УДК 004.036

В.А. Лахно,
А.С. Петров

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ DOS-АТАКИ НА СЕРВЕРЫ КОМПЬЮТЕРНЫХ СИСТЕМ

В статье изложены результаты исследований, позволяющие повысить уровень защиты автоматизированных информационных систем. Приведены математические модели, описанные с помощью цепей Маркова, а также результаты моделирования информационных систем, имеющих подключение к Интернету через различные каналы связи.

Ключевые слова: *Цепь Маркова, граф состояний, автоматизированные информационные системы, информационная безопасность.*

У статті викладено результати досліджень, що дозволяють підвищити рівень захисту автоматизованих інформаційних систем. Наведено математичні моделі, описані за допомогою ланцюгів Маркова, а також результати моделювання інформаційних систем, що мають підключення до Інтернету через різні канали зв'язку.

Ключові слова: *Ланцюг Маркова, граф станів, автоматизовані інформаційні системи, інформаційна безпека.*

Results of the researches allowing raising of the automated information systems security level are stated. Mathematical models described by means of Markov chains, as well as the results of the modelling of information systems having a connection to the Internet through various communication channels are resulted.

Keywords: *Markov chain, state graph, automated information systems, information security.*

В связи с постоянным ростом количества инцидентов в области информационной безопасности в последнее десятилетие все больше внимания стало уделяться разработке общих формальных моделей оценки и управления информационными рисками различных классов систем. Эта задача актуальна и при построении вероятностных моделей различных классов атак, включая риск-анализ компьютерных систем и разработку алгоритмов управления рисками по защите информационных ресурсов субъектов хозяйственной деятельности и физических лиц.

Как правило, под атаками отказа в обслуживании (Denial of Service attack – DoS-attack) понимают сетевые атаки, приводящие к невозможности для легитимного пользователя сети получить доступ к ресурсам сервера.

Наиболее известны следующие разновидности DoS-атак [1, 2]:

- TCP SYNflood, TCP FIN Flood;
- Ping of Death;

– Tribe Flood Network (TFN) и Tribe Flood Network 2000 (TFN2K);
– Stacheldracht и др.

И хотя в среде профессиональных взломщиков многие из DoS-атак считаются занятием "для начинающих", тем не менее, такие атаки способны принести достаточно большой экономический ущерб.

В настоящее время существует много работ, раскрывающих различные подходы к моделированию атак типа DoS: сети Петри [3], описательные модели сети на базе деревьев [4], модель запрос/ответ для DoS-атак [1, 2, 4], использование графов атак для анализа уязвимостей [5] и т.д.

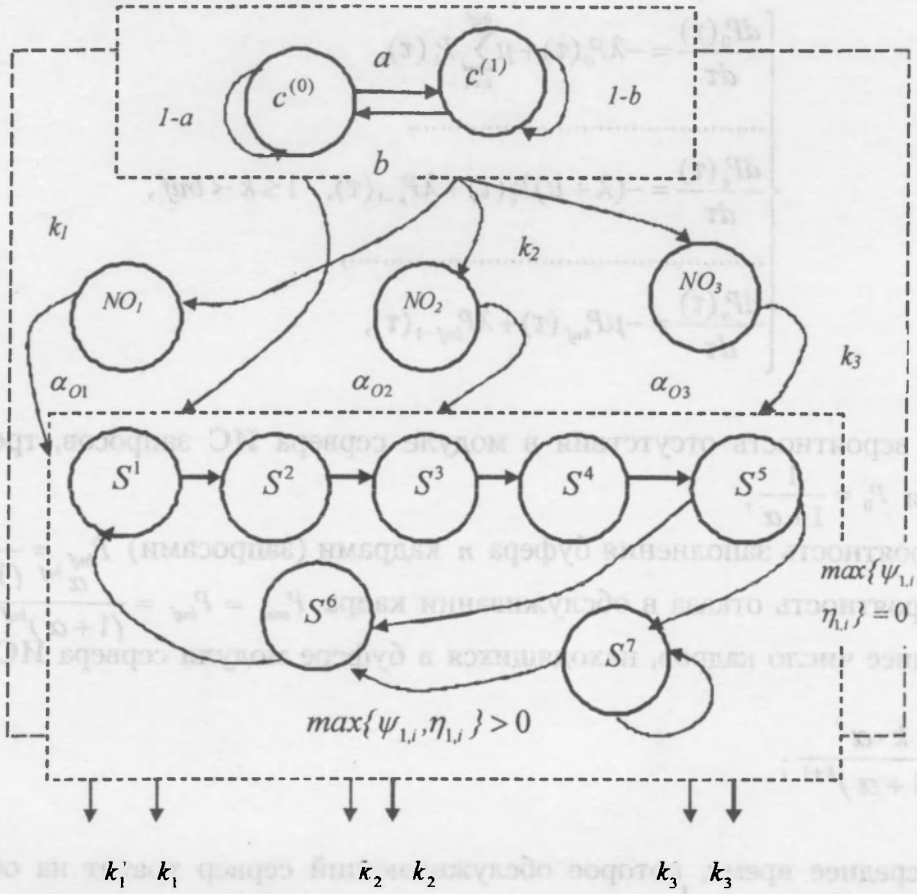
В отличие от большинства видов атак, атаки DoS используют не только ошибки программного обеспечения (ПО) сервера и пользователей. Часто используются программные ограничения операционной системы сервера и его ПО, а также аппаратные ограничения, накладываемые самим оборудованием, и потому очень трудно преодолеваемые (практически, аппаратные ограничения можно преодолеть только заменой оборудования сервера, что не только дорого, но и бессмысленно, поскольку новое оборудование тоже ограничено в возможностях). Задача предотвращения DoS-атак является достаточно сложной. Практически, она решается только отбрасыванием части данных, идущих на сервер от пользователей, причем это отбрасывание должно происходить не на самом сервере, а до него. Такая стратегия защиты имеет существенные недостатки:

– необходима координация действий с провайдером из вышестоящей сети, что не всегда возможно;
– существует вероятность отбросить данные легитимного пользователя, не участвующего в атаке.

Вторая проблема также достаточно серьезна, поскольку атакующий пользуется такими же протоколами обмена, какими и нормальные пользователи. Поэтому достаточно высока вероятность отказа в обслуживании не только атакующему, но и легитимному пользователю, создающему своими действиями достаточно высокую, но отнюдь не критическую, нагрузку на сервер.

Опираясь на приведенный выше анализ возможности моделирования атак на сервер ИС, функциональную схему системы можно представить следующим образом (рис. 1). Входные потоки k_1, k_2, k_3 формируются в некоторой случайной среде (СС), состояние которой определяет вероятностную структуру этих потоков. Если среда находится в состоянии $c^{(0)}$, то входные потоки представляют собой потоки типа Пуассона (потоки отдельных требований). При состоянии среды $c^{(1)}$ входные потоки являются потоками типа Бартлетта (потоки пачек) [8]. Заявки входных потоков поступают в накопители (очереди) $N\mathrm{O}_1, N\mathrm{O}_2, N\mathrm{O}_3$ с неограниченными емкостями. Далее будем считать, что поток k_1 – малоинтенсивный приоритетный поток; поток k_2 – малоинтенсивный поток; поток k_3 – приоритетный поток наибольшей интенсивности.

Информативность потока k_1 означает, что в динамике работы системы учитывается наличие заявок в накопителе $N\mathrm{O}_1$ и поступление требований по этому потоку. Его приоритетность – необходимость оперативного обслуживания поступающих требований. Приоритетность потока k_3 означает, что при отсутствии требований по потоку k_1 (разрыв) будет продолжено обслуживание по потоку k_3.



S^1 – начальное состояние входа в ИС; S^2 – начало сканирования доступных ресурсов; S^3 – ожидание ответа о наличии свободных ресурсов; S^4 – подключение к имеющимся ресурсам; S^5 – передача данных в КС (например, загрузка эксплойта); S^6 – передача данных на доступные АРМ; S^7 – загрузка/отправка запросов на сервер.

Рис. 1. Функциональная схема системы с конфликтными заявками на обслуживание

В соответствии с этим организована работа обслуживающего устройства (ОУ), например, сервера БД информационной системы предприятия, имеющего состояния, показанные на рис. 1 – $S^{(r)}, r = \overline{1,7}$, образующих множество $S = \{S^{(r)} : r = \overline{1,7}\}$. ОУ в состоянии $S^{(r)}$ находится в течении времени $\tau_r, r = \overline{1,7}$. ОУ выполняет функции по анализу и обслуживанию требований, по управлению входными потоками, по формированию очередей в накопителях и по отбору требований из очередей с помощью некоторых стратегий обслуживания $\alpha_{O1}, \alpha_{O2}, \alpha_{O3}$. Состояние $S^{(2j-1)}$ для $j = 1,2,3$ обслуживающего устройства соответствует обслуживанию требований потока k_j. В состоянии $S^{(2j)}$ для $j = 1,2,3$ не обслуживаются требования ни одного из входных потоков. В состоянии $S^{(7)}$ обслуживаются требования потока k_3. В соответствии с графом (1), при каждом $r = 1,2,3,4$ состояние $S^{(r)}$ переходит в состояние $S^{(r+1)}$.

Выходные потоки при работе системы с максимальной загрузкой, когда по любому потоку k_i злоумышленники, атакующие систему, могут создать очередь, а ОУ работает без простоев, назовём потоками насыщения и обозначим $k_1'k_2'k_3'$. Реальные выходные потоки в системе будем обозначать k_1, k_2, k_3.

Модель взаимодействия случайной среды передачи данных и сервера ИС можно описать следующей системой уравнений:

$$\begin{cases} \dfrac{dP_0(\tau)}{d\tau} = -\lambda P_0(\tau) + \mu \sum\limits_{k=1}^{buf} P_k(\tau), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \dfrac{dP_k(\tau)}{d\tau} = -(\lambda+\mu)P_k(\tau) + \lambda P_{k-1}(\tau), \quad 1 \le k \prec buf, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \dfrac{dP_n(\tau)}{d\tau} = -\mu P_{buf}(\tau) + \lambda P_{buf-1}(\tau), \end{cases} \quad (1)$$

где P_0 – вероятность отсутствия в модуле сервера ИС запросов, требующих обработки $P_0 = \dfrac{1}{1+\alpha}$;

P_{buf} – вероятность заполнения буфера n кадрами (запросами) $P_{buf} = \dfrac{\alpha^{buf}}{(1+\alpha)^{buf}}$;

$P_{отк}$ – вероятность отказа в обслуживании кадра $P_{отк} = P_{buf} = \dfrac{\alpha^{buf}}{(1+\alpha)^{buf}}$;

m – среднее число кадров, находящихся в буфере модуля сервера ИС

$$m = \sum_{k=1}^{buf} \frac{k \cdot \alpha^k}{(1+\alpha)^{k+1}};$$

$\langle \tau_{обс_1} \rangle$ – среднее время, которое обслуживающий сервер тратит на обработку одного кадра $\langle \tau_{обс_1} \rangle = \dfrac{1}{\mu \cdot m}$;

buf – необходимая емкость буферной памяти модуля ЛВС $buf = \dfrac{ln\, P_{отк}}{ln\,\alpha - ln(1+\alpha)}$,

$P_{обс}$ – вероятность своевременного обслуживания кадров с данными в зависимости от параметров N, $C_н$, $L_д$, $\tau_{p_{max}}$ $(P_{обс} = f(N, C_н, L_д, t_{p_{max}}))$.

N – число модулей ЛВС;

$C_н$ – номинальная пропускная способность протокола;

$L_д$ –длина поля данных;

λ – интенсивность входящего потока заявок;

$\tau_{p_{max}}$ – максимальное время реакции.

В рамках исследований мы не рассматриваем все варианты организации атакующими различных конфликтных типов потоков, поскольку этому вопросу посвящены отдельные исследования [7, 8]. Упомянем лишь небольшое количество вариантов, подтвердивших свою "высокую эффективность" при использовании злоумышленниками:

– низкоскоростные DoS-атаки;

– атаки с посылкой пакетов с нулевой частотой относительно временной шкалы времени прохождения пакетов по каналу связи до адресата и обратно;

– атаки, в которых злоумышленник может варьировать длительностью импульсов;

– атаки с минимальными случайными значениями относительно временной шкалы времени прохождения пакетов по каналу связи до адресата и обратно.

Все анализируемые случайные объекты, применяемые при построении математической модели и связанные с процессом обслуживания заявок, заданы на некотором полном вероятностном пространстве $(\Omega, A, P(*))$ элементарных случайных событий $\omega \in \Omega$ с вероятностной мерой $P(A)$. Для описания входных потоков заявок использовался нелокальный способ, то есть нашему рассмотрению подлежит не конкретное требование, а весь поток заявок.

Произвольный входной поток k_j описывается векторной случайной последовательностью $\{(\tau_i, \nu_i, \eta_{j,i}); i \geq 0\}$, где $\eta_{j,i}$ – число заявок типа ν_i, поступивших на промежутке времени $[\tau_i, \tau_{i+1})$ по этому потоку. Тип заявок определен меткой ν_i (состоянием случайной среды). Поведение случайной среды описывалось однородной Марковской последовательностью $\{\nu_i; i \geq 0\}$ с двумя состояниями: $c^{(0)}$ – поток заявок с малой интенсивностью, $c^{(1)}$ – большой поток заявок и вероятностями перехода $a, b \quad 0 \leq a < b << 1$. Данные ограничения означают, что смена интенсивности потока происходит редко и, что обычный режим работы сервера ИС с "малоинтенсивным" потоком заявок бывает чаще чем поток с большим количеством запросов, как это происходит при атаке типа DoS – "Отказ в обслуживании". Это позволяет считать, что за время τ_r, когда ОУ пребывает в состоянии $S^{(-)}$, интенсивность запросов не меняется. Известно, что случайные элементы $\nu_i; i \geq 0$ связаны соотношениями:

$$\nu_{i+1} = \phi_i(\nu_i, \omega_i),$$

где ϕ_i – некоторые измеримые отображения пространства $\{c^{(0)}, c^{(1)}\} \cdot \{0,1\}$ на $\{c^{(0)}, c^{(1)}\}$;

$\{\omega_i; i \geq 0\}$ – последовательность независимых случайных величин с некоторым распределением, в нашем случае, равномерным на интервале (0,1).

Протекающие процессы обслуживания имеют в нашей модели дискретный характер и рассматриваются на интервалах времени, порождаемых некоторым случайным точечным процессом $\tau = \{\tau_i; i \geq 0\}$ на оси времени.

Моменты $\tau_i; i \geq 0$, как правило, определенным образом связаны с моментами смены состояний обслуживающего устройства, их определение будет приведено ниже.

Обозначим через $\psi_{j,i}$ длину очереди в накопителе NO_j по потоку k_i в момент $\tau_i; i > 0$, $j = 1,2,3$.

В любой момент времени $\tau > 0$ обслуживающее устройство находится в некотором состоянии $S(\tau) \in S$. Управление входными потоками и трансформациями состояний ОУ с учетом вышеуказанных предварительных замечаний можно описать следующим образом:

$$S_{i+1} = u(S_i, \psi_{1,i}, \eta_{1,i}) = \begin{cases} S^{(1)} & при\ S_i = S^{(6)}; \\ S^{(r+1)} & при\ S_i = S^{(r)}\ r = \overline{1,4}; \\ S^{(6)} & при\ S_i \in \{S^{(5)}, S^{(7)}\} \& \max\{\psi_{1,i}, \eta_{1,i}\} > 0; \\ S^{(7)} & при\ S_i \in \{S^{(5)}, S^{(7)}\} \& \max\{\psi_{1,i}, \eta_{1,i}\} = 0; \end{cases} \quad (2)$$

для $i = 0,1,\ldots k$.

Для состояний ОУ предполагаем, что $S_i = S(\tau_i) = S(\tau_i + 0)$, $S(t) = S(\tau_i)$. Случайный точечный процесс $\{\tau_i; i \geq 0\}$ при $\tau_0 = 0$ определяется рекуррентным соотношением:

$$\tau_{i+1} = \tau_i + U(S_i), i \geq 0, \quad (3)$$

где $U(*)$ – отображение множества на числовое множество $\{T_1, T_2, \ldots T_7\}$ такое, что $T_r = U(S^{(r)}) > 0$, $r = 1,2,\ldots,7$.

Параметр T_r называется длительностью фазы (состояния) $S^{(r)}$ обслуживающего устройства, а величина $T = \sum_{r=1}^{7} T_r$ длительностью периода ОУ.

Рассмотрим, как изменяется эффективность DoS-атаки при большом количестве потоков и для ситуации, когда потоки имеют различные параметры. На рис. 2 показана модель очереди, составленная в Matlab 7, для различных типов потоков заявок к ОУ.

В ходе моделирования в качестве основной топологии атакуемой сети предприятия были приняты следующие парамстры:

– несколько потоков идут через узкий перегруженный канал емкостью до 1.5 Мб/с;

– размер буфера очереди выбран таким образом, что RTT (шкала отметки времени прохождения пакетов по каналу связи до адресата и обратно) лежит в пределах от 12 до 132 мс;

– одиночные заявки поступают (импульсы DOS-трафика) со скоростью 1.5 Мб/с, длительностью импульса 100 мс и размером пакета 50 байт.

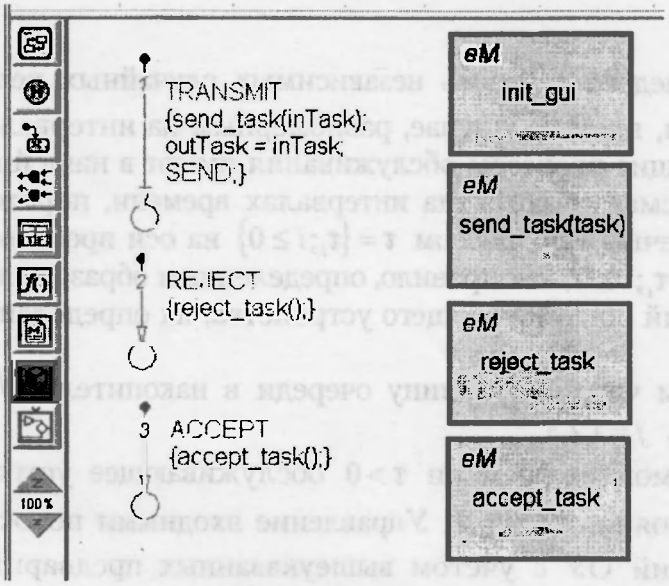


Рис. 2. Блок "Очередь заявок"

На рис. 3 показана нормализованная пропускная способность множества потоков от периода атаки T (фактически фазы обслуживания). Ситуация наблюдается аналогичная как и при одиночном потоке. Однако следует обратить внимание, что при $T=1/min$RTO пропускная способность не равна нулю (как было в случае с одним потоком), и некоторые потоки все же смогли "прорваться" в канале. Также стоит отметить, что при частоте $2/min$RTO пропускная способность практически сводится к нулю.

На рис. 4 показана нормализованная пропускная способность для нескольких потоков (от 1–3). Кривая, отмеченная "Атака отсутствует", показывает пропускную способность каждого потока без атаки. Кривая, обозначенная "Имитация атаки в MATLAB7+Simulink", показывает пропускную способность каждого потока в сумме с атакующим импульсом скоростью 5 Мб/с, длительностью 80–100 мс и периодом 0,5–1,0 сек.

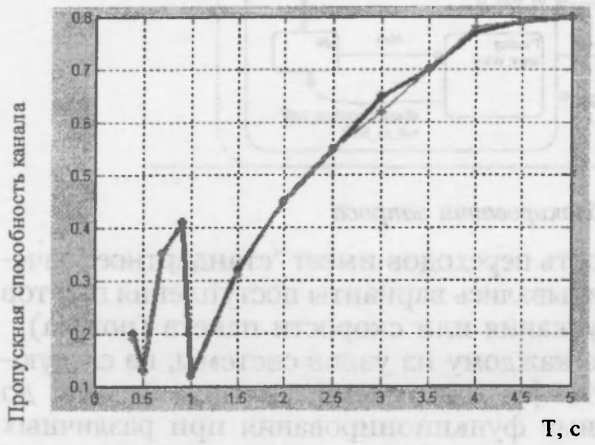


Рис. 3. Нормализованная пропускная способность множества потоков

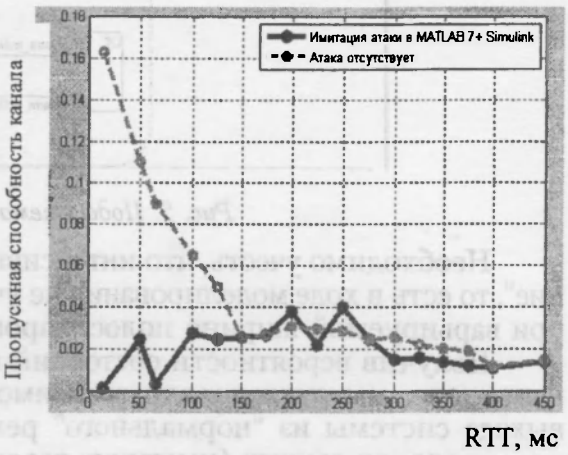


Рис. 4. Нормализованная пропускная способность множества потоков

В большинстве случаев после захвата управления злоумышленнику необходимо получать информацию с хоста (например, файлы паролей, личные документы пользователя, cookies и т.д.) и посылать команды на хост. Для обеспечения скрытой передачи данных достаточно часто используются скрытые каналы. Основная идея скрытых каналов заключается в том, чтобы передавать информацию в неиспользуемых полях сетевых протоколов, или изменять некритичную информацию в сетевом протоколе. Разработано достаточно большое количество программ для создания скрытых каналов [2, 3]. Поскольку задача детального проектирования СЗИ в рамках настоящих исследований не ставилась, ниже остановимся только на самой простой схеме моделирования блокировки запроса в сети предприятия при обнаружении атаки типа "отказ в обслуживании" при использовании скрытого канала. Сетевое обнаружение наличия скрытого канала представляется достаточно сложным [4, 9]. При обнаружении необходимо построить решающее правило, которое позволит разделять Intial Sequence Number (ISN), сгенерированные оригинальным стеком, и ISN, сгенерированные атакующим.

Построение разделяющего правила возможно при учете следующих факторов. В большинстве ОС ядро генерирует ISN не случайным образом. ISN является сложной функцией от текущего времени и предыдущего значения ISN. Атакующий генерирует ISN случайным образом, поскольку шифрует данные на случайном сеансовом ключе.

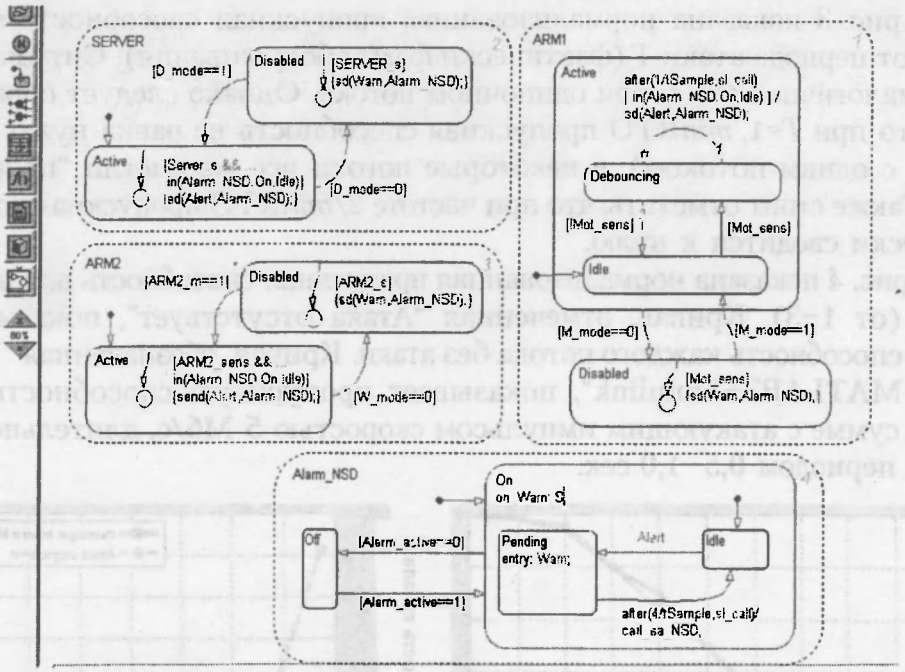
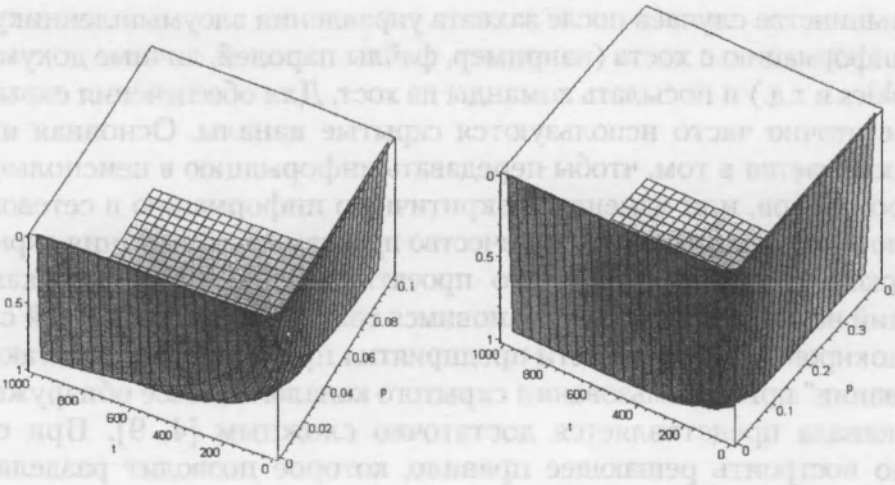


Рис. 5. Подсистема блокирования запроса

Необходимо учесть, что интенсивность переходов имеет "стандартное значение", то есть в ходе моделирования не учитывались варианты поступления пакетов при варьируемой ширине полосы пропускания или скорости пакета (потока).

Получив вероятности состояний по каждому из узлов системы, на следующем этапе, мы исследовали зависимость функции распределения времени до выхода системы из "нормального" режима функционирования при различных типах потоков данных (имеющих различную интенсивность и приоритет выполнения), в том числе конфликтных, и времени для проведения злоумышленником атаки. Основные результаты моделирования показаны на рис. 6.



$\tau = 0..1000, P(S_1)=0..0.1$

а)

$\tau = 0..1000, P(S_1)=0..0.4$

б)

Рис. 6. Зависимость функции распределения времени до выхода системы из "нормального" режима функционирования при различных типах потоков данных, и времени для проведения злоумышленником атаки

Как показывает анализ полученных зависимостей, а также данных, приведенных в работе "Имитационное моделирование систем защиты информации предприятия в MATLAB 7/2009 и SIMULINK" [10], при использовании злоумышленником тактики присваивания малоинтенсивному потоку заявок высокого приоритета, и достаточном времени проведения атаки, можно увеличить вероятность проникновения в систему и перемещения по состояниям $S_1 - S_7$, при этом не изменяя параметров потока, имеющего в системе наибольшую интенсивность и высокий приоритет.

Таким образом, для проведения успешной атаки на сервер ИС, в частности, типа "отказ в обслуживании", не обязательно создавать большое количество запросов к серверу или снижать полосу пропускания трафика. Можно с достаточно большой степенью вероятности успеха эксплуатировать уязвимости, связанные с созданием малоинтенсивного приоритетного потока, например, варьируя такими параметрами, как: скорость пакета (низкоскоростные DoS-атаки); количество пакетов с нулевой частотой относительно RTT; длительность импульса и др.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Классификация сетевых атак. [Электронный ресурс]. – Режим доступа : http :// www.cpress.ru.
2. *Лукацкий А.В.* Обнаружение атак / А.В. Лукацкий. – СПб. : BHV-Санкт-Петербург, 2001. – 624 с.
3. *Сабо Ю.И.* Применение сетей Петри с марковскими свойствами для анализа отказоустойчивости систем с резервированием / Ю.И. Сабо // Известия вузов. Приборостроение. – Т. 47. – 2004. – № 12. – С. 18–23.
4. *Воробьев А.А.* Анализ моделей процессов защиты информации от несанкционированного доступа в автоматизированных системах / А.А. Воробьев // Информатика-машиностроение. – 1999. – № 2. – С. 32–34.
5. *Iglun K.* State Transition Analysis : A Rule-Based Intrusion Detection System / K. Iglun, R.A. Kemmerer, P.A. Porras // IEEE Transactions on Software Engineering. – 1995. – 21 (3).
6. *Chung M.* Simulating Concurrent Intrusions for Testing Intrusion Detection Systems / M. Chung, B. Mukherjee, R.A. Olsson, N. Puketza // Proc. of the 18th NISSC, 1995.
7. *Smirniy M.F.* The research of the conflict request threads in the data protection systems / M.F. Smirniy, V.A. Lahno, A.S. Petrov // Праці Луганського відділення Міжнародної Академії інформатизації. – 2009. – № 2 (20). – Ч. 2. – С. 23–30.
8. *Лахно В.А.* Исследование конфликтных потоков заявок в системах защиты информации / В.А. Лахно, А.С. Петров, Н.Н. Чертунина // Вісник СНУ ім. В. Даля. – 2009. – № 6 (136). – С. 200–209.
9. *McNab C.* Network Security Assessment. O'Reilly Media, Inc., (2004).
10. *Лахно В.А.* Имитационное моделирование систем защиты информации предприятия в MATLAB 7/2009 и SIMULINK / В.А. Лахно, А.С. Петров // Сучасна спеціальна техніка. – 2010. – № 3 (22). – С. 63–73.
11. *Лахно В.А.* Построение дискретных процедур распознавания и поиска уязвимостей информации / В.А. Лахно, А.С. Петров, А.С. Скрипкина // Інформаційна безпека. – 2010. – № 2 (4). – С. 5–13.

Отримано 22.04.2011