УДК 004.056.5

О.С. Терновой, А.С. Шатохин

Методика обнаружения уязвимостей к DDOS атакам систем управления контентом на примере системы Wordpress

O.S. Ternovoy, A.S. Shatokhin

Methods to Detect Vulnerabilities to DDOS Attacks by Content Management Systems on the example of Wordpress System

Дается описание созданной на базе Алтайского государственного университета распределенной сети для проведения нагрузочных тестов серверов и исследования различных аспектов DDOS атак. В рамках использования данной сети приводится описание экспериментов, направленных на выявление потенциально уязвимых мест одной из систем управления содержанием. По итогам этих и других экспериментов выработана методика обнаружения уязвимостей к DDOS атакам систем управления содержанием.

Ключевые слова: DDOS атака, бот сеть, среднеквадратичное отклонение, статистический анализ, нагрузочный тест, Apache JMetr.

This paper considers the distributed network for stress tests and studies of various aspects of DDOS attacks. Distributed network was established on the basis of the Altai State University. The researchers describe experiments aimed to identify potential vulnerabilities in one of content management systems. As a result of this and the other experiments they developed a method to detect vulnerabilities in content management systems.

Key words: DDOS attack, bot network, standard deviation, statistical analysis, stress test, Apache JMetr.

Современные DDOS атаки – одна из самых опасных и распространенных угроз в сети Internet. В результате DDOS атаки компьютеры, входящие в зомби-сеть, начинают посылать значительное количество запросов к атакуемому серверу. Основой зомби-сети являются компьютеры обычных пользователей, которые в силу каких-то обстоятельств (отсутствие антивируса, брандмауэра и т.д.) были заражены вредоносными программами. Программа-клиент DDOS сети может работать на компьютере пользователя никак себя не проявляя до того момента, пока не получит от злоумышленника команду атаковать какую-то цель. В результате такой атаки расходуются процессорное время, оперативная память, ресурс канала связи и, как следствие, доступ к серверу становится невозможным или ограниченным. Даже самые крупные компании, такие как Microsoft, Amazon, Yahoo, периодически сталкиваются с DDOS атаками и не всегда могут им противостоять. DDOS атака может очень быстро истощить сетевые ресурсы и тем самым остановить работу целой компании. Если учесть, что на сегодняшний день для нормальной работы многим компаниям требуется постоянный доступ в Интернет, например, связь с филиалами или с удаленными базами данных, или постоянная работа сервиса, представленного на веб-сайте компании, то недоступность этих ресурсов может обернуться для компании крупными убытками [1, 2].

Компания Cisco, лидер в производстве и реализации сетевых решений, признает, что на сегодняшний день нет достаточных средств для борьбы с DDOS атаками. Это связано с тем, что атаки такого типа возникают неожиданно и у системных администраторов нет возможности проанализировать атаку до момента ее начала. Подбирать меры противодействия атакам приходится уже в момент проведения атаки, когда сетевой ресурс уже испытывает трудности. Кроме того, в каждом конкретном случае опыт прошлых атак может быть недостаточен для отражения новой атаки. Это связано с тем, что злоумышленники постоянно развивают средства атак, меняют стиль, конфигурацию пакетов и т.д. [2].

Исследование DDOS атак сходно с изучением природных явлений, таких как землетрясения, извержения вулканов, разряды молнии, т.е. которые могут быть напрямую исследованы только в момент их наступления. Можно создать математическую или компьютерную модель DDOS атаки и проводить ее исследования, но нет никаких гарантий, что эта модель будет отображать все нюансы следующей DDOS атаки, которую смогут придумать и реализовать злоумышленники. Аналогично и со средствами защиты, злоумышленник может найти элемент сетевого ресурса, атака на который приведет к отказу в обслуживании всего ресурса [3].

Для исследования DDOS атак необходим механизм, который бы в лабораторных условиях мог

повторять реальные DDOS атаки сколь угодное количество раз, эмулировать новые атаки, максимально соответствующие реальным, вносить изменения в основные параметры атаки и отслеживать результат. По своей сути механизм должен представлять распределенную зомби-сеть, максимально приближенную к реальным действующим бот-сетям.

Для реализации такого инструмента на базе компьютерных классов Алтайского госуниверситета создан аналог DDOS сети. В качестве клиентов сети и атакуемого сервера выступают физические компьютеры, имеющие следующие технические и системные характеристики:

– процессор: **Celeron Dual 2600 MHz**;
– размер оперативной памяти: **2Gb**;
– размер жесткого диска: **250Gb**;
– сетевой адаптер: **100 Mb/s**;
– операционная система: **Windows XP Professional, Service pack 3**.

Использование в качестве клиентов зомби-сети физических компьютеров позволило получать более точные данные, по сравнению с данными, получаемыми в сетях, состоящих из виртуальных компьютеров, запущенных на одной физической платформе. На каждом зомби-компьютере запущена консольная версия программа Apache JMetr для операционной системы Windows XP. Использование Windows версии позволяет использовать эту программу в фоновом режиме, что особенно актуально для компьютерных классов, так как это позволяет проводить эксперименты круглосуточно, не прерывая основной учебный процесс. Программа Apache JMetr позволяет проводить различные нагрузочные тесты. Для каждого нагрузочного теста может быть создан специальный сценарий, который сделает тест максимально приближенным к DDOS атаке. Если сценарий создается на основе лог-файлов сервера, соответствующих реальной атаке, то нагрузочный тест по сути будет являться копией DDOS атаки. Выбранное программное обеспечение позволяет проводить распределенные нагрузочные тесты, что еще более приближает их к реальным атакам.

В качестве атакуемого сервера используется выделенный физический сервер с указанными выше техническими и системными характеристиками. На сервере установлен следующий набор серверного программного обеспечения:

– web-сервер: **Apache, версия 2.4.2**;
– сервер баз данных: **MySQL, версия 5.5.24**.

При проведении экспериментов ставилась задача выработать методику поиска потенциально уязвимых мест в системе управления содержанием – CMS – Content Management System. В качестве системы управления содержанием была использована CMS Wordpress версии 3.3.2. На сегодняшний день это одна из самых популярных CMS с открытым исходным кодом.

Эксперимент проводился в несколько этапов, в ходе каждого этапа компьютеры зомби-сети посылали запросы к определенному скрипту системы управления содержанием. При этом фиксировалась нагрузка, которую генерировал скрипт. Оценка нагрузки проводилась по следующим критериям: загрузка процессора, использование памяти, время отклика сервера. Данные о загрузке процессора и использовании памяти фиксировались непосредственно на самом атакуемом компьютере. Время отклика фиксировалось с нейтрального компьютера, находящегося в той же подсети, что и компьютеры зомби-сети.

Сценарий проведения атаки был разработан на основании данных лог-файлов web-сервера соответствующих одной из реальных DDOS атак.

Основной трудностью при проведении эксперимента стало выявления целей-скриптов, на которые необходимо было проводить атаки. Это связано с тем, что один и тот же скрипт может использовать различные ресурсы в зависимости от передаваемых в него параметров. Для создания списка всех возможных целей была использована библиотека mnoGoSearch для языка PHP, которая представляет собой модуль полнотекстового поиска – реализацию поисковой машины. С помощью этого модуля был создан список всех возможных страниц сайта – список скриптов с определенными параметрами. Затем из списка были удалены однотипные страницы. Например, страницы, показывающие различные записи или новости [4].

Второй задачей, требующей решения в рамках проведения данного эксперимента, стала задача по выбору оптимального значения интенсивности теста. Если к серверу будет поступать слишком большое количество запросов, т.е. интенсивность нагрузочного теста будет высока, единственное, что можно будет установить, – это сервер не работает. Так как в этом случае абсолютно любой скрипт будет использовать всю доступную для него память и 100 процентов процессорного времени. Необходимо выбрать такое значение интенсивности, при котором можно будет оценить и сравнивать работу всех скриптов. Это должно быть такое значение, при котором найдутся скрипты, которые не используют 100% предложенных ресурсов.

Для выбора оптимального значения была проведена предварительная серия экспериментов, в ходе которых интенсивность нагрузки постепенно наращивалась.

Перед началом экспериментов были зафиксированы значения используемых ресурсов. При нулевой нагрузке, т.е. при отсутствии с сервером активных соединений, загрузка процессора составила 0%, использование памяти – 433 Mb, время отклика – меньше 1 ms. При среднесуточной нагрузке использование ресурсов составило: процессор – 28%, оперативная память – 450 Mb, время отклика – меньше 1 ms.

В ходе проведения экспериментов был зафиксирован размер потребляемых различными скриптами ресурсов (табл.). Все скрипты потребляют примерно одинаковое количество ресурсов. Это показывает, что в данном случае не выявлено скриптов, тре-

бующих отдельного внимания. Этот результат ожидаем, так как Wordpress – один из лидеров среди систем управления содержанием.

Потребление скриптами ресурсов сервера при проведении DDOS атаки

№	Страница	Использование памяти (Mb)	Загрузка процессора (%)	Время отклика (ms)
1	index.php *главная страницы*	475	50	<1
2	index.php?p=1 *страница с записью*	479	57	<1
3	wp-comments-post.php *добавление комментария*	467	52	<1
4	wp-login.php *авторизация*	482	49	<1
5	index.php?s=search_text *поиск*	492	53	<1
6	indx.php?feed=rss2 *RSS лента*	483	51	<1
7	index.php?m=201205 *архив за месяц*	470	54	<1
8	readme.html *статичный файл*	450	27	<1
9	readme.html *то же, при максимальной нагрузке*	459	38	3

Анализ промежуточных результатов экспериментов показал, что самым ресурсоемким процессом является процесс web-сервера. Основную нагрузку этот процесс создает при генерации динамических страниц при помощи интерпретатора языка PHP. Были проведены дополнительные тесты со статичными файлами. Установлено, что работа со статическими файлами использует гораздо меньше ресурсов сервера даже при максимальной нагрузке, как при нормально функционирующем и настроенном сервере. Выяснив, что в данном случае основную угрозу несет именно создание динамических страниц, можно в случае обнаружения атаки включить кэширование и тем самым снизить нагрузку. Если бы по результатам экспериментов было установлено, что основную нагрузку генерирует, к примеру, сервер баз данных, можно было бы рекомендовать перенести его на отдельный компьютер.

Знание уязвимых мест позволяет системному администратору заранее, не дожидаясь начала атаки, оптимизировать работу скриптов и программного обеспечения, включить средства кэширования и т.д. Кроме того, средства, которые используются для раннего обнаружения DDOS атак, могут быть настроены в соответствии с уровнем уязвимости конкретных скриптов. Например, у более уязвимых скриптов можно будет установить повышенную чувствительность к атаке.

Если оптимизировать работу определенных скриптов не представляется возможным, то при обнаружении атаки можно отключить скрипты, генерирующие максимальную нагрузку. Например, при обнаружении атаки, автоматически блокировать работу скрипта регистрации новых пользователей. С одной стороны, новые пользователи не смогут зарегистрироваться, но с другой – такая мера позволит сохранить доступность ресурса [5].

В рамках исследования уязвимостей системы управления содержанием Wordpress была выработана следующая методика по обнаружению уязвимостей web-ресурсов и систем управления содержанием:

– получение списка всех возможных URL адресов web-ресурса;

– обработка полученного списка. Исключение однотипных страниц, генерируемых одним и тем же скриптом при использовании одних и тех же ресурсов;

– замер основных параметров сервера: загрузка процессора, использование памяти, время отклика и т.д., соответствующих нормальной работе сервера;

– выбор значения интенсивности нагрузочного теста;

– проведение атак на выбранные скрипты. Фиксация результатов;

– обработка результатов, выделение уязвимых мест.

После проведения указанного исследования необходимо оптимизировать работу уязвимых скриптов и повторить исследование необходимое количество раз. Это позволяет на каждом шаге выявлять новые уязвимости и увеличивать стойкость сервера к будущим атакам. На каждом новом шаге будут определяться другие скрипты, расходующие ресурсов больше остальных. Процесс оптимизации можно считать законченным, когда все скрипты потребляют примерно одинаковое количество ресурсов и сократить это потребление уже не представляется возможным.

Библиографический список

1. DDOS атаки [Электронный ресурс]. – URL: http://localname.ru/soft/ataki-tipa-otkaz-v-obsluzhivanii-dos-i-raspredelennyiy-otkaz-v-obsluzhivanii-ddos.html/

2. Предотвращение атак с распределенным отказом в обслуживании (DDoS) [Электронный ресурс]. – URL: http://www.cisco.com/web/RU/products/ps5887/products_white_paper0900aecd8011e927_.html

3. Методы защиты от DDOS нападений [Электронный ресурс]. – URL: http://www.securitylab.ru/analytics/216251.php.

4. Бенкен Е.С. PHP, MySQL, XML. Программирование для Интернета. – СПб., 2011.

5. Терновой О.С. Ранее обнаружение DDOS атак методами статистического анализа // Перспективы развития информационных технологий. – Новосибирск, 2012.