3) $M_1 \to M_2$: $E_{M_2}(E_A E_B(m), r_2)$,
$S_A(m, t_A), t_A, S_B(r_0, t_b), t_B, S_{M_1}(r_1, t_{M_1}), t_{M_1}$.

4) $M_2 \to A$: $E_A E_B(m)$,
$S_A(m, t_A), t_A, S_B(r_0, t_b), t_B, S_{M_1}(r_1, t_{M_1}), t_{M_1}, S_{M_2}(r_2, t_{M_2}), t_{M_2}$.

5) Проверка: $t_{M_2} =_w t_{M_1}$, $t_{M_1} =_w t_B$, $t_B =_w t_A$. Если проверки пройдены:
$A \to B$: $E_B(m)$,
$S_A(m, t_A), t_A, S_B(r_0, t_b), t_B, S_{M_1}(r_1, t_{M_1}), t_{M_1}, S_{M_2}(r_2, t_{M_2}), t_{M_2}$.

Описанный протокол может использоваться как протокол транспортного уровня в распределённой сети соревнования CTF следующим образом. Команда организаторов, согласно протоколу, осуществляет новостную рассылку во время соревнования. Если участник хочет получать новостную рассылку от организаторов, ему необходимо выступать в качестве посредника при передаче сообщений других участников, выставляя на них корректную метку времени. В противном случае участник не сможет получать новостную рассылку, что равносильно отключению от соревнования и техническому поражению.

Каждый участник при получении ответа на задание должен доказать этот факт, отправив сообщение организаторам. Засвидетельствовать факт получения ответа в текущий момент времени можно с помощью модифицированного протокола, т. е. предполагая, что команда организаторов в данный момент может быть недоступна. Команды-соперники, выбранные в качестве посредников с помощью функции $H(m)$, подтвердят факт создания сообщения в текущий момент времени. Итоговое сообщение вместе с метками времени может быть отправлено организаторам позже. По окончании соревнования организаторы смогут сформировать таблицу результатов с учётом времени получения ответов.

ЛИТЕРАТУРА

1. *Анисеня Н. И.* Разработка безопасного протокола распределённой системы проведения соревнований CTF // Прикладная дискретная математика. 2015. № 2(28). С. 59–70.

УДК 519.7 DOI 10.17223/2226308X/9/31

О ДИСКРЕТНО-АВТОМАТНЫХ МОДЕЛЯХ АТАК В КОМПЬЮТЕРНЫХ СЕТЯХ[1]

Д. Е. Горбатенко, С. Е. Кочемазов, А. А. Семёнов

Предлагается новая модель развития атак в компьютерных сетях. Основу модели составляет дискретный автомат синхронного действия, задаваемый графом сети. Рассматриваются переходы между состояниями данного автомата, совершаемые в дискретные моменты времени. Вершинам графа (интерпретирующим хосты сети) в каждый момент времени приписываются двоичные векторы, называемые состояниями хостов. В каждый следующий момент все состояния хостов синхронно пересчитываются по фиксированным правилам. В рамках предложенной модели проанализировано развитие некоторых известных типов атак. Изучены возможности атак с нескольких хостов, а также описана техника предотвращения атак посредством решения комбинаторной задачи расстановки патчей. В вычислительных экспериментах рассматривались сети, сгенерированные случайным образом. Перечисленные выше задачи для этих сетей решались за счёт их сведения к проблеме булевой выполнимости.

[1]Работа выполнена при частичной поддержке РФФИ, проекты № 14-07-00403а и 15-07-07891а.

Ключевые слова: *дискретный автомат, граф атак, задача булевой выполнимости, SAT.*

Исследование атак в компьютерных сетях и способов защиты от них является актуальной и интенсивно развивающейся областью компьютерной безопасности. Для анализа сети на предмет возможности реализации в ней той или иной атаки, а также для определения способов защиты от атак используются формальные модели компьютерных сетей [1]. Обычно основой такой модели является ориентированный помеченный граф $G = (V, A)$, в котором V — множество вершин, A — множество дуг. Вершины графа G соответствуют хостам рассматриваемой сети, а дуги интерпретируют связи между хостами. В последние 10 лет весьма широкое распространение получил подход к анализу атак в сетях, базирующийся на графах атак [2 – 4]. Есть большое число различных подходов к построению графов атак, описание некоторых способов может быть найдено в [4]. В настоящей работе предлагается рассматривать развитие атаки в компьютерной сети как эволюцию некоторого дискретного автомата, происходящую в моменты времени $t \in \{0, 1, \ldots\}$. Этот подход имеет в сравнении с известными по меньшей мере два преимущества. Во-первых, он позволяет в простой и естественной форме ставить комбинаторные задачи, связанные с развитием и блокированием атак. Во-вторых, эти задачи допускают применение современных эффективных комбинаторных алгоритмов. Для получения представленных в работе результатов в роли таковых использованы алгоритмы решения проблемы булевой выполнимости (SAT). Далее кратко остановимся на основных полученных к настоящему моменту результатах.

Итак, рассматриваем граф $G = (V, A)$, интерпретирующий компьютерную сеть. Пусть $Q = \{q_1, \ldots, q_l\}$ — множество уязвимостей хостов, возможных в рамках рассматриваемой сети. Процесс развития атаки в сети — это процесс эксплуатации злоумышленником доступных уязвимостей, результатом чего в следующий момент времени является возможность эксплуатировать другие уязвимости. Соответственно можно рассмотреть сеть G как дискретный автомат, функционирующий в моменты времени $t \in \{0, 1, \ldots\}$, момент $t = 0$ назовём начальным. Для каждого t свяжем с произвольным хостом $v \in V$ булев вектор

$$\alpha^v(t) = \left(\alpha_1^v, \ldots, \alpha_l^v, \alpha_{l+1}^v(t), \ldots, \alpha_r^v(t)\right), r \geqslant l,$$

называемый состоянием хоста v в момент t. Первые l координат вектора $\alpha^v(t)$ образуют вектор, называемый вектором уязвимостей. Он интерпретируется в следующем смысле: $\alpha_j^v = 1$ тогда и только тогда, когда на хосте v имеется уязвимость q_j, $j \in \{1, \ldots, l\}$. Вектор уязвимостей хоста во все моменты времени остаётся неизменным. В координатах $\alpha^v(t)$ с номерами $l + 1, \ldots, r$ находится информация, которая может меняться с течением времени. В частности, в этих координатах отображается доступ с рассматриваемого узла на другие узлы сети. Вектор $\left(\alpha_{l+1}^v(t), \ldots, \alpha_r^v(t)\right)$ будем называть вектором возможностей хоста v в момент t.

Полагая, что в начальный момент для всех хостов сети заданы их состояния (например, все векторы возможностей при $t = 0$ могут быть нулевыми), можем рассматривать переходы сети в последующие состояния, изменяя векторы возможностей хостов в каждый момент в соответствии с фиксированными правилами. В качестве правил использованы пары вида (предусловие, постусловие), посредством которых в работе [5] определяются элементарные атаки. В рамках предлагаемой модели состояния для всех хостов пересчитываются синхронно. В этом случае по аналогии с синхронными булевыми сетями (SBN, они же сети Кауффмана [6]) функционирование автомата,

задаваемого графом G, порождает граф переходов (STG, State Transition Graph), обозначаемый через Γ_G. Каждой вершине графа Γ_G соответствует набор состояний всех хостов сети в рассматриваемый момент времени.

Поясним сказанное на ставшем хрестоматийным примере, впервые рассмотренном в работе [2]. Рассматривается сеть, состоящая из трёх хостов H_1, H_2, H_3. Состояние каждого хоста в момент времени $t \in \{0, 1, \ldots\}$ — это булев вектор из 9 координат. Первые четыре координаты соответствуют уязвимостям «write-noauth», «trust-login», «pe-noauth», «pe-local». Смысл этих уязвимостей описан в [5] (например, уязвимость «pe-noauth» на хосте v дает получившему доступ на v права суперпользователя на данном хосте). Пятая координата отвечает за активность опции «trust between the host and all». Активность данной опции на конкретном хосте H означает, что этот хост доверяет любому другому хосту сети. Если нарушитель может активировать данную опцию на H, то на следующем шаге он может подключиться к H без использования логина и пароля. Остальные координаты определяют права, которые по умолчанию имеет рассматриваемый хост на остальные узлы сети. На рис. 1 приведён фрагмент графа состояний дискретного автомата, задаваемого (в указанном выше смысле) сетью из работы [2]. Данный фрагмент соответствует одной атаке на сеть, рассмотренной в [5].

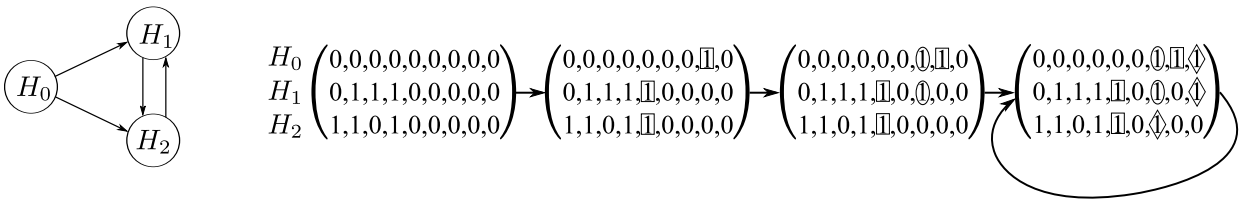

Рис. 1. Сеть из трёх хостов (слева) и фрагмент графа состояний автомата (STG), заданного этой сетью, интерпретирующий атаку, рассмотренную в [5]

Фрагмент графа состояний, приведённый на рис. 1, показывает два возможных пути развития атаки. В первом случае злоумышленник непосредственно со своего компьютера (хост H_0) проводит элементарную атаку, эксплуатирующую уязвимость «write-noauth» на хосте H_2, тем самым активируя на этом хосте опцию «trust between the host and all». Данная опция устанавливает доверительные отношения хоста H_2 со всеми остальными хостами в сети (на STG биты, отвечающие за включение данной опции, выделены квадратами). На следующем шаге, пользуясь полученной возможностью, злоумышленник получает права доступа пользователя на хосте H_2, эксплуатируя уязвимость «trust-login» (биты, соответствующие появлению прав пользователя у злоумышленника, выделены кругами). Далее, эксплуатируя уязвимость «pe-local» на хосте H_2, злоумышленник повышает свои права доступа до прав доступа суперпользователя (соответствующие биты выделены ромбами).

На фрагменте STG на рис. 1 виден также второй вариант получения злоумышленником прав суперпользователя на хосте 2. На первом шаге эксплуатируются уязвимости «write-noauth» и «pe-noauth» на хосте H_1, результат: доверие хоста H_1 всем остальным хостам сети и права суперпользователя у злоумышленника на хосте H_1. Далее злоумышленник проводит атаку на хост H_2 с хоста H_1, используя действия, аналогичные первому сценарию: последовательно эксплуатируются уязвимости «trust-login» и «pe-local» хостом H_1 на хосте H_2 в моменты времени $t = 2$ и 3 соответственно, благодаря чему хост H_1 получает права доступа суперпользователя на хосте H_2.

В рамках описанной модели мы рассмотрели ряд задач, связанных с анализом и блокированием атак в компьютерных сетях. В частности, рассмотрены задачи выбора злоумышленником нескольких хостов для атаки. При этом на возможности злоумышленника накладывались различные ограничения: например, требовалось получить root-право на некотором компьютере сети не более чем за фиксированное число моментов времени. Дополнительно предполагалось, что в процессе атаки злоумышленник не может в каждый момент времени иметь root-права более чем на заданном числе хостов сети. Такого рода задачи «подбора множеств хостов» являются комбинаторными из-за значительного в общем случае числа различных альтернатив, требующих проверки. Задачи описанного типа решались за счёт их сведения к задаче о булевой выполнимости (SAT). При этом использованы кодировки и общие идеи, представленные в работе [7], в которой методами SAT исследована активационная динамика в сетях. Если удавалось подобрать множество хостов, с которого злоумышленник успешно атаковал рассматриваемую систему, то для этой ситуации рассматривалась обратная задача: запретить те или иные уязвимости на некоторых хостах сети, чтобы в результате найденная атака стала невозможной. Эту задачу М. Данфорт называет задачей расстановки патчей. В рамках развитого вычислительного аппарата можно накладывать ограничения на число расставляемых патчей и их вид (например, предполагать, что блокирование некоторых уязвимостей невозможно). Все вычислительные эксперименты проводились на сетях, сгенерированных случайным образом в соответствии с моделью Барабаши — Альберт [8]. Перечисленные комбинаторные задачи удалось успешно решить для сетей с несколькими сотнями хостов.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Модели безопасности компьютерных систем: учеб. пособие для вузов. М.: Издательский центр «Академия», 2005.
2. *Jha S., Sheyner O., and Wing J.* Two formal analysis of attack graphs // Proc. 15th IEEE Workshop on Computer Security Foundations (CSFW '02). 2002. P. 49–63.
3. *Sheyner O., Haines J. W., Jha S., et al.* Automated generation and analysis of attack graphs // Proc. 2002 IEEE Symposium on Security and Privacy. 2002. P. 273–284.
4. *Колегов Д. Н.* Проблемы синтеза и анализа графов атак. `http://www.securitylab.ru/contest/299868.php`. 2009.
5. *Danforth M.* Models for Threat Assessment in Networks. PhD Thesis, University of California-Davis, 2006.
6. *Kauffman S.* Metabolic stability and epigenesis in randomly constructed genetic nets // J. Theoretical Biology. 1969. No. 22. P. 437–467.
7. *Kochemazov S. and Semenov A.* Using synchronous Boolean networks to model several phenomena of collective behavior // PLoS ONE. 2014. No. 9: e115156. P. 1–28.
8. *Barabasi A-L. and Albert R.* Emergence of scaling in random networks // Science. 1999. No. 286. P. 509–512.

УДК 004.94 DOI 10.17223/2226308X/9/32

О РЕЗУЛЬТАТАХ ФОРМИРОВАНИЯ ИЕРАРХИЧЕСКОГО ПРЕДСТАВЛЕНИЯ МРОСЛ ДП-МОДЕЛИ

П. Н. Девянин

«Монолитное» представление мандатной сущностно-ролевой ДП-модели, являющееся основой механизма управления доступом в отечественной защищённой опе-