Оглавление

Настройка Apache mod_ssl

Тестировалось на версии Apache 2.2.3 / mod_ssl 2.2.3 / openssl 0.9.8c

Необходимые настройки в /etc/apache2/mods-available/ssl.conf:

SSLSessionCache none

SSLSessionCacheTimeout 0

SSLProtocol all -SSLv2

SSLCipherSuite ALL:!EDH:!DH:RC4+RSA:+HIGH:+MEDIUM:+LOW:!SSLv2:+SSLv3:+EXP

Программа тестировалась только с этой последовательностью *CipherSuite.*

Запуск программы

Внимание! Для нормальной работы программы требуются права root.

Необходимые библиотеки функций

Для работы программы необходимы следующие библиотеки функций:

- libssl 0.9.8 и новее
- libdssl 1.2.0 и новее
- libpcap 0.8 и новее
- libnet 1.1.2.1 и новее

Более подробную информацию о зависимостях можно получить с помощью команды, в качестве единственного аргумента – путь к исполняемому файлу программы, в нашем случае – viewssl.

Аргументы командной строки

Чтобы посмотреть перечень необходимы аргументов и дополнительных опций, сделайте:

 # ssluncrypt -h

Все аргументы имеют длинный многобуквенный и краткий однобуквенный аналоги, например “--dst-interface” и просто “–d”, можно использовать любой из этих алиасов параметров.

Общие параметры

--config (её краткий аналог – **“-c”**)

По умолчанию программа не ищет конфигурационный файл и все параметры для работы ожидает от аргументов командной строки. Если указан этот аргумент. то будет обработан конфигурационный файл.

--no-daemon (её краткий аналог – “**-n**”)

По умолчанию программа работает как демон, т.е. отключается от терминала, в котором она запускается и переходит в фоновый режим работы, при этом все сообщения (ошибки, предупреждения, и пр.) отправляются в Syslog, а не на консоль. Такое поведение может быть изменено данной опцией – программа будет работать в интерактивном режиме, все сообщения будут выводиться в текущую консоль.

--verbose (её краткий аналог – “**-v**”)

По умолчанию программа не разговорчива, и сообщает только об ошибках в работе. Такое поведение может быть изменено, например, для отладки. При запуске программы с этой опцией она выводит чуть больше информации и считанной из конфигурации информации (полезно для отладки конфигурационного файла) и о ходе своей работы.

Важное замечание: эффект от этой опции почти не заметен при протоколировании в Syslog. Рекомендуется использовать эту опции вместе с опцией “–n”.

--pid (её краткий аналог – “**-f**”)

При запуске программа создает специальный файл, куда записывает свой PID. По умолчанию этот файл “/var/run/ssluncrypt.pid”, но с помощью этой опции можно указать произвольный файл для этой цели.

--help (её краткий аналог – “**-h**”)

Показывает справку по аргументам запуска.

--version (не имеет краткого аналога)

Показывает информацию о версии программы.

Параметры патока расшифровки

--src-interface (её краткий аналог - “**-s**”)

Определяет интерфейс, который будет "слушать" программа на предмет SSL-трафика.

--dst-interface (её краткий аналог - “**-d**”)

Определяет интерфейс, на который будут отправляться расшифрованные пакеты. Предполагается, что это будет dummy интерфейс, который можно создать командой "modprobe dummy" и

активировать “ifconfig dummy0 up”. Именно этот интерфейс нужно указать snort для прослушивания (что-то вроде *# snort -i dummy -c /etc/snort.conf*).

--ip (её краткий аналог – “**-i**”)

Программой будут обрабатываться пакеты только с указанным IP-адресом назначения. Сюда нужно указать IP-адрес HTTPS сервера.

--port (ей краткий аналог - “**-p**”)

Программой будут обрабатываться пакеты только с указанным TCP портом назначения. Сюда нужно указать HTTPS порт, по умолчанию 443.

--key (её краткий аналог – “**-k**”)

Здесь нужно указать путь к закрытому ключу HTTPS сервера.

Если ключ защищен парольной фразой – следует её указать параметром **“–pwd**” (её краткий аналог – “**-w**”).

При отсутствии вышеперечисленных аргументов программа выдает сообщение об ошибке с указанием каких именно аргументов не хватает.

Конфигурационный файл

Общая информация

В конфигурационном файле все опции делятся на два вида:

- Общие параметры (common options).
- Параметры патока расшифровки (capture otions).

Общие параметры должны находиться вначале файла - выше, чем параметры потоков расшифровки, последние, в свою очередь, если будет использовано несколько параллельных потоков, должны начинаться с заключенного в квадратные скобки ([<title>]) заголовка. Этот заголовок определяет начала блока параметров отдельного патока и (соответственно) конец другого блока параметров патока, или конец блока общих параметров. Более подробно о заголовках и блоках написано в части “**Заголовки блоков**”.

В типовом случае конфигурационный файл должен выглядеть следующим образом:

…общие параметры…

[заголовок патока 1]

…параметры патока 1…

[заголовок патока 2]

…параметры патока 2…

…..

В случае однопоточного режима (когда можно расшифровывать информацию только по одному блоку условий – интерфейс, ip адрес, порт, ключ) нет нужды в заголовках, вполне корректной является такая структура конфигурационного файла:

…общие параметры…

… параметры патока…

В случае такой упрощенной конфигурации можно смешивать общие параметры с параметрами патока, в случае “блочной” структуры конфигурационного файла смешивать параметры потоков и общие параметры недопустимо.

Все параметры в конфигурационном файле имеют следующий синтаксис:

Имя_параметра = значение параметра

Прабелы необязательны, более того – они игнорируются.

Имена всех параметров –регистронезависимы.

Во всех параметра пробелы игнорируются, если необходимо чтобы пробел был интерпретирован как часть опции – необходимо его экранировать символом обратной косой черты “\ ”.

Что бы символ “\” воспринимался именно как символ, а не как управляющая последовательность - его также следует экранировать самим собой – “\\”.

Каждый новый параметр в конфигурационном файле должен начинаться с новой строки!

Заголовки блоков

Программа может работать в 2х режимах – в однопоточном и многопоточном, выбор конкретного режима работы зависит от конфигурации. Если параметры для работы передаются только через аргументы командной строки – то режим работы однозначно будет однопоточным, т.к. в аргументах нельзя передать более одной конфигурации патока расшифровки, см. подчасть “Параметры патока расшифровки” части “Аргументы командной строки”.

Если для конфигурирования используется конфигурационный файл, то есть возможность сконфигурировать программу так, что бы она работала в несколько потоков, т.е. расшифровывала трафик сразу из нескольких источников и с помощью разных криптографических ключей. Это

достигается помещением конфигурации отдельных потоков в отдельные блоки, которые образуются посредством т.н. заголовков блоков.

Структура заголовка:

[title]

Где title – непосредственно текст заголовка, который в дальнейшем будет использоваться как “имя” потока при выводе информации, например информации о конфигурации (посредством опции “-v”) или в лог-файл (syslog). Заголовок может содержать исключительно печатные символы (непечатные символы игнорируются), также в заголовок могут быть включены пробелы, что не рекомендуется, но все же возможно с помощью экранирования пробела “\ ”. не экранированные пробелы игнорируются парсером конфигурационного файла.

Title может содержать произвольное имя, кроме зарезервированного “common”, т.к. блок с заголовком “common” может использоваться только для хранения общих конфигурационных параметров, хотя это не обязательно, т.к. общие параметры можно поместить в начало файла, вне блоков […].

Список параметров конфигурационного файла

Общие параметры

Общие параметры – это системные параметры программы, значения которых распространяется на все потоки.

Pid =<path>

Указывает путь к PID-файлу, в которые будет записан PID-процесса приложения, в случае многопоточного режима – PID мастер-процесса, который является родителем (parent) дочерних процессов, в которых выполняются потоки.

Аналог аргумента –pid (-f).

Daemon = on/off

Соответственно включить/выключить демонизацию процесса после запуска. Демонизация – перевод в фоновый режим исполнения. В таком режиме все сообщения процесса (ошибки, исключительные ситуации, информационные сообщения) попадают не на консоль, а в syslog.

Посмотреть сообщения можно командой:

 # cat /var/log/daemon.log | grep viewssl

Loglevel = [0|1]

Уровень детализации сообщений от программы (уровень протоколирования). На данный момент поддерживается всего 2 уровня:

Уровень 1 (normal): Только сообщения о критических ошибках. Если во время передачи или расшифровки данных происходит ошибка протокола, то сообщения об ошибках на этом уровне протоколирования не фиксируются в syslog и не выводятся на экран. Такое поведение обусловлено большим количеством подобных ошибок в определенных ситуациях, и сообщения о таких ошибках могут “захломить” Syslog или консоль. Этот уровень рекомендуется использовать в “нормальной” штатной работе программы, когда все отлажен и стабильно работает.

Уровень 2 (debug): Этот уровень может быть включен аргументом командной строки “**-v**”. Информационно-отладочные сообщения, вывод информации о конфигурации на консоль пользователя во время запуска программы без демонизации (-n). Если происходит сбой протокола во время передачи данных, сообщения об ошибках SSL фиксируются в Syslog или выводятся на консоль (при “**–n**”). Этот уровень рекомендуется использовать только для отладки.

Параметры потоков расшифровки

Параметры потоков расшифровки – это индивидуальные для каждого потока параметры, определяющие такие важные вещи, как ключ для расшифровки, исходный и целевой интерфейсы и др.

Src = <имя интерфейса>

Пример:

SRC = eth0

Параметр определяет интерфейс, который будет “слушаться” на предмет зашифрованного SSL-трафика.

Аналог аргумента командной строки **--src-interface (-s)**.

Dst = <имя интерфейса>

Параметр определяет интерфейс, на который будут посланы расшифрованные пакеты, допускается использование того же интерфейса, что и в параметре “**src**”.

Аналог аргумента командной строки **--dst-interface (-d)**.

Ip = <IP адрес в формате x.x.x.x>

Параметр определяет IP адрес, пакеты на который будут расшифровываться. Сюда нужно указать IP адрес, например, HTTPS сервера.

Аналог аргумента командной строки **--ip (-i)**.

Port = <номер TCP порта>

Определяет номер TCP порта назначение пакетов, которые будут расшифровываться. Для HTTPS протокола порт, скорее всего, 443.

Аналог аргумента командной строки **–port (-p)**.

Key = <path>

Путь до закрытого ключа.

Аналог аргумента командной строки **–key (-k)**.

Pwd = < passphrase>

Парольная фраза закрытого ключа.

Аналог аргумента командной строки **–pwd (-w)**.