

УДК 004.7:004.056.55

Метод организации стеганографического канала связи с использованием протокола HTTP

Е.А. ДРЫБИН,
аспирант кафедры
интеллектуальных систем

В.С. САДОВ,
профессор кафедры интеллектуальных
систем, к. т. н., доцент

Белорусский государственный
университет

В статье разработан метод построения скрытого канала передачи данных в телекоммуникационных сетях TCP/IP с использованием протокола HTTP. Методы сетевой стеганографии востребованы в современных задачах обеспечения конфиденциальности и целостности передаваемой информации. Разработанный метод основан на использовании механизмов кеширования протокола HTTP и позволяет осуществлять внедрение скрытой информации в различных режимах.

Введение. В связи с повсеместной цифровизацией обеспечение конфиденциальности информационных обменов на сегодняшний день является одним из наиболее актуальных и приоритетных направлений развития информационных технологий и, в частности, сетей передачи данных. В настоящее время в большинстве случаев защита информации от несанкционированного доступа обеспечивается методами криптографии, основанными на сложных математических задачах (считается, что для этих задач не существует эффективных методов решения). Знание третьей стороной эффективного метода решения базовой математической задачи делает информационный обмен с использованием такой криптографической защиты фактически незащищенным от несанкционированного доступа. Таким образом, наибольшей надежности в задачах обеспечения защиты информации можно достичь методами стеганографии, скрывая сам факт передачи сообщения. Кроме того, методы скрытой передачи информации позволяют пересылать данные при введении ограничений на передачу информации определенного вида на законодательном уровне.

Построение стеганографического канала связи. Протокол HTTP является наиболее используемым протоколом прикладного уровня в современных компьютерных сетях (как в рамках локальной сети, так и в рамках глобальной сети интернет), а доля трафика этого протокола приближается к половине от всего передаваемого объема данных. Скрытый канал связи на основе HTTP за счет повсеместного распространения этого протокола является практически

универсальным и позволит осуществлять защищенный обмен между корреспондентами, расположенными в любой точке мира и подключенными к сети интернет.

Наиболее интересными с точки зрения возможности сокрытия в них данных выступают стандартные заголовки HTTP, используемые механизмом кеширования. Поскольку единственным назначением механизма кеширования является снижение нагрузки на сеть и уменьшение количества сетевого трафика, то изменение значений этих заголовков (в результате встраивания секретного сообщения) не влияет на результат и процесс (для внешнего наблюдателя) информационного взаимодействия между клиентом и сервером.

Заголовок ETag представляет собой уникальный идентификатор (тег), присваиваемый сервером версии ресурса, располагающегося по определенному URI. Метод, с помощью которого генерируется значение ETag, не определен спецификацией протокола HTTP. Однако стандарт дает общие рекомендации к методу создания ETag, которые включают использование устойчивой к коллизиям хеш-функции содержимого ресурса, последнего времени изменения или номера версии [1].

Таким образом, методы, используемые для генерации ETag, должны гарантировать, что каждый тег является уникальным (или создание одинаковых тегов происходит «приемлемо редко»). На практике это означает, что функция, используемая для генерации тегов, должна обладать равномерным распределением, однозначно определяемым входными

параметрами (будь то содержимое ресурса, его версия или последнее время изменения).

Разработанный метод организации двунаправленного скрытого канала в сетевом трафике заключается в сокрытии данных сообщения в поле значения заголовка запроса If-Non-Match и поле значения заголовка ответа ETag протокола HTTP.

Принцип организации однонаправленного скрытого канала от клиента к серверу заключается в следующем:

1) на принимающей стороне (сервере) имеется произвольный ресурс, находящийся по определенному адресу;

2) на передающей стороне (клиенте) имеется секретное сообщение, которое преобразуется в последовательность байт в шестнадцатеричном виде (аналогичном виду значения заголовка If-Non-Match) сообразным целям методом кодирования;

3) на передающей стороне выделяют фрагмент преобразованного секретного сообщения, длина которого равна длине поля значения заголовка If-Non-Match; в случае необходимости выбранный фрагмент дополняют произвольными данными до длины поля значения заголовка If-Non-Match;

4) на передающей стороне формируется запрос HTTP к определенному на первом шаге URI с использованием заголовка If-Non-Match, в поле значения которого записывается выделенный на предыдущем шаге фрагмент преобразованного сообщения;

5) на передающей стороне запрос HTTP инкапсулируется в сегмент TCP и передается по открытой сети принимающей стороне;

6) на принимающей стороне запрос HTTP деинкапсулируется и, если он обращен к определенному на первом шаге URI, из сообщения извлекается значение заголовка If-Non-Match;

7) на принимающей стороне формируется ответ HTTP (согласно стандарту), который отправляется принимающей стороне и служит сигналом об успешном получении сообщения отправленного ранее запроса;

8) на принимающей стороне производится обратное преобразование (согласно выбранному методу кодирования) содержимого поля значения заголовка If-Non-Match и восстанавливается фрагмент исходного сообщения;

9) шаги 3–8 повторяются до тех пор, пока сообщение не будет передано целиком.

Для организации скрытого канала в обратном направлении (от сервера к клиенту) необходимо выполнить симметричные действия с заголовком ETag.

Пропускная способность единицы контейнера (в данном случае сообщения HTTP) будет равна

длине поля значения заголовка If-Non-Match или заголовка ETag соответственно.

Наиболее типичной реализаций клиента HTTP является веб-браузер, который обычно установлен на любом пользовательском компьютере или сервере. В качестве примера рассмотрим стандартный заголовок запроса HTTP, формируемого браузером Internet Explorer, который встраивается и используется по умолчанию в ОС семейства Windows. Структура сообщения HTTP для такого запроса приведена на рис. 1.

```
GET /home HTTP/1.1
(заголовки HTTP)
If-None-Match: "A5D255EEB1B235EAA9FF320A92D5A8F0"
(заголовки HTTP)
(пустая строка)
```

Рисунок 1 – Запрос HTTP, выполненный веб-браузером Internet Explorer 11

Длина заголовка запроса HTTP в этом случае составляет $L_{H1} = 364$ байта, а пропускная способность скрытого канала в направлении от клиента к серверу в сети передачи данных на основе технологии Gigabit Ethernet с общей пропускной способностью канала $C = 1$ Гбит/с составит:

$$C_{s1} = \frac{8L_s}{8t_b(L_{H1} + L_{TCP} + L_{IP} + L_{ETH}) + t_f} =$$

$$= \frac{(8 \cdot 16) \text{ бит}}{(8 \cdot 1 \cdot (364 + 20 + 20 + 26) + 96) \cdot 10^{-9} \text{ с}} =$$

$$= 3,62 \cdot 10^7 \text{ бит/с}, \quad (1)$$

где L_s – длина тега в байтах;

L_{H1} – длина запроса HTTP;

L_{TCP} – длина заголовка сегмента TCP;

L_{IP} – длина заголовка пакета IP;

L_{ETH} – длина заголовка кадра Ethernet;

t_f – время задержки между передачей кадров Ethernet (межкадровый интервал);

t_b – время сериализации одного бита.

В качестве эталонной реализации сервера HTTP выберем программный пакет Nginx, который представляет собой простой и широко применяемый веб-сервер, использующий теги длиной 128 бит. Структура сообщения HTTP для стандартного ответа веб-сервера Nginx со статусом 200 (OK) приведена на рис. 2.

```
HTTP/1.1 200 OK
(заголовки HTTP)
(пустая строка)
<html><body><h1>200 OK</h1></body></html>
```

Рисунок 2 – Ответ HTTP, выполненный веб-сервером Nginx

Длина заголовка ответа HTTP в этом случае составляет $L_{H2} = 271$ байт, а пропускная способность скрытого канала в направлении от сервера к клиенту в сети передачи данных на основе технологии Gigabit Ethernet с общей пропускной способностью канала $C = 1$ Гбит/с составит:

$$C_{s2} = \frac{8L_s}{8t_b(L_{H2} + L_{TCP} + L_{IP} + L_{ETH}) + t_f} = \frac{(8 \cdot 16) \text{ бум}}{(8 \cdot 1 \cdot (271 + 20 + 20 + 26) + 96) \cdot 10^{-9} \text{ с}} = 4,58 \cdot 10^7 \text{ бум/с.} \quad (2)$$

В случае использования полной пропускной способности единицы контейнера преимущество в защищенности вложения от применения стеганографии отсутствует, т. к. все биты поля значения заголовка тега (ETag или If-Non-Match) замещаются битами секретного сообщения, а стеганографический путь является одинаковым для всех контейнеров:

$$\frac{P_s}{P_0} = 1. \quad (3)$$

Получить преимущество в стойкости вложения за счет применения стеганографии можно путем использования следующего способа встраивания данных.

Перед сокрытием данных необходимо сгенерировать так называемую маску внедрения, которая (в идеальном случае) представляет собой последовательность бит с равномерным распределением, однозначно определяемую стеганографическим ключом. В качестве маски внедрения с подобными свойствами можно использовать ПСП, которую получают, как правило, при помощи ГПСЧ, инициализация которого производится с помощью секретного ключа.

Выбор статистических характеристик маски внедрения определяется необходимостью добиться максимальной неопределенности значений бит маски, а значит, и распределения скрытого содержимого внутри контейнера. Как известно, максимум неопределенности значений бит маски внедрения достигается при равномерном распределении ее бит.

Длина маски внедрения совпадает с длиной передаваемого вложения. Для встраивания секретного сообщения маска внедрения разделяется на фрагменты длиной, равной длине поля значения заголовка тега:

$$L_m = L_t = 128 \text{ бит}, \quad (4)$$

где L_m – длина фрагмента маски внедрения; L_t – длина поля значения заголовков уникального идентификатора ресурса.

Встраивание информации в значение заголовка ETag или If-Non-Match производится следующим образом. Сначала в разряды тега, которым соответствуют единичные биты фрагмента маски внедрения, помещаются биты секретного сообщения, как это изображено на рис. 3.

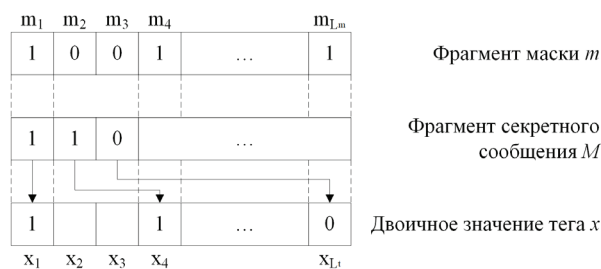


Рисунок 3 – Встраивание бит секретного сообщения в значение тега

После этого в разряды тега, которым соответствуют нулевые биты фрагмента маски внедрения, помещают произвольные биты таким образом, чтобы добиться их равномерного распределения, как это изображено на рис. 4.

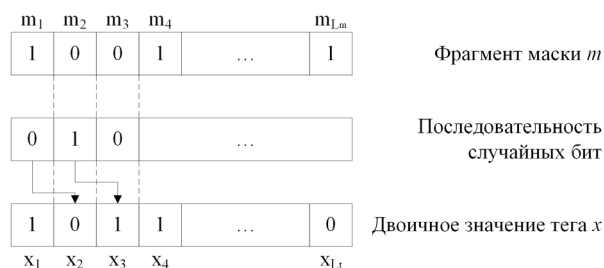


Рисунок 4 – Формирование оставшихся бит значения тега

Полученное двоичное значение тега преобразуется в шестнадцатеричный вид и встраивается в виде строки в значение заголовка ETag или If-Non-Match очередного сообщения HTTP для передачи по сети.

Показанный выше подход позволяет установить однозначную зависимость размещения бит секретного сообщения в контейнере от стеганографического ключа. Преимущество в защищенности вложения, обеспечиваемое стеганографическим методом, при таком подходе и ключе длиной L_k составит:

$$\frac{P_s}{P_0} = 2^{L_k}. \quad (5)$$

Другими словами, для проведения успешной атаки исчерпывающего поиска на секретное вложение атакующему потребуется выполнить

в 2^{L_k} раз больше переборов (наихудший для атакующего исход).

Эффективная пропускная способность скрытого канала в каждом направлении при использовании маски внедрения с вероятностным распределением бит составит:

$$C_{s1k} = \frac{8L_s p(1)}{8t_b (L_{H1} + L_{TCP} + L_{IP} + L_{ETH}) + t_f} =$$

$$= \frac{(8 \cdot 16 \cdot 1/2) \text{ бит}}{(8 \cdot 1 \cdot (364 + 20 + 20 + 26) + 96) \cdot 10^{-9} \text{ с}} =$$

$$= 1,80 \cdot 10^7 \text{ бит/с}; \quad (6)$$

$$C_{s2k} = \frac{8L_s p(1)}{8t_b (L_{H2} + L_{TCP} + L_{IP} + L_{ETH}) + t_f} =$$

$$= \frac{(8 \cdot 16 \cdot 1/2) \text{ бит}}{(8 \cdot 1 \cdot (271 + 20 + 20 + 26) + 96) \cdot 10^{-9} \text{ с}} =$$

$$= 2,29 \cdot 10^7 \text{ бит/с}, \quad (7)$$

где $p(1)$ – вероятность появления единичного бита в маске внедрения, в случае равномерного

распределения и использования алфавита $A = \{0, 1\}$ значение $p(1) = 1/2$.

В случае возникновения искажений и потерь в канале передачи данных часть сообщений на принимающей стороне может быть искажена. Описанный метод организации скрытого канала защищен от подобных негативных эффектов за счет использования надежного транспорта, который предоставляет TCP протоколам прикладного уровня (в т.ч. и протоколу HTTP). Механизм кэширования TCP осуществляет повторную передачу информации в случае возникновения искажений или потерь данных в канале связи и упорядочивает данные, если сегменты TCP были доставлены получателю с нарушением порядка отправления.

Заключение. В настоящей статье были рассмотрены новые методы организации скрытого канала в телекоммуникационных сетях на основе стека протоколов TCP/IP. В качестве контейнера предложено использовать сетевой трафик протокола прикладного уровня, а именно поля значения заголовков If-Non-Match и ETag сообщений HTTP. Разработанный стеганографический метод сокрытия информации позволяет также изменить ее пространственное распределение по элементам контейнера, что увеличивает преимущество в защищенности скрытого сообщения средствами стеганографии.

ЛИТЕРАТУРА

1. Hypertext Transfer Protocol (HTTP/1.1): Conditional Request [Electronic resource]: RFC 7232 / R. Fielding, J. Reschke. – Publ. date June 2014. – Mode of access: <https://tools.ietf.org/html/rfc7232#section-2.3>. – Date of access: 04.02.2018.
2. **Олифер, В. Г.** Компьютерные сети. Принципы, технологии протоколы / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2006. – 958 с.

The method to create a hidden data transmission channel in TCP/IP networks using the HTTP protocol is developed in the article. Network steganography methods are in a high demand in modern environment to ensure confidentiality and integrity of transmitted information. The developed method is based on the HTTP caching mechanisms and allows several modes to hide sensitive data.

Получено 28.06.2019.