DOI: 10.25586/RNU.V9187.20.01.P.105 УДК 004.94+517

А.Г. Певнева, А.И. Зимовец, Г.А. Санжаревский

СИСТЕМНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ КРИТИЧЕСКОЙ НАГРУЗКИ НА ИНФОРМАЦИОННЫЕ РЕСУРСЫ

Приводятся результаты предварительного анализа ситуации критической нагрузки, дается описание методик нагрузочного тестирования информационных ресурсов в ситуации переполнения буфера запросов по различным протоколам.
Ключевые слова: нагрузочное тестирование, системный анализ, критическая нагрузка.

A.G. Pevneva, A.I. Zimovets, G.A. Sanzharevskij

SYSTEM ANALYSIS AND MODELING OF CRITICAL LOAD ON INFORMATION RESOURCES

The results of a preliminary analysis of the critical load situation are given, and methods for load testing information resources in situations of request buffer overflows using various protocols are described.
Keywords: load testing, system analysis, critical load.

Введение

Системное исследование любого объекта или процесса есть последовательность взаимно влияющих этапов: сбор и анализ данных, математическое моделирование, оптимальное проектирование. Целью настоящего исследования служит выявление проблематики *первого этапа,* представляющего собой программное моделирование ситуации критической нагрузки на информационные ресурсы и предмодельную обработку полученной статистической информации.

Критическая нагрузка на информационный ресурс возникает в том числе в результате распределенной сетевой атаки типа «отказ в обслуживании» (DDoS-атаки). Как объект системного исследования она интересна для изучения, потому что, во-первых, угроза подобных атак остается актуальной на протяжении многих лет, даже такие мощные сетевые ресурсы, как eBay или Yahoo, страдают от DDoS-атак [4]. Известно три различных метода организации атак такого типа: превышение пропускной способности полосы, превышение возможности сервера, использование ошибок ресурса. Первый вид атаки предполагает, что на ресурс сети направляется большое количество запросов по различным протоколам, превышая их пропускную способность. Атаки второго типа вызывают «падение» конкретной службы сервера. Третий тип атаки требует кропотливого исследования программной организации ресурса для выявления уязвимости. Примером уязвимостей считаются скрипты, выполняющие огромное количество запросов к базе данных и использующие процессорное время.

По этому направлению существует достаточно открытых источников, содержащих статистические данные. Эти данные, несмотря на их разнородность и даже в некоторых

случаях противоречивость [2; 4], пригодны для вычисления количественных оценок по известным методикам при разработке макромоделей в области сетевой безопасности.

Атака типа «отказ в обслуживании» (или DoS-атака) представляет собой «критический» случай. В математическом моделировании этого явления используется методология теории массового обслуживания. Развернутое описание данного аппарата приводится, в частности, в литературных источниках (см., например: [1; 2; 6]).

Описанные в настоящей работе программные модели фактически представляют собой методики нагрузочного тестирования, а в статистических расчетах используются методы анализа временных рядов. Заметим, что в дальнейшем при построении математической модели не учитывается формализованное поведение «нарушителя», которое включается в общую модель безопасности информационных систем [2]. Отсутствие этого компонента в данном исследовании объясняется тем, что в информационных системах специального назначения само понятие «нарушителя» значительно нивелируется установкой нескольких уровней доступа. Именно в этом состоит сходство описанных методик с методиками нагрузочного тестирования, когда необходимо, подобно «злоумышленнику», устраивать мини-атаки, чтобы выявлять уязвимости ресурса.

Программное моделирование при исследовании отказа в обслуживании для сбора и анализа данных

На первом этапе системного анализа необходимо выработать программную методику проведения эксперимента и архитектуру хранилища данных, удобную для статистической обработки, основные параметры, которые будут количественно оцениваться: количество запросов за определенный период, время между запросами, количество запросов к определенному пункту назначения [5].

Все работы будут производиться с виртуальными машинами, которые должны быть объединены в единую сеть. В качестве сервера, на который будет производиться атака, была выбрана ОС Ubuntu с веб-сервером Apache.

В первом варианте используются операционная система Ubuntu Desktop 18.04 LTS и утилита Netdata для сбора статистики. Прежде чем приступать к установке Netdata, необходимо установить все необходимые зависимости и клонировать официальный репозиторий netdata. Далее выполним команду для запуска скрипта автоматической сборки и установки Netdata: *cd ~/netdata sudo ./netdate-installer.sh.* По завершении установки инструмент Netdata доступен по 19999 порту виртуальной машины, на которую был установлен. В браузере необходимо перейти по ссылке *http://ip_address:19999*, где ip_address – IP-адрес виртуальной машины, на которую установлена утилита.

На виртуальной машине, с которой будет осуществляться DoS-атака, будет установлена операционная система Kali Linux Light. Чтобы осуществить настройку репозиториев для возможности установки дополнительных пакетов, нужно в файл sources.list добавить ссылки на репозитории и выполнить обновление списка пакетов: *sudo apt-get update.* Далее необходимо объединить виртуальную машину с сервером и виртуальную машину с Kali Linux в единую сеть и установить их IP-адреса.

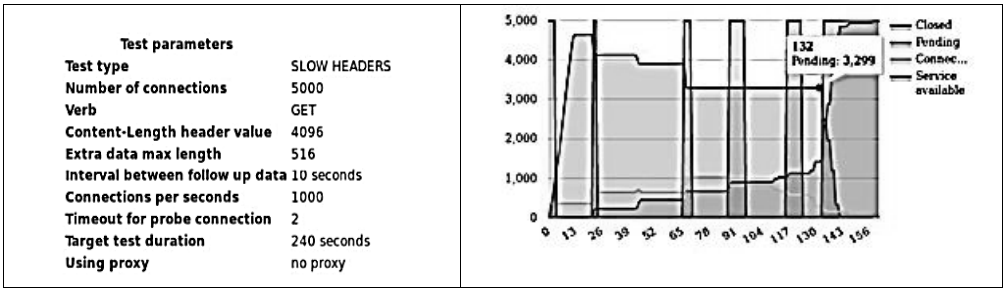
Для симулирования атаки DoS используются инструменты для стресс-тестирования сети. Одним из таких инструментов является утилита SlowHTTPTest, входящая в состав ОС Kali Linux. Проведем атаку, направленную на создание множества открытых http-со-

единений, в результате чего сервер не сможет отправить ответ на все запросы, произойдет переполнение стека очереди. Для запуска теста выполним

slowhttptest -c 5000 -H -g -o outputfile -i 10 -r 2000 -t GET -u http://10.0.2.15 -x 240 -p,

где *c* – общее количество соединений; *H* – Slowloris mode (создание множества открытых http-соединений); *g* – генерация статистики; *o* – путь к файлу, содержащему статистику; *i* – время ожидания данных; *r* – количество запросов с типом, указанным в *t* (в нашем случае GET-запросы); *u* – URL-адрес; *x* – размер запроса, байт; *p* – время ожидания ответа на запрос.

Пример вывода после выполнения команды представлен на рисунке 1 слева. Справа отображается график из выходного файла, который получился в результате исполнения программы SlowHTTPTest. Этот график показывает количество соединений и моменты времени, когда веб-сервер был недоступен. На рисунке 2 приводится зависимость обрабатываемых пакетов от времени, что с 11:23:30 до 11:25:00 сервер не обрабатывает запросы, также на графике отражается разрыв в соединении в том же промежутке времени.



Test parameters	
Test type	SLOW HEADERS
Number of connections	5000
Verb	GET
Content-Length header value	4096
Extra data max length	516
Interval between follow up data	10 seconds
Connections per seconds	1000
Timeout for probe connection	2
Target test duration	240 seconds
Using proxy	no proxy

Рис. 1. Статистика, собираемая SlowHTTPTest

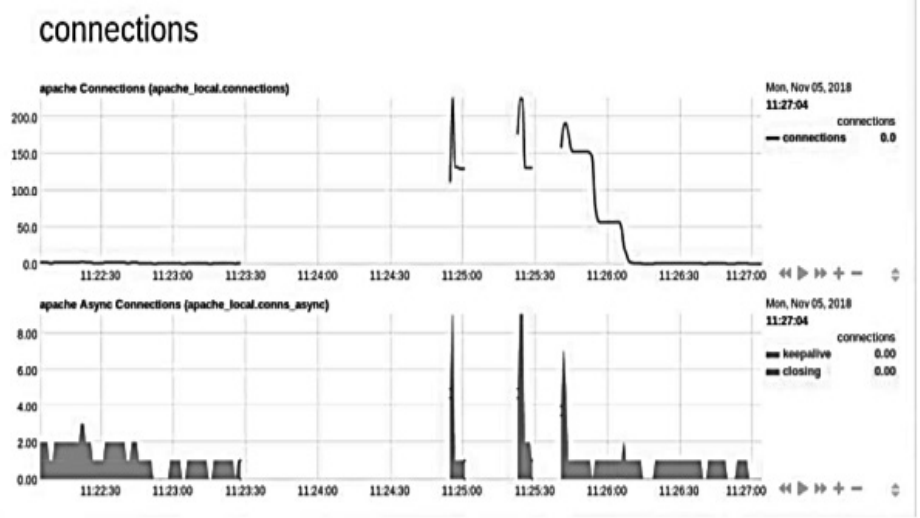


Рис. 2. Статистика сервера

Недостаток представленной методики заключается в невозможности анализа данных при повторяющихся экспериментах. Поэтому далее приводится описание другого подхода к сбору статистических данных при эмуляции атаки «отказ в обслуживании».

Этот подход основан на анализе лог-файлов сервера. Известно, что лог-файлы записаны в определенном формате, поэтому для проведения анализа написан скрипт, извлекающий из лог-файла данные для экспорта в базу данных. Скрипт реализован с помощью языка Python из-за наличия удобных средств работы со строками и регулярными выражениями. В качестве системы управления базами данных выбрана свободно распространяемая СУБД MySQL версии 8.0.16.

Статистику по запросам с группировкой по IP можно получить с помощью утилиты logtop:

tail -f access.log | awk {'print $1; fflush();'} | logtop.

Эта статистика работает в режиме реального времени и выводит новые значения с каждым запросом.

Для эмуляции DoS-атаки используется утилита для тестирования GoldenEye. На вход подается адрес атакуемого сервера. Время работы скрипта задается в переменной *x*. В примере время равно 30 секундам.

Результат проведения и визуальную нагрузку запросов во время атаки иллюстрирует рисунок 3. Как видим, запросы поступают только на один сайт большим потоком.

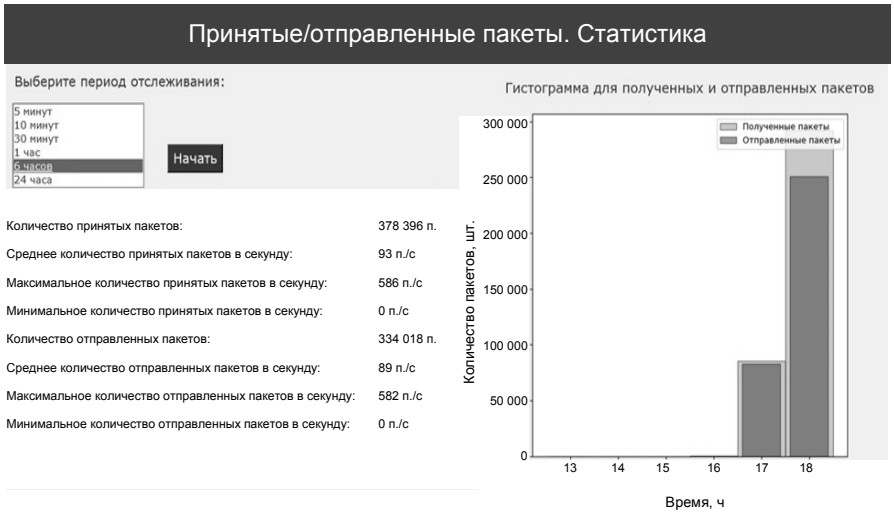


Рис. 3. Результат работы скрипта

Особенности статистического анализа и математического моделирования полученной информации

Для выбора архитектуры базы данных для хранения результатов эксперимента нужно не только выделить необходимые атрибуты, но и предусмотреть удобство доступа к динамически обновляемой информации, чтобы оценить интенсивность работы ресурса для дальнейшего построения математической модели. Очевидно, что для обработки получа-

емых данных следует использовать аппарат временных рядов для исключения так называемой сезонной компоненты. В [3] описана эмпирическая процедура исключения недельных и суточных циклов, которая применяется для расчетов доверительных интервалов для количества запросов, чтобы определить ранее начало атаки.

Именно необходимость учета активности в различные периоды работы сервера накладывает специфику на структуру базы данных. Если статистика собирается в течение нескольких суток, причем сутки разделены на *n* периодов, то необходимо данные по каждому измеряемому фактору хранить в отдельной таблице. Кроме этого, при проектировании такого хранилища данных приходится учитывать, что скорость его наполнения весьма велика. Накопленные в результате вычислительных экспериментов данные устаревают медленно, они могут быть востребованы для различных методик анализа.

В процессе оценки параметров потоков данных в вычислительных сетях используется аппарат сетей массового обслуживания (СеМО). В [6] для обнаружения атак предложен метод оценки вероятности потери произвольного запроса в узле СеМО.

Подход, основанный на предположении о стационарности входящих потоков, дает оценку частоты потери запросов в сети в случае, если СеМО находится в стационарном режиме. При атаке СеМО в некоторый момент времени выходит из стационарного режима.

Необходимо сделать предположение о независимости атак прикладного уровня на различные сетевые службы. В этом случае можно моделировать одноканальную систему массового обслуживания с очередью длины *m*. Также считается, что входящий поток запросов является пуассоновским с параметром λ.

Пусть λ – вектор интенсивностей входящих потоков запросов; μ – вектор интенсивностей обработки запросов и стохастическая матрица вероятностей переходов *P*.

Состоянием считается упорядоченная пара чисел (i, d), где i – номер узла, в котором находится запрос, $1 < i < J$; d – длина очереди, $1 < d < m_i + 1$. Состояние $(i, m_i + 1)$ соответствует переполненной очереди в узле *i*.

Далее, состояние *S* соответствует ответу на запрос, а состояние *F* соответствует потере запроса. Попав в состояние *F*, процесс из него не выходит. Начальные вероятности этих состояний равны нулю, а начальное распределение системы можно рассчитать как

$$\hat{p}\{(i,d)\} = \frac{\lambda_i \frac{\rho_i}{\mu_i}\left(1-\frac{\rho_i}{\mu_i}\right)}{\left(\sum_{i=1}^{J}\lambda_i\right)\left(1-\left(\frac{\rho_i}{\mu_i}\right)^{m_i}\right)}. \qquad (1)$$

В формулу (1) входят элементы вектора суммарных (извне и из других узлов) интенсивностей ρ, входящих в узлы исходной сети потоков. Вероятность перехода из состояния (i, d) в состояние (j, w) можно рассчитать по формуле

$$p\{(i,d \to j,w)\} = \frac{\frac{\rho_j^{\,w}}{\mu_j^{\,w}}\left(1-\frac{\rho_j}{\mu_j}\right)}{\left(1-\left(\frac{\rho_j}{\mu_j}\right)^{m_j}\right)}\hat{p}_{i,j}. \qquad (2)$$

Ясно, что $p\{(i,d) \to S)\}$ – вероятность успешной обработки запроса, а если запрос находится в переполненной очереди, то $p\{(i,m_i+1) \to F)\} = 1$ – вероятность его потери.

Программное моделирование стресс-нагрузки, описанное выше, дает возможность сбора необходимого статистического материала. Однако внимательное изучение источников [2; 5] требует указать на некоторые детали.

В формуле (1) все интенсивности должны быть рассчитаны с помощью итерационных процедур, подробно описанных в [6]. По существу, эти процедуры реализуют метод скользящего среднего по времени наблюдения. В формуле (2) элементы $\hat{p}_{i,j}$ необходимо корректировать, так как матрица P определяется для СеМО без потерь запросов.

Установка параметра k – числа шагов марковской цепи – происходит с помощью дополнительной итерационной процедуры. Идея этой процедуры состоит в том, что вероятность того, что на k-м шаге заданной цепи Маркова запрос все еще находится в сети, не должна превышать некоторую заданную точность [Там же].

Заключение

Зачастую доступные средства нагрузочного тестирования не предусматривают полного анализа высокой скорости накопления запросов разных видов к сетевому ресурсу. Чтобы провести полный цикл системного исследования такой ситуации, необходимо не только владеть инструментами нагрузочного тестирования, но и создать отдельные скрипты для сохранения результатов статистической обработки и численного определения параметров теста. Представленные средства являются удобным инструментом программного моделирования, но сравнительный расчетный анализ и оптимальный синтез системы нагрузочного тестирования видятся отдельной задачей.

В системном понимании задача нагрузочного тестирования ресурса является обратной к задаче выявления атаки типа «отказ в обслуживании», если прямую задачу обнаружения атаки понимать как критическое увеличение нагрузки без выделения задачи фильтрации по IP-адресам и анализа поведения пользователя. Поэтому периодичность можно учитывать при построении модели нагрузки, фактически плана эксперимента.

Также в заключение отметим, что в процессе проектирования такой системы программного моделирования критической нагрузки необходимо учитывать все особенности, которые указаны в настоящей статье.

Литература

1. *Алиев Т.И.* Основы моделирования дискретных систем: учебное пособие. СПб.: Изд-во ИТМО, 2009. 363 с.
2. *Гатчин Ю.А. и др.* Теория информационной безопасности и методология защиты информации: учебное пособие. СПб.: Изд-во ИТМО, 2015. 100 с.
3. *Терновой О.С., Шатохин А.С.* Раннее обнаружение DDoS-атак статистическими методами // Доклады ТУСУРа. Серия «Математическое обоснование и теоретические аспекты информационной безопасности». 2012. № 1 (25), ч. 2. С. 104–108.
4. Тренды 2018 года в области безопасности и развития интернет-инфраструктуры в России и мире // Qrator Labs. URL: https://qrator.net/ru/company/news/trendy-2018-goda-v-oblasti--bezopasnosti-i-razvitiia-internet-infrastruktury-v-rossii-i-v-mire (дата обращения: 30.05.2019).

5. *Щерба Е.В., Волков Д.А.* Разработка системы обнаружения распределенных сетевых атак типа «отказ в обслуживании» // Омский научный вестник. Серия «Прикладная дискретная математика». 2012. № 3. С. 67–70.
6. *Щерба Е.В., Щерба М.В.* Архитектура программной реализации методики обнаружения сетевых атак типа «отказ в обслуживании» // Динамика систем, механизмов и машин. 2012. № 1. С. 296–299.

Literatura

1. *Aliev T.I.* Osnovy modelirovaniya diskretnykh sistem: uchebnoe posobie. SPb.: Izd-vo ITMO, 2009. 363 s.
2. *Gatchin Yu.A. i dr.* Teoriya informatsionnoj bezopasnosti i metodologiya zashchity informatsii: uchebnoe posobie. SPb.: Izd-vo ITMO, 2015. 100 s.
3. *Ternovoj O.S., Shatokhin A.S.* Rannee obnaruzhenie DDoS-atak statisticheskimi metodami // Doklady TUSURa. Seriya "Matematicheskoe obosnovanie i teoreticheskie aspekty informatsionnoj bezopasnosti". 2012. № 1 (25), ch. 2. S. 104–108.
4. Trendy 2018 goda v oblasti bezopasnosti i razvitiya internet-infrastruktury v Rossii i mire // Qrator Labs. URL: https://qrator.net/ru/company/news/trendy-2018-goda-v-oblasti--bezopasnosti-i-razvitiia-internet-infrastruktury-v-rossii-i-v-mire (data obrashcheniya: 30.05.2019).
5. *Shcherba E.V., Volkov D.A.* Razrabotka sistemy obnaruzheniya raspredelennykh setevykh atak tipa "otkaz v obsluzhivanii" // Omskij nauchnyj vestnik. Seriya "Prikladnaya diskretnaya matematika". 2012. № 3. S. 67–70.
6. *Shcherba E.V., Shcherba M.V.* Arkhitektura programmnoj realizatsii metodiki obnaruzheniya setevykh atak tipa "otkaz v obsluzhivanii" // Dinamika sistem, mekhanizmov i mashin. 2012. № 1. S. 296–299.

DOI: 10.25586/RNU.V9187.20.01.P.111 УДК 614.842.83.07/.08:004.942

А.А. Рыженко, О.М. Прошина

МОДЕЛЬ ИНФОРМАЦИОННО-АНАЛИТИЧЕСКОЙ СИСТЕМЫ ПОДДЕРЖКИ УПРАВЛЕНИЯ ПОЖАРНОЙ БЕЗОПАСНОСТЬЮ ОБРАЗОВАТЕЛЬНЫХ КОМПЛЕКСОВ КРУПНЫХ ГОРОДОВ И АГЛОМЕРАТОВ

Отмечено, что проектирование системы пожарной безопасности образовательных комплексов с централизованным управлением существенно отличается от аналогичной для образовательного учреждения. Предложена модель системы поддержки управления пожарной безопасностью образовательных комплексов, объединяющих территориально-распределенные объекты разного уровня сложности.

Ключевые слова: моделирование, управление, образовательные комплексы, пожарная безопасность.