УТВЕРЖДЕНО

RU.09445927.425530-06 32 01-ЛУ

СИСТЕМА INVGUARD CS-01

Программный комплекс invGuard CS-SW-01

Руководство системного программиста

RU.09445927.425530-06 32 01

Листов 44

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата
0063	15.11.2014			

2014

АННОТАЦИЯ

В данном программном документе приведено руководство системного программиста по настройке и использованию программного комплекса invGuard CS-SW-01 системы invGuard CS-01 (далее Очиститель), предназначенного для исследования и фильтрации вредоносного трафика в сетях передачи данных операторов связи.

В данном программном документе в разделе «Общие сведения о программе» указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы, а также требования к персоналу.

В разделе «Структура программы» приведены сведения о структуре программы, ее составных частях, о связях между составными частями и о связях с другими программами.

В данном программном документе в разделе «Настройка программы» приведено описание действий по настройке программы на условия конкретного применения.

Оформление программного документа «Руководство системного программиста» произведено по требованиям ЕСПД (ГОСТ 19.101-77, ГОСТ 19.103-77, ГОСТ 19.104-78, ГОСТ 19.105-78, ГОСТ 19.106-78, ГОСТ 19.503-79, ГОСТ 19.604-78).

СОДЕРЖАНИЕ

АННОТАЦИЯ 2
СОДЕРЖАНИЕ 3
1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ 5
1.1 Назначение программы 5
1.2 Функции программы 5
1.3 Необходимый состав технических средств 5
1.4 Компоненты, необходимые для функционирования программы 5
1.5 Требования к персоналу (системному программисту) 6
2. СТРУКТУРА ПРОГРАММЫ 6
2.1 Структура программы с описанием функций составных частей и связи между ними 6
2.2 Расположение системы на дисках 11
2.3 Конфигурационные файлы syn 13
2.4 Форматы и структуры данных 14
2.5 Реализации и использование Intel Bypass 25
2.5.1 Модуль Watchdog 25
2.5.2 Модуль проверки совместимости с Bypass 26
2.5.3 Модуль перехода из режима Bypass в режим normal 26
2.5.4 Модуль интеграции с DPDK Engine 26
3. НАСТРОЙКА ПРОГРАММЫ 27
3.1 Установка операционной системы 27
3.2 Процесс установки invGuard CS-SW-01 29
3.2.1 Требования и порядок установки компонентов и драйверов для возможности выполнения инсталляции 29
3.2.1.1 Настройка портов управления для доступа к системе 29
3.2.1.2 Установка драйвера DPDK 29
3.2.2 Процесс установки invGuard CS-SW-01 30
3.2.3 Конфигурация invGuard CS-SW-01 31
3.2.4 Настройка драйвера DPDK 31
3.2.5 Запуск invGuard CS-SW-01 32
3.2.6 Порядок действий по настройке программного комплекса для готовности к работе 32
3.2.7 Порядок контрольных проверок для определения готовности инсталлированного программного комплекса 32

3.3 Работа с электронными ключами SenseLock 33
3.4 Обновление invGuard CS-SW-01 34
3.4.1 Автоматическое обновление 34
3.4.2 Обновление в ручном режиме 36
3.5 Логирование внутреннего состояния invGuard CS-SW 36
4. ПРОВЕРКА ПРОГРАММЫ 39
5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ 39
ПРИЛОЖЕНИЕ 1 41
ПЕРЕЧЕНЬ ТЕРМИНОВ 41
Приложение 2 43
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ 43
Лист регистрации изменений 44

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1 Назначение программы

Функциональным назначением Очистителя является исследование и фильтрация вредоносного трафика, направленного на очистку (исследование).

Программный комплекс invGuard CS-SW-01 разработан для применения в составе системы invGuard CS-01, входящей в СЗСА invGuard.

1.2 Функции программы

Основные функции программы состоят в сборе статистики по трафику и нагрузке на сетевое оборудование с целью обнаружения и отражения различных атак на сеть передачи данных оператора связи.

1.3 Необходимый состав технических средств

Необходимый состав используемых технических (аппаратных) средств:

1) сервер, имеющий минимум 2 многоядерных процессора Intel с 6 и более вычислительными ядрами на каждый и частотой не менее 2,0 ГГц;
2) оперативная память объемом не менее 32 Гб;
3) жесткий диск объемом 500 Гб и выше;
4) двухпортовая сетевая карта Intel, поддерживающая технологию DPDK. Плата должна быть реализована на чипсете из списка http://dpdk.org/doc/nics.

1.4 Компоненты, необходимые для функционирования программы

Для функционирования программы необходимо следующее программное обеспечение:

1) Локализованная и сертифицированная по требованиям безопасности операционная система (например, РОСА SX «КОБАЛЬТ» 1.0);

1.5 Требования к персоналу (системному программисту)

Системный программист должен иметь минимум высшее техническое образование.

В перечень задач, выполняемых системным программистом, должны входить:

1) задача поддержания работоспособности технических средств;
2) задача установки (инсталляции) и поддержания работоспособности системных программных средств – операционной системы;
3) задача установки (инсталляции) и поддержания работоспособности Очистителя трафика.

2. СТРУКТУРА ПРОГРАММЫ

2.1 Структура программы с описанием функций составных частей и связи между ними

Программный комплекс CS-SW-01 состоит из следующих модулей:

– synctl;
– модуль управления;
– модуль вывода;
– хранилище файлов;
– модуль ввода;
– модуль фильтров;
– модуль статистики;
– модуль логирования.

Взаимодействие модулей показано на схеме (см. рис. 1). Синими стрелками показан путь прохождения трафика через систему, красными – управляющие сигналы. Слова ingress и egress подразумевают направление трафика по отношению к контролируемой сети.

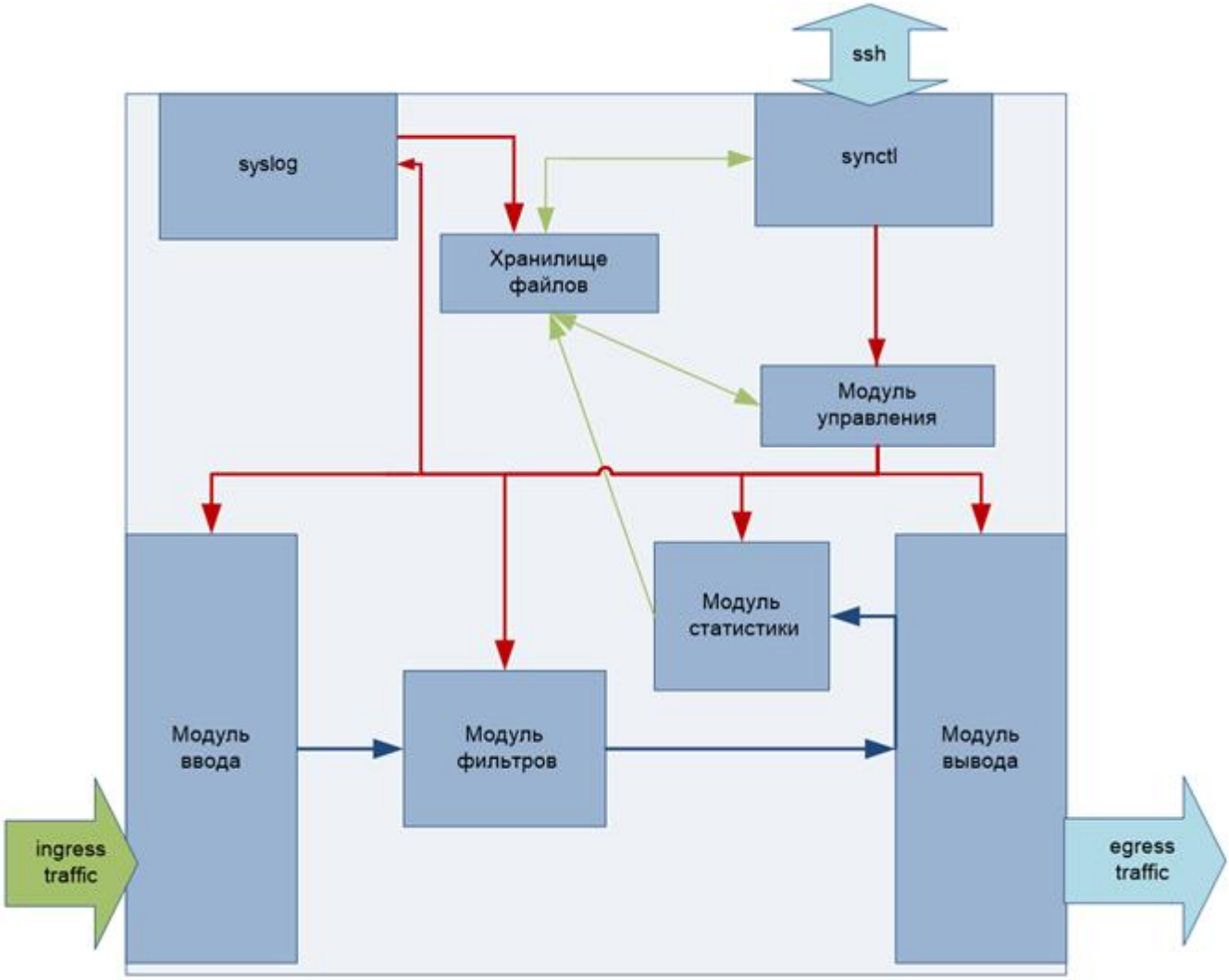

Рисунок 1 – Модульная архитектура системы invGuard CS-01

Механизм обмена данными между модулями может быть выполнен в виде следующей схемы. В системе содержится четыре состояния буфера пакетов (см. рис. 2). В один и тот же момент времени первый буфер заполняется, второй анализируется блоком фильтров, третий используется модулем вывода и по нему же рассчитывается статистика, четвертый стоит в очереди на заполнение. В случае, если первый буфер полностью заполняется, либо проходит определенный период времени, (определенный в конфигурационном файле, см. ниже), буферы меняются местами (что можно эффективно реализовать, используя обмен представлениями – без копирования данных). Таким образом, входные пакеты помещаются в буфер модулем ввода, далее этот буфер исследуется модулем фильтров, который сопоставляет с каждым проанализированным пакетом результат фильтрации – выбросить пакет (с указанием того, кто принял решение) или направить на

выходной интерфейс. Вся информация о пакете, добытая модулем фильтрации (например, результат распарсивания DNS-запроса), передаётся в виде некоторой структуры в модуль статистики для того, чтобы не делать анализ пакета дважды. После модуля фильтрации буфер направляется модулю вывода, который пересылает пакеты на выходной интерфейс. Данный буфер также используется для расчета статистики.

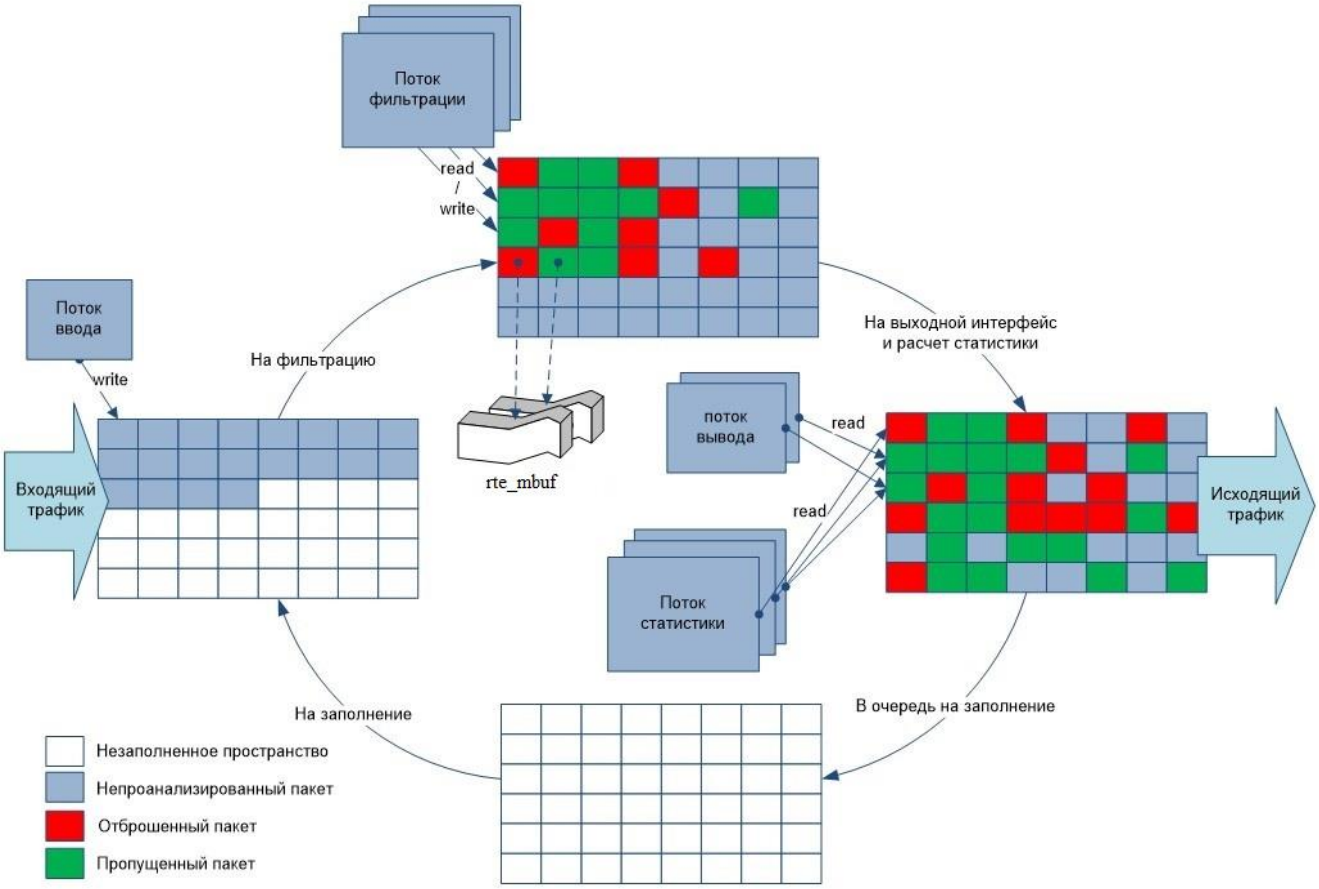

Рисунок 2 – Возможная схема обмена данными между модулями в разрезе одного процессора

Исходя из этой схемы, можно предложить модель распределения модулей по процессам, см. таблицу 1.

Таблица 1 – Распределение модулей по процессам

Процесс/модуль ядра	**Название модуля**
процесс syn	Модуль статистики
	Модуль фильтров
	Модуль логирования
	Модуль управления
процесс syn, dpdk	Модуль ввода
	Модуль вывода
операционная система	

МОДУЛЬ SYNCTL

Утилита synctl предназначена для управления системой SYN, запуска/остановки заданий очистки и вывода пользователю информации о состоянии текущих заданий очистки.

МОДУЛЬ УПРАВЛЕНИЯ

Модуль управления необходим для обеспечения корректного взаимодействия остальных модулей и выполняет следующие функции:

- осуществляет мониторинг процессоров и перезапускает их при необходимости;
- осуществляет запуск/остановку системы;
- протоколирует работу системы;
- осуществляет взаимодействие между модулями системы.

ХРАНИЛИЩЕ ФАЙЛОВ

Данный модуль предназначен для предоставления пользователю доступа к статистическим данным, событиям и конфигурационным файлам в виде объектов файловой системы, а также служит средством обмена сообщениями между модулями системы.

Интерфейс к хранилищу файлов представляется средствами операционной системы.

Возможности хранилища файлов:

- Хранилище файлов имеет возможность контроля доступа к файлам на чтение/запись для указанных пользователей;
- Хранилище файлов позволяет создавать файлы в директориях входящих сообщений для модулей за время менее 100 миллисекунд;

МОДУЛЬ ВЫВОДА

Данный модуль предназначен для обеспечения доставки трафика на устройство-получатель согласно указанным правилам доставки.

Трафик может быть доставлен на заранее сконфигурированное устройство в том виде, в котором он поступил на входной интерфейс. Однако такой метод возврата трафика стоит применять лишь в том случае, если есть уверенность, что не возникнет петель маршрутизации. Ответственность за возникновения петель маршрутизации в случае выбора такого способа доставки трафика лежит на администраторе сети.

МОДУЛЬ ВВОДА

Модуль ввода предназначен для ввода данных с сетевого интерфейса и предоставления полученного трафика остальным модулям для фильтрации, вывода и анализа при помощи механизма входных очередей DPDK.

МОДУЛЬ ФИЛЬТРОВ

Модуль фильтрации предназначен для очистки направленного на Очиститель трафика согласно заданным правилам. Модуль фильтрации получает пакеты для анализа от модуля ввода, и принимает решение о действии, которое необходимо совершить над пакетом.

МОДУЛЬ СТАТИСТИКИ

Модуль сбора статистики выполняет следующие задачи:

- анализ сырого трафика, проходящего через Очиститель;
- формирование статистики по результатам работы фильтров;
- мониторинг состояния системы;
- формирование отчётов в формате xml;
- дампинг сырого трафика.

МОДУЛЬ SYSLOG

Модуль syslog предназначен для сбора информационных сообщений и сообщений об ошибках от модулей системы.

Лог-файл системы находится в /syn/log/*.log, где * – номер процессора

Модули системы при вызове функции syslog указывают категорию каждой лог-записи.

2.2 Расположение системы на дисках

Исполняемые и конфигурационные файлы системы располагаются в следующих каталогах, см. таблицу 2.

Таблица 2 – Расположение системы на дисках

Название каталога	**Назначение**
/usr/bin/syn	Исполняемые файлы системы synctl.
/syn /syn/config	Конфигурационные файлы системы.
/syn /syn	Исполняемые файлы для Очистителя.
/syn	Домашний каталог системы.

Объекты данных

Для взаимодействия пользователя с системой служит домашний каталог системы, создаваемый в момент установки системы. Структура каталога syn описана в таблице 3.

Каталог /syn

Таблица 3 – Структура каталога /syn

Название директории	**Назначение**
/syn/	Домашний каталог пользователя syn.
/syn /syn/config/	Конфигурационные файлы системы.
/syn/.mitigs/	Параметры заданий очистки, хранимые Очистителем. Модуль управления сохраняет параметры заданий очистки в этот каталог в момент запуска задания и удаляет из него, как только очистка трафика прекращается.

Название директории	Назначение
/syn/stat/	Статистика, предоставляемая Очистителем.
/syn/stat/mitig/	Статистика по очистке трафика.
/syn/stat/tc/	Статистика по использованию ресурсов Очистителем.
/syn/stat/raw/	Статистика, собираемая Очистителем по сырому трафику.
/syn/stat/mitig/ms_XXX_TS.xml	Статистика по процессу очистки с номером XXX в момент времени TS. Файлы создаются при попытке запуска очистки и через каждые 60 секунд для активного процесса очистки. Файл автоматически перемещается в архив через определенный в конфигурационном файле период времени. По умолчанию, 24 часа.
/syn/stat/tc/tc_TS.xml	Информация об использовании ресурсов Очистителем в момент времени TS. Файлы создаются каждые 60 секунд. Файл автоматически перемещается в архив через определенный в конфигурационном файле промежуток времени. По умолчанию, 1 час.
/syn/stat/raw/raw_stat_TS.xml	Статистика, собираемая Очистителем по сырому трафику в момент времени TS – в данном релизе не поддерживается. Файлы создаются каждые 5 минут (если ведется сбор статистики). Файл автоматически перемещается в архив через определенный в конфигурационном файле период времени. По умолчанию, 1 час.
/syn/stat/archive/	Архив статистики.
/syn/stat/archive/mitig/	Архив статистики по заданиям очистки. Содержит файлы, перемещенные из папки /syn/stat/mitig/. Файлы за сутки архивируются в файл формата zip. Файлы текущих суток не архивируются. Каждый zip-файл удаляется через определенный в конфигурации промежуток времени.
/syn/stat/archive/tc/	Архив статистики по использованию ресурсов Очистителя. Содержит файлы, перемещенные из папки /syn/stat/tc/. Файлы за сутки архивируются в файл формата zip. Файлы текущих суток не архивируются. Каждый zip-файл удаляется через определенный в конфигурации промежуток времени.
/syn/stat/archive/raw/	Архив статистики по сырому трафику. Содержит файлы, перемещенные из папки /syn/stat/raw/. Файлы за сутки архивируются в файл формата zip. Файлы текущих суток не архивируются. Каждый zip-файл удаляется через определенный в конфигурации промежуток времени.

Название директории	Назначение
/syn/log/	log-файлы Очистителя. Данный каталог также предназначен для вывода отладочной информации Очистителя.
/syn/log/	Логи Очистителя. Ведется при помощи демона syslog.
/syn/alerts/	Оповещения Очистителя. Одно оповещение представлено одним файлом. Именем файла является *alert_TS_XXX.xml*, где *TS* – время создания, XXX – число для устранения неоднозначности. Файлы удаляются пользователем (анализатором), однако, во избежание переполнения диска, раз в сутки модуль управления удаляет все файлы, старше чем 24 часа.
/syn/config/statparams.xml	Файл с параметрами сбора статистики по сырому трафику.
/syn/.msg/	Директория для обмена сообщениями между модулями.
/syn/.msg/synctl	Директория входящих сообщений для утилиты synctl.
/syn/.msg/control	Директория входящих сообщений для модуля управления.
/syn/.msg/filter	Директория входящих сообщений для модуля фильтрации.
/syn/.msg/stat	Директория входящих сообщений для модуля статистики.
/syn/.msg/input	Директория входящих сообщений для модуля ввода.
/syn/.msg/output	Директория входящих сообщений для модуля вывода.

TS – момент времени в формате yyyymmdd_hhmmss. Если файл предоставляет статистику за интервал времени от *start_time* до *end_time*, то TS должен быть равен *end_time*.

Далее, для краткости, будем ссылаться на вышеописанные файлы без указания временной метки TS и номера задания очистки. То есть, файл ms_XXX_TS.xml будем называть *ms.xml*, tc_TS – *tc.xml,* raw_stat_TS.xml – *raw_stat.xml*.

2.3 Конфигурационные файлы syn

Конфигурация Очистителя представлена следующими файлами, расположенными в syn/syn/config:

– config.xml – содержит общие параметры Очистителя;

– config.txt – содержит настройки логирования;

– statparams.xml – описывает параметры собираемой по сырому трафику статистики.

Конфигурация модуля вывода

Конфигурация модуля вывода содержится в конфигурационном файле Очистителя config.xml. Файл содержит параметры, влияющие на функционирование модуля вывода, см. таблицу 4.

Таблица 4 – Конфигурационные параметры модуля вывода

Название параметра	Определение параметра	Описание параметра
Схема включения Очистителя	Атрибут *type* элемента *deployment* файла *config.xml*	Схема включения Очистителя определяет способ возврата трафика.
Устройство для возврата трафика	Атрибут *next_hop* элемента *tcparams* файла *config.xml*	Определяет next-hop ip для возврата трафика в случае inline схемы включения.
Интерфейс	Представлен атрибутом *name* элемента *iface* со значением *output* атрибута *type*, файл *config.xml*	Интерфейс Очистителя, используемый для вывода трафика.

Параметры модуля вывода содержатся в конфигурационном файле *config.xml*.

2.4 Форматы и структуры данных

Форматы файлов

config.xml

```
<?xml version="1.0" encoding="utf-8"?>

<!--
    Параметры очистителя.
    drop_fragmented     - пропускает ли очиститель
                          фрагментированные пакеты.
    resume_mitigs_on_error - стоит ли перезапускать задания
                             очистки в случае ошибки.
    debug               - отладочная опция - включается в случае
```

```
                        необходимости сбора отладочной
                        информации о системе
-->
<tcparams
  drop_fragmented="true"
  resume_mitigs_on_error="true"
  debug="false">     - разрешает или запрещает логирование

  <!--
      Параметры размещения очистителя.
      type              - схема включения очистителя
                          o - offramp
                          o - inline
                          o - portspan
      next_hop          - ip адрес устройства, на которое
                          пересылается трафик в случае
                          offramp и inline схем включения
                          очистителя
  -->
<deployment type="inline">

    <!--
        Физические интерфейсы очистителя.
    -->
    <interfaces>
      <!--
        Описание физического канала.
        input - имя входного порта
        output - имя выходного порта
        mac_input - входной mac-адрес
        ip_input - входной ip-адрес
        mac_output - выходной mac-адрес
        ip_output - выходной ip-адрес
        next_hop_forward - ip-адрес устройства для
перенаправления трафика
      -->
    <iface input="gbe1" ip_input="192.168.17.3"
ip_output="192.168.18.3" mac_input="00:00:00:00:00:00"
mac_output="00:00:00:00:00:00" next_hop_forward="192.168.18.1"
output="gbe3"/>
<!-
     bsx_rx_io - длина входной очереди модуля ввода
     bsx_tx_io - длина выходной очереди модуля вывода
     bsx_wx_io - длина очереди модуля фильтрации
     coremask - маска доступных процессоров
      dedicated - очереи сетевой карты, выдеоенный для ARP задач
     input_ports - описание входных очередей с привязкой к
процессорам
```

```
     output_ports - описание выходных очередей с привязкой к
процессорам
     workers - описание тредов филтрации с привязкой к
процессорам
→
    <dpdk>
      <config bsx_rx_io="(144,144)"
              bsx_tx_io="(144,144)"
              bsx_wx_io="(144,144)"
              coremask="0xffffff"
              dedicated="(port1_tx,1,0),(port2_tx,1,0)"

input_ports="(port1_rx,0,1),(port1_rx,1,2),(port1_rx,2,4),(port1_
rx,3,5),(port1_rx,4,6),(port1_rx,5,7),(port1_rx,6,8),(port1_rx,7,
9),(port1_tx,0,0)"
              output_ports="(port2_rx,0,0),(port2_tx,0,3)"
workers="21"/>
    </dpdk>
```

```
  <iface input="gbe2" ip_input="192.168.17.4" ip_output="192.168.18.4"
mac_input="00:00:00:00:00:00" mac_output="00:00:00:00:00:00"
next_hop_forward="192.168.18.2" output="gbe4"/>
  </interfaces>
  <dpdk>
   <config input_ports="(portx,1,0),(portx,1,0)" output_ports="(portx,1),(portx,1)"
workers="2,3"/>
  </dpdk>
  <!--
  routers содержит ip  адреса легетимных роутеров, которым разрешено сообщать свой
MAC-адрес
  -->
<routers><ip>192.168.17.1</ip><ip>192.168.17.2</ip></routers>
```

```
  <!-- Параметры хранения информации в каталоге /syn/stat/ -->
  <storage>

    <!--
        Параметры хранения информации в подкаталоге
        path             - путь к подкаталогу относительно
                           /syn/stat
        minutes_to_keep    - время хранения информации в
                           неизменном виде, в минутах.
        days_to_keep_archived - время хранения заархивированной
                           информации, в днях.
    -->
    <dir path="/mitig/" minutes_to_keep="1440"
         days_to_keep_archived="30"/>
```

```
    <dir path="/tc/" minutes_to_keep="60"
days_to_keep_archived="30"/>
    <dir path="/raw/" minutes_to_keep="60"
days_to_keep_archived="30"/>
  </storage>

  <!--
      Параметры модулей.
  -->
  <modules>

    <!--
        Параметры модуля управления.
        ping_timeout    - допустимое время ответа модуля системы
                          на ping-сообщение, в секундах
        ping_interval   - интервал опроса модулей на доступность
                          при помощи ping-сообщений, в секундах
    -->
    <control ping_timeout="10" ping_interval="60" />

    <!--
        Параметры модуля ввода.
        buffer_size      - размер пакетного буфера, в Мб
        swap_time        - время переключения между буферами, мс
       max_mtu          - максимальный размер ethernet фрейма,
                          принимаемого модулем ввода
    -->
    <input buffer_size="32" swap_time="100" max_mtu="2000"/>

    <!--
        Глобальные параметры для блока фильтров
              enabled         - включена фильтрация или нет.
                                Здесь глобальные настройки
                                переопределяют локальные.
    -->
    <filters enabled="true">

      <!--
          Глобальный список исключений. Содержит правила на языке
          фингерпринтов, применяемые на входе очистителя
      -->
      <exception_list>
        <filter>drop proto 0</filter>
        <filter>drop proto icmp</filter>
        <filter>drop net 127.0.0.0/8</filter>
        <filter>drop net 10.0.0.0/8</filter>
        <filter>drop net 172.16.0.0/12</filter>
        <filter>drop net 192.168.0.0/16</filter>
        <filter>drop net 224.0.0.0/4</filter>
```

```
        <filter>drop net 240.0.0.0/5</filter>
        <filter>drop tflags /SAFRPUEW</filter>
        <filter>drop tflags FUP/FUP</filter>
        <filter>drop tflags SR/SR</filter>
        <filter>drop tflags SF/SF</filter>
      </exception_list>

      <!--
          Параметры фильтра "черный и белый списки"
      -->
      <bwlist />

      <!--
          Параметры фильтра "динамический черный список"
      -->
      <dynamic_filters />

      <!--
          Параметры фильтра "исследование содержимого пакетов"
      -->
      <payload />

      <!--
          Параметры фильтра "исследование заголовков http-
пакетов"
      -->
      <http_hdr />

      <!--
            Параметры фильтра "Выравнивание тренда по /24
адресам"
      -->
      <baseline_24 />

      <!--
          Параметры фильтра "Выравнивание тренда по протоколам"
      -->
      <baseline_proto />

      <!--
          Параметры фильтров в блоке "контрмеры"
      -->
      <countermeasures>

        <!--
            Параметры фильтра "TCP-аутентификация"
            time_to_block     - время блокировки
                                неаутентифицированного хоста,
                                секунд
```

```
            white_list_size   - максимальное количество элементов
                                в белом списке
            gray_list_size    - максимальное количество элементов
                                в сером списке
            connection_credit - масимальное число соединений,
                                после которого аутентифицированны
                                хост вновь
                                подвергается аутентификации
            trust_time        - время, по истечении которого
                                аутентифицированный хост вновь
                                подвергается аутентификации
        -->
        <tcp_auth time_to_block=”60” white_list_size=”10000000”
                  gray_list_size=”30000000”
connection_credit="1000”
                  trust_time=”300” />

        <!--
            Параметры фильтра "Сброс TCP-соединений"
        -->
        <tcp_reset />

        <!--
            Параметры фильтра "Блокирование зомби"
        -->
        <zombie />



        <!--
            Параметры фильтров в блоке http
        -->
        <http>

          <!--
              Параметры фильтра "Фильтрация вредоносных
              HTTP-запросов"
          -->
          <http_rfc />

          <!--
              Параметры фильтра "Ограничение числа HTTP-запросов
от объекта"
          -->
          <request_limit />

          <!--
              Параметры фильтра "Ограничение числа HTTP-запросов
к объекту"
```

```
            -->
            <objects_limit />
          </http>

          <!--
              Параметры фильтров в блоке DNS
          -->
          <dns>
            <!--
                Параметры фильтра "Фильтрация вредоносных DNS-
запросов"
            -->
            <dns_rfc />

            <!--
                Параметры фильтра "DNS-аутентификация"
            -->
            <dns_auth />
          </dns>

          <!--
              Параметры фильтров в блоке VoIP
          -->
          <voip>

            <!--
                Параметры фильтра "Фильтрация вредоносных SIP-
запросов"
            -->
            <sip_rfc  />

            <!--
                Параметры фильтра "Ограничение числа SIP-запросов"
            -->
            <sip_src_limit  />
          </voip>
        </countermeasures>
      </filters>

      <!--
          Параметры шейпера.
      -->
      <shaping>
      </shaping>

      <!--
          Параметры модуля вывода.
          drop_threshold  - количество непереданных пакетов, в
                            процентах от общего, при котором
```

```
                        генерируется сообщение, что модуль
                        вывода не успевает обрабатывать пакеты.
    -->
    <output drop_threshold="5"/>
  </modules>

  <!--
      Параметры дампинга сырого трафика.
      max_file_size     - максимальный размер файла, в
мегабайтах.
     max_sessions       - максимальное количество одновременно
                          проводимых процессов дампинга сырого
                          трафика.
  -->
  <rawsampling max_file_size="10" max_sessions="5" />
</tcparams>
```

statparams.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  Параметры статистики
  Общие замечания:
  1. Если нескольким элементам из одной группы (TCP, UDP)
     соответствует одно и то же имя name, то результирующая
     статистика для данного имени возвращается одной строчкой,
     как суммарная статистика для данного имени.

  enabled - стоит ли собирать статистику по сырому трафику.
-->
<statparams enabled="true">

  <!--
      Параметры расчета статистики по DNS
  -->
  <dns>
    <!--
        TOP FQDN - количество наиболее запрашиваемых полных
        доменных имен в контролируемой подсети
        enabled         - собирать ли статистику такого рода
        count           - количество самых запрашиваемых имен
        host_count      - количество самых запрашивающих хостов
                          для одного самого запрашиваемого имени
        name_count      - количество имен, возвращаемых для
                          каждого хоста.
    -->
    <dns_topfqdn enabled="true" count="100" host_count="10"
                 name_count="10" />
```

```
  <!--
       TOP RDN - количество наиболее запрашиваемых коротких
       доменных имен в контролируемой подсети
       enabled         - собирать ли статистику такого рода
       count           - количество самых запрашиваемых хостов
       host_count      - количество самых запрашивающих хостов
                         для одного самого запрашиваемого хоста
       name_count      - количество имен, возвращаемых для
                         каждого хоста.
  -->
  <dns_toprdn enabled="true" count="100" host_count="10"
              name_count="10"/>

  <!--
      Отслеживание частоты запросов к DNS-серверам.
      enabled          - отслеживать ли частоту запросов к
                         DNS-серверам
      percentile       - процентиль для вычисления порога
      multiplier       - множитель для вычисления порога
      host_count       - количество возвращаемых top-активных
                         хостов
  -->
  <baseline_alerts enabled="true"
                   percentile="95"
                   multiplier="1.1"
                   sensitivity="150"
                   host_count="100" />

</dns>

<!--
    Параметры сбора статистики по приложениям.
    tcp_enabled        - собирать статистику по tcp-приложениям
    udp_enabled        - собирать статистику по udp-приложениям
    icmp_enabled       - собирать статистику по icmp-приложениям
    count              - количество возвращаемых наиболее
                         распространенных приложений для каждого
                         протокола.
-->
<apps tcp_enabled="true" udp_enabled="true" icmp_enabled="true"
      count="1000">

  <!--
      Сигнатура одного из dpi-приложений.
      name               - название приложения
      enabled            - собирать статистику по этому
                           приложению.
      proto              - протокол. Разрешенные значения: tcp и
                           udp
```

```
        port              - порт.
    -->
    <dpi name="tcp_app1" enabled="true" proto="tcp" port="80">

      <!--
          Необязательное выражение, уточняющее характеристики
          трафика, соответствующего приложению.
      -->
      <fingerprint>fingerprint expression</fingerprint>

      <!--
          Необязательное выражение, определяющее содержимое
пакетов трафика, соответствующего приложению.
      -->
      <payload>payload expression</payload>
    </dpi>
  </apps>

  <!--
      Статистика по HTTP-запросам
  -->
  <http>
    <!--
        Статистика по наиболее запрашиваемым URL, определенными
        полными доменными именами (www.google.com).
        enabled           - собирать ли статистику такого рода.
        count             - количество возвращаемых записей по
                            наиболее запрашиваемым URL
        host_count        - количество наиболее запрашивающих
                            хостов для каждого URL
    -->
    <http_topfqdn enabled="true" count="100" host_count="10" />

    <!--
        Статистика по наиболее запрашиваемым URL, определенными
        сокращенными доменными именами (*.google.com).
        enabled           - собирать ли статистику такого рода.
        count             - количество возвращаемых записей по
                            наиболее запрашиваемым URL
        host_count        - количество наиболее запрашивающих
                            хостов для каждого URL
    -->
    <http_toprdn enabled="true"  count="100" host_count="10" />

    <!--
        Статистика по наиболее запрашиваемым документам.
        enabled           - собирать ли статистику такого рода.
        count             - количество возвращаемых записей по
                            наиболее документам
```

```
      host_count        - количество наиболее запрашивающих
                          хостов для каждого документа
  -->
  <http_topdocs enabled="true" count="100" host_count="10"  />

  <!--
      Статистика по наиболее запрашиваемым типам MIME
      enabled           - собирать ли статистику такого рода.
      count             - количество возвращаемых записей по
                          наиболее запрашиваемым типам MIME
      host_count        - количество наиболее запрашивающих
                          хостов для каждого типа MIME
  -->
  <http_topmime enabled="true"  count="100" host_count="10" />
</http>

<!--
    Распределение пакетов по размерам.
    enabled           - собирать ли статистику такого рода.
-->
<stat_by_size enabled="true" />

<!--
    Распределение пакетов по протоколам.
    enabled           - собирать ли статистику такого рода.
-->
<stat_by_proto enabled="true" />

<!--
    Распределение пакетов по TOS.
    enabled           - собирать ли статистику такого рода.
-->
<stat_by_tos enabled="true" />

<!--
    Статистика по VoIP
    enabled           - собирать ли статистику такого рода.
    count             - количество возвращаемых звонков.
-->
<voip enabled="true" count="2000"/>

<!--
    Параметры сбора статистики по выравниванию тренда по
    /24 адресам.
    ret_count       - количество возращаемых записей в
                      статистике по заданиям очистки
-->
<baseline_24 ret_count="100" />
```

```
  <!--
      Параметры сбора статистики по выравниванию тренда по
      протоколам.
      ret_count       - количество возвращаемых записей в
                        статистике по заданиям очистки
  -->
  <baseline_proto ret_count="100" />
</statparams>
```

2.5 Реализации и использование Intel Bypass

Для программного комплекса invGuard CS-SW-01 реализована поддержка технологии Intel Bypass. Режим Bypass включается после истечения заданного интервала времени, при условии, что в течение данного интервала invGuard CS-SW-01 не сбросит таймер байпасного Watchdog. Сброс Watchdog timer производится самим программным комплексом один раз в заданный интервал времени. Таким образом, технология Intel Bypass реализована при помощи Intel DPDK таймера и Intel Bypass Watchdog. Для поддержки технологии Intel Bypass реализованы следующие программные компоненты:

1) Модуль Watchdog для ActionScheduler;
2) Модуль проверки совместимости с Bypass;
3) Модуль перехода из режима Bypass в режим normal;
4) Модуль интеграции с DPDK Engine.

2.5.1 Модуль Watchdog

Данный модуль используется для проверки включения режима Intel Bypass. Реализован в коде в файлах watchdog.cpp, watchdog.h, ActionScheduler.cpp, ActionScheduler.h.

По сигналу таймера из ActionScheduler вызывается данный модуль, обновляя время наступления следующего события для захода в режим Bypass. При этом удаляется текущий статус перехода в Bypass.

Если таймер не валидный (некорректный), то режим Bypass уже включен и не может быть сброшен. При этом сетевая карта не восстанавливает нормальный

режим функционирования, что означает либо зависание Очистителя, либо режим «голодания» или отсутствие процесса в файловой системе /proc.

Для понимания, что Очиститель завис была реализована следующая функциональность: по истечении времени таймера invGuard CS-SW-01 каждый раз записывает в файл heartbeat.txt текстовую строку «I am alive», которую будет проверять хостовый Watchdog.

2.5.2 Модуль проверки совместимости с Bypass

В модуль работы с DPDK Engine внедрён код, проверяющий сетевую карту на поддержку технологии Intel Bypass. Если карта поддерживает данную технологию, то выполняется инициализация карты в данном режиме. В противном случае происходит обычная инициация сетевой карты в режиме DPDK. Данная функциональность внесена в файлы dpdk_engine.cpp и dpdk_engine.h

2.5.3 Модуль перехода из режима Bypass в режим normal

При запуске invGuard CS-SW-01 всегда проверяется поддержка сетевой картой режима Bypass. Проверка наличия поддержки встроена в код файлов dpdk_engine.cpp и dpdk_engine.h. Если карта поддерживает данный режим, то выполняется переход из режима Bypass в режим нормального функционирования, чтобы Очиститель начал/продолжил работу по заданным алгоритмам. В режиме Bypass функционал invGuard CS-SW-01 не работает.

2.5.4 Модуль интеграции с DPDK Engine

Для интеграции модуля Bypass с модулем работы с DPDK Engine был создан интерфейс IWatchdog, реализованный в коде watchdog.cpp, watchdog.h, ActionScheduler.cpp, ActionScheduler.h, dpdk_engine.cpp и dpdk_engine.h. Данный интерфейс реализует функционал DPDK Engine, тем самым связывая алгоритм реализации поддержи Intel Bypass с внешними потребителем – ActionScheduler. ActionScheduler ничего не знает о реализации Bypass в DPDK Engine, он просто

использует интерфейс IWatchdog у объекта DPDK Engine для использования Intel Bypass.

Переход в режим Bypass происходит по таймеру с интервалом в 32 секунды. Исходя из соображений, что процедура сброса таймера притормаживает процесс обработки пакетов самой картой (согласно документации на Intel DPDK) и делать её желательно как можно реже. Интервал выбран и не может быть изменён через конфигурационные файлы. Очиститель в течение этого времени сбросит таймер или перейдёт в режим Bypass. После перехода из режима Bypass модуль Watchdog перезапускает invGuard CS-SW-01 и выключает режим Bypass.

3. НАСТРОЙКА ПРОГРАММЫ

3.1 Установка операционной системы

В качестве операционной системы используется дистрибутив локализованной и сертифицированной по требованиям безопасности операционной системы РОСА SX «КОБАЛЬТ» 1.0.

Процесс установки операционной системы зависит от конкретной аппаратной платформы, ниже описаны основные этапы.

Замечание: В процессе инсталляции все данные, находящиеся на сервере, будут утеряны.

1) Перед установкой BIOS сервера должен быть настроен на следующий порядок загрузки:
   а) CD-DVD ROM;
   б) Жесткий диск.
2) Включите сервер, вставьте стандартный диск установки ОС РОСА, перезагрузить сервер. Загрузится программа установки с компакт диска.
3) В меню «Welcome to POCA SX64 "COBALT"» выберите пункт
   – Install or upgrade an existing system

   и нажмите клавишу Enter. Начнется загрузка программы установки.

4) В появившемся экране программы установки в диалоге «Disk Found» выберите «Skip».
5) Запустится графический интерфейс пользователя программы установки с возможностью работы с мышью. На экране приветствия нажмите «Next».
6) Следующий диалог – диалог выбора языка для процесса инсталляции. Выберите «Russian (Русский)», нажмите «Next».
7) Диалог выбора раскладки клавиатуры. Выберите «Русская», если не выбрана. Нажмите «Next».
8) «Какой тип устройств будет использоваться при установке?» – выберите «Стандартные накопители», нажмите «Далее».
9) В диалоге «Присвойте этому компьютеру имя...» поставьте имя по умолчанию.
10) В диалоге выбора часового пояса выберите часовой пояс, например, «Европа/Москва», нажмите «Далее».
11) Введите пароль для пользователя root в верхней строке диалога и в строке подтверждения второй раз. Нажмите «Далее».

Замечание: пароль пользователя root необходимо запомнить, так как он необходим далее в процессе установки.

12) Диалог «Какой тип установки вы предпочитаете?» – выберите «Все пространство», нажмите «Далее», подтвердите действие нажатием кнопки «Сохранить изменения на диск» в появившемся диалоге. Происходит создание файловой системы.
13) Выберите тип системы: «Software Dewelopment workstation», нажмите «Далее». Запустится процесс установки пакетов и конфигурации, который может занять некоторое время.
14) После окончания установки нажмите кнопку «Перезагрузка».
15) ОС РОСА загрузится с жесткого диска. На экране приветствия нажмите кнопку «Вперед».
16) Экран информации о лицензии, нажмите «Вперед».

17) Экран добавления пользователя, введите в форму информацию о пользователе: имя (логин), полное имя, пароль и подтверждение пароля. Нажмите «Вперед».
18) Экран установки времени - установите время, нажмите «Готово».
19) Возникнет приглашение для входа в систему. Нажмите на имя пользователя, по запросу введите пароль. Вход выполнен, появится рабочий стол пользователя.
20) Перезагрузите серевер.

3.2 Процесс установки invGuard CS-SW-01

3.2.1 Требования и порядок установки компонентов и драйверов для возможности выполнения инсталляции

3.2.1.1 Настройка портов управления для доступа к системе

Настройка портов системы производиться путём редактирования файлов в соответствии с техническим решением:

1) /etc/udev/rules.d/70-persistent-net.rules
2) /proc/net/vlan/config
3) /etc/sysconfig/network-scripts/ifcfg-eth*
4) /etc/sysconfig/iptables

3.2.1.2 Установка драйвера DPDK

Выполните вход в систему под пользователем «root». Подключите и смонтируйте диск с операционной системы ROSA SX64 "COBALT" (mount -t iso9660 /dev/sr0 /media/ROSA-SX64-1.0).

Выполните команды:

1) yum clean all
2) yum install glibc.i686

Для установки invGuard CS-SW-01:

- Проверьте контрольную сумму дистрибутива. Контрольная сумма должна соответствовать значению, приведенному в документе RU.09445927.425530-03 30 01 «Формуляр».
- Откройте консоль пользователя «root». Скопируйте и распакуйте дистрибутив с Очистителя в каталог /opt/cs-x86

Выполните набор команд для версии очистителя invGuard CS-SW-01 1.2

1) mkdir -p /opt/cs-x86
2) cd /opt/cs-x86
3) tar -xvf install_cleaner_x86_1.2.tar.gz

Дистрибутив invGuard CS-SW-01 версии 1.2 имеет следующую структуру:

1) install_cleaner_x86_1.2/
2) install_cleaner_x86_1.2/cleaner_x86_11Nov2014_usr_bin_syn.tar.bz2
3) install_cleaner_x86_1.2/cleaner_x86_11Nov2014_syn_syn_syn.bz2

Выполните набор команд:

1) cd install_cleaner_x86_1.2
2) tar -xvf cleaner_x86_11Nov2014_usr_bin_syn.tar.bz2
3) tar -xvf cleaner_x86_11Nov2014_syn_syn_syn.bz2

Структура исходных файлов и директорий после разархивирования должна быть следующей:

1) syn
2) usr

Откройте консоль пользователя «root» и введите команды:

1) cd /opt/cs-x86/install_cleaner_x86_1.2
2) \cp -rf syn /
3) \cp -rf usr /

Выполните установку драйвера DPDK:

1) cd /syn/syn
2) tar -xvf dpdk-[версия DPDK].tar.gz
3) Проверьте grep -i numa /var/log/dmesg
4) Выполните /syn/syn /sc_dpdk_startup.sh
5) Откройте файл /etc/rc.local
6) Добавьте в файл строку /syn/syn/sc_dpdk_startup.sh

3.2.2 Процесс установки invGuard CS-SW-01

Выполните установку дополнительных библиотек используя команды:

1) cd /syn/syn
2) tar -xvf re2-20140304.tgz

3) cd re2
4) make clean
5) make
6) make install
7) cp ./re2/obj/so/libre2.so.0 /usr/lib64
8) unzip libxml2-2.9.1.zip
9) cd libxml2-2.9.1
10) ./autogen.sh
11) ./configure --build=x86_64-unknown-linux-gnu
12) make clean
13) make
14) make install
15) cp libxml2-2.9.1/.libs/libxml2.so.2.9.1 /usr/lib64/
16) unlink libxml2.so.2
17) ln -s libxml2.so.2.9.1 libxml2.so.2

3.2.3 Конфигурация invGuard CS-SW-01

Выполните следующие действия:

1) cd /var/spool/cron/
2) touch root
3) В созданный файл запишите: 0 0 * * * /usr/bin/syn/rotate (добавить в конце перевод строки)
4) Откройте файл /etc/rc.local
5) Добавьте в него строку /usr/bin/syn/synctl startsystem
6) mkdir /var/log/syn
7) touch /var/log/syn/syn.log
8) Произведите синхронизацию времени между Очистителем и Анализатором (проверьте часовой пояс).
9) rm -rf /etc/localtime
10) ln -s /usr/share/zoneinfo/Europe/Moscow /etc/localtime

3.2.4 Настройка драйвера DPDK

1) Укажите DPDK-конфигурационные данные в /syn/config/config.xml в соответствии с количеством ядер/процессоров.

Например,

```
<iface input="gbe1" ip_input="127.0.0.1" ip_output="127.0.0.1"
mac_input="64:66:B3:04:1D:EE" mac_output="64:66:B3:04:1D:EE"
next_hop_forward="127.0.0.1" output="gbe2"/>
</interfaces>
```

```
<dpdk>
<config bsx_rx_io="(14,14)" bsx_tx_io="(14,14)" bsx_wx_io="(14,14)"
coremask="0xff" input_ports="(port1_rx,0,0),(port1_tx,0,0)"
output_ports="(port2_rx,0,1),(port2_tx,0,1)" workers="2,3,4"/>
 </dpdk>
```

– для восьми процессоров атрибут coremask="0xff";
– номер 1 в значении атрибута input="gbe1" должен соответствовать номеру 1 в значении атрибута input_ports="(port1_rx,0,0),(port1_tx,0,0)",
где port1_rx – входной порт, port1_tx – выходной порт, 0 – номер очереди сетевой карты, 0 – номер ядра.
– номер 2 в значении атрибута output="gbe2" должен соответствовать номеру 2 в значении атрибута output_ports="(port2_rx,0,1),(port2_tx,0,1)",
где port2_rx - входной порт, port2_tx – выходной порт, 0 – номер очереди сетевой карты, 1 – номер ядра.
– значение атрибута workers="2,3,4" должно начинаться с номера следующего ядра.

3.2.5 Запуск invGuard CS-SW-01

Выполните следующие действия:

1) перезапустите систему
2) вставьте ключ SenseLock
3) после перезапуска выполните команду: /usr/bin/syn/synctl ping
4) вывод должен содержать строку «Ping succeeded» - это подтверждает, что invGuard CS-SW-01 запущен

Для дальнейшей настройки Системы должен использоваться веб-интерфейс.

Остановка Системы может быть завершена с помощью команды (пользователь root):

1) /usr/bin/syn/synctl stopsystem

3.2.6 Порядок действий по настройке программного комплекса для готовности к работе

Порядок подготовки к работе:

Выполните сбор данных для настройки очистителя через веб-интерфейс (форма Администрирование / Подавление атак / Управление очистителями).

3.2.7 Порядок контрольных проверок для определения готовности инсталлированного программного комплекса

Перечень проверок системы:

- Через веб-интерфейс:
  1) Состояние системы «Очиститель»: запущен
- Через ssh-подключение:
  1) Состояние запуска необходимых «демонов» для invGUARD CS-SW:
  2) /opt/tilera/TileraMDE-4.2.4.174600/tilegx/bin/tile-monitor
  3) /usr/bin/syn/synctl ping (ответ Ping succeeded)

3.3 Работа с электронными ключами SenseLock

Работа с комплексом invGuard CS-SW-01 невозможна без использования электронного ключа SenseLock, служащего для защиты комплекса от несанкционированного использования и копирования.

Утилита licenseTool.x, поставляемая совместно с системой invGuard, предназначена для работы с электронными ключами SenseLock. Данная программа предназначена для удаленного обновления ключа, а так же для вывода информации о действующей лицензии.

Использование:

licenseTool.x -p userPin --c2v <filename>

licenseTool.x -p userPin --v2c <filename>

licenseTool.x -p userPin –info

где userPin – пин-код пользователя.

Параметры команд:

-i, --info вывод информации о действующей лицензии.

-C, --c2v генерировать c2v-файл из SenseLock-ключа (от пользователя (customer) к производителю (vendor) в c2v-файле (customer to vendor)).

-V, --v2c загрузить v2c-файл в SenseLock-ключ (v2c – vendor to customer).

Опции:

-v, --verbose

-q, --quite
-h, --help

Пример генерации запроса с секретными данными для лицензии и обновление лицензии:

licenseTool.x -C license.c2v – экспорт лицензии с ключа, для формирования обновлённой лицензии.

licenseTool.x -V license.v2c – обновление лицензии на ключе из v2c-файла.

Сведения о возникших в ходе работы предупреждениях или ошибках фиксируются в системном журнале.

3.4 Обновление invGuard CS-SW-01

3.4.1 Автоматическое обновление

Для проведения обновления программного комплекса invGuard CS-SW-01 необходимо использовать интерфейс invGuard AS-SW. Для обновления в автоматическом режиме необходимо использовать скрипт updater.sh Данный скрипт должен запускаться как бинарный файл с параметрами. Параметры должны быть следующими:

1) updater.sh version cleaner – показывает текущую версию установленного ПК invGuard CS-SW-01;
2) updater.sh install cleaner – устанавливает текущую базовую версию invGuard CS-SW-01;
3) updater.sh list cleaner – распечатывает список доступных версий invGuard CS-SW-01 в репозитории.

Скрипт updater.sh необходимо запускать из консоли invGuard AS-SW

После запуска updater.sh считывает свой конфигурационный файл настроек, в котором должны быть прописаны пути установки системы, адрес сервера репозитория, логин и пароль доступа к репозиторию, настройки доступа по сетевому протоколу SSH.

Конфигурацию updater.sh необходимо хранить в файле update_config.xml. Данный файл необходимо расположить в одном каталоге с файлом updater.sh. Файл updater.sh необходимо хранить в каталоге /home пользователя Системы. Дополнительные бинарные файлы и библиотеки должны быть расположены в каталоге /lib/update/*.

Пример файла конфигурации update_config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <cleaner major_version="000" minor_version="000" build="0001" host="192.168.20.19" port="22" type="x86"/>
  <analyzer major_version="001" minor_version="001" build="0007" db_host="127.0.0.1" db_port="3306" db_name="install_syn"/>
  <connection host="192.168.20.19" port="22"/>
  <path>
      <remote main_folder="/opt/updater_check/"/>
      <local backup_folder="/opt/updater/backup/" tmp_folder="/opt/updater/temp/"/>
      <cleaner backup_folder="/opt/updater/backup2/" tmp_folder="/opt/updater/temp2/"/>
  </path>
</config>
```

Узел <remote main_folder = “”> описывает путь к файлам репозитория на удаленном сервере.

Узел <local backup_folder=””> описывает путь, куда сохраняется резервная копия текущей версии.

Узел <local tmp_folder =””> описывает путь, куда временно скачиваются файлы обновления из репозитория, для дальнейшей локальной работы с ними.

Узлы <cleaner> и <analyzer> описывают текущие версии, установленных компонентов Системы invGuard.

В командной строке требуется выбрать нужную версию для установки. При запуске скрипта с параметром install устанавливается только базовая версия, находящаяся в репозитории. Если базовая версия ниже, чем предлагается для обновления, то после установки базовой, необходимо выполнять пошаговое обновление до максимальной (см. п. 3.4.2).

Во время работы скрипта updater.sh и при возникновении ошибочных ситуаций на консольный вывод выводятся информационные сообщения с причинами ошибки, которые должны анализироваться и устраняться администратором.

3.4.2 Обновление в ручном режиме

Для обновления Очистителя в ручном режиме необходимо использовать скрипт updater.sh Данный скрипт должен запускаться как бинарный файл с параметрами. Параметры должны быть следующими:

1) updater.sh version cleaner – показывает текущую версию установленного ПК invGuard CS-SW-01;
2) updater.sh update cleaner – обновляет версию ПК invGuard CS-SW-01;
3) updater.sh downgrade cleaner – понижает версию ПК invGuard CS-SW-01;
4) updater.sh list cleaner – распечатывает список доступных версий invGuard CS-SW-01 в репозитории.

Скрипт updater.sh запускается из консоли invGuard AS-SW.

После запуска updater.sh должен считывает свой конфигурационный файл настроек, в котором должны быть прописаны пути установки системы, адрес сервера репозитория, логин и пароль доступа к репозиторию, настройки доступа по сетевому протоколу SSH.

Конфигурацию updater.sh необходимо хранить в файле update_config.xml. Данный файл необходимо расположить в одном каталоге с файлом updater.sh. Файл updater.sh необходимо хранить в каталоге /home пользователя Системы. Дополнительные бинарные файлы и библиотеки должны быть расположены в каталоге /lib/update/*. Файл должен соответствовать приведённому в п. 3.4.1.

3.5 Логирование внутреннего состояния invGuard CS-SW

Логирование (журналирование) внутреннего состояния – инструмент для детализации сведений о происходящих в Системе событиях, диагностики поведения

программных модулей и возникающих в них ошибок с возможностью переключения уровней без рестарта и со сжатием логов.

Существуют следующие уровни логов: TRACE, ALERT, CRITICAL, ERROR, WARNING, NOTICE, DEBUG, HIGH_DEBUG. При выборе уровня существует возможность выбора одного, нескольких одновременно или всех уровней сразу. Например, DEBUG,ERROR,WARNING – выдает логи отладки, ошибок и предупреждений. Если ни один уровень не задан, значит все события игнорируются.

Созданы следующие категории (группы логирования) ПК invGuard CS-SW:

1) COMMON
2) ACTION_SCHEDULER
3) ARP_ANNOUNCER
4) CONTROL_MESSAGE_PROCESSOR
5) TRAFFIC_CAPTURER
6) TRAFFIC_DUMPER
7) MAC_ADDRESS_CHECKER
8) XML_RPC_RESPONDER
9) CONFIG
10) CONFIG_PARSER
11) MAIN
12) CONTROL_MITIGATION
13) UPDATER_MITIGATION
14) CONTROL_PACKET_DIGESTER
15) PACKET_INPUT
16) PACKET_OUTPUT
17) PACKET_GENERATOR
18) BITS_ANALYZER
19) DATA_LIST
20) NEW_TOP_MAKER
21) SORT_MAKER
22) SEARCHING_TREE
23) TCP_SESSION
24) PACKET_SIP_TOP
25) PACKET_CHECK
26) PACKET_PREPROCESSOR
27) PACKET_FINGERPRINT
28) PACKET_REGEXP_WRAPPER
29) PACKET_MITIGATION_FILTER_DESTINATION
30) PACKET_MITIGATION_FILTER_RULES
31) PACKET_MITIGATION_FILETR_SET

32) PACKET_FILTER_GLOBAL
33) PACKET_ANALIZ_PROT_HTTP
34) PACKET_ANALIZ_PROT_DNS
35) PACKET_ANALIZ_DNS_RFC
36) PACKET_DNS_TOP
37) PACKET_MITIGATION_FILTER_BLACKLIST
38) PACKET_MITIGATION_PAYLOAD
39) PACKET_MITIGATION_HTTP_RFC_REGEXP
40) PACKET_MITIGATION_HTTP_HDR_PAYLOAD
41) PACKET_MITIGATION_TREND_24
42) PACKET_MITIGATION_BASELINE_PROTO
43) PACKET_MITIGATION_COUNTERMEASURES
44) PACKET_MITIGATION_SYN_AUTH
45) PACKET_MITIGATION_SYN_AUTH_ALT
46) PACKET_MITIGATION_ZOMBIE
47) PACKET_MITIGATION_TCP_SESSIONS
48) PACKET_MITIGATION_HTTP_PARSER
49) PACKET_MITIGATION_HTTP_RFC
50) PACKET_MITIGATION_HTTP_REQUEST_LIMIT
51) PACKET_MITIGATION_DNS_RFC
52) PACKET_MITIGATION_DNS_AUTH
53) PACKET_MITIGATION_SIP_RFC
54) PACKET_MITIGATION_SIP_LIMIT
55) PACKET_MITIGATION_MS_STATISTIC
56) PACKET_MITIGATION_RAW_STATISTIC
57) PACKET_TC_STATISTIC
58) PACKET_MITIGATION_SHAPER
59) DPDK_ENGINE
60) ALL

Группы и уровни логирования задаются через запятую в файле /syn/conf/config.txt

```
LogLevel=DEBUG

LogCatigories=ALL
```

Допускается перечисление нескольких категорий и уровней через запятую. Пример перечисления категорий и уровней логирования:

```
LogLevel=ERROR,WARNING,DEBUG

LogCatigories= DPDK_ENGINE, COMMON, MAIN
```

Для смены уровня логирования без перезапуска программного комплекса предусмотрена команда переконфигурации invGuard CS-SW:

```
/usr/bin/syn/synctl reconfigure
```

Общая активация и отключения модуля логирования внутри программного комплеса invGuard CS-SW производится при помощи редактирования параметра “debug” файла /syn/conf/config.xml

Пример строки параметров из файла config.xml:

```
<tcparams debug="false" drop_fragmented="false" resume_mitigs_on_error="false">
```

Где:

- параметр debug=”false” отключает систему логирования;
- параметр debug=”true” включает систему логирования.

При активированной системе логирования все логи сбрасываются в каталог /syn/log в виде текстовых файлов. Во время эксплуатации комплекса invGuard CS-SW конечными потребителями не предусматривается возможность включения логирования из-за существенного влияния на производительность.

4. ПРОВЕРКА ПРОГРАММЫ

Полное описание проверки работоспособности Очистителя приведено в разделе «Методы испытаний» документа RU.09445927.425530-06 51 01 «Программа и методика испытаний»

5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

Для выдачи пользователю диагностических сообщений о возникающих ошибках системы используется каталог /syn/syn/alerts/, содержащий файлы с информацией о событиях системы.

Оповещение представляет собой xml-файл с именем *TS.xml*, где *TS* отражает время создания оповещения. Корневой элемент файла называется *alert* и содержит атрибуты *type*, *severity*, *name*, и др. Тривиальным будем называть оповещение,

которое описывается атрибутами *type*, *severity*, *name*, *ts* и, при необходимости, параметром *description*.

Типы оповещений задаются атрибутом *name* элемента *alert* и описаны в таблице 5.

Таблица 5 – Атрибуты событий системы

Name	**Type**	**Severity**	**Описание**
output_fault	error	hi	Модуль вывода не успевает обрабатывать пакеты. Тривиальное оповещение.
mitig_run_failed	error	hi	Не удалось запустить задание очистки трафика. Параметр “description” содержит описание причины, по которой не удалось начать процесс очистки трафика. Тривиальное сообщение.
mitig_stop_failed	error	hi	Не удалось остановить задание очистки. Параметр «description» содержит описание причины, по которой не удалось остановить задание очистки. Тривиальное сообщение.
config_error	error	hi	Ошибка конфигурации системы. Параметр «description» содержит описание причины, по которой не удалось применить новые параметры системы. Тривиальное сообщение.
start_failed	error	hi	Ошибка запуска системы. Параметр «description» содержит описание причины, по которой не удался запуск системы. Тривиальное сообщение.
module_error	error	hi	Ошибка в модуле системы. Параметр «description» содержит описание ошибки в модуле. Тривиальное сообщение.
restart_module_failed	error	hi	Не удалось перезапустить модуль системы. Параметр «description» содержит описание причины, по которой не удался перезапуск системы. Тривиальное сообщение.
dns_baseline	warning	hi	Частота DNS-запросов отличается от тренда.

ПРИЛОЖЕНИЕ 1

ПЕРЕЧЕНЬ ТЕРМИНОВ

В настоящем документе применяют следующие термины с соответствующими определениями.

Автономная система	Система IP-сетей и маршрутизаторов, управляемая одним или несколькими операторами и имеющая единую политику маршрутизации с Интернетом.
Наблюдаемый объект	Совокупность объектов сети, потоков трафика и сетевых сервисов, рассматриваемая анализатором трафика как единое целое в контексте задач мониторинга обнаружения сетевых угроз.
Очистка трафика	Совокупность механизмов и алгоритмов фильтрации трафика с целью отбрасывания пакетов, классифицированных как аномальные.
Сигнатура трафика / угрозы	Описание существенных характеристик трафика (произвольного или аномального) в виде выражения на специальном языке.
Зомби	дочерний процесс в Unix-системе, завершивший своё выполнение, но ещё присутствующий в списке процессов операционной системы, чтобы дать родительскому процессу считать код завершения.
Сетевые сервисы	Приложение или функциональность, поддерживаемая и обеспечиваемая инфраструктурными элементами СПД.
IP-адрес	Уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP.

ПРИЛОЖЕНИЕ 2

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

AS	Autonomous system (автономная система).
BIOS	Basic input/output system (базовая система ввода-вывода). Предназначается для предоставления операционной системе API-доступа к аппаратуре компьютера и подключенным к нему устройствам.
SSH	Secure Shell — «безопасная оболочка». Сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений.
TCP	Transmission Control Protocol – протокол управления передачей.
UDP	User Datagram Protocol — протокол пользовательских датаграмм.
XML	eXtensibe Markup Language, универсальный текстовый формат для хранения и передачи структурированных данных.

Лист регистрации изменений

Номера листов (страниц)					Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного документа и дата	Подп.	Дата
Изм	измененных	замененных	новых	аннулированных					