7. *Ramalingam G., Field J., Tip F.* Aggregate structure identification and its application to program analysis // POPL, 1999.
8. *Cui W., Peinado M., Chen K., Wang H. J., Irun-Briz L.* Tupni: Automatic Reverse Engineering of Input Formats // Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS). Alexandria, VA. October 27—31, 2008.
9. *Caballero J., Yin H., Liang Z., Song D.* Polyglot: Automatic Extraction of Protocol Message Format using Dynamic Binary Analysis // Proceedings of the 14th ACM Conference on Computer and Communications Security. Alexandria, VA. October 2007.
10. *Cui W., Kannan J., Wang H. J.* Discoverer: Automatic Protocol Reverse Engineering from Network Traces // Proceedings of the 16th USENIX Security Symposium. Boston, MA. August 2007.
11. *Wang R., Wang X. F., Zhang K., Li Z.* Towards automatic reverse engineering of software security configurations // Proceedings of the 15th ACM conference on Computer and communications security. October 27—31, 2008. Alexandria, Virginia, USA.

Г. Н. Мальцев, И. В. Прохоров

ВОПРОСЫ БЕЗОПАСНОСТИ ПРИ ОРГАНИЗАЦИИ ЗАЩИТЫ СИСТЕМЫ ЭЛЕКТРОННЫХ ПЛАТЕЖЕЙ

В настоящее время широкое распространение получили системы для оплаты повседневных услуг (системы моментальных платежей) с использованием различных технологических платформ: платежные системы в сети Интернет, банкоматы, терминалы самообслуживания, карточки пополнения. Поскольку более половины населения России пользуется услугами операторов сотовой связи или Интернета, как минимум, ежемесячно, а то и ежедневно значительное количество людей сталкивается с необходимостью пополнения баланса или оплаты счета.

Создание современной и защищенной системы, позволяющей принимать платежи от населения и обеспечивать высокий уровень безопасности, является на сегодняшний день новой и еще не изученной задачей. В России рынок подобных систем начал формироваться только после 2000 г., стали активно развиваться два направления: карточки пополнения и платежные системы Интернета. Каждое направление поначалу было ориентировано на различные рынки: карточки пополнения выпускались поставщиками услуг для увеличения баланса пользователя в системе, а платежные системы осуществляли выпуск электронных денег, позволяющих приобретать товары в интернет-магазинах. Однако основной проблемой в обоих случаях была сложность приема денег от населения, платежные системы также выпускали карточки пополнения. Но уже к 2004 г. на рынке появились первые платежные системы, которые предоставляли широкий спектр решений для приема платежей в пользу различных операторов услуг (также именуемых провайдерами), тем самым вытеснив с рынка карточки пополнения, а также предоставляли удобный способ конвертации наличных денег в электронные. На сегодняшний день основным инструментом для зачисления денег на счет является платежный терминал — аппарат, позволяющий совершить платеж наличными деньгами в пользу любого провайдера.

Конечно же новое направление не осталось без внимания различных мошенников с целью получения незаконной выгоды. Сейчас количество попыток незаконно списать деньги со счетов пользователей только увеличивается. Можно выделить два основных направления атак преступников: на сервера платежной системы и на ее пользователей.

В первом случае производится анализ архитектуры системы с целью нахождения ошибок в программном обеспечении и незаконного доступа к счетам пользователей или выведение из строя системы, но при этом необходимы профессиональные навыки специалиста высокого уровня, а также большие временные затраты и огромное количество попыток, которые зачастую не приносят желаемого результата. Попытки атак можно разделить на следующие виды:

1. Нацеленная на отказ системы DDoS-атака (Distributed Denial of Service, распределенный отказ в обслуживании). Цель этих атак — довести систему до отказа, т. е. создать такие условия, при которых легитимные (правомерные) пользователи системы не могут получить доступ к предоставляемым системой ресурсам либо этот доступ затруднен [7].

2. Нацеленные на нахождение ошибок в логике работы системы, которые предоставляют возможности получения доступа к персональным данным пользователей или незаконного списания денег со счетов.

3. Получение доступа в систему в обход защиты обычно производится с использованием эксплойтов (exploit) — это общий термин в сообществе компьютерной безопасности для обозначения фрагмента программного кода, который, используя возможности, предоставляемые ошибкой, отказом или уязвимостью, ведет к повышению привилегий или отказу в обслуживании компьютерной системы [8].

Наиболее часто встречающейся атакой является DDoS-атака с использованием значительного количества компьютеров, объединенных в Botnet. Botnet — это компьютерная сеть, состоящая из некоторого количества хостов с запущенными ботами — автономным программным обеспечением. Чаще всего бот в составе Botnet скрытно устанавливается на компьютере «жертвы» и позволяет злоумышленнику выполнять некие действия с использованием ресурсов зараженного компьютера [9].

В случае атак на клиентов системы ставится цель получить реквизиты, необходимые для авторизации в системе, или заставить пользователя самого заплатить мошенникам. В обоих случаях не требуется высокого профессионализма атакующего, поскольку пользователи зачастую не уделяют внимания безопасности собственного компьютера или не обладают достаточными знаниями, чтобы противостоять атакующему.

В первом случае у пользователя воруются пароли доступа в систему с использованием специальных программ, перехватывающих нажатые пользователем клавиши (кейлогеров), или вирусов, анализирующих данные, передаваемые в Интернет. Чаще всего эти программы попадают на компьютер пользователя через ошибки в браузерах Интернета при посещении сомнительных сайтов или пользователи сами запускают файлы, скачанные из Интернета.

Во втором случае используются приемы социальной инженерии. Это метод (атак) несанкционированного доступа к информации или системам хранения информации без использования технических средств. Метод основан на использовании слабостей человеческого фактора и считается очень разрушительным. Злоумышленник получает информацию, например, путем сбора данных о служащих объекта атаки с помощью обычного телефонного звонка или путем проникновения в организацию под видом ее сотрудника [10].

К сожалению, атаки на пользователей имеют широкую распространенность из-за простоты в организации и не требуют специальных навыков атакующего. Предотвращение таких атак со стороны платежной системы может сводиться только к информированию пользователей о недопустимости передачи личных данных кому-либо или к рекомендациям по защите персонального компьютера.

Исходя из перечисленных выше типов атак и распространенности их применения к платежным системам в Интернете рассмотрим самую распространенную DDoS-атаку и способы защиты от нее.

Главной целью платежной системы является доступность, надежность проведения платежа и отказоустойчивость. С точки зрения конечного пользователя, система должна в приемлемые

сроки принимать платежи, а также предоставлять отчеты, показывающие движение средств на счете пользователя. Таким образом, на первый план при DDoS-атаке выходит время отклика системы и создание гибкой архитектуры, позволяющей балансировать нагрузку. Выделяют следующие виды DDoS-атак [11]:

1. UDP flood — отправка на адрес системы-мишени множества пакетов UDP (User Datagram Protocol). Этот метод использовался в ранних атаках и в настоящее время считается наименее опасным.

2. TCP flood — отправка на адрес мишени множества TCP-пакетов, что также приводит к «связыванию» сетевых ресурсов.

3. TCP SYN flood — посылка большого количества запросов на инициализацию TCP-соединений с узлом-мишенью, которому в результате приходится расходовать все свои ресурсы на то, чтобы отслеживать эти частично открытые соединения.

4. Smurf-атака — пинг-запросы ICMP (Internet Control Message Protocol) по адресу направленной широковещательной рассылки с использованием в пакетах фальшивого адреса источника. Таким образом создается большой объем данных, которые будут посланы в ответ.

5. ICMP flood — атака, аналогичная Smurf, но без использования рассылки.

Самое большое распространение получила DDoS-атака с использованием TCP SYN flood, поскольку в данном случае атакующим посылается всего один пакет на открытие соединения. Такие атаки не нуждаются в обратной связи с атакующим, и поэтому можно не использовать настоящий адрес источника. Потому в отличие от остальных видов атак нет возможности построить список «плохих» IP-адресов, с которых производится атака.

Но в любом случае для защиты от DDoS-атак требуется высокоскоростной канал доступа к Интернет, желательно дублированный. Одним из решений для защиты от атак может служить приобретение автономной системы (АС) Интернет. Это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом [12]. Таким образом, приобретая АС, мы получаем свой собственный сегмент сети, который не зависит от остальной части Интернета. Данный сегмент глобальной сети может полностью перестраиваться под конкретные задачи. Внутри АС можно применять любой тип маршрутизации, как основанный на статических правилах, так и использующий семейство протоколов динамической маршрутизации. Например: RIP, OSPF и др. Каждая система имеет информационные каналы с другими системами, которые входят в состав базовой (backbone) сети Интернет. В нашем случае роль соседей исполняют провайдеры Интернета. Каждая АС имеет свой уникальный номер, назначаемый централизованно. Именно его применяет для идентификации автономных систем протокол BGP (Border Gateway Protocol). Сетям, входящим в состав автономной системы корпоративного клиента, присваивают провайдеро-независимые IP-адреса. На сегодняшний день можно зарегистрировать в Интернете до 4096 адресов, принадлежащих одной автономной системе. Маршрутизаторы на границах автономных систем общаются между собой с помощью одного из протоколов внешней маршрутизации BGP.

Используя протокол BGP, можно в любой момент времени своими силами управлять маршрутизацией своих сетей в Интернете. Главной особенностью данного протокола является возможность построения автоматически включаемых резервных каналов, на каждый из которых можно направить часть нагрузки. Поскольку может быть сколько угодно провайдеров, к которым присоединена автономная система, то при атаке надо будет заблокировать все эти каналы (рис. 1), чтобы таким образом сделать ресурс недоступным для пользователей. Поэтому задача атакующего усложняется, так как ему теперь надо полностью заблокировать не один канал связи, а несколько. При этом не известны общее количество возможных подключений автономной системы к сети

Интернет и совокупная пропускная способность всех каналов. Также имея большое пространство неиспользуемых IP-адресов, платежная система может изменять адрес сайта, тем самым уходя из-под атаки, до того момента пока атакующие не заметят, что адрес сменился, и не перенастроят Botnet, а это требует дополнительного времени.

Для предотвращения атак на серверы системы используется отдельный Firewall, который способен балансировать нагрузку между серверами приложений, а также контролировать запросы, приходящие в систему. Трафик от серверов приложений к внутренним ресурсам сети также отдельно контролируется, образуя демилитаризованную зону. Демилитаризованная зона — технология обеспечения защиты информационного периметра, при которой серверы, отвечающие на запросы из внешней сети или направляющие туда запросы, находятся в особом сегменте сети и ограничены в доступе к основным сегментам [13].

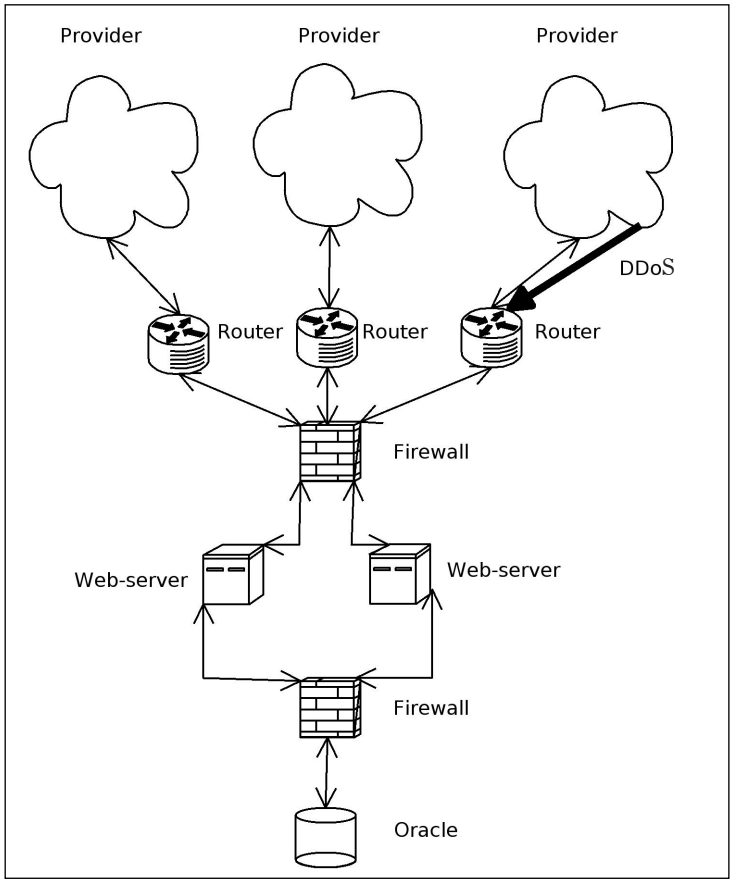

Рис. 1. Диаграмма сетевой архитектуры платежной системы

Также возможны атаки, целью которых является вывод из строя системы по приему платежей путем создания огромного потока фиктивных платежей и в итоге увеличения времени обработки запроса до неприемлемого.

Рассмотрим систему, когда пользователь использует платежный терминал для осуществления платежей, покупок в интернет-магазинах, а также для последующего просмотра отчетов по движению средств на балансе. Также существует ряд дополнительных функций, не связанных с платежами, которые могут привлечь новых пользователей. Благодаря широкому распространению терминалов одновременно значительное количество запросов должно обрабатываться системой с приемлемой задержкой, которая для удобного взаимодействия с человеком не должна превышать 5 секунд. Для более полного понимания всех этапов, на которых происходит задержка в обработке запроса,

рассмотрим диаграмму последовательности при работе с системой моментальных платежей на примере запроса на совершение платежа. Этот запрос является самым приоритетным для системы.

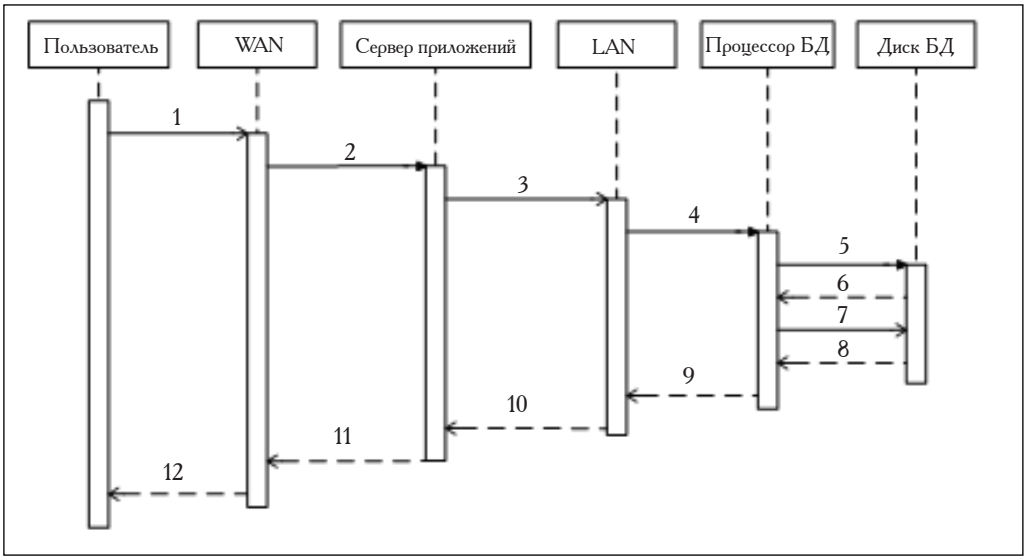

Рис. 2. Диаграмма последовательности платежной системы

Общее время отклика системы складывается из следующих задержек:

1. Пользователь, решив, что он хочет получить от системы, инициирует запрос на получение данных, посылает HTTP-пакет в глобальную сеть (WAN).

2. Некоторое время спустя запрос, пройдя по глобальной сети, достигает сервера приложений.

3. Сервер приложений, выполнив некоторые прикладные операции промежуточного уровня, формирует запрос к базе данных и отправляет его по протоколу SQL*Net в локальную сеть (LAN).

4. Запрос за некоторое время (быстрее, чем в глобальной сети) проходит по локальной сети и поступает на сервер базы данных.

5. Процессор сервера БД выполняет ряд вычислений, и ядро формирует системный вызов с запросом на чтение с диска.

6. Через некоторое время системный вызов заканчивает чтение и возвращает управление процессу сервера БД.

7. Выполнив ряд вычислений на сервере, ядро формирует еще один вызов на чтение.

8. Затратив еще некоторое время на выполнение дисковых операций, системный вызов снова возвращает управление процессу сервера БД.

9. Ядро, затратив часть процессорного времени на завершающую обработку, формирует ответ на запрос сервера приложений и отправляет его по протоколу SQL*Net в локальную сеть.

10. Ответ проходит по локальной сети и поступает на сервер приложений.

11. Сервер приложений преобразует полученные из базы данных результаты и отправляет их по глобальной сети по протоколу HTTP.

12. Результат запроса через глобальную сеть возвращается клиенту.

Из данной диаграммы видно, что возможна оптимизация системы на 3 уровнях:

1. На сетевом уровне, который формирует задержки при передаче информации в распределенной системе. Это задержки в глобальной сети (2, 12) а также задержки в локальной сети системы (4, 10).

2. На уровне сервера приложений, который обрабатывает запросы от пользователей, берет на себя выполнения основы алгоритмов по обработке запроса и формирует ответ (3, 11).

3. На уровне базы данных, которая является самой важной частью системы, в ней ведутся все расчеты по движению денег, хранится баланс для каждого пользователя. Потеря этой информации является критической для предприятия. При запросе на БД приходится основная задержка (5, 6, 7, 8, 9).

Указанные пути оптимизации надо применять в комплексе, поскольку каждый уровень вносит свою часть в общее время выполнения запроса. И надо рассмотреть среднюю задержку на каждом уровне. Непрогнозируемым временем является время доставки запроса в систему и обратно пользователю, тут возможно его уменьшить только с помощью увеличения пропускной способности канала связи системы и глобальной сети Интернет. Остальное время запрос обрабатывается системой, и тут оптимизация приобретает особую важность.

Рассмотрим, из чего складывается время обработки запроса к базе данных. В качестве СУБД мы будем использовать Oracle 10, которая является признанным лидером на рынке промышленных баз данных, а также отлично документирована. Oracle позволяет получить как подробную статистику по выполнению отдельного запроса, так и общую системную статистику по загруженности системы и задержкам на различных этапах выполнения запросов.

Время ожидания Oracle, измеренное при выполнении системного вызова, — это общее фактическое время, прошедшее с момента последней инструкции, следующей за возвратом вызова ОС, и до первой инструкции, следующей за возвратом вызова ОС [3]. Рассмотрим графическое представление времени ожидания.

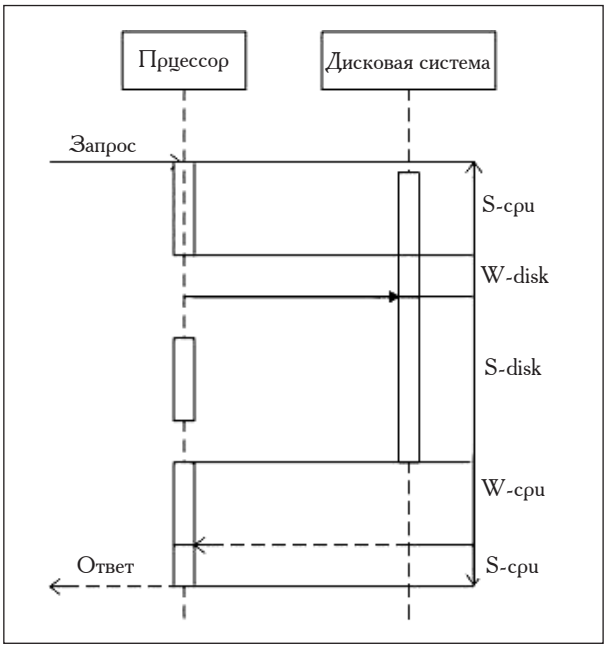

Рис. 3. Время ожидания Oracle — фактическое время отклика

Из графика видно, что время выполнения запроса состоит из:

1. S-cpu — времени процессора, потраченного на подготовку системного вызова.

2. W-disk — задержки в очереди к дисковому устройству, включает в себя задержку передачи запроса от процессора.

3. S-disk — времени работы дискового устройства, включая задержку поиска, задержку передачи данных.

4. W-cpu — задержки в очереди к процессору.

5. S-cpu — времени, необходимого процессору для завершения запроса.

Таким образом, для оптимизации работы с БД можно уменьшить задержки, описанные выше. Это может достигаться различными путями: как наращиванием производительности оборудования, на котором работает СУБД, так и оптимизацией запросов, посылаемых сервером приложений.

Рассмотренная классификация атак была выработана на основе опыта эксплуатации реально существующей платежной системы. За это время были выявлены атаки как на саму систему, так и на ее пользователей. В последнее время в связи с распространением платежных систем все больше пользователей становятся жертвами обмана со стороны мошенников, использующих приемы социальной инженерии. Для предотвращения атак на систему была разработана вышеописанная архитектура, позволяющая эффективно защищаться от различного типа DDoS-атак, используя автономную систему Интернет с резервированием каналов связи и балансировкой нагрузки. Также был произведен анализ задержек, возникающих как на уровне серверов приложений, так и на уровне базы данных при нагрузках, превышающих среднее значение в несколько раз. Анализ самых важных запросов дал необходимые данные о потоке запросов, загрузке процессоров сервера СУБД, а также о среднем времени выполнения операций.

СПИСОК ЛИТЕРАТУРЫ:

1. *Donal O. Mahony, Michael Peirce, Hitesh Tewari*. Electronic Payment Systems for E-Commerce. Second Edition. Artech House Boston, 2001.
2. *Голдовский И. М*. Безопасность платежей в Интернете. СПб.: ИД «Питер», 2001
3. *Миллсан Хольт*. Oracle оптимизация производительности. М.: Символ-Плюс, 2005.
4. *Конноли Т., Бегг К*. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. М.: Вильямс, 2003.
5. *Шаньгин В. Ф*. Информационная безопасность компьютерных систем и сетей. М., 2008.
6. *Стрельцов А. А*. Правовое обеспечение информационной безопасности России: Теоретические и методологические основы. Мн., 2003.
7. http://ru.wikipedia.org/wiki/DoS.
8. http://ru.wikipedia.org/wiki/Exploit.
9. http://ru.wikipedia.org/wiki/Botnet.
10. http://ru.wikipedia.org/wiki/Социальная_Инженерия.
11. http://argon.com.ru/internet/attack.
12. http://ru.wikipedia.org/wiki/Автономная_система(интернет).
13. http://ru.wikipedia.org/wiki/DMZ_(Компьюреные_сети).