Шаг 4. На последнем шаге теста, в случае аварийного завершения исследуемого процесса, восстанавливается корректность его последующих запусков для независимого выполнения очередного теста.

Применение стрессового тестирования позволяет обнаружить программные ошибки за короткое время.

Первоначально входные данные для тестирования формировались в статичном режиме, на основе различных эвристических правил и без восстановления алгоритмов работы исследуемого ПО.

Для повышения эффективности тестирования было решено сочетать стрессовое тестирование и динамический анализ на основе трассировки исследуемого процесса. Трассировка помогает сгенерировать входные данные.

При этом возникают две взаимосвязанные задачи. Первая — сопоставление входных данных и результатов трассировки, для чего строится граф потока данных.

Вторая задача — анализ собранной трассы, при этом можно выделить различные подзадачи: обнаружение функций, «схлопывание» циклов, минимизация объемов хранимых данных (поскольку объемы трассы измеряются гигабайтами) и другие. Для решения данной задачи предполагается использовать возможности среды *Ida Pro*. Интеграция со средой *Ida Pro* позволит минимизировать объем трассы и решить ряд подзадач с помощью штатных средств дизассемблера.

Таким образом, выявление программных ошибок в ПО без исходных текстов путем дополнительного тестирования механизмов обработки входных данных позволяет повысить надежность разрабатываемых программных комплексов.

ЛИТЕРАТУРА

1. *Козиол Д., Личфилд Д., Эйтэл Д., и др.* Искусство взлома и защиты системы. СПб.: Питер, 2006. 416 с.
2. *Ховард М., Лебланк Д.* Защищенный код. 2-е изд. М.: Издательско-торговый дом «Русская редакция», 2005. 704 с.
3. *Макаров А. Н.* Метод автоматизированного поиска программных ошибок // Безопасность информационных технологий. Вып. 2. М.: МИФИ, 2008. С. 101–104.
4. *Хогланд Г., Мак-Гроу Г.* Взлом программного обеспечения: анализ и использование кода. М.: Издательский дом «Вильямс», 2005. 400 с.
5. *Eilam E.* Reversing: Secrets of Reverse Engineering. Wiley Publishing, 2005. 589 p.

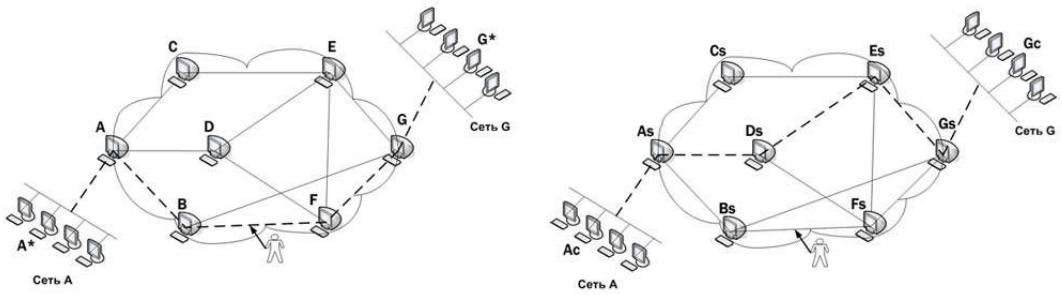
УДК 004.738

МАРШРУТИЗИРУЕМЫЙ СЕРВИС ПЕРЕДАЧИ ДАННЫХ

В. И. Никонов

Настоящая работа продолжает исследование [1], посвященное разработке алгоритмов разделения данных в распределенных сетях. Этот метод выступает в качестве альтернативы снижению вычислительных затрат при использовании шифрования.

Одним из видов активных сетевых атак является класс атак, основанных на снифинге [2]. Приведем пример, в котором злоумышленник, обладая знаниями, что некоторая организация регулярно передает данные из A в G, может довольно точно определить маршрут от A до G в момент времени Δt и осуществить перехват на каком-нибудь из участков следования трафика (см. рис. 1,a).



а *б*

Рис. 1. Работа протоколов маршрутизации между A и G в момент Δt. Вариант возможной атаки на участке $B - F$ (*а*). Изменение маршрута трафика за счет использования доверенных серверов D_S,E_S (*б*)

Разработан маршрутизируемый сервис S_M передачи данных через распределенные сети. S_M — клиент-серверное приложение, позволяющее пользователю передавать данные специфичным маршрутом. Характер маршрута определяется базой критериев S_M. В данной статье приведено описание работы, посвященной исследованию критерия безопасности передачи.

В роли маршрутизаторов для S_M выступает некоторое множество доверенных серверов распределенной сети. Под доверенным сервером будем понимать некоторый многофункциональный сервер распределенной сети, к которому злоумышленник не имеет доступа.

На доверенных серверах A_S, B_S, C_S, D_S, E_S, F_S, $G_S \in F$ устанавливается серверная часть сервиса — S_{MS}, выполняющая автоматическую «интеллектуальную» маршрутизацию трафика. Обозначим через F множество всех доверенных серверов с S_{MS}, через F_i — конкретный доверенный сервер $i \in [1, n]$.

На рис. 1,*б* показано, что использование S_M позволило избежать прохождения трафиком подконтрольного злоумышленнику участка. Данное решение S_M (итоговый маршрут) является вероятностным с вероятностью принятия p_j, $j \in [1, k]$. Здесь k — количество различных маршрутов от A_S до G_S на графе с вершинами A_S, B_S, C_S, D_S, E_S, F_S, G_S и ребрами, определяемыми текущей топологией сети. Расчет значений p_j будет рассмотрен далее.

Напомним, что в процессе передачи с помощью S_M данные проходят через некоторое число доверенных серверов, равное *f*. Выбор каждого следующего сервера происходит динамически и описывается гипергеометрическим распределением $HG(c; a_i, n, c)$. Параметры распределения: n — число всех используемых доверенных серверов; $c = 1$ (в случае использования инструмента мультиплексирования трафика $c > 1$), a_i — число недоступных для F_i серверов из числа всех серверов (определяется динамически). Таким образом, итоговый маршрут трафика от отправителя до получателя при использовании S_M и f доверенных серверов (из n доступных) будет выбран с вероятностью

$$p_j = \binom{n - a_0}{c} \cdot \binom{n - 1 - a_1}{c} \cdot \ldots \cdot \binom{n - f - a_f}{c}, j \in [1, k],$$

где a_i — число недоступных серверов для F_i при выборке F_{i+1} доверенного сервера на $i + 1$ шаге. Оценим вероятность успешной атаки p_A, когда злоумышленник контролирует участок между доверенными серверами F_t и F_{t+1}. При неизвестном про-

странственном расположении F_i считаем атаку успешной, если при работе сервиса S_M передатчики F_t и F_{t+1} были выбраны на i и $i+1$ этапе передачи, $t \in [1, n]$, $i \in [1, f]$:

$$p_A = \frac{2}{n - a_0} \cdot \frac{1}{n - 1 - a_1} + \frac{2}{n - 1 - a_1} \cdot \frac{1}{n - 2 - a_2} + \cdots + \frac{2}{n - (f-1) - a_{f-1}} \cdot \frac{1}{n - f - a_f}.$$

Эта формула легко распространяется на случай подконтрольных злоумышленнику участков между s доверенными серверами F_t, F_{t+1},...,F_{t+s}. Так, при достаточно большом n и достаточно малых a_i и f, причем $n >> f$ и $n >> a_i$, оценка p_A представляется в виде

$$p_A = O\left(\frac{1}{n^2}\right).$$

При использовании мультиплексирования ($c > 1$) задача злоумышленника еще более усложняется. Варианты атак злоумышленника на разнесенный трафик рассматриваются в [3].

ЛИТЕРАТУРА

1. *Ефимов В. И., Файзуллин Р. Т.* Система мультиплексирования разнесенного TCP/IP трафика // Вестник Томского госуниверситета. Приложение. 2005. № 14. С. 115–118.
2. *Avi Kak.* Port Scanning, Vulnerability Scanning, and Packet Sniffing // Computer and Network Security. 2008. № 23. С. 29–38.
3. *Ефимов В. И.* Атака на систему разнесенного TCP/IP трафика на основе анализа корреляции потоков // Информационные технологии моделирования и управления. 2005. № 6(24). С. 859–863.

УДК 519.8

АППРОКСИМАЦИЯ СЕТЕВОГО ТРАФИКА МОДЕЛЬЮ АЛЬТЕРНИРУЮЩЕГО ПОТОКА СОБЫТИЙ

О. В. Ниссенбаум, И. Б. Пахомов

Трафик пользователя в сети, как правило, имеет переменную интенсивность и с той или иной степенью достоверности может быть представлен потоком событий с кусочно-постоянной стохастически изменяемой интенсивностью [1]. Рассмотрим трафик пользователя компьютерной сети с точки зрения соответствия модели асинхронного альтернирующего потока. Такой поток имеет два состояния, в первом из которых наблюдается пуассоновский поток с параметром λ, а во втором события потока отсутствуют. Интервалы, на которых поток находится в первом или втором состоянии, распределены по экспоненциальному закону с параметром α_1 и α_2 соответственно. Сравним полученные результаты с результатами для модели пуассоновского потока с интенсивностью λ_P.

Статистика трафика в виде временных моментов получения пакетов данных была собрана с компьютеров одной локальной сети. Данные сгруппированы по 1 мин, в каждой группе производилась оценка параметров. Использованы оценки параметров асинхронного альтернирующего потока, полученные в [2], и оценка моментов для интенсивности пуассоновского потока [3]. На рис. 1 черными штрихами обозначены моменты поступления пакетов данных за два отрезка времени по 2 мин. На верхней части рисунка (эксп. 1) трафик достаточно равномерен, на нижней (эксп. 2) выделяются периоды высокой интенсивности трафика и периоды «молчания». Оценки параметров