щённой ОССН *Astra Linux Special Edition* и заключающийся в теоретическом обосновании алгоритмически проверяемых и реализуемых на практике условий безопасности в ОССН информационных потоков по памяти и по времени.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учеб. пособие для вузов. 2-е изд., испр. и доп. М.: Горячая линия — Телеком, 2013. 338 с.
2. *Девянин П. Н.* Адаптация мандатной сущностно-ролевой ДП-модели к условиям функционирования ОС семейства Linux // Системы высокой доступности. 2013. № 3. С. 98–102.
3. *Девянин П. Н.* Администрирование системы в рамках мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками в ОС семейства Linux // Прикладная дискретная математика. 2013. № 4(22). С. 22–40.
4. Операционные системы Astra Linux. `http://www.astra-linux.ru/`
5. *Девянин П. Н.* Условия безопасности информационных потоков по памяти в рамках МРОСЛ ДП-модели // Прикладная дискретная математика. Приложение. 2014. № 7. С. 82–85.
6. *Девянин П. Н., Куликов Г. В., Хорошилов А. В.* Комплексное научно-обоснованное решение по разработке отечественной защищенной ОССН Astra Linux Special Edition // Методы и технические средства обеспечения безопасности информации: Материалы 23-й науч.-технич. конф. 30 июня–03 июля 2014 г. СПб.: Изд-во Политехн. ун-та, 2014. С. 29–33.

УДК 004.94 DOI 10.17223/2226308X/8/31

О ВОМОЖНОСТИ РЕАЛИЗАЦИИ СКРЫТЫХ КАНАЛОВ ПО ВРЕМЕНИ НА ОСНОВЕ ЗАГОЛОВКОВ КЭШИРОВАНИЯ ПРОТОКОЛА HTTP В ОБЛАЧНЫХ СЕРВИСАХ ХРАНЕНИЯ ФАЙЛОВ

Д. Н. Колегов, О. В. Брославский, Н. Е. Олексов

Показывается, как скрытые каналы по времени на основе заголовков кэширования протокола HTTP могут быть реализованы в облачных сервисах хранения файлов.

Ключевые слова: *HTTP, скрытые каналы, безопасность веб-приложений, ботсети.*

В [1] впервые предложено семейство скрытых каналов по времени на основе заголовков кэширования протокола HTTP, а также базовые сценарии их реализации. В [2] исследуются практические вопросы реализации таких скрытых каналов в современных компьютерных системах. Одним из сценариев реализации рассматриваемых скрытых каналов является сценарий в модели M_1, в рамках которой рассматривается взаимодействие доверенного веб-сервера и двух субъектов-нарушителей, кооперирующих друг с другом путём обращения к этому серверу с целью обмена вредоносными данными. Данный сценарий обеспечивает свойство анонимности скрытого канала, но может обеспечить лишь пропускную способность не более 1 бит/с. В данной работе в рамках рассматриваемого сценария предлагается метод реализации скрытого канала по времени, позволяющий повысить его пропускную способность и сохранить свойство анонимности.

Основным фактором, существенно ограничивающим пропускную способность скрытого канала в рамках модели M_1, является частота обновления заголовков кеширования веб-сервером. В результате проведенных экспериментов установлено, что веб-серверы облачных сервисов хранения файлов обновляют заголовки кэширования с более высокой частотой, чем традиционные веб-серверы. Это связано с необходимостью поддержание актуальной информации в файлах и их метаданных. Большинство крупных облачных файловых хостингов предоставляют программный интерфейс (API), облегчающий загрузку и скачивание файлов, а также управление уже размещённой на сервере информацией: изменение параметров и прав доступа к файлам, их систематизацию и обновление метаданных. Возможность обновления метаданных, в частности времени последнего изменения файла, позволяет говорить о реализации скрытого канала по времени на основе заголовков кеширования.

Рассмотрим схему реализации данного скрытого канала (рис. 1). Пусть e_1 — сущность-файл, содержащая передаваемую информацию и доступная на чтение субъекту s_1; e_3 — сущность-ресурс, расположенная на серверах облачного файлового хостинга и доступная на запись субъекту s_1 через предоставляемый программный интерфейс файлового хостинга s_2; e_2 — сущность HTTP-запрос; e_4 — сущность HTTP-ответ; e_5 — сущность-файл, доступная на запись субъекту s_3.

Для передачи одного бита информации из e_1 субъект s_1 осуществляет HTTP-запрос к программному интерфейсу сервиса хранения файлов s_2, обновляющий время последней модификации сущности-ресурса e_3. В тот же временной интервал субъект s_3 при помощи HTTP-запроса к программному интерфейсу или непосредственно к сущности e_3 получает значение используемого заголовка либо иным из описанных в [2] способов определяет факт модификации сущности e_3. Субъект s_3 интерпретирует полученные сведения о модификации в соответствии с выбранным способом кодирования и записывает полученный бит в e_5. Процесс передачи повторяется через выбранный интервал времени.

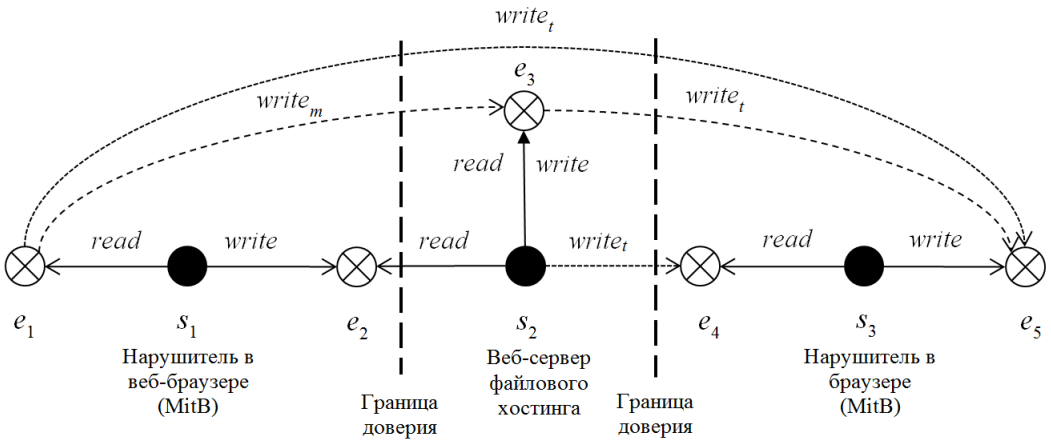

Рис. 1. Схема функционирования скрытых каналов по времени на основе заголовков кэширования HTTP в облачном файловом сервисе

Предложенный метод реализации скрытых каналов по времени на основе заголовков кеширования протокола HTTP позволяет существенно повысить пропускную способность скрытых каналов, сохраняя высокий уровень анонимности: крупные файловые сервисы, как правило, являются доверенными с точки зрения сетевых средств

контроля, а обнаружение скрытого канала, реализованного подобным образом, также затруднительно, так как им используется только стандартная функциональность, предоставляемая API сервиса. Теоретическая пропускная способность такого скрытого канала составляет 1 бит за $(L + S)$ секунд, где L — время, необходимое s_3 для выполнения запроса к e_3, а S — время, уходящее на обработку запроса на серверах хостинга.

Рассмотрим пример реализации предложенного метода на примере облачного сервиса хранения файлов Google Drive. Субъект s_1 через заданные промежутки времени осуществляет чтение одного бита из сущности e_1 и, в зависимости от полученных данных, совершает или не совершает POST-запрос e_2 по адресу `www.googleapis.com/drive/v2/files/`**fileId**`/touch`, где **fileId** — идентификатор сущности e_3. Данный запрос обновляет время последней модификации сущности-ресурса e_3. В тот же временной интервал субъект s_3 выполняет GET-запрос по адресу `www.googleapis.com/drive/v2/files/`**fileId** и получает ответ e_4, содержащий значение entity-tag сущности e_3. Субъект s_3 записывает в e_5 бит 1, если ресурс был модифицирован с момента последнего запроса, и бит 0 в противном случае.

В результате тестирования реализации скрытого канала через сервис Google Drive достигнута пропускная способность 3 бит/с при точности передачи 99,8 %, где под точностью понимается отношение числа правильно переданных бит к общему числу бит. Установлено, что основным фактором, влияющим на пропускную способность реализации, является время обработки запроса серверами сервиса. Таким образом, можно говорить о достижении в эксперименте максимальной пропускной способности канала.

ЛИТЕРАТУРА

1. *Колегов Д. Н., Брославский О. В., Олексов Н. Е.* Об информационных потоках по времени, основанных на заголовках кэширования протокола HTTP // Прикладная дискретная математика. Приложение. 2014. № 7. С. 89–91.
2. *Колегов Д. Н., Брославский О. В., Олексов Н. Е.* Исследование скрытых каналов по времени на основе заголовков кэширования протокола HTTP // Прикладная дискретная математика. 2015. № 2. С. 71–85.

УДК 004.94 DOI 10.17223/2226308X/8/32

НЕИНВАЗИВНЫЙ МЕТОД КОНТРОЛЯ ЦЕЛОСТНОСТИ COOKIE В ВЕБ-ПРИЛОЖЕНИЯХ

Д. Н. Колегов, О. В. Брославский, Н. Е. Олексов

Предлагается метод контроля целостности cookie в веб-приложениях, построенный на основе криптографических протоколов с ключевыми хеш-функциями. Метод может быть использован для реализации неинвазивных механизмов защиты от атак на веб-приложения через cookie.

Ключевые слова: *криптографические протоколы, хеш-функции, веб-приложения, web cookie.*

Термином *cookie* в протоколе HTTP обозначается набор данных, хранимый веб-клиентом (веб-браузером) и отправляемый на сервер в специальном заголовке Cookie в HTTP-запросах. Cookie первоначально могут быть сгенерированы на веб-сервере и переданы веб-клиенту в заголовке Set-Cookie в HTTP-ответе либо сгенерированы