УДК 004.9

Д.В. Сергеев, Р.А. Хади

Алгоритм и реализация фильтра сетевой информации со скрытыми от внешнего наблюдателя правилами

Рассматривается применение алгоритма фильтрации сетевой информации с учетом присутствия одного или нескольких внешних наблюдателей, заинтересованных во вскрытии данного алгоритма. Строится модель угроз и приводится пример реализации.
Ключевые слова: фильтрация, внешний наблюдатель, сетевая информация, скрытая запись.

В настоящее время в сети Интернет можно найти объявления большого числа хостинг-провайдеров, предоставляющих услуги:

- размещения сайтов на серверах хостинг-провайдера (хостинг);
- аренды физических выделенных серверов (dedicated server);
- аренды виртуальных выделенных серверов (VDS);
- аренды бизнес-приложений.

Пользование услугами хостинг-провайдеров в части аренды выделенных серверов (как физических, так и виртуальных) предоставляет следующие преимущества:

- отсутствие начальных инвестиций в сервер;
- возможность перехода на более производительный арендованный сервер;
- техническим обслуживанием сервера, включая ремонт и профилактику, занимается провайдер.

Задачи, решаемые с помощью применения выделенных серверов, могут быть различными, однако надо понимать, что физически сервер находится во владении хостинг-провайдера, а значит, эта организация имеет полный доступ к тем данным, которые хранятся и обрабатываются на предоставляемых выделенных серверах. Помимо хостинг-провайдера, доступ к этим данным могут получить хакеры, а также (по юридическому запросу) и спецслужбы. Таким образом, можно сделать предположение, что и хостинг-про-вайдер, и хакеры, и спецслубы могут являться внешними наблюдателями, заинтересованными в негласном получении информации, хранимой в пределах хостинг-сервера. В данном контексте под внешним наблюдателем будем понимать любое заинтересованное лицо или организацию, имеющих доступ к данным, хранящимся или обрабатываемым на арендуемых выделенных серверах.

Целью данного исследования является создание такого программного обеспечения, ко-торое позволило бы в процессе функционирования на хостинг-серверах исключить возможность негласного получения обработанной информации.

Схема работы такого программного средства (обозначим его через аббревиатуру «ВВ» – от англ. «black box») достаточно проста. На вход программного модуля ВВ (рис. 1) поступают сетевые данные, которые по установленным правилам фильтруются. Отобран-ная информация сохраняется в контейнер.

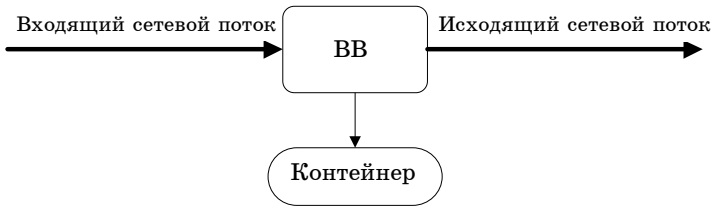

Рис. 1. Общая схема работы программного модуля ВВ

Программный модуль ВВ и контейнер представляют собой файлы, расположенные на удаленном сервере. Как уже было отмечено раньше, в случае использования выделенных серверов мы предполагаем, что любой файл доступен стороннему наблюдателю для анализа. Для нас это означает, что, размещая файлы на удаленном сервере, мы осознанно подвергаемся риску быть атакованными. Для получения полной картины возможных атак на программный модуль ВВ осуществим построение модели угроз.

Построение модели угроз

Для внешнего наблюдателя программный модуль ВВ является своего рода «черным ящиком», т.е. объектом исследования, внутреннее устройство которого неизвестно. Мы предполагаем, что у внешнего наблюдателя имеются следующие стратегии поведения:

1) попытаться изучить логику работы программного модуля ВВ методом «грубой силы», т.е. дизассемблировать исполняемый код;

2) попытаться изучить поведение программного модуля ВВ, исследуя реакции на разнообразные внешние воздействия, выполняя программу в пошаговом режиме, получая доступ к регистрам, областям памяти и т.д., полностью абстрагируясь от внутреннего устройства этого модуля. Это так называемая трассировка программы.

Первая из описанных стратегий сама по себе является угрозой. Что же касается второй, то для выявления возможных угроз необходимо рассмотреть входные и выходные данные программного модуля ВВ, манипуляция которыми позволит проводить определенные исследования.

Для построения модели угроз определим точки воздействия на ВВ.

1. Программный модуль ВВ является исполняемой программой, а значит, существует угроза анализа исполнимого кода ВВ, что может привести к вскрытию используемых алгоритмов.

2. Для корректного исполнения программного модуля ВВ требуются ресурсы ОС (процессорное время, оперативная память, дисковое пространство), что приводит к угрозе нехватки ресурсов.

3. На вход программного модуля ВВ поступает поток сетевых данных, что позволяет стороннему наблюдателю осуществлять выборочную передачу данных, а это ведет к угрозе контроля входящего и исходящего потоков.

4. Результат работы программного модуля ВВ записывается в контейнер, который может быть подвержен анализу сторонним наблюдателем. Данный факт говорит о существовании угрозы анализа содержимого контейнера.

Все рассмотренные варианты сгруппированы и представлены на рис. 2.

С учетом построенной модели угроз реализуем программный модуль ВВ с минимальным риском реализации описанных угроз.

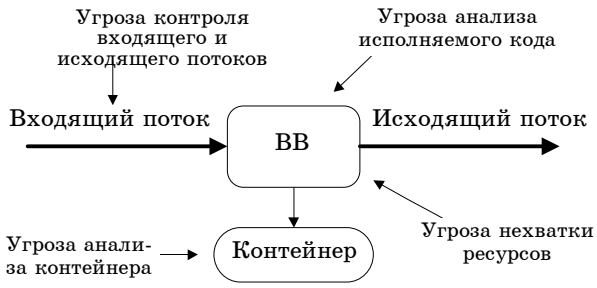

Рис. 2. Модель угроз для программного модуля ВВ

Программный модуль ВВ получает на вход буферы данных произвольного размера d_{data}, не превышающего значения D_{data}, и сохраняет их в контейнер, имеющий размер L_{cont}. Величины d_{data}, D_{data} и L_{cont} измеряются в блоках. Блок является минимальной единицей измерения и обозначается S_{block}.

Пусть $L_{cont} = S_{block} \cdot N$ и $D_{data} = S_{block} \cdot M$, где N и M – количество блоков. Также предположим, что для d_{data}, D_{data} и L_{cont} справедливо неравенство $d_{data} < D_{data} \ll L_{cont}$.

Первый вопрос, возникающий после получения данных (о природе входных данных никаких предположений не строится), состоит в следующем: каким образом осуществлять запись поступающих данных в контейнер? Сделаем предположения касательно этого вопроса, которые смогут усложнить анализ контейнера злоумышленниками:

– данные записываются в контейнер с добавлением «мусора»;

– данные не записываются в контейнер последовательно;

– блоки данных шифруются;

– контейнер имеет «срок жизни», по истечении которого создается новый контейнер.

Каждый блок контейнера имеет метку занятости. Для сохранения отдельного блока поступивших данных вычисляется позиция размещения в контейнере с помощью функции $P(L_{cont})$, которая оперирует метками занятости. Если выбранный блок контейнера свободен, то в него помещается блок данных, иначе выбирается новое место (как вариант, смещением в сторону на следующий блок). Каждый сохраняемый блок данных зашифровывается индивидуальным ключом [1].

После сохранения блоков данных в контейнере осуществляется случайный выбор блоков контейнера с помощью той же функции P, к выбранным блокам применяется сначала операция расшифрования, а затем операция зашифрования с новым ключом. Полученные блоки помещаются на те же места в контейнере. Количество блоков, подвергающихся операциям расшифрования/зашифрования, вычисляется через функ-

цию $f(d_{data}, L_{rest})$, где L_{rest} – остаток свободного места в контейнере. Данный подход (назовем его операцией запутывания) используется для затруднения обнаружения места сохранения поступивших данных.

Помимо сохраняемых данных, в контейнере также содержится таблица с метаинформацией, описывающая каждый блок контейнера. Метаинформация состоит из текущего ключа шифрования, метки занятости и смещения блока. На этапе заполнения контейнера данными метаинформация постоянно изменяется. При сохранении данных в блок меняются его метка занятости и ключ шифрования. Ключ шифрования меняется и в том случае, когда к блоку применяется операция запутывания.

Кроме размера, контейнер характеризуется фиксированным «временем жизни». Наличие временного параметра предоставляет возможность ограничить внешнего наблюдателя в проведении анализа изменений контейнера.

Обозначим через T «время жизни» контейнера. В силу того обстоятельства, что контейнер имеет ограниченную вместимость, может возникнуть ситуация, когда контейнер будет заполнен до истечения времени T. В этом случае ВВ не обязательно сразу прекращает работать с контейнером, все зависит от состояния поступающих данных. Если контейнер полон и данные не поступают, то используется тот же контейнер, который изменяется с помощью применения операций расшифрования/зашифрования случайных блоков. Как только поступают блоки данных, ВВ создает новый контейнер.

В заключение проанализируем риски реализации возможных угроз применительно к представленной реализации программного модуля ВВ.

Преодоление угрозы анализа исполнимого кода решается методами противодействия дизассемблированию [2]:

– архивация;

– шифрование;

– использование самогенерирующих кодов.

Угроза нехватки ресурсов преодолевается контролем за состоянием оперативной памяти и дискового пространства. При нехватке ресурсов программный модуль завершает свое выполнение.

Избежать угрозы контроля входящих и выходящих потоков данных не удастся в силу того обстоятельства, что внешний наблюдатель имеет полный доступ к программному модулю ВВ. Это означает, что он сможет в любой момент подавать на вход специальным образом сформированные данные и пошагово фиксировать результаты работы ВВ.

Проблема существования угрозы анализа содержимого контейнера решается реализацией в ВВ следующими способами:

– данные записываются в контейнер с добавлением «мусора»;

– данные не должны записываться в контейнер последовательно;

– блоки данных должны шифроваться;

– контейнер имеет «срок жизни», по истечении которого создается новый контейнер.

Литература

1. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

2. Касперски К. Искусство дизассемблирования / К. Касперски, Е. Рокко. – СПб.: БХВ-Петербург, 2008. – 896 с.

Сергеев Дмитрий Васильевич
Мл. науч. сотр. ФГНУ «НИИ «Спецвузавтоматика», г. Ростов-на-Дону
Тел.: (+7 863) 201-28-24
Эл. адрес: sva@rsu.ru, sergeevdv@gmail.com

Хади Роман Ахмедович
Канд. техн. наук, зам. директора по научной работе
ФГНУ «НИИ «Спецвузавтоматика», г. Ростов-на-Дону
Тел.: (+7 863) 201-28-24
Эл. адрес: sva@rsu.ru

D.V. Sergeev, R.A. Hady

Algorithm and implementation of the network data filter with concealed from the outside observer rules

It is shown the application of network information filtering algorithm taking into account the presence of one or more outside observers interested in the showdown of the algorithm. It is constructed a model of threats and an example implementation.

Keywords: filtration, outside observer, network information, the hidden record.