УДК 004.773

С.А. ВАХНИН, А.М. БРАТАНИН

БЕЗОПАСНОЕ УПРАВЛЕНИЕ УДАЛЕННЫМ ХОСТОМ ЧЕРЕЗ НЕЗАЩИЩЕННЫЙ СЕРВЕР

Анотація. *Розглянуто проблему безпечного управління комп'ютером за умов відсутності реальних ip-адрес. Запропоновано та описано структуру програмного комплексу, що вирішує поставлену проблему.*
Ключові слова: *віддалене управління, комп'ютерні мережі, захист інформації.*

Аннотация. *Рассмотрена проблема безопасного удаленного управления компьютером при условии отсутствия реальных ip-адресов. Предложена и описана структура программного комплекса, решающего поставленную проблему.*
Ключевые слова*: удаленное управление, компьютерные сети, защита информации.*

Abstract. *The problem of safe computer management under the conditions of the absence of real ip-addresses is regarded. The structure of bundled software solving this problem was proposed and described.*
Keywords: *remote management, computer networks, information security.*

1. Вступление

Большинство компьютеров, подключенных к сети Интернет, не имеют возможности принимать входящие подключения, что обусловлено нехваткой реальных ip-адресов.
Эта проблема может быть решена переходом на IPv6, но на данный момент прогнозы по внедрению этого протокола отсутствуют [1].

Для решения проблемы нехватки ip-адресов в Интернете широко применяется технология создания виртуальных хостов, которая предусматривает привязку множества доменных имен к одному ip-адресу, эта технология оправдывает себя и широко используется лишь для протокола прикладного уровня HTTP (HyperText Transfer Protocol). Провайдеры, в свою очередь, используют механизм преобразования сетевых адресов Network Address Translation (NAT), который дает возможность нескольким пользователям работать от имени одного ip-адреса, но ограничивает возможность приема входящих tcp-подключений.

Все перечисленные методы не предусматривают возможность устанавливать прямое подключение между двумя компьютерами сети Интернет, которое необходимо для создания безопасного соединения и управления удаленным хостом. Далее в статье речь пойдет о проблеме управления удаленным хостом в условиях отсутствия реальных ip-адресов на обеих (управляемой и управляющей) сторонах.

2. Постановка задачи

Можно сделать вывод, что связь между двумя компьютерами, подключенными к сети Интернет и не имеющими реальных ip-адресов, физически невозможна. Таким образом, следует ввести между ними третий компьютер, который будет исполнять роль сервера. На таком принципе основано множество сетевых протоколов высокого уровня: SMTP/POP3, ICQ, Skype, VPN. В случае размещения сервера в Интернете нельзя с полной уверенностью гарантировать то, что данные, хранящиеся на сервере или передающиеся через него, не будут скомпрометированы со стороны администрации сервера. Таким образом, будем считать, что сервер не является защищенным и шифрование данных должно выполняться на клиентах.

Задача состоит в разработке программного комплекса, который обеспечивает передачу и выполнение команд между двумя точками: управляемым и управляющим компьютерами.

Программный комплекс состоит из четырех частей:

– программное обеспечение (ПО) сервера, которое используется как промежуточное звено для передачи данных между клиентами с отсутствующим реальным ip-адресом;

– программа Master, которая устанавливается на управляющем компьютере и выполняет необходимые функции для передачи команд управляемому компьютеру через сервер;

– программа Slave, выполняющая прием и выполнение команд от программы Master, использующая сервер как промежуточное звено передачи данных;

– программа для управления ключами, обеспечивающая безопасную генерацию и обмен ключами.

Важным является также вопрос выбора сервера, который будет выполнять роль промежуточного звена при передаче команд от управляющего компьютера управляемому.

3. ПО сервера

Для определения средств, которые будем применять при разработке программного обеспечения сервера, следует определить его тип. Можно условно выделить пять типов предоставляемого хостинга [2]:

1. Общий виртуальный хостинг (Shared-hosting) – веб-сервер, на котором размещено несколько веб-сайтов. Для определения клиентом необходимого сайта используется технология виртуальных хостов.

2. Виртуальный выделенный сервер (VPS – Virtual private server, VPS-хостинг) – сервер делится на несколько частей и клиенту предоставляется отдельная виртуальная машина с автономной операционной системой, которой он может распоряжаться по своему усмотрению.

3. Выделенный сервер (VDS – Virtual Dedicated server, Dedicated-hosting) – в аренду предоставляется отдельный компьютер.

4. Облачный сервер (Cloud Server, Cloud-hosting) – сервер с возможностью неограниченного расширения.

5. Colocation – подразумевает аренду места на серверной площадке для размещения компьютера клиента.

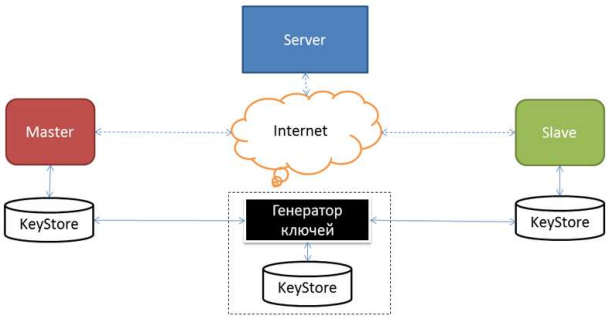

Рис. 1. Структурная схема программного обеспечения

В данном случае мы выберем общий виртуальный хостинг с поддержкой PHP и MySql, так как он самый распространённый и самый доступный. Недостатком такого решения является обязательное использование HTTP протокола, который в данном случае является избыточным. Структурная схема всего программного комплекса приведена на рис. 1.

Как показано на рисунке, все элементы системы, кроме генератора ключей, взаимодействуют через сеть Интернет. Также, у управляемого и управляющего компьютеров есть хранилище ключей, которое используется для хранения закрытых ключей и сертификатов. Генератор ключей используется для передачи ключей между участниками диалога и взаимодействует с ними только через безопасные каналы. В данном случае хранилище и генератор ключей должны находиться на отдельном накопителе. Основной задачей сервера является пересылка со-

общений между участниками диалога и хранение необработанных сообщений. Также сервер выполняет функции авторизации и аутентификации.

В отличие от управляемого и управляющего компьютеров, сервер может выполнять два вида диалога: диалог с управляемым компьютером и диалог с управляющим компьютером, каждый из которых имеет свои особенности.

Диалог с управляющим компьютером состоит из следующих этапов:

1. Авторизация – сервер принимает и проверяет логин и пароль программы Master и возвращает ей идентификатор сессии, по которому эта программа будет идентифицирована на остальных этапах диалога.

2. Прием команды.

3. Отправка следующего в очереди статуса или ответа.

4. Прием статуса «Закрыто». Получив такой статус, сервер удаляет сообщение, для которого установлен этот статус.

Диалог с управляемым компьютером:

1. Авторизация.

2. Прием запроса на следующую команду или статус.

3. Отправка команды или статуса.

4. Прием подтверждения статуса.

5. Прием ответа на команду.

Общая схема взаимодействия компонентов системы представлена на рис. 2 в виде диаграммы последовательности.

Диаграмма отображает такие действия:

1. Управляющий компьютер передает на сервер свой логин и пароль, сервер проверяет их и, если логин и пароль верны, возвращает идентификатор сессии, который далее будет отправляться управляющим компьютером вместе с каждым последующим сообщением. Такой подход дает возможность минимизировать количество передач логина и пароля.

2. Управляющий компьютер передает на сервер текстовую команду и двоичные данные, которые ее дополняют и могут содержать набор параметров команды. Также передается электронная цифровая подпись (ЭЦП) команды и данных, что предотвращает возможность их подмены или модификации. Двоичные данные зашифрованы открытым ключом управляемого компьютера и могут быть расшифрованы только им.

3. Аутентификация и авторизация управляемого компьютера происходят аналогичным образом, как для управляющего компьютера.

4. Управляемый компьютер запрашивает список невыполненных команд и выполняет их. Результат выполнения каждой команды отправляется на сервер в виде текстового результата выполнения команды, зашифрованных двоичных данных и ЭЦП. Также существует сквозной счетчик команд, значение которого также подписывается каждой из сторон при передаче команды или результата ее выполнения. Такой подход исключает возможность ретрансляции команды злоумышленником.

5. Сервер сохраняет результат выполнения команды в базе данных до приема результатов и закрытия команды управляющим компьютером.

6. Управляющий компьютер посылает запрос на список выполненных (но не закрытых команд), сохраняет их в локальной базе данных, после чего отправляет запрос на закрытие каждой принятой команды, что приводит к их удалению на сервере.

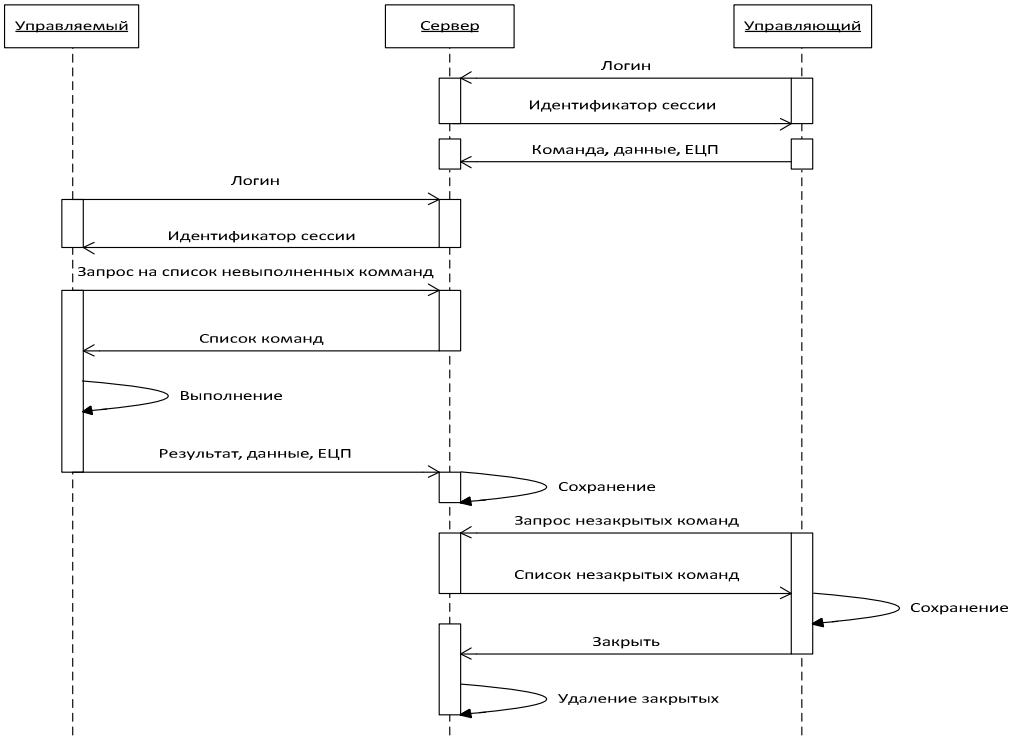

Рис. 2. Последовательность взаимодействия компонентов

Как видно из приведённого списка, сервер не выполняет никаких действий, связанных с криптографическими преобразованиями, и не хранит (или передает) данные, критичные к безопасности в открытом виде. Таким образом, даже получив доступ к серверу, максимальный ущерб, который может быть причинен злоумышленником, это обрыв связи между управляющим и управляемым компьютерами.

4. ПО Клиентов

Для реализации программного обеспечения управляющего и управляемого компьютера может быть использован любой язык программирования, поддерживающий выполнение криптографических преобразований. Но для конкретной реализации выбран язык программирования Java, так как, кроме поддержки алгоритмов шифрования, он также является мультиплатформенным, что значительно расширяет сферу применения программного комплекса. Программы Master и Slave целесообразно реализовывать в виде консольных приложений, управляемых через конфигурационные файлы и socket-интерфейс. Это дает возможность запуска на платформах без оконного графического интерфейса и возможность для разработки графического интерфейса сторонними разработчиками. Для хранения данных на стороне управляемого компьютера предполагается использование встроенной базы данных, которая не требует установки отдельного сервера баз данных [3]. В роли такой базы данных может служить SQLite, она также поддерживается и некоторыми мобильными устройствами.

5. Задача генератора ключей

Генератор ключей – это часть программного комплекса, которая обеспечивает генерацию и передачу ключей между клиентом и сервером. Для высокой надежности генератор должен быть реализован в виде отдельного устройства [4], но в конкретном случае предпола-

гается использование флеш накопителя с записанной на него виртуальной Java-машиной и программой генерации ключей.

Процедура генерации и обмена ключами через безопасный канал состоит из следующих шагов:

1. Генерация основного закрытого ключа и его сертификата (CA).

2. Генерация закрытого ключа и сертификата программы Master, подпись сгенерированного сертификата основным закрытым ключом.

3. Передача закрытого ключа и сертификата CA в хранилище ключей на компьютер программы Master, уничтожение переданного закрытого ключа на флеш накопителе.

4. Генерация закрытого ключа и сертификата программы Slave, подпись сгенерированного сертификата основным закрытым ключом.

5. Передача закрытого ключа и сертификата CA в хранилище ключей на компьютер программы Slave, уничтожение переданного закрытого ключа на флеш накопителе.

После выполнения такой процедуры обмен открытыми ключами между программами Master и Slave может быть выполнен через незащищенный канал, без угрозы MITM-атаки (Man in the middle), так как сертификат CA был передан по защищенному каналу, а открытые ключи были подписаны им.

6. Выводы

В статье рассмотрена проблема взаимодействия компьютеров при отсутствии реального ip-адреса. Поставлена задача разработки программного комплекса по обеспечению передачи команд между двумя компьютерами через незащищенный сервер. Описаны алгоритмы и принципы работы программного комплекса, решающего поставленную задачу. Продолжением рассмотренной темы может быть взаимодействие между группой компьютеров.

СПИСОК ЛИТЕРАТУРЫ

1. http://habrahabr.ru/blogs/internet/117926.
2. http://www.infohost.net.ru/articles/vidy_host.htm.
3. http://www.sqlite.org.
4. Мельников В.П. Информационная безопасность и защита информации / Мельников В.П., Клейменов С.А., Петраков А.М.; под ред. С.А. Клейменова. – [3-е изд.]. – М.: Издательский центр «Академия», 2008. – 336 с.
5. Нильс Ф. Практическая криптография / Ф. Нильс, Ш. Брюс. – М.: Издательский дом «Вильямс», 2005. – 416 с.
6. Heffelfinger D.R. Java EE 6 Development with NetBeans 7 / Heffelfinger D.R. – Birmingham, UK, 2011. – 392 p.

Стаття надійшла до редакції 08.11.2011