эффективного правового регулирования экономического сектора и другие направления.

Список использованных источников

1. Указ Президента РФ от 31.12.2015 г. № 683 «О Стратегии национальной безопасности Российской Федерации» // Собрание законодательства РФ. 2016. № 1 (часть II), ст. 212.
2. Гладких В.И., Сухаренко А.Н. Криминальные угрозы экономической безопасности России: состояние и меры нейтрализации // Российский следователь. 2018. № 2. С. 47 - 52.
3. Горохова С.С. О системе управления рисками в сфере обеспечения экономической безопасности в Российской Федерации // Современный юрист. 2018. № 1. С. 30 - 39.
4. Земскова А. Е. С., Горин В. А. Особенности экономического развития в контексте национальной экономической безопасности //Национальные интересы: приоритеты и безопасность. № 4 (61). 2019. С. 70-71.
5. Трошин Д.В. Подход к типологии и классификации угроз и рисков экономической безопасности Российской [i]Федерации // Безопасность бизнеса. 2018. № 1. С. 18 - 24.

УДК 003.26+004.9

П.П. Урбанович, И.В. Калоша, Н.П. Шутько
Белорусский государственный технологический университет
Минск, Беларусь

ИСПОЛЬЗОВАНИЕ СКРЫТЫХ КАНАЛОВ ДЛЯ ПЕРЕДАЧИ ИНФОРМАЦИИ НА ОСНОВЕ СТЕГАНОГРАФИЧЕСКИХ МЕТОДОВ В СТЕКЕ ПРОТОКОЛОВ TCP/IP

Аннотация. *Проанализированы некоторые особенности создания и использования скрытых стеганографических каналов на основе сетевых протоколов. Показан пример их практической реализации.*

P.P. Urbanovich, I.V. Kalosha, N.P. Shutko
Belarusian State Technological University
Minsk, Belarus

USE OF THE COVERT CHANNELS FOR INFORMATION TRANSMISSION BASED ON STEGANOGRAPHIC METHODS IN THE TCP/IP PROTOCOLS STACK

Abstract. *Some features of the creation and use of covert steganographic channels based on network protocols are analyzed. An example of their practical implementation is shown.*

Стеганография является одним из основных средств для скрытой передачи информации и для защиты контента от несанкционированного использования или модификации [1, 2]. Классические методы цифровой стеганографии основаны на использовании пространственно-геометрических, цветовых или иных параметров файлов-контейнеров для размещения тайной информации.

Относительно новым направлением в рассматриваемой предметной области является создание скрытых стеганоканалов на основе использования элементов и параметров сетевых протоколов (в основном – TCP/IP) для передачи информации [3]. Такой подход является весьма привлекательным для реализации различных деструктивных действий. В последнее время наблюдается новый и опасный тренд: все больше разработчиков вредоносного ПО и средств кибершпионажа прибегают к использованию стеганографии. В этих случаях сетевое взаимодействие является ключевой функцией вредоносной программы. Существующие средства защиты не в состоянии обрабатывать все объекты, которые потенциально могут быть заполненными контейнерами или стеганоконтейнерами.

Вместе с тем следует отметить, что сетевая стеганография может также использоваться и легитимными пользователями. Например, сетевыми администраторами – для реализации различных операций аудита защищенности локальной сети.

Далее проанализируем некоторые теоретические и практические аспекты реализации рассматриваемых методов.

Большинство основополагающих и более поздних публикаций по стеганографии на основе сетевых протоколов сосредоточено на верхних уровнях стека протоколов TCP/IP. В частности, в [4] описаны три метода стеганографического размещения информации в заголовке TCP/IP (сетевой и транспортный уровни): в поле идентификации IP, в поле начального порядкового номера и в поле порядкового номера

подтверждения протокола TCP. Автор акцентирует внимание на поле *Identification* IP-протокола, а также на полях в заголовке TCP: *SequenceNumber* и *AcknowledgmentNumber*.

Поле *SequenceNumber* позволяет, например, клиенту установить надежное согласование протокола с удаленным сервером. Как часть процесса согласования для TCP/IP, можно реализовать несколько шагов в так называемом «трехстороннем рукопожатии». Для реализации скрытого канала поле порядкового номера служит хорошим средством для передачи тайных данных из-за его размера (32-битное число). В этом свете существует ряд возможных методов. Самый простой – сгенерировать порядковый номер символа ASCII, который мы хотим закодировать, и далее разместить этот символ в указанном поле. Слабым местом использования поля является то, что оно генерируется специальными алгоритмами, которые генерируют случайные последовательности.

Поле *AcknowledgmentNumber* предназначено для корректной адресации между пакетами: по его содержимому сервер чаще всего «понимает» очередность пакетов и правильность их дальнейшей обработки. Однако каждый запрос на сервер начинается с аналогичного запроса, только с несуществующим номером в данном поле. Такой запрос используется для прослушивания порта сервера и, по сути, нужен только для того, чтобы получить от сервера ответ о доступности в сети. С точки зрения стеганографии же не важно, будет ли понимать сервер дальнейший порядок пакетов и будет ли он понимать правильность содержимого пакета. При большой нагрузке на сервер заметить такие прослушивания порта практически невозможно.

Принципиально новой является идея, изложенная в [5]. Здесь предлагается использовать несовершенства среды передачи данных: влияние помех и шумов в канале связи, т. е. естественную восприимчивость к искажению данных. Ключом метода являются ошибки, возникающие в передаваемых данных под влиянием помех. Соответственно создаваемая на этой основе система получила название HICCUPS (HIdden Communication system for CorrUPted networkS).

Предлагаемая система предназначена для реализации в сетевой среде со следующими тремя свойствами: сеть реализована в разделяемой среде с возможностью перехвата кадров; используется общеизвестный метод инициации шифрования; используются механизмы целостности для зашифрованных кадров, например, односторонняя хэш-функция, циклический избыточный код, CRC

(один из элементов стандарта IEEE 802.11). В [3–5] классифицированы и описаны также иные методы сетевой стеганографии.

На основе всестороннего изучения и анализа основных из существующих подходов в реализации сетевой стеганографии нами разработано клиент-серверное приложение, развернутое на независимых виртуальных машинах, со встроенным анализатором сетевых пакетов между клиентом и сервером, позволяющее «вживую» реализовать систему обмена одноразовыми сообщениями посредством скрытого стеганографического канала.

На каждую из виртуальных машин необходимо установить ОС *Linux* на основе образа *Ubuntu*. Вся разработка велась на стороне *Windows*в *IDEVisualStudio.* Для сборки по сети необходимо иметь все библиотеки и компилятор (*gcc*) на виртуальной машине. Для запуска конечного приложения установка библиотек не требуется, так как все необходимые компоненты формируются сборщиком в исполняемый файл. Необходимо открыть 22 порт в ssh-конфигурации Linux для удаленного подключения в машине, а также для возможности удаленной разработки с помощью *VisualStudio.*

Для настройки среды необходимо получить IP виртуальной машины, используя команду *ipaddr*.

В настройках виртуальных сред разработки необходимо добавить виртуальную машину, указав IP-адрес, а также логин и пароль для пользователя. Сигнатуры функций отправителя сообщения представлены на рис. 1.

```
⊞int main(int argc, char* argv[]) { ... }

⊞unsigned short csum(unsigned short* ptr, int nbytes) { ... }

⊞size_t writeCallback(void* contents, size_t size, size_t nmemb, void* userp) { ... }

⊞char* findMessage() { ... }
```

Рис. 1 - Сигнатуры функций отправителя

Функция *csum*, в частности, необходима для вычисления хэша TCP- или IP-заголовка. Этот параметр является обязательным для отправки пакета. Создана специальная структура, отвечающая за IPзаголовок. Для ее инициализации нужна библиотека *netinet/ip.h*. Поля данной структуры заполняются необходимой информацией, соответствующей размеру полей. В поле *Identification* встраивается часть секретного сообщения, после чего вычисляется хэш и записывается в поле *check*. Таким образом, получается стандартный, не поврежденный IP-

заголовок, заполненный вручную. Похожим способом, используя библиотеку *netinet/tcp.h*, заполняется TCP-заголовок.

Сигнатуры функций получателя представлены на рис. 2.

```
⊞int main(int argc, char* argv[]) { ... }

⊞unsigned short csum(unsigned short* ptr, int nbytes) { ... }

⊞size_t writeCallback(void* contents, size_t size, size_t nmemb, void* userp) { ... }

⊞char* findMessage() { ... }
```

Рис. 2 -Сигнатуры функций отправителя

Большая часть кода относится к проверке соответствия отправителя и получателя и обеспечения надежности и отказоустойчивости канала. На рис. 3 показан процесс отправки пакетов и их количество, на рис. 4 – соответствующая информация на стороне получателя.

```
Введите секретное сообщение (до 65к символов):
Hello World
message%5Bbody%5D=Hello World
curl_easy_perform() failed: Couldn't resolve host name
Пакет отправлен. "messag"
Пакет отправлен. "e%5Bbo"
Пакет отправлен. "dy%5D="
Пакет отправлен. "Hello "
Пакет отправлен. "World"
```

Рис. 3 - Работа отправителя

```
Ожидание сообщения...

Ссылка на секретное сообщение:
https://tmwsd.ws/message%5Bbody%5D=Hello World
Ошибка сервиса TMWSD: Couldn't resolve host name
message%5Bbody%5D=Hello World
Ожидание сообщения...
```

Рис. 4 - Работа отправителя

Предварительно перед отправкой сообщения (Hello World) был включен снифинг пакетов в WireShark. Применив несколько фильтров и настроек, удалось найти нужные пакеты с полезным содержимым. Они показаны на рис. 5.

26	84.005016190	127.0.0.1	127.0.0.1	TCP	54 20 → 9380 [SYN] Seq=0 Win=5840 Len=
27	84.015236312	127.0.0.1	127.0.0.1	TCP	54 20 → 7738 [SYN] Seq=0 Win=5840 Len=
28	84.018061369	127.0.0.1	127.0.0.53	DNS	79 Standard query 0x1ae8 A tmwsd.ws OP
29	84.018216798	127.0.0.1	127.0.0.53	DNS	79 Standard query 0xa3e5 AAAA tmwsd.ws

Рис. 5 - Пакеты с секретным сообщением

На рис. 6 выделена первая часть секретного сообщения, которое было отправлено по скрытому каналу, организованному в TCP-пакетах.

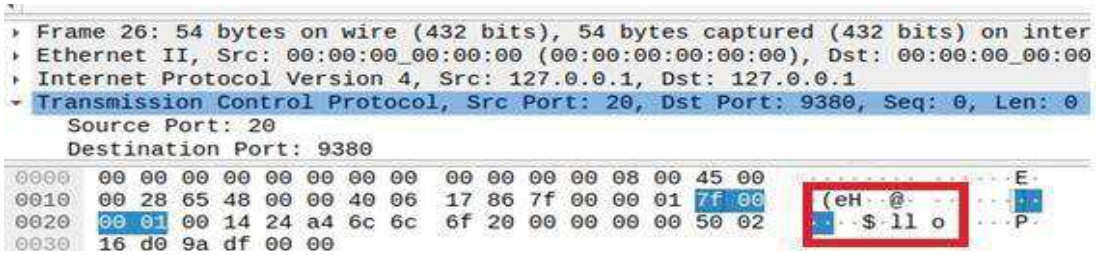

Рис. 6 - Первая часть секретного сообщения на основе TCP

По результатам тестирования работы приложения сделан вывод: при увеличении нагрузки сети начинают появляться ошибки и коллизии. Установлено также, что указанная особенность проявляется на маршрутах с числом узлов более 4. Однако прямой зависимости успешной передачи от количества хостов нет. Это объясняется наличием прокси и других защитных сервисов на пути передачи пакета.

Список использованных источников

1. N. Shutko, P. Urbanovich, P. Zukowski. A method of syntactic text steganography based on modification of the document-container aprosh. Przegląd Elektrotechniczny. 94 (2018), 6: 82–85. doi:10.15199/48.2018.06.15.
2. Blinova E. A., Urbanovich P. P. Steganographic method based on hidden messages embedding into Bezier curves of SVG images. Journal of the Belarusian State University. Mathematics and Informatics. 2021; 3: 68–83. https://doi.org/10.33581/2520-6508-2021-3-68-83.
3. Ласык, Я. Использование сетевых протоколов и стеганографии для тайной передачи информации / Я. Ласык, Д. М. Романенко, П. П. Урбанович // Информационные технологии: материалы 86-й научно-техн. конф. профессорско-препод. состава, научных сотр. и аспирантов, Минск, 31 января – 12 февраля 2022 г. – Минск: БГТУ, 2022. – С. 158–163.
4. Rowland, C. H. Covert channels in the TCP/IP protocol suite. First Monday. 1997, 2(5). https://doi.org/10.5210/fm.v2i5.528.
5. Szczypiorski, K. HICCUPS: Hidden communication system for corrupted networks. In Proc. of: The Tenth International Multi-Conference on Advanced Computer Systems ACS'2003, October 22-24, 2003.