где $A(q_i)$ – влияние состояния, $H'(q_i)$ – изменение, вносимые внешними факторами. Учитывая приведенные определения, пример модели из пяти состояний и шести переходов, можно изобразить, как показано на рис. 1.

Можно предложить различные варианты выбора $g(.)$ и $h(.)$. Для определенности в данной работе предлагается следующее:

1. Функция $g(.)$ представляют собой вероятностные нейронные сети (Probabilistic Neural Networks, PNN) в режиме классификации. Данный тип нейронных сетей позволяет проводить классификацию данных на основе оценки вероятности принадлежности данных к некоторому классу. Как и другие типы нейронных сетей PNN должны быть обучены на классификацию данных, однако, время обучения PNN чрезвычайно мало.
2. Значение $A(q_i)$ вычисляется на основе многослойных нейронных сетей прямого распространения с гладкими активационными функциями (Multilayer Preceptrons, MLP). Данный тип нейронной сети позволяет приблизить любую гладкую функцию от входных аргументов. Недостатком данной сети является большое время обучения.
3. Функция $h(.)$ представляет собой расчет вероятностей на основе Марковских моделей (Markov models). Данный метод позволяет быстро построить долговременную вероятностную модель системы и оценить вероятность некоторого состояния. Другим возможным способом представле-

ния $h(.)$ могут стать вероятностные суффиксные деревья (Probabilistic suffix trees, PST), которые обеспечивают подобные возможности.

Возвращаясь к задаче моделирования атак, опишем приведенные величины.

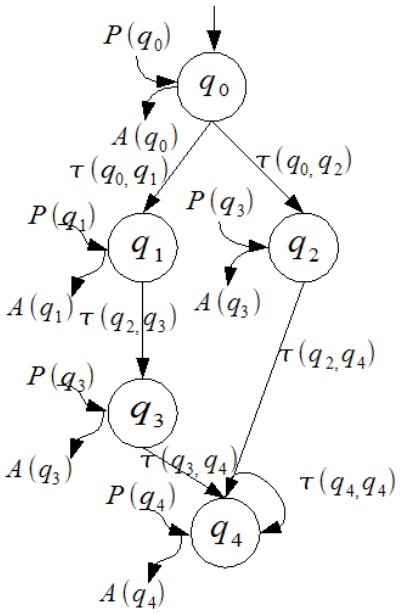

Рис. 1. Пример предлагаемой модели

Состояние представляет собой этап атаки. Формальный признак отдельного этапа атаки – законченное воздействие на атакуемую систему. Пример – переполнение буфера для захвата управления, расширение привилегий и т.д. Воздействие состояния – влияние перехода в некоторое состояние на переменные системы. В целях упрощения расчета при описании воздействия состояния имеет смысл использовать модель общей памяти, т.е. состояние системы представляется в виде глобального пула переменных, который модифицируется атакующими воздействиями в ходе атаки. Переход представляет собой вероятностную взаимосвязь между различными вариантами атаки.

Функция $g(.)$ обеспечивает получение номера и вероятности следующего состояния по переменным системы. Эта функция учитывает *кратковременные 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, как отмечалось ранее, вероятностные нейронные сети (PNN) допускают чрезвычайно быстрое обучение. $A(q_i)$ – предложенный способ вычисления $A(q_i)$ позволяет абстрагироваться от фактической реализации атаки, которая может быть существенно различной (например, применение набора инструкций процессора, команд интерпретируемого языка, получение вредоносного пакета и т.д.) и оценивать изменения в различных типах переменных системы. 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 $h(.)$ учитывает *долговременные изменения* в ходе атаки на основе вероятностных параметров. Данная функция должна быть рассчитана для каждого сценария атаки. При этом выбранные методы (марковские модели и суффиксные деревья) позволяют производить построение данной функции чрезвычайно быстро.

Заключение

Предложенная в работе модель обеспечивает получение описания сценариев различных атак с использованием общей понятийной базы и общего метода моделирования. Как было упомянуто ранее, модель обеспечивает учет параметров, ко-

торые динамически меняются в ходе выполнения сценария атаки, а также параметров, которые изменяются самим атакующим воздействием.

Дальнейшим продолжением данной работы станет разработка методов выполнения преобразований для предложенной модели, например, сложения, вычитания, упрощения, сравнения и т.д. Разработка таких методов позволит построить эффективные средства обнаружения атак как ранее известных, так и новых. Основной проблемой при этом является сбор данных о сценарии проведения атаки в виде, пригодном для использования в данной модели. В настоящее время базы данных атак, содержащие такие данные, в открытом доступе не представлены. Однако существует ряд проектов по унификации описания и сбору унифицированных данных атак, например, Common Weakness Enumeration. Развитие таких проектов, видимо, позволит получить необходимую информацию о проведении атак.

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.

УДК 681.3.06

А.В. Благодаренко

ПОСТОРОЕНИЕ МАРШРУТОВ ВЫПОЛНЕНИЯ ПО БЕЗ ИСХОДНЫХ КОДОВ*

Существуют ситуации, когда анализ ПО на наличие уязвимостей предпочтительнее делать по исполняемому коду [1,2]. Для этого необходимо провести ряд процедур, в том числе и построение маршрутов выполнения.

Под маршрутом понимается последовательность команд, выполняющихся при отработке какого-либо действия, входящего в функциональность программы. Например, если ПО обрабатывает файл определенного формата, то можно говорить о маршрутах, выполняемых при его разборке и выполнении. Для удобства последующего анализа, выявленный маршрут должен быть представлен на фоне общей картины ПО. Так, если исследование производится с использованием исходных кодов, можно показывать принадлежность участка кода к структурной единице (функции, модулю, классу). Такая функциональность требует проведения

* Работа поддержана грантами РФФИ № 07-07-00138, №06-07-89010.