ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ №1(24), 2015, с. 147–156

УДК 519.682.3

А. А. Кузнецов

Исследование устойчивости сетевого протокола BotikKey к подбору пароля доступа методом прямого перебора

Аннотация. В работе приведено исследование устойчивости к различного рода атакам сетевого протокола BotikKey, который используется в системе телекоммуникаций «Ботик» г. Переславля-Залесского для аутентификации абонентских подключений. Протокол разработан в рамках подхода Ботик-технологий, согласно которому все программно-аппаратное обеспечение сети «Ботик» является либо свободнораспространяемым, либо разработано собственными усилиями компании-провайдера. В работе представлено назначение протокола, понятие пароля, ключа доступа, региона доступа, и схема выполнения сетевых обменов между клиентом и сервером BotikKey. Перечислены уязвимости протокола BotikKey: подбор ключа доступа методом прямого перебора на параллельных вычислительных системах, либо на «облачных» сервисах, подбор BotikKey-пароля с использованием радужных таблиц для функции хеширования MD5, SSL-атака на пользователей протокола с целью подбора пароля доступа к сети Интернет, описание последствий кражи пароля из файловой системы. Даны рекомендации провайдеру услуг связи системы телекоммуникаций «Ботик» по отказу от системы BotikKey, либо переходу на более актуальные средства аутентификации абонентских подключений.

Ключевые слова и фразы: протокол BotikKey, доступ к сети, безопасная аутентификация абонентов, радужные таблицы, прямой перебор пароля.

Введение

В системах, построенных по технологии Ethernet, серьезной проблемой является привязка услуги к абоненту. Очень часто ключом такой привязки является IP-номер, присвоенный подключениям абонента. Но нельзя рассчитывать на то, что этот IP-номер не присвоит себе кто-либо посторонний. Это может произойти либо по злому умыслу, либо по недосмотру, по ошибке. В этом случае произойдет кража

трафика: передача данных, выполненная в интересах одного абонента, будет оплачена другим абонентом. Таким образом, требуется система, не позволяющая осуществить подобную кражу трафика. Кроме того, будет очень полезно, если эта система управляла бы уровнем доступа абонента к сети. Это связано с тем, что очень часто сетевые обмены с компьютером абонента происходят неожиданно для самого абонента. На современных компьютерах могут работать программные средства, которые обращаются в сеть Интернет без ведома абонента. Они могут быть доброкачественными средствами (например, системы автообновления программных продуктов), либо вредоносными средствами (вирусы, трояны), которые без ведома абонента ведут сетевые обмены на его компьютере. Чтобы этого не происходило, можно дать абоненту возможность управления уровнем доступа: запрет всех сетевых обменов, либо разрешение только на бесплатные локальные (внутригородские) обмены, либо разрешение на доступ ко всей сети Интернет. В этом случае абонент может включать глобальный доступ тогда, когда это нужно только ему. Все эти задачи решаются при помощи клиент-серверной системы BotikKey.

В телекоммуникационной сети «Ботик» г. Переславля-Залесского система BotikKey применяется с 2003 года для аутентификации абонентов. С 2007 года использование программы BotikKey из набора программ «BotikTools» [1] стало обязательным для постоянных подключений абонентов СТ «Ботик».

В работе [3] приведено описание протокола Botikkey. Данная работа посвящена описанию защиты протокола Botikkey от компрометации. Здесь под компрометацией понимается ситуация, при которой злоумышленнику удалось выдать себя за добросовестного абонента и аутентифицировать свой компьютер на BotikKey-сервере. Исследованы проблемы защиты протокола от различных видов атак, таких, как прямой перебор, «человек посередине» и кража ключа доступа с компьютера абонента (либо с принадлежащих ему съемных носителей).

1. Личный кабинет абонента и задание паролей BotikKey

При подключении нового устройства (компьютера) абонента к сети «Ботик» сотрудники компании-провайдера на странице подключения в личном кабинете на веб-сайте «Надмин: Абоненту» заводят новый BotikKey-пароль для аутентификации данного абонентского подключения. Как правило, устанавливается один пароль для региона

доступа «WORLD» (означает полный доступ к ресурсам сети Интернет). Режимы доступа и интерфейс клиентской программы подробнее рассмотрены в [2].

После установки пароля и ключа доступа эта информация рассылается всем территориально-распределенным шлюзам сети «Ботик», на которых установлен BotikKey-сервер. Провайдер рекомендует выбирать достаточно сложный пароль доступа: от 8 знаков, содержащий различные символы алфавита в разных регистрах, цифры и спецсимволы. После установки паролей в личном кабинете абонента сотрудник компании-провайдера производит настройку клиентской программы BotikKey из пакета программ BotikTools. Эта реализация клиента инсталлируется на компьютер абонента по умолчанию. На сайте BotikTools представлены также альтернативные реализации клиента BotikKey для иных программно-аппаратных платформ. Поддерживаются различные программно-аппаратные конфигурации (персональный компьютер, серверы, роутеры, смартфоны, планшеты и др.).

В окне настроек клиентской программы BotikKey указывается (помимо прочего) пароль доступа и способ хранения ключа доступа (MD5-сумма от пароля) — в переменной среды, в умолчательном файле на компьютере, либо в специальном хранилище (например, на съемном USB-носителе). Пароль указывается при настройке программы всего один раз, а для аутентификации абонента используется только ключ доступа — MD5-свертка от пароля.

2. Способы компрометации протокола BotikKey

Сокет-соединение между клиентом и сервером небезопасно. Если на абонента будет произведена атака типа «Человек посередине» («Man in the Middle» [5]), то через злоумышленника будет проходить весь Интернет-трафик абонента, в том числе все коммуникации между клиентом и сервером BotikKey. После каждого сеанса обмена данными будут перехвачены следующие данные:

- строка *challenge*;
- строка *random*;
- строка *region*;
- строка *secret*.

Есть ненулевая вероятность компрометации протокола BotikKey на основе полученных данных. Противник должен за разумное время (несколько месяцев, пока абонент не сменил пароль) подобрать

такой ключ *fakedKey*, чтобы MD5(*salt*·*fakedKey*) и MD5(*salt*·*accessKey*) совпали.

Один раз подобрав пароль BotikKey либо ключ доступа, противник сможет впоследствии продолжительное время оставаться незамеченным, пользоваться чужими TCP/IP-настройками и выдавать себя за абонента. Как будет показано ниже, на данном этапе развития вычислительной техники задача подбора пароля не является трудноразрешимой.

2.1. Подбор ключей доступа

1-й вариант — прямой перебор всех 32-значных хеш-сумм с целью подбора **ключа доступа**. Требуется перебрать $16^{32} \approx 3.4 \cdot 10^{38}$ хеш-сумм, что займет значительное время даже на высокопроизводительном вычислительном кластере, оборудованном графическими ускорителями. Автором был проведен следующий эксперимент. С помощью общедоступной утилиты «Hashcat» (CUDA-версия) [4] на одном из узлов вычислительного Linux-кластера, содержащем графический ускоритель NVidia Tesla K20c, была произведена попытка обращения хеш-суммы. Программе в качестве опций запуска были указаны:

(1) алфавит (0123456789abcdef);
(2) количество символов в пароле (с помощью маски «?1?1…?1» (всего 32 позиции));
(3) *salt*;
(4) некоторая хеш-сумма: MD5(*salt*·*accessKey*);
(5) режим подбора: md5($salt.$pass).

Утилита перебирала все возможные хеш-суммы для *fakedKey* до тех пор, пока свертка MD5(*salt*·*fakedKey*) не совпадет с заданной хеш-суммой MD5(*salt*·*accessKey*). Через некоторое время после запуска программа выдала прогноз завершения, согласно которому на устройстве NVidia Tesla K20c перебор всех вариантов хеш-суммы займет более 10 лет. Можно сделать вывод об абсолютной бесперспективности такого метода перебора.

2.2. Подбор паролей BotikKey

2-й вариант — можно попытаться скомпрометировать протокол BotikKey через подбор всех возможных паролей *fakedKey* так, чтобы хеш-сумма MD5(*salt*·*fakedKey*) совпала с перехваченной хеш-суммой MD5(*salt*·*accessKey*). Вычислительная сложность такого перебора

Таблица 1. Прогнозируемое время подбора пароля

	7 знаков	8 знаков	9 знаков	10 знаков
L.U.	41 минута	1 день 12 часов	78 дней	$>$ 10 лет
L.U.D.	2 часа 22 мин.	6 дней	1 год 15 дней	$>$ 10 лет
L.U.D.S.	1 день 22 часа	187 дней	$>$ 10 лет	$>$ 10 лет

прямо пропорциональна сложности пароля, указанного пользователем либо сотрудником компании-провайдера в личном кабинете «Надмин: Абоненту».

С помощью утилиты «Hashcat» (CUDA-версия) на том же вычислительном кластере, содержащем один графический ускоритель NVidia Tesla K20c, было произведено несколько попыток подбора пароля. Программе в качестве опций запуска были указаны:

(1) алфавиты: L.U., L.U.D., L.U.D.S., где:
  L. означает строчные латинские символы;
  U. означает заглавные латинские символы;
  D. означает цифры;
  S. означает специальные символы;
  например: L.U. означает алфавит, состоящий из знаков a–zA–Z;
(2) количество символов в пароле: 7, 8, 9, 10;
(3) строка *salt*;
(4) перехваченная хеш-сумма: MD5(*salt*·*accessKey*);
(5) режим перебора: md5($salt.$pass).

Было произведено 12 тестовых запусков программы для подбора пароля на разных комбинациях пар (алфавит, количество знаков пароля). Во время работы программа печатает прогноз завершения полного перебора всех вариантов пароля (всех перестановок знаков). Результаты прогнозирования представлены в таблице 1.

3. Другие методы компрометации протокола BotikKey

3.1. Проникновение в файловую систему

Протокол BotikKey может быть скомпрометирован, если злоумышленник тем или иным образом получит доступ к ключу *accessKey*, который вычислен как хеш-сумма от BotikKey-пароля и хранится в файловой системе пользователя, либо на съемном носителе. Имея этот ключ, будет очень просто сформировать правильный ответ (*response*)

на запрос (*challenge*), как следствие атакующему будет предоставлен доступ к сети и он без труда будет выдавать себя за абонента.

Зная ключ доступа и используя такие утилиты как «Hashcat», злоумышленник может подобрать BotikKey-пароль. Этот пароль может использоваться данным абонентом и для других сервисов в сети Интернет. Таким образом, криптоустойчивость протокола BotikKey напрямую влияет на защищенность других сервисов и систем, используемых абонентом.

3.2. Сервис Информер

В клиент BotikKey встроена функциональность для периодичного запроса данных об абоненте от сервиса «Информер». Задача сервиса состоит в том, чтобы авторизовать клиента по протоколу HTTPS и переслать ему важную информацию: остаток на лицевом счету, порог блокировки, текущий режим тарификации для абонентского подключения. Для получения этих данных клиент формирует HTTP POST-запрос вида random=$random&digest=$digest, где $random — это произвольная строка, а $digest представляет собой хеш-сумму MD5(random·*accessKey*), *accessKey* — ключ доступа (хеш-сумма от BotikKey-пароля).

Эта возможность подвержена MITM-атаке на SSL [5], при успешном исходе которой злоумышленник перехватит следующие данные:

(1) строка random;
(2) строка digest.

Осталось перебрать все возможные хеш-суммы *fakedKey* так, чтобы MD5(random·*fakedKey*) совпал с digest. Это весьма ресурсоемкая операция, как было показано выше.

Можно также перебрать все возможные пароли $pass так, чтобы строка MD5(random·md5($pass)) совпала с digest. Как уже было сказано, сложность такого перебора варьируется и прямо зависит от сложности пароля $pass.

3.3. Атака «Человек посередине»

Атака «Man in the Middle» [5] — метод компрометации канала связи, при котором злоумышленник получает возможность подслушивать и искажать информацию, передаваемую/получаемую абонентом по данному каналу связи. Противник может выполнить врезку в сеть между клиентом и сервером, установив специальный аппаратный

модуль на линию связи. Устройство осуществляет туннелинг трафика абонентского подключения, кроме того (после успешного подбора ключа BotikKey) выполняет свои собственные задачи в сети Интернет от имени абонента.

3.4. Облачные сервисы

Существует ряд «облачных» веб-сервисов, предоставляющих своим клиентам вычислительные ресурсы (в том числе на основе CUDA-архитектуры) на платной основе. Мощность ресурсов (количество узлов вычислительных кластеров) может быть весьма значительной и зависит от объема финансовых средств, которыми располагает злоумышленник. На основе этих веб-сервисов может быть организован прямой перебор (в том числе распределенный), с ненулевым шансом на успешный исход.

3.5. Радужные таблицы

Радужные таблицы [6] — это специальный вариант таблиц поиска, одно из средств быстрого обращения хеш-сумм. Они представляют собой структуры данных, содержащие предвычисленные хеш-суммы для всех сочетаний символов заданного алфавита для заданной длины цепочки. Эти структуры разработаны исходя из принципа компромисса между временем поиска по таблице и занимаемой памятью.

Радужная таблица создается построением цепочек возможных паролей. Каждая цепочка начинается со случайного возможного пароля, затем подвергается действию хеш-функции и функции редукции. Данная функция преобразует результат хеш-функции в некоторый возможный пароль. Промежуточные пароли в цепочке отбрасываются и в таблицу записываются только первый и последний элементы цепочек. Во время генерации радужных таблиц расходуется больше вычислительных ресурсов, чем на создание обычных таблиц поиска, но значительно меньше памяти на жестком диске.

В целях эксперимента автором были сгенерированы радужные таблицы для хеш-функции MD5 для L.U.D.S.-алфавита (см. выше) для цепочек (паролей) длиной от 1 до 7 символов. Генерация заняла 47 дней и производилась с использованием программного инструментария «Rainbow crack» [7] на небольшом Linux-кластере. Процесс был размножен на несколько узлов кластера — механизм генерации радужной таблицы позволяет разбить ее на несколько файлов, каждый

из которых порождается на своем узле вычислительного кластера. Используемый инструментарий имеет свойство многопоточности, что позволило задействовать все ядра процессора на каждом из узлов. Таблицы на жестком диске занимают 128 ГБайт в несжатом формате (.rt). При необходимости процесс сжатия в формат .rtc позволит сократить на 50% занимаемое место на диске и ускорить время поиска.

Попытка обращения любой MD5-суммы с использованием полученных радужных таблиц и инструментария «Rainbow crack» будет успешной в 99,9% случаев. Если попытаться перенастроить параметры генерации таблиц, можно добиться 100% вероятности успешного поиска, но при этом будет нарушен вышеупомянутый компромисс и возникнет сильный перевес в сторону занимаемой таблицами памяти.

Поиск пароля по MD5-сумме с использованием радужных таблиц производится утилитой rcrack, которая входит в инструментарий «Rainbow crack». В наших тестах обращение хеш-сумм занимает в среднем 9 минут на один хеш (количество обращенных хеш-сумм: 20). Открытые тексты (пароли) сгенерированы при помощи вызова утилиты pwgen с опциями -c -n -y 7 20.

Заключение

По результатам проведенного исследования можно сделать следующие выводы:

(1) Вероятность компрометации протокола BotikKey тем выше, чем проще BotikKey-пароль у абонента.
(2) С развитием вычислительных технологий будет проще выполнять обращение хеш-сумм, что может повлечь за собой угрозу пользователям системы и протокола BotikKey.
(3) От протокола BotikKey рекомендуется либо полностью отказаться, либо перейти на использование более криптостойких хеш-функций [8].

Каким бы ни был алгоритм аутентификации, нельзя исключать человеческий фактор. Персонал провайдера во время настройки подключения устанавливает достаточно стойкий к подбору BotikKey-пароль. Но абонент может в личном кабинете «Надмин: Абоненту» по разным причинам сменить пароль на более простой. В этом случае он берет на себя ответственность за собственную безопасность.

Если атаку производит группа хакеров, то приведенные выше методы будут использованы одновременно несколькими взломщиками.

В этом случае вероятность успешной компрометации протокола существенно выше.

Благодарности. *Работы, положенные в основу данной статьи, были выполнены в рамках НИР «Методы и программные средства разработки параллельных приложений и обеспечения функционирования вычислительных комплексов и сетей нового поколения». Автор выражает благодарности разработчикам системы и протокола BotikKey (С. М. Абрамов, Ю. В. Шевчук) и администратору локального Linux-кластера (М. Р. Коваленко) за полезные комментарии по содержанию статьи.*

Список литературы

[1] *Пакет программ BotikTools*, URL http://www.botik.ru/~botik/tools/index.ru.html ↑ 148.

[2] С. М. Абрамов, А. А. Кузнецов, «BotikTools — пакет программ для Абонентов научно-образовательной сети г. Переславля-Залесского», *Международная конференция «Программные системы: теория и приложения»*. Т. 1 (Переславль-Залесский, октябрь 2006), Наука. Физматлит, М., 2006, с. 135–154 ↑ 149.

[3] А. А. Кузнецов. «Исследование криптостойкости протокола аутентификации Botikkey к компрометации уязвимостей алгоритма хеширования MD5», *Программные системы: теория и приложения*, **6**:1(24) (2015), с. 135–144, URL http://psta.psiras.ru/read/psta2015_1_135-145.pdf ↑ 148.

[4] *Программа Hashcat*, URL http://hashcat.net/oclhashcat/ ↑ 150.

[5] URL https://en.wikipedia.org/wiki/Man-in-the-middle_attack ↑ 149, 152.

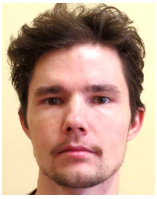
[6] URL https://en.wikipedia.org/wiki/Rainbow_table ↑ 153.

[7] *Проект RainbowCrack*, URL http://project-rainbowcrack.com/ ↑ 153.

[8] *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, URL http://csrc.nist.gov/publications/drafts/fips202/fips_202_draft.pdf ↑ 154.

Рекомендовал к публикации *д.ф.-м.н. С. В. Знаменский*

Об авторе:



Антон Александрович Кузнецов

Научный сотрудник ИЦМС ИПС им. А.К. Айламазяна РАН. Разработчик всех программ пакета BotikTools. Один из разработчиков системы OpenTS. Область научных интересов: системы параллельного программирования, распределенные вычисления в гетерогенных средах, геоинформационные системы.

e-mail: tonic@pereslavl.ru

Пример ссылки на эту публикацию:

А. А. Кузнецов. «Исследование устойчивости сетевого протокола BotikKey к подбору пароля доступа методом прямого перебора», *Программные системы: теория и приложения*, 2015, **6**:1(24), с. 147–156.

URL http://psta.psiras.ru/read/psta2015_1_147-156.pdf

Anton Kuznetsov. *Research into the issue of BotikKey protocol resistance to the brute force attack.*

Abstract. In this paper vulnerabilities of the BotikKey network protocol are described. The protocol is being used in the "Botik" telecommunication system of Pereslavl-Zalessky for secure subscriber authentication. Protocol was developed as part of Botik-technologies initiative, according to which all software and hardware is based on open source, or on the inhouse developments. We outline the purpose and implementation details of the protocol. It is pointed out that majority of protocol vulnerabilities arise from weaknesses of MD5 cryptographic hash function being used. BotikKey protocol can be compromised in several ways: brute force attack for recovering plain network password using specific software on high-performance computing devices and cloud services, password attack using rainbow tables for MD5 hash function, and the password theft. It is noted that "Botik" network service provider should use more contemporary cryptographic methods for subscriber authentication, or even avoid using the BotikKey system. (*In Russian*).

Key Words and Phrases: BotikKey protocol, network access, subscribers secure authentication, rainbow tables, brute force, password search.

Sample citation of this publication

Anton Kuznetsov. "Research into the issue of BotikKey protocol resistance to the brute force attack", *Program systems: theory and applications*, 2015, **6**:1(24), pp. 147–156. (*In Russian.*) URL http://psta.psiras.ru/read/psta2015_1_147-156.pdf