Microolap EtherSensor

Руководство Администратора

2020

Содержание

1. Microolap EtherSensor

Аннотация

Microolap EtherSensor, v. 6.1.

Программное обеспечение Microolap EtherSensor зарегистрировано в "Едином реестре российских программ для электронных вычислительных машин и баз данных" Минкомсвязи Российской Федерации за № 5034.

Руководство Администратора.

В документе описано назначение и применение Microolap EtherSensor, сформулированы его общие характеристики как объекта администрирования, определены основные технические операции и мероприятия по подготовке объектов администрирования ко вводу в действие, определен порядок обновления ПО, определен регламент работы объекта администрирования в аварийных ситуациях.

О Microolap EtherSensor

EtherSensor – инфраструктурная платформа для автоматизации анализа сетевого трафика (Network Traffic Analysis, NTA). EtherSensor извлекает из трафика и анализирует объекты от канального уровня до уровня приложения включительно: пакеты, сессии, файлы, сообщения, события и их метаданные. По завершении анализа EtherSensor доставляет результаты одной или нескольким системам-потребителям.

EtherSensor применяется как поставщик данных и метаданных извлечённых из трафика объектов для NDR, DLP, eDiscovery, Enterprise Archiving систем, а также для различных подсистем SOC (SIEM-системы, U(E)BA, высоконагруженные DLP-системы, системы Network Detection and Response, Threat Intelligence/Management, Asset Management, Application Management и пр.).

Отличительными особенностями Microolap EtherSensor являются:

- Высокая производительность обработки сетевого трафика для данного класса решений

(20Gbps "из коробки" на серийном оборудовании, масштабируется до 50Gbps)

- Отсутствие ограничений на количество поддерживаемых Интернет/Интранет-сервисов благодаря открытости системы детектирования и захвата объектов сетевого трафика.

Microolap EtherSensor поставляется с 2008 г. Применяется в основном в системах информационной безопасности.

Как работает EtherSensor:

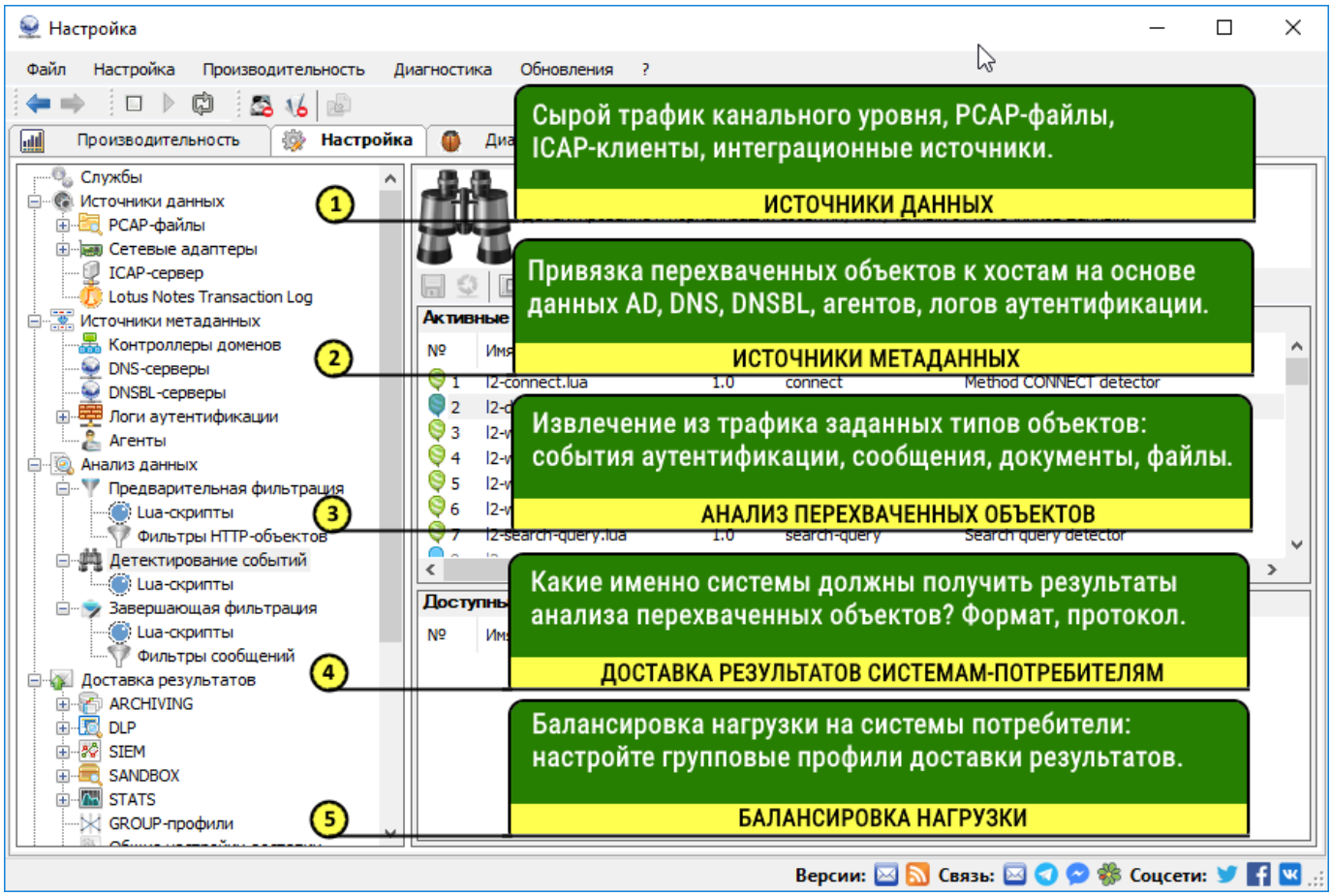

Рис.1. Схема работы EtherSensor.

Сопровождение рабочего экземпляра EtherSensor

Для сопровождения и поддержки рабочего экземпляра EtherSensor используйте EtherSensor PCAP Edition [12].

EtherSensor PCAP Edition не требует трудоёмких подготовительных мероприятий по развёртыванию в сетевой инфраструктуре организации и имеет полностью идентичную с полной версией EtherSensor функциональность в части анализа сетевого трафика и доставки результатов

системам-потребителям.

С этой редакцией вы сможете отлаживать и тестировать фильтры, детекторы и профили доставки на заранее заготовленных PCAP-файлах. После отладки и тестирования вы сможете перенести их на рабочий экземпляр EtherSensor.

Если в Сети не удаётся найти PCAP-файл для нужного вам случая (Google: "pcap files collection"), воспользуйтесь TCPDUMP или WireShark для записи своего собственного трафика. EtherSensor PCAP Edition поддерживает форматы pcap и pcapng.

> Совет: для записи PCAP-файлов в среде Windows используйте TCPDUMP for Windows.

1.1. Краткое описание возможностей

Microolap EtherSensor – высокопроизводительная платформа извлечения событий и сообщений из сетевого трафика в реальном времени со следующими свойствами:

- Извлечение из сетевого трафика событий и сообщений на уровнях от L2 до L7 согласно модели OSI
- Значительное количество (несколько тысяч) распознаваемых программным обеспечением EtherSensor Интернет-сервисов
- Высокая производительность: 20Gbps "из коробки" на серийном оборудовании, масштабируется до 50Gbps
- Доставка событий, сообщений и метаданных в любые системы-потребители (архивы, подсистемы SOC,DLP, SIEM, eDiscovery и т.п.)
- Высокие показатели непрерывной работы без обслуживания
- Работа на серийном оборудовании с низким потреблением ресурсов.

EtherSensor состоит из нескольких служб ОС Windows, в совокупности решающих задачу перехвата и анализа сообщений уровня приложения и их метаданных (как правило, это – сообщения пользователей сети). Затем сообщения, их метаданные или извлеченные из сообщений данные EtherSensor доставляет системам-потребителям [6].

Отличительной особенностью и главным принципом работы EtherSensor является его неучастие в процессе передачи сетевого трафика контролируемой сети и, как следствие этого – независимость надежности сети от его работы. При этом EtherSensor гарантированно обеспечивает полный контроль трафика в сетях с нагрузкой 20Gbps и выше, детектируя сообщения нескольких тысяч Интернет-сервисов.

Методы фильтрации сетевого трафика на всех уровнях, – как на уровне IP пакетов, так и на уровне уже реконструированных объектов уровня приложения, – позволяют добиться минимального расхода ресурсов для контроля сетевых коммуникаций.

Расширяемость EtherSensor позволяет как принимать данные от внешних источников (SPAN/TAP трафик, ICAP-клиенты, лог транзакций Lotus Notes, PCAP-файлы), так и поставлять реконструированные сообщения любому количеству внешних систем-потребителей.

Ниже приведено описание функций EtherSensor, сгруппированных по функциональным модулям.

Веб-почта (WM)

Извлечение из трафика исходящих сообщений сервисов веб почты: Mail.RU, Yandex.RU, Pochta.RU, GMail и т.п. (более 40 доменов), а также все сервисы на базе движка "SquirrelMail". Для получения сообщений, отправляемых по зашифрованному каналу, следует использовать SSLSplitter, либо сторонние решения для расшифровки SSL, либо EtherSensor ICAP.

Социальные сети (SN)

Извлечение из трафика различных типов пользовательских данных (аутентификационные данные, текстовые сообщения, комментарии и т.п.) следующих веб-сервисов:

- Социальных сетей Vk.com, Facebook, LinkedIn, Mamba.ru и др.
- Форумов, функционирующих на базе движков phpbb, ipb, vbulletin, mybb
- Сервисов отправки SMS/MMS сообщений.

Для извлечения сообщений, отправляемых по зашифрованному каналу, используйте SSLSplitter, либо сторонние решения для расшифровки SSL, либо EtherSensor ICAP.

Email (EM)

Извлечение из трафика сообщений электронной почты, передаваемых по протоколам SMTP, POP3, IMAP и Lotus Notes.

Для извлечения сообщений, отправляемых по зашифрованному каналу, следует использовать SSLSplitter, либо сторонние решения для расшифровки SSL, либо EtherSensor ICAP.

ICAP-сервер (IS)

Позволяет использовать в качестве источника трафика для извлечения сообщений HTTP- и FTP-трафик, передаваемый в EtherSensor по протоколу ICAP внешними системами, такими как: SQUID, Blue Coat Proxy SG, Cisco WSA, Webwasher, Websense, McAfee Web Gateway, FortiGate, Entensys UserGate и т.п.

Мгновенные сообщения (IM)

Извлечение из трафика сообщений, отправляемых и получаемых через сервисы мгновенных сообщений, работающие по протоколам Skype, XMPP/Jabber, IRC, MSN, YAHOO и OSCAR.

Для извлечения сообщений, отправляемых по зашифрованному каналу, следует использовать SSLSplitter, либо сторонние решения для расшифровки SSL, либо EtherSensor ICAP.

Lotus Notes (LN)

Извлечение из трафика событий системы Lotus Notes, таких как сообщения, события календаря и другие. Если трафик Lotus Notes зашифрован, EtherSensor извлекает сообщения из Lotus Notes Transaction Log. Оба метода никак не влияют на работу Lotus Notes.

Передача файлов (FT)

Извлечение из трафика файлов, передаваемых по протоколам SMB, HTTP, FTP и WebDAV.

Для извлечения сообщений, отправляемых по зашифрованному каналу, следует использовать SSLSplitter, либо сторонние решения для расшифровки SSL, либо EtherSensor ICAP.

Поиск работы (CV)

Извлечение из трафика событий (регистрация, аутентификация, отклик на вакансии, актуализация резюме), размещаемых на сервисах вакансий и поиска работы, таких как HH.ru, SuperJob.ru, Job.ru и т.п. (более 150 доменов).

Для извлечения сообщений, отправляемых по зашифрованному каналу, следует использовать SSLSplitter, либо сторонние решения для расшифровки SSL, либо EtherSensor ICAP.

EtherSensor Agent (AG)

EtherSensor Agent устанавливается на рабочих станциях в случае, если установка полноценного end-point DLP решения по каким-либо причинам невозможна или нежелательна.

EtherSensor Agent предназначен для привязки сетевых соединений, создаваемых локальными процессами на рабочих станциях, к именам пользователей и именам рабочих станций при использовании в сети организации NAT, терминальных серверов и т.п.

Кроме того, EtherSensor Agent решает задачи отслеживания изменений (оборудование, процессы и т.п.) на рабочей станции и передачи данных о таких событиях на серверы EtherSensor и EtherStat.

1.2. Область применения EtherSensor

EtherSensor используется для анализа трафика от уровня L2 до уровня L7 согласно модели OSI. Результатом анализа являются контент и метаданные сообщений и соответствующих им событий.

При разработке EtherSensor были поставлены требования:

Адаптивность к системе-потребителю:

EtherSensor должен уметь передавать любой системе-потребителю как данные, так и метаданные перехваченного объекта в любом требуемом формате и любым транспортным протоколом. Тип и назначение потребителя могут быть любыми: SIEM, DLP, eDiscovery, UEBA, Enterprise Search, файловая система, облако, СУБД и т.п.

Одновременная работа с многими источниками и потребителями:

Количество систем-потребителей, с которыми одновременно работает EtherSensor, не должно быть ограничено.

Количество источников данных, с которыми одновременно работает EtherSensor, также не должно быть ограничено.

Полнота контроля:

EtherSensor должен обнаруживать любой объект уровня приложения, переданный по сети вне зависимости от его типа. Специализированная система-потребитель должна получить результат обработки такого объекта.

Реальное время:

EtherSensor должен работать в реальном времени на самых скоростных каналах передачи данных, доступных организациям. В текущей версии EtherSensor уверенно обрабатывает потоки данных 20Gbps "из коробки" на серийном оборудовании, масштабируется до 50Gbps.

Работа "из коробки":

EtherSensor должен развертываться "из коробки" и не должен требовать постоянного внимания ни разработчика, ни Заказчика. В идеале – полностью необслуживаемый элемент информационной инфраструктуры.

Наиболее часто Microolap EtherSensor находит применение при решении следующих групп задач:

Архивирование сообщений (eDiscovery и Compliance Archiving):

Извлечение из трафика документов и других объектов электронной почты, мессенджеров, социальных сетей, блогов, форумов и других средств коммуникации.

Примеры систем-потребителей:

- Bloomberg Vault
- Global Relay
- Google Vault (help)
- IBM Content Collector
- Mailarchiva
- Smarsh
- Somansa Mail-i
- Veritas eDiscovery Platform (ранее Clearwell)
- Veritas Enterprise Vault.

SIEM-системы, U(E)BA, анализаторы логов:

Извлечение из трафика метаданных перехваченных объектов уровня приложения и связанных с ними событий (в том числе и в реальном времени из контента объекта) для регистрации в SIEM-системах.

Примеры систем-потребителей:

- AlienVault
- ArcSight Enterprise Security Manager (ESM) (ранее HP ArcSight)
- Elastic Stack (ранее ELK Stack: Elasticsearch, Logstash, Kibana)
- EventTracker
- FortiSIEM (ранее AccelOps)
- Graylog
- IBM QRadar
- LogRhythm
- ManageEngine EventLog Analyzer
- McAfee Enterprise Security Manager (ESM)
- Micro Focus Sentinel (ранее NetIQ)
- RuSIEM
- SolarWinds Log & Event Management (LEM)
- Splunk
- SQLstream

- Trustwave SIEM
- Любое программное обеспечение, принимающее данные по SYSLOG, NETFLOW или другому TCP/UDP-протоколу.

Предотвращение утечек конфиденциальных данных (DLP-системы)

Извлечение контента сообщений из трафика уровня приложения (L7 согласно модели OSI) внешних и внутренних коммуникационных сервисов с последующей отправкой их системе-потребителю: DLP-системе, системе архивирования почтовых сообщений, системе документооборота, локальной поисковой системе и любой другой системе, имеющей функцию архивирования документов.

Примеры систем-потребителей:

- DeviceLock DLP
- Forcepoint DLP
- Proofpoint Email DLP
- Symantec DLP
- McAfee DLP
- Trustwave DLP
- Falcongaze SecureTower
- InfoWatch Traffic Monitor
- Любое другое DLP-решение.

1.3. Системные требования

Минимальные системные требования к серверу для функционирования Microolap EtherSensor:

Data flow	50 Mbps	150 Mbps	250 Mbps	500 Mbps	750 Mbps	1.5 Gbps	2.5 Gbps	5 Gbps	10 Gbps	15 Gbps	20 Gbps
Users	100	300	500	1000	1500	3000	5000	10000	20000	30000	40000
CPU	1CPU*2 Cores	1CPU*4 Cores	1CPU*4 Cores	1CPU*4 Cores	1CPU*4 Cores	1CPU*6 Cores	1CPU*8 Cores	2CPU*6 Cores	2CPU*10 Cores	4CPU*10 Cores	4CPU*12 Cores
RAM	1 GB	4 GB	4 GB	8 GB	8 GB	32 GB	64 GB	128 GB	128 GB	128 GB	256 GB
HDD	75 GB SATA	2x75 GB SATA RAID1	2x75 GB SATA RAID1	2x150 GB SATA RAID1	2x150 GB SATA RAID1	2x146 GB SAS RAID1	4x300 GB SSD RAID10	4x500 GB SSD RAID10	6x500 GB SSD RAID10	8x500 GB SSD RAID10	8x900 GB SSD RAID10
NIC	1 x 1Gbps	1 x 1Gbps	2 x 1Gbps	4 x 1Gbps	4 x 1Gpbs	1 x 10Gbps	2 x 10Gbps	4 x 10Gbps	4 x 10Gbps	6 x 10Gbps	6 x 10Gbps

> Для исключения проблем с захватом трафика используйте сетевые адаптеры, не имеющие проблем с поддержкой RSS (Receive Side Scaling) в драйверах.
>
> Гарантированно подходят сетевые карты на чипсетах Intel серий X520, X710, XL710 и XXV710.

Дисковая подсистема используется только в частных случаях для кэширования перехваченных сетевых объектов, а также для хранения результатов анализа в случае временной недоступности системы-потребителя. Поэтому объем дисковой подсистемы следует рассчитать самостоятельно, исходя из возможного времени простоя системы-потребителя.

EtherSensor работает в среде операционной системы Windows Server 2012, Windows Server 2016 x64 или Windows Server 2019 x64, файловая система – только NTFS. Для максимального быстродействия рекомендуем использовать Windows Server 2016 или Windows Server 2019 на платформе x64.

> Вы можете установить EtherSensor и на десктопные x64 версии Windows: Windows 8, Windows 8.1 и Windows 10. Однако, для EtherSensor Full Edition следует учитывать возможное снижение производительности, обусловленное ограничениями архитектуры таких ОС.

Для корректной работы EtherSensor необходимо наличие последних обновлений ОС Windows. При отключенной службе обновления полноценная работа EtherSensor не гарантируется.

1.4. Описание функционирования

Microolap EtherSensor состоит из набора служб, работающих на отдельном аппаратном или виртуальном сервере ("сенсоре").

Сенсор подключен к Ethernet-порту активного сетевого устройства, на котором настроено дублирование сетевого трафика (mirroring, RX и TX пакеты) с заданных портов, либо к сетевому ответвителю (network tap). Для захвата трафика с нулевыми потерями пакетов при нагрузке в 50Gbps и выше используется технология захвата трафика, разработанная в Общество с ограниченной ответственностью Микроолап Текнолоджис.

Сенсор – это сервер с несколькими сетевыми интерфейсами, один из которых является административным, а другие используются для приема копии сетевого трафика. Сетевой стек ОС на сенсоре сконфигурирован только на административном сетевом интерфейсе, который служит также для передачи перехваченных сообщений внешним системам-потребителям по протоколам, установленным в профилях доставки результатов.

Политика сенсора хранится в его конфигурационных файлах и является набором правил, определяющих захват IP пакетов, анализ контента перехваченных сообщений уровня

приложения, и, в зависимости от результата, формат, способ доставки и направление транспортировки результата перехвата.

На сенсоре работают следующие службы:

Службы, работающие с источниками данных.

Служба извлечения сообщений уровня приложения из PCAP-файлов (EtherSensor PCAP, sensor_pcap.exe)

Предназначена для анализа PCAP-файлов в форматах tcpdump/libpcap/pcapng. Служба EtherSensor EtherCAP извлекает из обрабатываемых файлов объекты уровня приложения (L7 согласно модели OSI), и передаёт эти объекты для дальнейшей обработки в службу EtherSensor Analyser.

Служба извлечения сообщений уровня приложения из Ethernet-трафика (EtherSensor EtherCAP, sensor_ethercap.exe)

Предназначена для пассивного перехвата трафика на одном или нескольких Ethernet-адаптерах. Служба EtherSensor EtherCAP извлекает из обрабатываемого трафика сообщения, события и другие объекты уровня приложения, и затем передаёт эти объекты для дальнейшей обработки в службу анализа сообщений EtherSensor Analyser.

Служба извлечения сообщений уровня приложения из данных, предоставляемых ICAP клиентами (EtherSensor ICAP, sensor_icap.exe)

Предназначена для получения трафика по протоколу ICAP в режиме REQMOD+RESPMOD от любых ICAP-клиентов (Squid, Blue Coat Proxy SG, Cisco WSA и т.п.). Полученные от ICAP-клиентов объекты EtherSensor ICAP передаёт службе EtherSensor Analyser для дальнейшего анализа.

Служба извлечения сообщений из Lotus Notes Transaction Log (EtherSensor LotusTXN, sensor_lotustxn.exe)

Предназначена для мониторинга и реконструкции сообщений системы Lotus Notes методом извлечения их из Lotus Notes Transaction Log (журнала транзакций Lotus Notes). EtherSensor LotusTXN извлекает сообщения из файлов Lotus Notes Transaction Log, и затем передаёт эти сообщения для дальнейшей обработки в службу EtherSensor Analyser.

Служба накопления метаданных для обогащения сетевых событий (EtherSensor Identity, sensor_identity.exe)

Предназначена для накопления и обработки метаданных, которые используются в EtherSensor для решения задачи безагентной привязки перехваченного объекта сетевого трафика к конкретному хосту или пользователю.

Служба анализа сообщений, извлеченных из Ethernet-трафика (EtherSensor Analyser, sensor_analyser.exe)

Предназначена для детектирования, анализа и фильтрации перехваченных сообщений и сетевых событий. EtherSensor Analyser анализирует объекты протоколов уровня приложения, полученные от служб EtherSensor PCAP, EtherSensor EtherCAP, EtherSensor ICAP и EtherSensor LotusTXN с целью детектирования сетевых событий и сообщений, передаваемых по сети.

Затем на основе настраиваемых пользователем правил фильтрующий механизм службы EtherSensor Analyser принимает решение:

1. О прекращении обработки сообщения

2. О доставке сообщения системе-потребителю (DLP, U(E)BA, архив, eDiscovery-система, Enterprise Search и т.п.)

3. О формировании строки с заранее заданной структурой на основе данных, извлечённых из сообщения и его метаданных. Для SIEM-систем это, как правило, syslog-строка в формате CEF.

Также служба EtherSensor Analyser отвечает за взаимодействие с экземплярами EtherSensor Agent, маркирующими сессии для привязки к конкретной рабочей станции в случае использования NAT, терминальных сервисов и т.п.

Служба доставки результатов анализа трафика (EtherSensor Transfer, sensor_transfer.exe)

EtherSensor Transfer является подсистемой EtherSensor, предназначенной для доставки результатов анализа перехваченных объектов (само сообщение, либо заранее сконфигурированная syslog-строка) различным системам-потребителям по различным протоколам в соответствии с заранее определенными профилями доставки результатов.

Служба журналирования EtherSensor (EtherSensor Watcher, sensor_watcher.exe)

EtherSensor Watcher является подсистемой EtherSensor, предназначенной для логирования текущего состояния и событий самого EtherSensor.

Служба обновления EtherSensor (EtherSensor Updater, sensor_updater.exe)

Предназначена для загрузки и установки файлов обновлений и лицензий Microolap EtherSensor при выходе новых версий и/или исправлений.

Для просмотра статистики работы сенсора используйте Консоль управления EtherSensor (sensor_console.exe) из директории установки EtherSensor.

Для управления политикой сенсора также используйте Консоль управления EtherSensor, либо редактируйте конфигурационные файлы служб сенсора в директории [INSTALLDIR]\config.

1.5. Уровень подготовки администратора

Системный администратор Microolap EtherSensor должен:

- Иметь базовые знания о стеке протоколов TCP/IP, в частности по прикладным протоколам SMTP, HTTP
- Иметь базовые знания по администрированию сетевых интерфейсов и системных служб ОС семейства Windows Server
- Иметь базовые знания по использованию tcpdump и WireShark
- Обладать знаниями и навыками по администрированию служб Microolap EtherSensor и уметь с их помощью реализовывать корпоративную политику безопасности в части, относящейся к использованию внешних коммуникационных сервисов.

1.6. Перечень эксплуатационных документов

Для эксплуатации и администрирования Microolap EtherSensor, а также для понимания основных операций по вводу служб EtherSensor в эксплуатацию, администратору необходимо ознакомиться со следующими документами:

- Настоящее Руководство
- Документация на внешние системы-потребители сообщений и/или связанных с сообщениями событий (в зависимости от целей использования Microolap EtherSensor).

1.7. Об EtherSensor PCAP Edition

EtherSensor PCAP Edition – продукт-компаньон Microolap EtherSensor, предназначенный для:

- Сопровождения полной рабочей версии EtherSensor: разработки и тестирования фильтров, правил и детекторов, а также анализа PCAP и HAR-файлов без рискованных экспериментов на его рабочем экземпляре.
- Изучения функциональных возможностей EtherSensor без трудоёмкого создания для этого специальных условий.

EtherSensor PCAP Edition не содержит кода для обработки в реальном времени сетевого трафика, полученного с сетевых адаптеров, кроме того, производительность этой редакции ограничена 30 Mbps ИЛИ примерно 75 пользователей.

EtherSensor PCAP Edition лицензируется по принципу "один специалист – одна лицензия", количество инсталляций не ограничено.

1.8. Доставка результатов анализа трафика

Сразу после инсталляции методом доставки результатов анализа трафика по умолчанию является профиль FILEDROP[208]. Он сохраняет результаты на локальную файловую систему. Каталог, в который сохраняются результаты, назначается в свойствах профиля (поле ***Путь***).

Вы также можете сразу же настроить и другие транспортные профили, доставляющие события в системы-потребители по вашему желанию в требуемом для них формате.

Никаких ограничений на работу профилей доставки ни в полной версии EtherSensor, ни в PCAP Edition нет. Но мы рекомендуем при самом первом знакомстве с EtherSensor начинать с самого простого – работы с профилем FILEDROP[208].

Подробнее о настройках профилей доставки результатов анализа трафика можно узнать в разделе Доставка результатов[204].

2. Установка и настройка Microolap EtherSensor

Для установки Microolap EtherSensor запустите инсталлятор ethersensor_pcap_setup_x64_v6.1_ru-ru.exe, входящий в дистрибутив. В процессе работы инсталлятора будет установлено программное обеспечение, необходимое для его корректной работы, а именно:

- Microsoft .Net Framework 4.0
- Microsoft Visual C++ 2019 redistributable runtime.

> Для корректной установки EtherSensor необходимы права администратора в том виде, в каком они установлены по умолчанию при инсталляции ОС Windows.
>
> Дополнительные ограничения прав администратора могут привести к некорректной работе.

В процессе инсталляции Microolap EtherSensor устанавливаются следующие службы:

EtherSensor PCAP (процесс sensor_pcap.exe)

Предназначена для анализа PCAP-файлов в форматах tcpdump/libpcap/pcapng.

EtherSensor EtherCAP (процесс sensor_ethercap.exe):

Предназначена для пассивного перехвата трафика на сетевых адаптерах

EtherSensor ICAP (процесс sensor_icap.exe):

Предназначена для получения и обработки сетевого трафика по протоколу ICAP от ICAP-клиентов и извлечения сообщений

EtherSensor LotusTXN (процесс sensor_lotustxn.exe):

Предназначена для получения сообщений Lotus Notes из журнала транзакций (Lotus Notes Transaction Log)

EtherSensor Analyser (процесс sensor_analyser.exe):

Предназначена для детектирования, анализа и фильтрации перехваченных сообщений

EtherSensor Transfer (процесс sensor_transfer.exe):

Предназначена для доставки результатов анализа извлечённых из трафика объектов системам-потребителям

EtherSensor Watcher (процесс sensor_watcher.exe):

Предназначена для логирования EtherSensor. Запускается первой и от нее зависят остальные службы EtherSensor.

Служба обновления EtherSensor (процесс sensor_updater.exe):

Предназначена для автоматического обновления EtherSensor.

Помимо служб устанавливаются приложения:

Консоль управления EtherSensor (процесс sensor_console.exe):

Консоль управления EtherSensor, конфигурирования, наблюдения за производительностью, а также сбора информации о работе EtherSensor для создания диагностических отчётов.

После завершения настройки EtherSensor следует провести настройку установленных служб.

2.1. Состав и содержание дистрибутива ПО

Для подготовки Microolap EtherSensor к работе предварительно необходимо иметь следующие дистрибутивы:

- Дистрибутив Microolap EtherSensor.
- ОС Windows Server 2012, Windows Server 2012 x64, Windows Server 2016 или Windows Server 2019. Рекомендуемая операционная система – Windows Server 2019.

Вы можете установить EtherSensor и на десктопные x64 версии Windows: Windows 8, Windows 8.1 и Windows 10. Однако, для EtherSensor Full Edition следует учитывать возможное снижение производительности, обусловленное ограничениями архитектуры таких ОС.

В состав дистрибутива EtherSensor входят:

- Инсталлятор EtherSensor для платформы Windows x64
- Инсталляторы EtherSensor Updater (EtherSensor Updater) для платформы Windows x64
- Документация.

Инсталлятор EtherSensor содержит в себе все зависимости и инсталляторы необходимых для его работы платформ. Его следует использовать для первоначальной установки продукта на новый сервер. Это гарантирует отсутствие проблем, связанных с зависимостями от другого программного обеспечения или пререквизитов.

Служба обновления не выполняет обновление до актуальной версии, если не установлена первоначальная базовая версия. Её установка требуется в любом случае, только после этого Служба обновления установит актуальную версию.

2.2. Подключение Microolap EtherSensor к Ethernet

Для подключения сенсора к сети организации необходимо выполнить следующие действия:

1. Подключить сетевой интерфейс управления [15]
2. Подключить сетевые интерфейсы прослушивания трафика [16]
3. Настроить коммутаторы [16]
4. Обеспечить совместимость со сторонними средствами защиты информации. [17]

2.2.1. Интерфейс управления

EtherSensor для работы требуется подключение к сети одного сетевого интерфейса управления и не менее одного сетевого интерфейса, на который направлена копия сетевого трафика.

Интерфейс управления представляет собой стандартный сетевой интерфейс, сконфигурированный с адресом TCP/IP, маской подсети и, при необходимости, адресом маршрутизатора по умолчанию.

Для работы EtherSensor желательно, чтобы при его настройке были указаны серверы DNS, на которых работает обратное распознавание адресов (от IP к hostname) хостов локальной сети: это позволит определять имена хостов, которые участвовали в соединении, особенно в случае применения для настройки адресов TCP/IP на хостах локальной сети протокола DHCP.

Данные, получаемые от DNS-сервера, кэшируются, что позволяет сократить дополнительную нагрузку на DNS-серверы. Параметры кэширования настраиваются утилитой Консоль управления EtherSensor (sensor_console.exe).

Скорость подключения интерфейса управления должна быть достаточной, чтобы пропускать через себя весь объем перехваченных сообщений, не приводя к их накоплению на сенсоре.

Допускается использование сетевых интерфейсов 100/1000/10000 Mbit/s.

При конфигурации сетевого интерфейса управления необходимо учесть особые настройки EtherSensor, при которых будет производиться фильтрация распознанных данных, отправляемых через сетевой интерфейс управления. Подробно про данные настройки указано в разделе Служба EtherSensor EtherCAP.

2.2.2. Интерфейс прослушивания

Интерфейс прослушивания сетевого трафика не требует настройки стека протоколов TCP/IP, а также других протоколов. С этого сетевого интерфейса происходит захват пакетов для дальнейшего анализа и реконструкции перехваченных объектов. Трафик на этот интерфейс может подаваться с Mirror-порта коммутатора, с ответвителя (network tap), либо с концентратора (hub).

Для нормальной работы требуется, чтобы трафик на интерфейсе прослушивания содержал все сетевые пакеты: как от сервера (RX), так и от клиента (TX).

> Помните:
>
> Наличие или отсутствие тегов VLAN в трафике для работы EtherSensor значения не имеет, так как они удаляются драйвером сетевого адаптера.

Пропускная способность интерфейса прослушивания должна быть на 20% больше поступающего потока данных. Только в этом случае можно иметь уверенность, что пакеты не будут теряться при захвате на сетевом оборудовании и качество перехвата трафика будет высоким.

Сенсор может работать с несколькими интерфейсами прослушивания одновременно. Максимальное их количество на сервере EtherSensor никак не ограничено, но зависит от производительности сетевых интерфейсов, а также от производительности сервера EtherSensor в целом (количества оперативной памяти, производительности процессоров и т.п.).

2.2.3. Настройка коммутаторов

Ниже приведен пример настроек Mirror-порта (SPAN) для коммутаторов Cisco Catalyst (такие коммутаторы позволяют настроить две активные SPAN-сессии):

```
monitor session 1 source interface gigabitEthernet 1/0/24 both
monitor session 1 destination interface gigabitEthernet 1/0/23
```

Здесь "1/0/24" – порт коммутатора, копию трафика с которого необходимо направлять на анализ в EtherSensor, а "1/0/23" – порт коммутатора, к которому подключен сенсор.

Ниже приведена типовая схема включения сенсора в локальной сети:

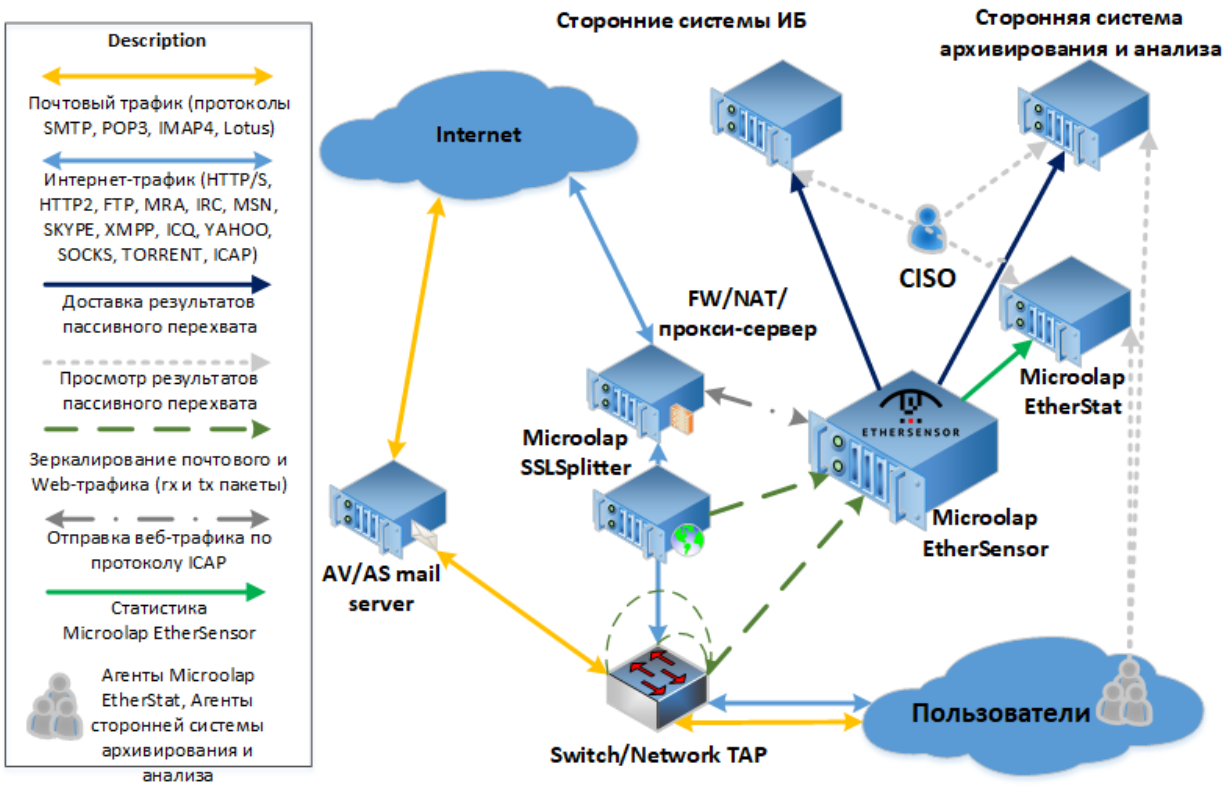

Рис.2. Типовая схема включения Microolap EtherSensor в локальной сети.

2.3. Работа со сторонними СЗИ

Для обеспечения стабильной работы Microolap EtherSensor следует учитывать совместимость со сторонними системами/средствами защиты информации и другими компонентами информационной инфраструктуры организации:

- В директории, где установлен EtherSensor, расположены рабочие директории. Следует исключать данный путь и все его поддиректории из контроля средств, аналогичных антивирусам, поисковым индексаторам, а также средствам контроля целостности. Никакое программное обеспечение не должно блокировать файлы в данных директориях от создания, удаления, перемещения и изменения.
- EtherSensor включает в себя следующие службы Windows: EtherSensor EtherCAP, EtherSensor ICAP, EtherSensor LotusTXN, EtherSensor Analyser, EtherSensor Transfer, EtherSensor Watcher, а также службу обновления EtherSensor Updater. Эти службы должны запускаться с правами локальной системы, так как им требуется доступ к функциям ядра.
- EtherSensor требует для нормальной работы отправку сообщений по протоколу SMTP, FTP, SMB, SYSLOG или IMAP на удаленный сервер. Средства защиты информации не должны проверять, модифицировать, или ограничивать соединения на сервер, куда EtherSensor отправляет данные. Аналогично, средства защиты информации не должны препятствовать

службе EtherSensor Transfer открывать соединения на порты, используемые для отправки сообщений.

- EtherSensor требует для своей работы подключения к системному модулю NDIS. Сторонние средства защиты информации не должны препятствовать службе EtherSensor EtherCAP иметь доступ к функциям этого модуля.
- При установке и функционировании процессы программного обеспечения EtherSensor используют вызовы, требующие высоких привилегий. Политика безопасности ОС должна позволять операции над драйверами, управление процессами и доступ к сетевым интерфейсам для EtherSensor.
- В процессе перехвата EtherSensor способен оперировать большими объемами файлов в рабочих директориях. Файловая система EtherSensor должна располагать достаточным свободным объемом для записи и хранения данных.
- При работе с большими сетевыми потоками рекомендуется монтировать директорию [INSTALLDIR]\data как отдельную партицию на рейд-контроллере с оптимизацией по скорости доступа и записи (raid10).

2.4. Настройки сенсора

Перед началом эксплуатации сенсора следует произвести следующие настройки:

1. **Обеспечить наличие файла лицензии в директории установки Microolap EtherSensor** [279] (для PCAP Edition не нужен)
2. **Настроить источники данных** [18]
3. **Настроить источники метаданных** [54]
4. **Настроить доставку результатов перехвата** [204]
5. **Настроить службу логирования** [249]
6. **Настроить службу анализа сообщений** [89]
7. **Настроить фильтр HTTP-объектов** [166]
8. **Настроить службу обновления** [261].

3. Источники данных и метаданных

В текущей версии Microolap EtherSensor (6.1) на сенсоре функционируют следующие службы, предназначенные для работы с источниками данных: EtherSensor PCAP [21], EtherSensor EtherCAP [30], EtherSensor ICAP [47] и EtherSensor LotusTXN [51].

Кроме того, EtherSensor включает в себя службу EtherSensor Identity[54], которая отвечает за получение метаданных, позволяющих привязать перехваченные объекты к пользователю или хосту.

Общая схема работы служб выглядит следующим образом:

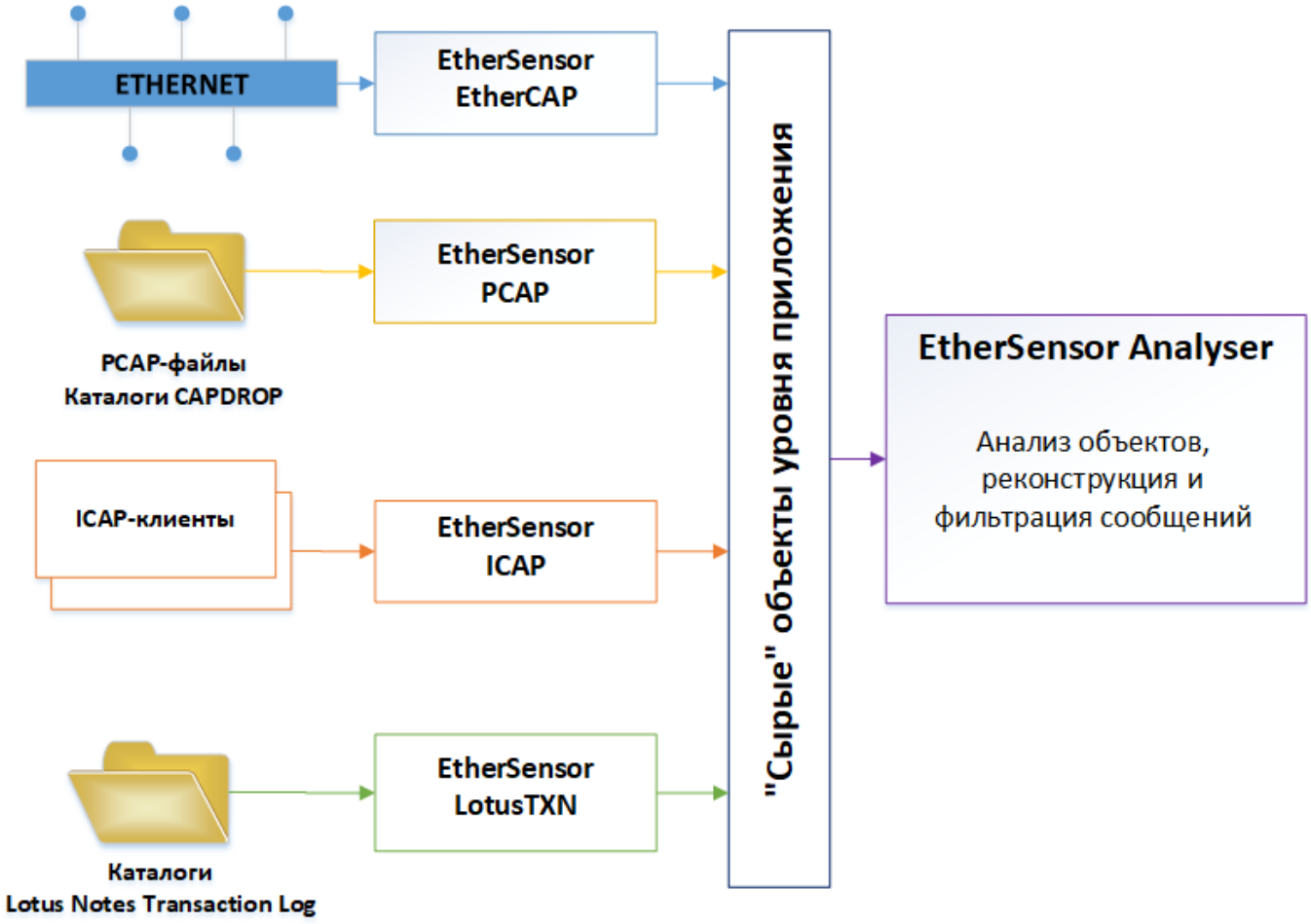

Рис.3. Схема работы служб EtherSensor PCAP, EtherSensor EtherCAP, EtherSensor ICAP, EtherSensor LotusTXN.

Служба EtherSensor PCAP[21] отвечает за обработку трафика из PCAP-файлов в форматах tcpdump/libpcap/pcapng.

Служба EtherSensor EtherCAP[30] отвечает за пассивный захват трафика на сетевых адаптерах и реконструкцию сессий протоколов уровня приложения.

Служба EtherSensor ICAP[47] отвечает за получение трафика по протоколу ICAP от любых ICAP-клиентов и дальнейшую передачу полученных объектов в службу EtherSensor Analyser[89].

Служба EtherSensor LotusTXN [51] отвечает за извлечение сообщений из файлов Lotus Notes Transaction Log, используется только в случае отсутствия доступа к нешифрованному трафику Lotus Notes.

Результатом работы этих служб являются реконструированные объекты, переданные в службу анализа результатов EtherSensor Analyser [89].

Для запуска и остановки служб можно использовать как штатную оснастку Службы ОС Windows, так и Консоль управления EtherSensor (утилита sensor_console.exe) из директории установки EtherSensor:

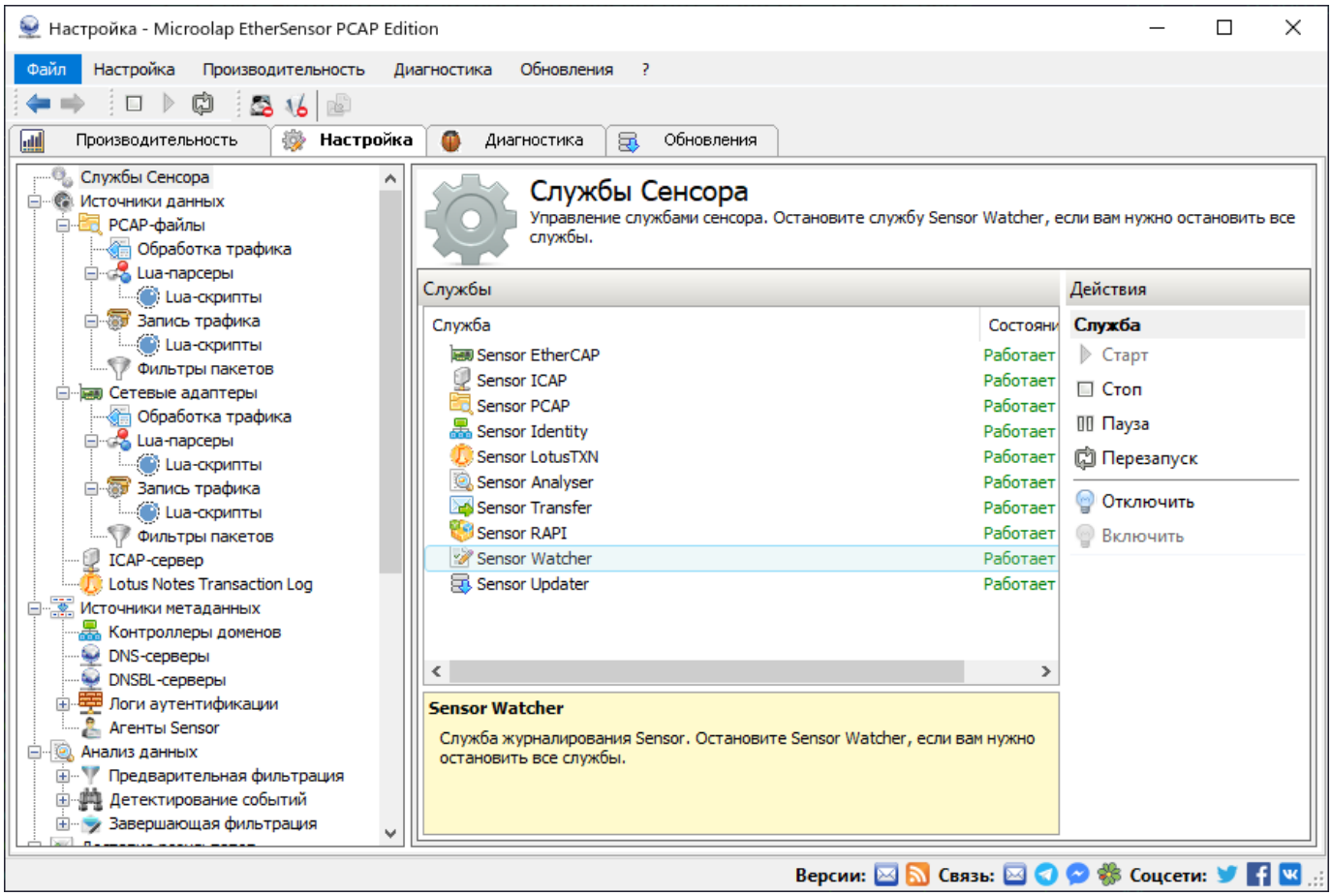

Рис.4. Запуск и остановка служб-источников данных.

Между службой подсистемы логирования EtherSensor Watcher и другими службами EtherSensor установлены зависимости. Поскольку ни одно событие работы EtherSensor не должно остаться незамеченным подсистемой логирования, ни одна служба EtherSensor не может быть запущена до запуска службы логирования EtherSensor Watcher.

И наоборот: чтобы остановить все службы EtherSensor, достаточно остановить только службу логирования EtherSensor Watcher.

3.1. Служба EtherSensor PCAP

Служба EtherSensor PCAP отвечает за обработку трафика из PCAP-файлов в форматах tcpdump/libpcap/pcapng.

EtherSensor PCAP извлекает из обрабатываемых файлов объекты уровня приложения (L7 согласно модели OSI), и передаёт эти объекты для дальнейшей обработки в службу EtherSensor Analyser [89].

EtherSensor PCAP может обрабатывать все те же самые протоколы третьего уровня модели OSI, что и служба EtherSensor EtherCAP.

Взаимодействие EtherSensor PCAP со службой EtherSensor Analyser (а через неё и со службой EtherSensor Transfer) происходит точно так же, как и в службе EtherSensor EtherCAP.

Общая схема работы службы EtherSensor PCAP:

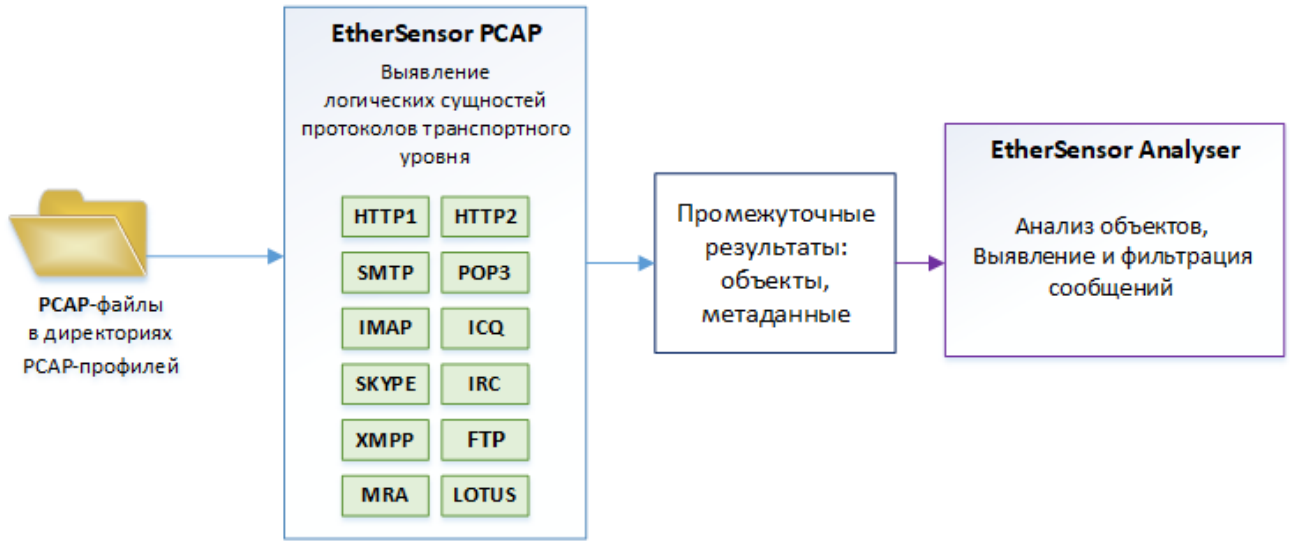

Рис.5. Схема работы службы EtherSensor PCAP

Варианты организации работы EtherSensor PCAP:

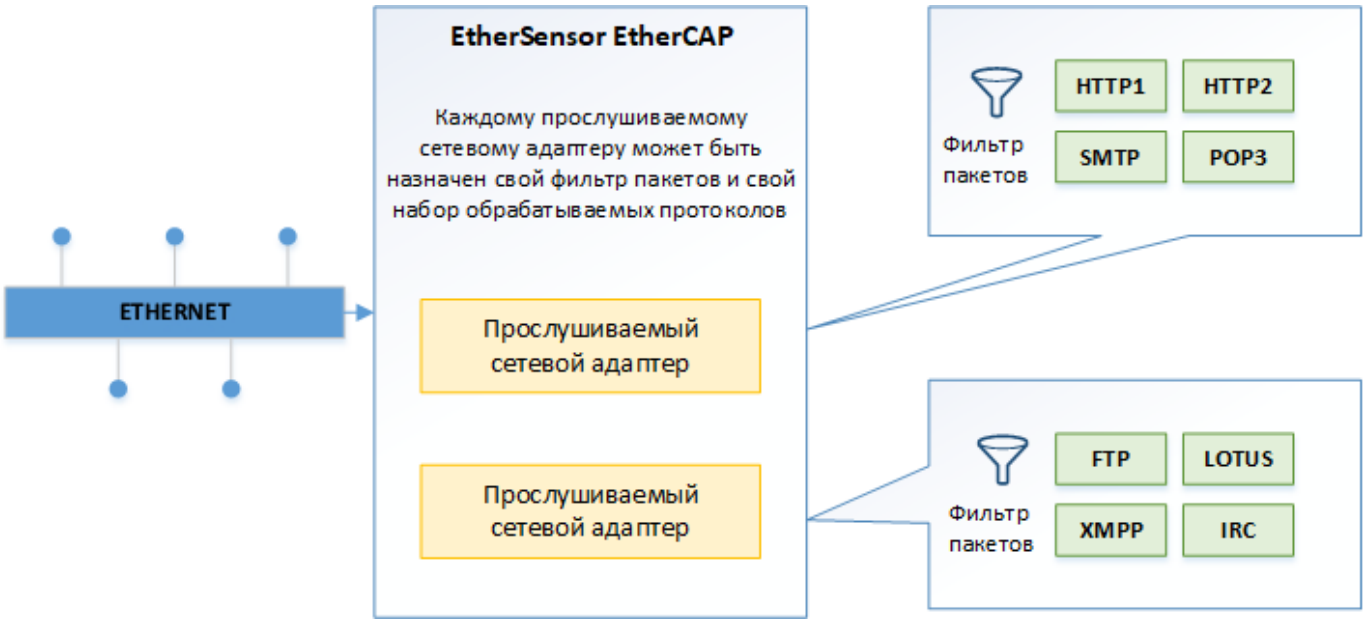

Рис.6. Варианты организации работы службы EtherSensor PCAP.

Служба EtherSensor PCAP позволяет одновременно отслеживать несколько локальных директорий для обработки помещённых в них PCAP-файлов. Также она позволяет назначать для каждой отслеживаемой директории пакетный фильтр и сетевые протоколы, которые необходимо анализировать.

Параметры командной строки

Служба Windows EtherSensor PCAP в ходе инсталляции EtherSensor устанавливается с автоматическим запуском. Однако, при необходимости процесс sensor_pcap.exe можно запустить как приложение Windows со следующими параметрами командной строки:

/process

Запустить процесс sensor_pcap.exe как обычный Windows Win32-процесс (возможно использование для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

3.1.1. Настройка EtherSensor PCAP

Служба EtherSensor PCAP работает со специальным виртуальным устройством, предназначенным для обработки трафика, предварительно сохраненного в PCAP-файлах.

Настройки EtherSensor PCAP немного отличаются от настроек службы EtherSensor EtherCAP[30]:

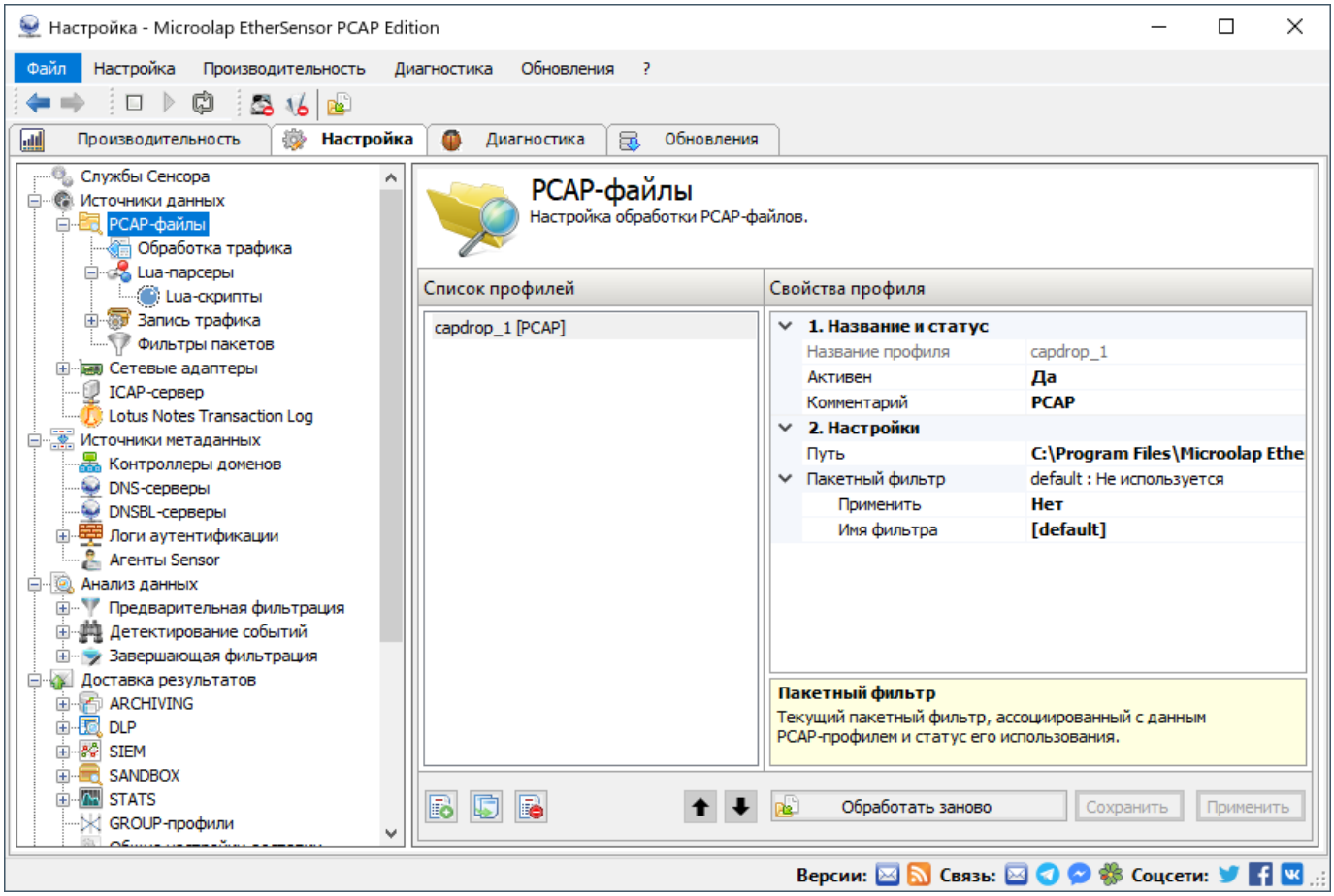

Рис.7. Настройки службы EtherSensor PCAP.

Для обработки PCAP-файлов скопируйте их в директорию PCAP-профиля, и затем разрешите обработку PCAP-файлов в каталоге, указанном в ***Путь***. Для этого установите статус данного PCAP-профиля ***Активен*** в состояние ***Да***.

PCAP-файлы "прослушиваются" в порядке FIFO: файл, помещенный в директорию PCAP-профиля первым (т.е., имеющий более раннюю дату модификации с точки зрения файловой системы), будет "прослушан" первым, помещенный туда вторым будет "прослушан" вторым и т.д.

Если флаг профиля службы ***Активен*** находился в состоянии ***Да***, директория PCAP-профиля была пуста, и в неё был помещен PCAP-файл, то этот файл немедленно начнет "прослушиваться".

Если флаг профиля службы ***Активен*** находился в состоянии ***Нет***, в директории PCAP-профиля имелись PCAP-файлы, то "прослушивание" файлов начнется немедленно после перевода данного профиля службы в активное состояние.

> Все обработанные таким образом PCAP-файлы помещаются в поддиректорию .\processed директории PCAP-профиля без каких-либо изменений. Это позволяет при отладке фильтров и детекторов повторять процесс анализа PCAP-файлов сколько угодно раз, используя кнопку ***Обработать заново***.

Объекты, извлечённые службой EtherSensor PCAP из PCAP-файлов, далее обрабатываются обычным образом: анализируются службой EtherSensor Analyser, которая их передаёт в службу EtherSensor Transfer для доставки результатов анализа системам -потребителям.

> Совет: если вам нужно просто проверить или отладить детекторы или фильтры, наиболее удобным типом системы-потребителя является локальная файловая система (профиль доставки FILEDROP).

Конфигурация службы EtherSensor PCAP содержится в XML-файле pcap.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

3.1.2. Файл конфигурации EtherSensor PCAP

Конфигурация службы EtherSensor PCAP содержится в XML-файле pcap.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

Пример файла конфигурации pcap.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<PcapConfig version="1.0">
 <PcapAdapters>
  <PcapAdapter enabled="true" id="capdrop_1" description="PCAP files processing adapter">
   <Comment>PCAP</Comment>
   <CapDrop>[[INSTALLDIR]]\data\capdrop</CapDrop>
   <Filter enabled="false" name="default" />
   <Protocol enabled="true" name="ftp" />
   <Protocol enabled="true" name="http" />
   <Protocol enabled="true" name="http2" />
   <Protocol enabled="true" name="icap" />
   <Protocol enabled="true" name="imap4" />
   <Protocol enabled="true" name="irc" />
   <Protocol enabled="true" name="lotus" />
   <Protocol enabled="true" name="pop3" />
   <Protocol enabled="true" name="skype" />
   <Protocol enabled="true" name="smb" />
   <Protocol enabled="true" name="smtp" />
   <Protocol enabled="true" name="socks" />
   <Protocol enabled="true" name="websocket" />
   <Protocol enabled="true" name="xmpp" />
  </PcapAdapter>
 </PcapAdapters>

 <Filters>

  <Filter name="default">
   <RuleGroup enabled="true" name="">
    <Rule type="accept" src="any" srcport="any" dst="any" dstport="any" proto="tcp" />
   </RuleGroup>
  </Filter>
  <Filter name="internet">
   <RuleGroup enabled="true" name="">
    <Rule
     type="reject"
     src="192.168.0.1"
     srcport="any"
     dst="any"
     dstport="any"
     proto="tcp" />
    <Rule
     type="reject"
     src="*"
     srcport="any"
     dst="192.168.0.1"
     dstport="any"
     proto="tcp" />
    <Rule
     type="accept"
     src="any"
     srcport="any"
     dst="any"
     dstport="any"
     proto="tcp" />
   </RuleGroup>
  </Filter>

 </Filters>
</PcapConfig>
```

Тег PcapConfig

Корневой тег конфигурации EtherSensor PCAP. Атрибут "version" содержит версию конфигурации. В Microolap EtherSensor версии 6.X она всегда должна быть равна 1.0.

Тег PcapAdapters

Определяет настройки отслеживаемых каталогов с PCAP-файлами.

Тег PcapAdapter

Тег PcapAdapter является вложенным в тег PcapAdapters и содержит настройки PCAP-интерфейса. Атрибут "enabled" содержит статус активности PCAP-интерфейса: если он выставлен в false, то PCAP-интерфейс не будет использоваться. Атрибут "id" используется для указания имени PCAP-интерфейса. Этот атрибут не должен изменяться, и предназначен только для чтения. Атрибут "description" содержит ваше описание PCAP-интерфейса.

Тег Comment

Тег Comment является вложенным в тег PcapAdapter и содержит ваш комментарий к настройке профиля PCAP-интерфейса.

Тег CapDrop

Тег CapDrop является вложенным в тег PcapAdapter и содержит абсолютный путь к отслеживаемому каталогу с PCAP-файлами.

Тег Filter

Тег Filter является вложенным в тег PcapAdapter и содержит описание используемого IP-фильтра для данного PCAP-интерфейса. Атрибут "enabled "содержит статус использования IP-фильтра: если он установлен в false, то для данного PCAP-интерфейса этот фильтр использоваться не будет.

Атрибут "name" содержит имя профиля IP-фильтра. Профили IP-фильтров указываются в теге Filters (см. ниже).

Тег Protocol

Тег Protocol является вложенным в тег PcapAdapter и содерит имя Интернет-протокола. Атрибут "enabled" содержит статус использования Интернет-протокола: если этот атрибут равен "false", то для данного PCAP-интерфейса данный протокол будет игнорироваться. Атрибут "name" содержит имя профиля Интернет-протокола. Этот атрибут используется только для чтения и не подлежит изменению.

Пример настройки PCAP-интерфейса для перехвата сообщений клиентов HTTP, HTTP2, SMTP, WEBSOCKET:

```
<PcapAdapter enabled="true" id="capdrop_1" description="PCAP files processing adapter">
 <Comment>PCAP</Comment>
  <CapDrop>[[INSTALLDIR]]\data\capdrop</CapDrop>
   <Filter enabled="false" name="default" />
    <Protocol enabled="false" name="ftp" />
    <Protocol enabled="true" name="http" />
    <Protocol enabled="true" name="http2" />
    <Protocol enabled="false" name="icap" />
    <Protocol enabled="false" name="imap4" />
    <Protocol enabled="false" name="irc" />
    <Protocol enabled="false" name="lotus" />
    <Protocol enabled="false" name="pop3" />
    <Protocol enabled="false" name="skype" />
    <Protocol enabled="false" name="smb" />
    <Protocol enabled="true" name="smtp" />
    <Protocol enabled="false" name="socks" />
    <Protocol enabled="true" name="websocket" />
    <Protocol enabled="false" name="xmpp" />
</PcapAdapter>
```

Тег Filter

Тег Filter является вложенным в тег Filters и содержит описание настроек IP-фильтра. Атрибут "name" содержит имя профиля IP-фильтра. Значение этого атрибута может быть использовано в качестве значения атрибута "PcapAdapter/Filter/name" для указания IP-фильтра PCAP-интерфейсу.

Тег RuleGroup

Тег RuleGroup является вложенным в тег Filter и служит для группировки правил фильтрации, относящихся к конкретной решаемой задаче фильтрации трафика.

Атрибут "name" содержит ваше описание группы правил фильтрации. Значение этого атрибута может быть пустым.

Тег Rule

Тег Rule является вложенным в тег RuleGroup и содержит ваше описание правила фильтрации PCAP-трафика.

Атрибут "type" содержит тип правила: если он равен accept, то сетевые пакеты, подходящие под данное правило, будут приняты для дальнейшей обработки. Иначе, если он равен reject, то сетевые пакеты, подходящие под данное правило, будут отклонены.

Атрибуты "src" и "dst" содержат IP адрес, диапазон IP адресов, или же параметры сети для фильтрации IP адресов, подходящих под указанное значение.

Пример:

Отклонить пакеты, проходящие между компьютерами 10.1.5.10, 10.1.5.15-10.1.5.59 и сетью 10.1.6.0/255.255.255.0:

```
<Rule
 type="reject"
 src="10.1.5.10, 10.1.5.15-10.1.5.59"
 dst="10.1.6.0/255.255.255.0"
 proto="tcp" />

<Rule
 type="reject"
 src="10.1.6.0/255.255.255.0"
 dst="10.1.5.10, 10.1.5.15-10.1.5.59"
 proto="tcp" />
```

В атрибутах "srcport" и "dstport" укажите необходимые для фильтрации TCP-порты или диапазоны TCP-портов.

Пример:

Отклонить пакеты, проходящие между компьютерами 10.1.5.10, 10.1.5.15-10.1.5.59 и сетью 10.1.6.0/255.255.255.0 на портах 80, 443-1024:

```
<Rule
 type="reject"
 src="10.1.5.10, 10.1.5.15-10.1.5.59"
 srcport="80, 443-1024"
 dst="10.1.6.0/255.255.255.0"
 proto="tcp" />

<Rule
 type="reject"
 src="10.1.6.0/255.255.255.0"
 dst="10.1.5.10, 10.1.5.15-10.1.5.59"
 dstport="80, 443-1024"
 proto="tcp" />
```

Правила применяются линейно сверху вниз. Верхняя строка – это первая инструкция фильтра, нижняя – последняя. Каждая строка отклоняет или принимает только тот тип пакетов, который она описывает.

Пример:

Отклонить пакеты соединения между двумя хостами или группой хостов. При этом должны быть отклонены пакеты, передаваемые в обе стороны соединения:

```
<Rule
 type="reject"
 src="10.31.5.212"
 dst="10.31.5.57"
 dstport="1025"
 proto="tcp" />

<Rule
 type="reject"
 src="10.31.5.57"
 srcport="1025"
 dst="10.31.5.212"
 proto="tcp" />
```

Также необходимо помнить, что если нет правил фильтрации, то будет принят весь трафик. И наоборот, если есть правила фильтрации, то обрабатывается только трафик, удовлетворяющий правилам фильтрации.

Пример:

Получить трафик всех соединений с единственным хостом 10.31.5.57:

```
<Rule
 type="accept"
 src="10.31.5.57"
 srcport="*"
 dst="*"
 dstport="*"
 proto="tcp" />

<Rule
 type="accept"
 src="*"
 srcport="*"
 dst="10.31.5.57"
 dstport="*"
 proto="tcp" />
```

Пример:

Отсечь группу хостов. Для этого необходимо сначала отсечь эту группу, а потом обязательно принять все остальные пакеты, иначе мы ничего не получим вообще:

```
<Rule
 type="reject"
 src="10.31.5.212"
 dst="10.31.5.57"
 dstport="1025"
 proto="tcp" />

<Rule
 type="reject"
 src="10.31.5.57"
 srcport="1025"
 dst="10.31.5.212"
 proto="tcp" />

<Rule
 type="accept"
 src="*"
 srcport="*"
 dst="*"
 dstport="*"
 proto="tcp" />
```

Пример:

Получить трафик только двух определённых хостов, остальные игнорировать:

```
<Rule
 type="accept"
 src="10.31.5.212"
 dst="10.31.5.57"
 dstport="1025"
 proto="tcp" />

<Rule
 type="accept"
 src="10.31.5.57"
 srcport="1025"
 dst="10.31.5.212"
 proto="tcp" />
```

3.1.3. Пакетные фильтры EtherSensor PCAP

ВНИМАНИЕ:
В версии 6.1 изменился формат пакетных фильтров. Если вы используете пакетные фильтры, их следует преобразовать в новый формат (tcpdump/libpcap) в ручном режиме. Старые фильтры сохранены в каталоге \backup\DD.MM.YYYY\config.

Пакетные фильтры для службы EtherSensor PCAP абсолютно идентичны пакетным фильтрам для EtherSensor EtherCAP [42].

Поэтому после отладки фильтров на PCAP-файлах их можно скопировать из конфигурационного файла pcap.xml в конфигурационный файл ethcap.xml [37] службы EtherSensor EtherCAP, и затем использовать в её дальнейшей работе.

Используйте в этом процессе EtherSensor PCAP Edition: это сэкономит вам много времени и усилий.

3.2. Служба EtherSensor EtherCAP

Служба EtherSensor EtherCAP отвечает за пассивный захват трафика на сетевых адаптерах.

EtherSensor EtherCAP извлекает из обрабатываемого трафика объекты уровня приложения (L7 согласно модели OSI), и затем передаёт эти объекты для дальнейшей обработки в службу EtherSensor Analyser.

EtherSensor EtherCAP может обрабатывать следующие протоколы третьего уровня модели OSI:

IP:

Обычный трафик

GRE:

Туннелированный трафик: например, соединения нешифрованного протокола Ethernet over GRE или IP-over-IP

IPv6-over-IPv4:

Инкапсулированный протокол IPv6. Как правило, встречается в крупных сетях и на магистральных каналах.

Во всех случаях EtherSensor обрабатывает в основном TCP- и UDP-потоки.

В текущей версии EtherSensor (6.1) служба EtherSensor EtherCAP может распознавать и обрабатывать следующие наиболее часто используемые интернет протоколы передачи данных уровня приложения:

HTTPv1/HTTPv2:

Все виды запросов, передача сообщений, передача файлов.

SMTP:

Передача исходящих почтовых сообщений.

POP3:

Передача входящих почтовых сообщений.

IMAP4:

Передача входящих и исходящих почтовых сообщений.

ICQ:

Передача входящих и исходящих мгновенных сообщений, списков контактов, файлов.

DC:

Передача мгновенных сообщений по протоколам NMDC и ADC (DC++).

LOTUS:

Передача почтовых сообщений, календарей, задач системы Lotus Notes.

MRA:

Передача мгновенных сообщений, списков контактов и файлов при помощи утилиты Mail.Ru Агент.

MSN:

Передача мгновенных сообщений, списков контактов и файлов при помощи утилиты MSN.

XMPP:

Передача мгновенных сообщений, списков контактов и файлов при помощи XMPP-клиентов (Google Talk и др.)

IRC:

Передача мгновенных сообщений и файлов.

SKYPE:

Детектирование факта использования клиентов skype, извлечение сообщений.

SSL:

Детектирование факта использования протокола SSL.

TORRENT:

Детектирование факта использования Torrent-клиентов.

FTP:

Передача файлов.

YAHOO:

Передача мгновенных сообщений, файлов.

ICAP:

Перехват данных, передаваемых по протоколу ICAP.

> Microolap EtherSensor включает в себя службу EtherSensor ICAP: это ICAP-сервер, предназначенный для работы с ICAP-клиентами.
>
> Пассивный перехват протокола ICAP – другая функция Microolap EtherSensor.

SOCKS:

Перехват данных, передаваемых по протоколу SOCKS.

WEBSOCKET:

Перехват данных, передаваемых по протоколу WEBSOCKET.

Общая схема работы службы EtherSensor EtherCAP:

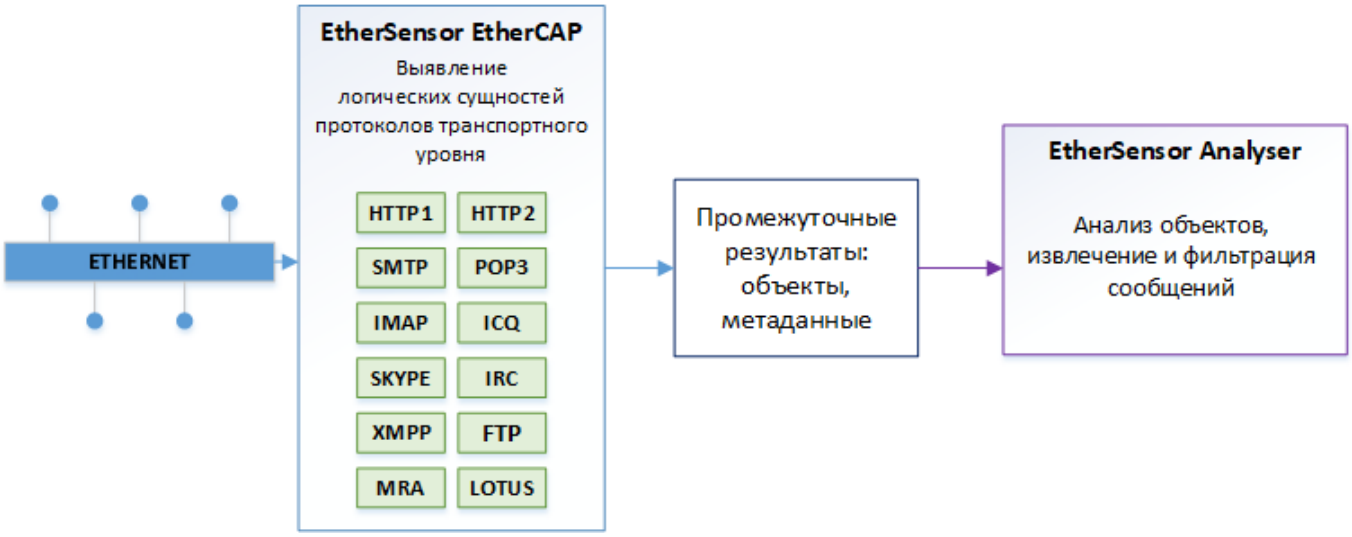

Рис.8. Схема работы службы EtherSensor EtherCAP

Служба EtherSensor EtherCAP позволяет вести одновременный перехват трафика со всех доступных Ethernet интерфейсов, а также отслеживать локальные директории для обработки помещённых в них PCAP-файлов.

Размер необходимых аппаратных ресурсов на каждый прослушиваемый сетевой интерфейс можно рассчитать исходя из того, что среднее количество памяти, необходимое для обработки одного TCP соединения составляет приблизительно 40 Кбайт:

Пропускная способность сетевого интерфейса	Расход оперативной памяти на кэш обрабатываемых пакетов
10000 Mbps	2000 МБ
5000 Mbps	1000 МБ
1000 Mbps	200 МБ
100 Mbps	50 МБ
10 Mbps	10 МБ

Таким образом, для одновременного отслеживания 10000 TCP сессий через сетевой интерфейс с пропускной способностью 1 Gbps серверу EtherSensor необходимо иметь следующее количество доступной физической памяти:

200 МБ + 10000 * 40 КБ – около 600 МБ

Служба EtherSensor EtherCAP позволяет для оптимизации обработки трафика назначать для каждого прослушиваемого сетевого интерфейса пакетный фильтр, а также сетевые протоколы, которые необходимо отслеживать.

Варианты организации работы службы EtherSensor EtherCAP:

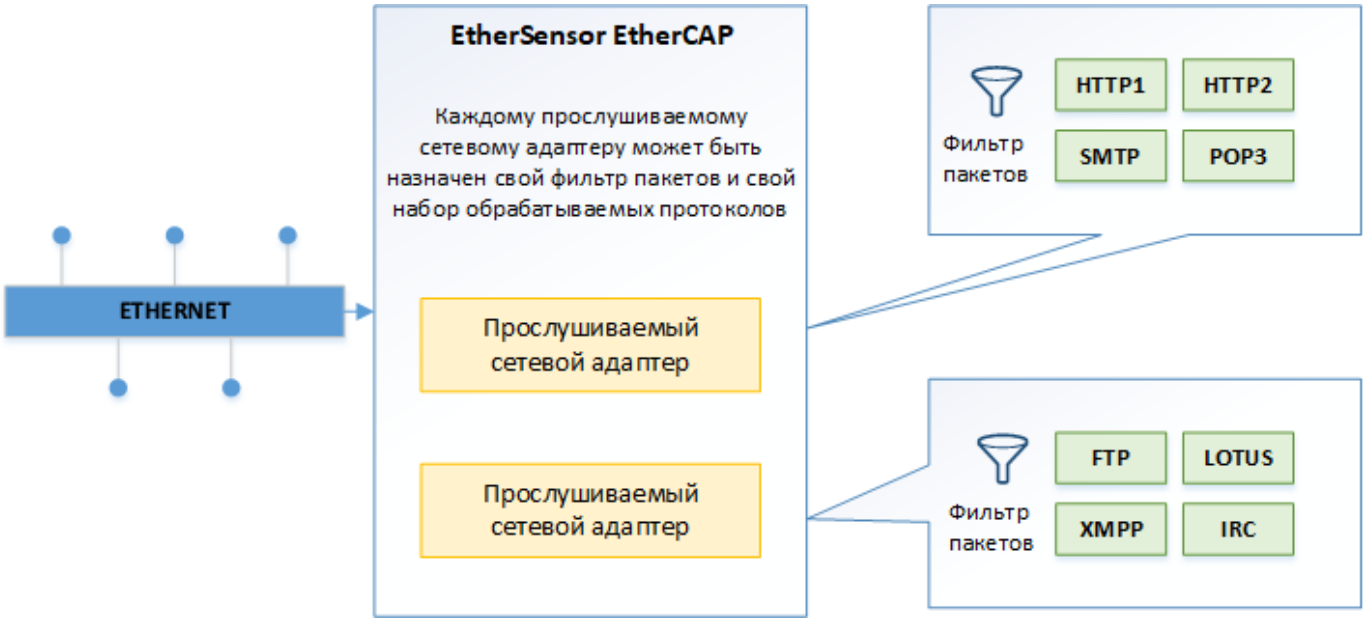

Рис.9. Варианты организации работы службы EtherSensor EtherCAP.

Параметры командной строки

Служба Windows EtherSensor EtherCAP в ходе инсталляции EtherSensor устанавливается с автоматическим запуском. Однако, при необходимости процесс sensor_ethercap.exe можно запустить как приложение Windows со следующими параметрами командной строки:

/process

Запустить процесс sensor_ethercap.exe как обычный Windows Win32-процесс (возможно использование для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

3.2.1. Настройка EtherSensor EtherCAP

Служба EtherSensor EtherCAP позволяет прослушивать аппаратные сетевые интерфейсы, установленные на сервере EtherSensor.

Настройки аппаратных сетевых интерфейсов

Информация о настройках сетевых интерфейсов и пакетных фильтрах содержится в файле ethcap.xml, находящемся в директории [INSTALLDIR]\config, который может быть отредактирован как в консоли управления, так и любым текстовым редактором.

В консоли управления (файл sensor_console.exe) настройка службы EtherSensor EtherCAP происходит следующим образом:

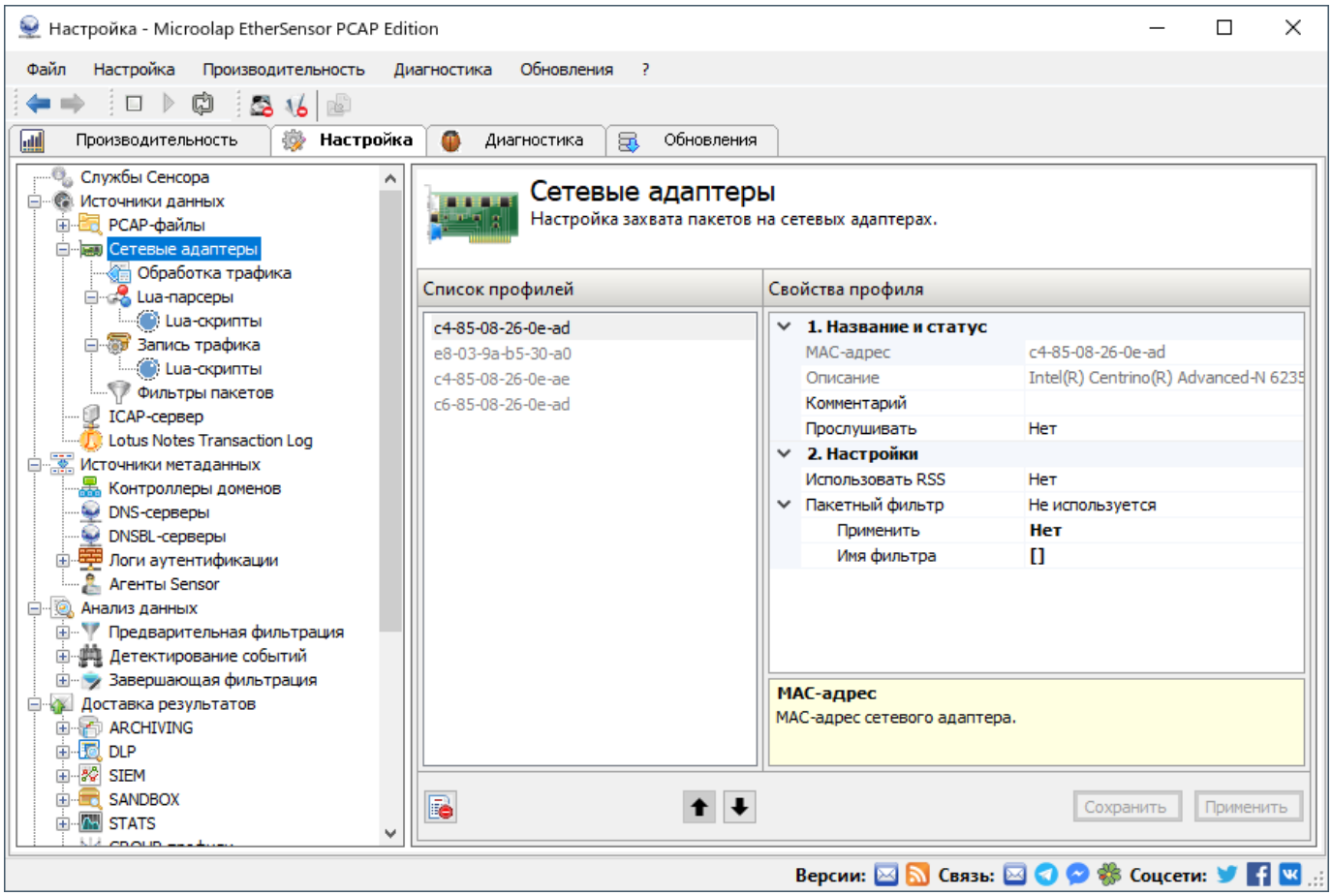

Рис.10. Настройка службы EtherSensor EtherCAP.

MAC-адрес:

MAC-адрес сетевого адаптера, которые вы настраиваете. Значение MAC-адреса будет присутствовать в сообщениях, отправляемых EtherSensor системе-потребителю.

Прослушивать:

Включение/выключение прослушивания данного интерфейса службой EtherSensor EtherCAP. Возможно одновременное прослушивание нескольких интерфейсов. По умолчанию служба EtherSensor EtherCAP прослушивает все интерфейсы, для которых не сконфигурирован сетевой стек ОС.

Чтобы прослушивать интерфейс, для которого сконфигурирован сетевой стек ОС, необходимо установить для него флаг ***Прослушивать*** в ***Да***. Чтобы не прослушивать те интерфейсы, на которых не сконфигурирован сетевой стек ОС, необходимо их флаги ***Прослушивать*** установить в ***Нет***.

Использовать RSS:

Включение/выключение использования технологии RSS . Receive Side Scaling (RSS) – технология, которая равномерно распределяет нагрузку по обработке сетевых пакетов между ядрами процессора, позволяя оптимизировать производительность.

Пакетный фильтр:

Отображается название фильтра, ассоциированного с данным интерфейсом и статус его использования (***Используется/Не используется)***. Вы можете определить заранее сколь угодно много фильтров, но применять одновременно на одном и том же интерфейсе вы можете только один из них.

Имя фильтра:

Поле выбора пакетного фильтра из числа уже имеющихся. Фильтры хранятся в файле **[INSTALLDIR]\config\ethcap.xml** [37].

Применить:

Включение/выключение использования указанного фильтра на данном сетевом интерфейсе.

Протоколы:

Позволяет в целях экономии ресурсов отключить неиспользуемые фильтры протоколов. Например, если нет необходимости в работе с мгновенными сообщениями, следует отключить фильтры для протоколов DC, ICQ, IRC, MRA, MSN, SKYPE и YAHOO.

Файл конфигурации EtherSensor EtherCAP

Конфигурация службы EtherSensor EtherCAP содержится в XML-файле ethcap.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

3.2.2. Файл конфигурации EtherSensor EtherCAP

Конфигурация службы EtherSensor EtherCAP содержится в XML-файле ethcap.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config

```
<?xml version="1.0" encoding="utf-8"?>
<EtherCapConfig version="4.2"
 flow_count="16"
 flow_buff_count="512"
 flow_buff_size="524288">
  <NetworkAdapters>
   <NetworkAdapter enabled="true" rss="true" mac="00-1F-C6-2D-EA-40"
    description="Marvell Yukon 88E8056 PCI-E Gigabit Ethernet Controller">
     <Filter enabled="true" name="internet" />
     <Protocol enabled="true" name="dc" />
     <Protocol enabled="true" name="ftp" />
     <Protocol enabled="true" name="http" />
     <Protocol enabled="true" name="icq" />
     <Protocol enabled="true" name="imap4" />
     <Protocol enabled="true" name="irc" />
     <Protocol enabled="true" name="lotus" />
     <Protocol enabled="true" name="mra" />
     <Protocol enabled="true" name="msn" />
     <Protocol enabled="true" name="pop3" />
     <Protocol enabled="true" name="skype" />
     <Protocol enabled="true" name="smtp" />
     <Protocol enabled="true" name="ssl" />
     <Protocol enabled="true" name="xmpp" />
     <Protocol enabled="true" name="yahoo" />
   </NetworkAdapter>
  </NetworkAdapters>

 <Filters>
  <Filter name="default">
   <RuleGroup enabled="true" name="">
    <Rule type="accept" src="any" srcport="any" dst="any" dstport="any" proto="tcp"
comment="Comment to the rule" />
   </RuleGroup>
  </Filter>

  <Filter name="internet">
   <RuleGroup enabled="true" name="">
    <Rule type="reject" src="192.168.0.1" srcport="any" dst="any" dstport="any" proto="tcp"
comment="Comment to the rule" />
    <Rule type="reject" src="*" srcport="any" dst="192.168.0.1" dstport="any" proto="tcp" />
    <Rule type="accept" src="any" srcport="any" dst="any" dstport="any" proto="tcp" />
   </RuleGroup>
  </Filter>
 </Filters>
</EtherCapConfig>
```

Тег EtherCapConfig

Корневой тег конфигурации EtherSensor EtherCAP. Атрибут "version" содержит версию файлов конфигурации.

Атрибут "flow_count" содержит количество потоков, одновременно обрабатывающих трафик. Весь перехватываемый трафик равномерно распределяется между потоками обработки.

Распределение трафика происходит на уровне ядра операционной системы, тем самым обеспечивается параллелизм обработки данных.

Атрибут "flow_buff_count" содержит количество буферов данных в потоке обработки трафика.

Этот параметр совместно с атрибутом "flow_buff_size" задаёт статический объём памяти, используемой для одного потока обработки.

Атрибут "flow_buff_size" содержит размер буфера данных в потоке обработки трафика (байты). Этот параметр совместно с атрибутом "flow_buff_count" задаёт статический объём памяти, используемой для одного потока обработки. Объём памяти для одного потока составляет "flow_buff_count" * "flow_buff_size" = 512 * 524288 = 256 МБ.

> Указанные выше атрибуты используются для тонкой инженерной настройки Microolap EtherSensor и требуют глубокого понимания функционирования продукта. Не экспериментируйте с ними на рабочей системе.

Тег NetworkAdapters

Определяет настройки прослушиваемых сетевых интерфейсов.

Тег NetworkAdapter

Тег NetworkAdapter является вложенным в тег NetworkAdapters и содержит настройки конкретного сетевого интерфейса. Атрибут "enabled" содержит статус активности сетевого интерфейса, и если этот атрибут равен "false", то трафик, поступающий с этого интерфейса, будет игнорироваться.

Атрибут "rss" (true/false) содержит статус использования технологии Receive Side Scaling (RSS). Это технология, которая равномерно распределяет нагрузку по обработке сетевых пакетов между ядрами процессора, позволяя оптимизировать производительность.

Атрибут "mac" содержит MAC-адрес сетевого интерфейса. Этот атрибут не должен изменяться, и предназначен только для чтения. Атрибут "description" содержит описание сетевого интерфейса и заполняется на усмотрение администратора.

Тег Filter

Тег Filter является вложенным в тег NetworkAdapter и указывает на описание используемого IP-фильтра для данного сетевого интерфейса. Атрибут "enabled" содержит статус использования IP-фильтра, и если данный атрибут равен "false", то для данного сетевого интерфейса не будет использоваться IP-фильтр в обработке данных. Атрибут "name" содержит имя профиля IP-фильтра. Профили IP-фильтров указываются в теге Filters [37].

Тег Protocol

Тег Protocol является вложенным в тег NetworkAdapter и содержит описание интернет-протокола, данные которого требуется обрабатывать. Атрибут "enabled" содержит статус

обработки интернет-протокола, и если данный атрибут равен "false", то для данного сетевого интерфейса этот интернет-протокол будет игнорироваться. Атрибут "name" содержит имя интернет-протокола. Этот атрибут используется только для чтения и не подлежит изменению.

Пример:

Настройки сетевого интерфейса для перехвата сообщений клиентов DC, ICQ, IRC, MRA, MSN, XMPP/Jabber, YAHOO:

```
<NetworkAdapter enabled="true" mac="00-1F-C6-2D-EA-40"
  description="Marvell Yukon Controller
  88E8056 PCI-E Gigabit Ethernet">
 <Filter enabled="true" name="default" />
 <Protocol enabled="false" name="ftp" />
 <Protocol enabled="false" name="http" />
 <Protocol enabled="true" name="icq" />
 <Protocol enabled="true" name="irc" />
 <Protocol enabled="true" name="mra" />
 <Protocol enabled="true" name="msn" />
 <Protocol enabled="false" name="pop3" />
 <Protocol enabled="false" name="skype" />
 <Protocol enabled="false" name="smtp" />
 <Protocol enabled="true" name="xmpp" />
</NetworkAdapter>
```

Тег Filters

Определяет настройки профилей IP-фильтров.

Тег Filter

Тег Filter является вложенным в тег Filters и содержит настроек IP-фильтра. Атрибут "name" содержит имя профиля IP-фильтра. Значение данного атрибута может быть использовано в качестве значения атрибута "NetworkAdapter/Filter/name" для указания IP-фильтра сетевому интерфейсу.

Тег RuleGroup

Тег RuleGroup является вложенным в тег Filter и служит для группировки правил фильтрации, относящихся к конкретной решаемой задаче фильтрации трафика. Атрибут "name" содержит имя (или описание) группы правил фильтрации. Значение данного атрибута может оставаться пустым.

Тег Rule

Тег Rule является вложенным в тег RuleGroup и содержит описание правила фильтрации сетевого трафика. Атрибут "type" содержит тип правила, и если данный атрибут равен "accept", то сетевые пакеты, подходящие под данное правило, будут пропущены для дальнейшей обработки, иначе, если он равен "reject", сетевые пакеты, подходящие под данное правило будут отклонены. Атрибуты "src" и "dst" содержат IP-адрес, диапазон IP-адресов или параметры сети для фильтрации IP-адресов, подходящих под указанное значение. В атрибуте "comment" вы можете написать комментарий к правилу фильтрации пакетов.

Пример:

Отклонить пакеты, проходящие между компьютерами 10.1.5.10, 10.1.5.15-10.1.5.59 и сетью 10.1.6.0/255.255.255.0:

```
<Rule
 type="reject"
 src="10.1.5.10, 10.1.5.15-10.1.5.59"
 dst="10.1.6.0/255.255.255.0"
 proto="tcp"
 comment="" />

<Rule type="reject"
 src="10.1.6.0/255.255.255.0"
 dst="10.1.5.10, 10.1.5.15-10.1.5.59"
 proto="tcp" />
```

В атрибутах "srcport" и "dstport" укажите необходимые для фильтрации TCP порты или диапазоны TCP портов.

Пример:

Отклонить пакеты, проходящие между компьютерами 10.1.5.10, 10.1.5.15-10.1.5.59 и сетью 10.1.6.0/255.255.255.0 на портах 80, 443-1024:

```
<Rule
 type="reject"
 src="10.1.5.10, 10.1.5.15-10.1.5.59"
 srcport="80, 443-1024"
 dst="10.1.6.0/255.255.255.0"
 proto="tcp" />

<Rule
 type="reject"
 src="10.1.6.0/255.255.255.0"
 dst="10.1.5.10, 10.1.5.15-10.1.5.59"
 dstport="80, 443-1024"
 proto="tcp" />
```

Правила применяются линейно сверху вниз. Верхняя строка – это первая инструкция фильтра, нижняя – последняя. Каждая строка отклоняет или принимает только тот тип пакетов, который она описывает.

Пример:

Отклонить пакеты соединения между двумя хостами или группой хостов. При этом должны быть отклонены пакеты, передаваемые в обе стороны соединения:

```
<Rule
 type="reject"
 src="10.31.5.212"
 dst="10.31.5.57"
 dstport="1025"
 proto="tcp" />

<Rule
 type="reject"
 src="10.31.5.57"
 srcport="1025"
 dst="10.31.5.212"
 proto="tcp" />
```

Также необходимо помнить, что если не определено ни одно правило фильтрации, то мы получаем весь трафик. И наоборот, если есть правила фильтрации, то обрабатывается только трафик, описанный этими правилами.

Пример:

Получить все соединения только с хостом 10.31.5.57:

```
<Rule
 type="accept"
 src="10.31.5.57"
 srcport="*"
 dst="*"
 dstport="*"
 proto="tcp" />

<Rule
 type="accept"
 src="*"
 srcport="*"
 dst="10.31.5.57"
 dstport="*"
 proto="tcp" />
```

Пример:

Для того, чтобы отсечь группу хостов, необходимо сначала отклонить пакеты этой группы, затем обязательно пропустить все остальные пакеты, иначе весь трафик будет пропускаться без анализа:

```
<Rule
 type="reject"
 src="10.31.5.212"
 dst="10.31.5.57"
 dstport="1025"
 proto="tcp" />

<Rule
 type="reject"
 src="10.31.5.57"
 srcport="1025"
 dst="10.31.5.212"
 proto="tcp" />

<Rule
 type="accept"
 src="*"
 srcport="*"
 dst="*"
 dstport="*"
 proto="tcp" />
```

Таким образом можно отсекать трафик с нужной стороны прокси-сервера.

Пример:

Если описать правила, пропускающие трафик только для определённых хостов, и запретить остальной трафик, будет обрабатываться трафик только этих хостов:

```
<Rule
 type="accept"
 src="10.31.5.212"
 dst="10.31.5.57"
 dstport="1025"
 proto="tcp" />

<Rule type="accept"
 src="10.31.5.57"
 srcport="1025"
 dst="10.31.5.212"
 proto="tcp" />
```

3.2.3. Пакетные фильтры EtherSensor EtherCAP

> **ВНИМАНИЕ:**
> В версии 6.1 изменился формат пакетных фильтров. Если вы используете пакетные фильтры, их следует преобразовать в новый формат (tcpdump/libpcap) в ручном режиме. Старые фильтры сохранены в каталоге \backup\DD.MM.YYYY\config.

Для создания и редактирования пакетных фильтров используйте консоль управления, секция ***Фильтры пакетов***:

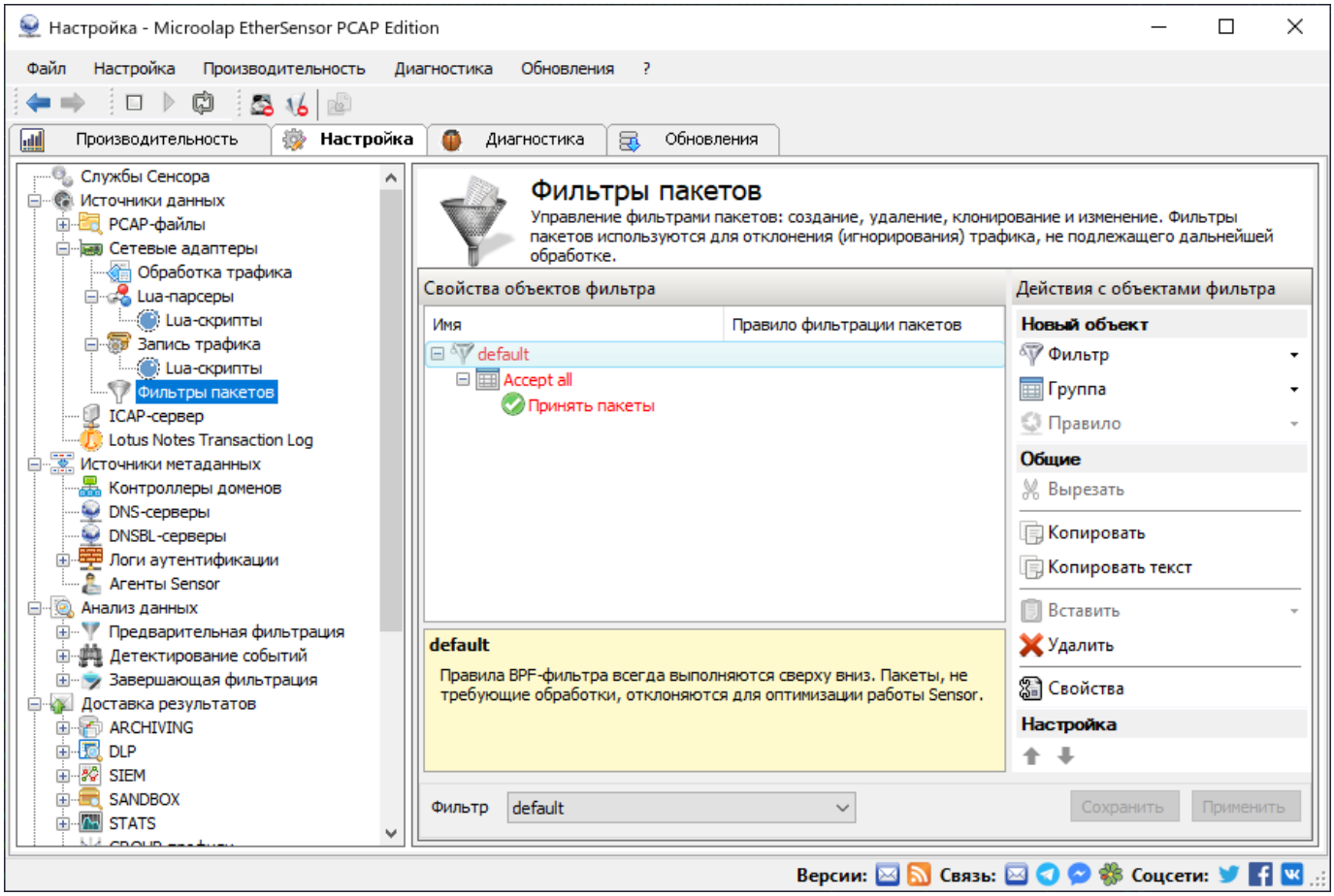

Рис.11. Настройки пакетных фильтров.

Панель Действия с объектами фильтра:

Создание, клонирование и удаление фильтров и их объектов: правил и групп правил.

Панель Свойства объектов фильтра

Редактирование свойств самого фильтра, групп и правил.

Панель редактирования свойств объекта вызывается двойным щелчком мыши по соответствующему объекту: фильтру, группе правил или правилу:

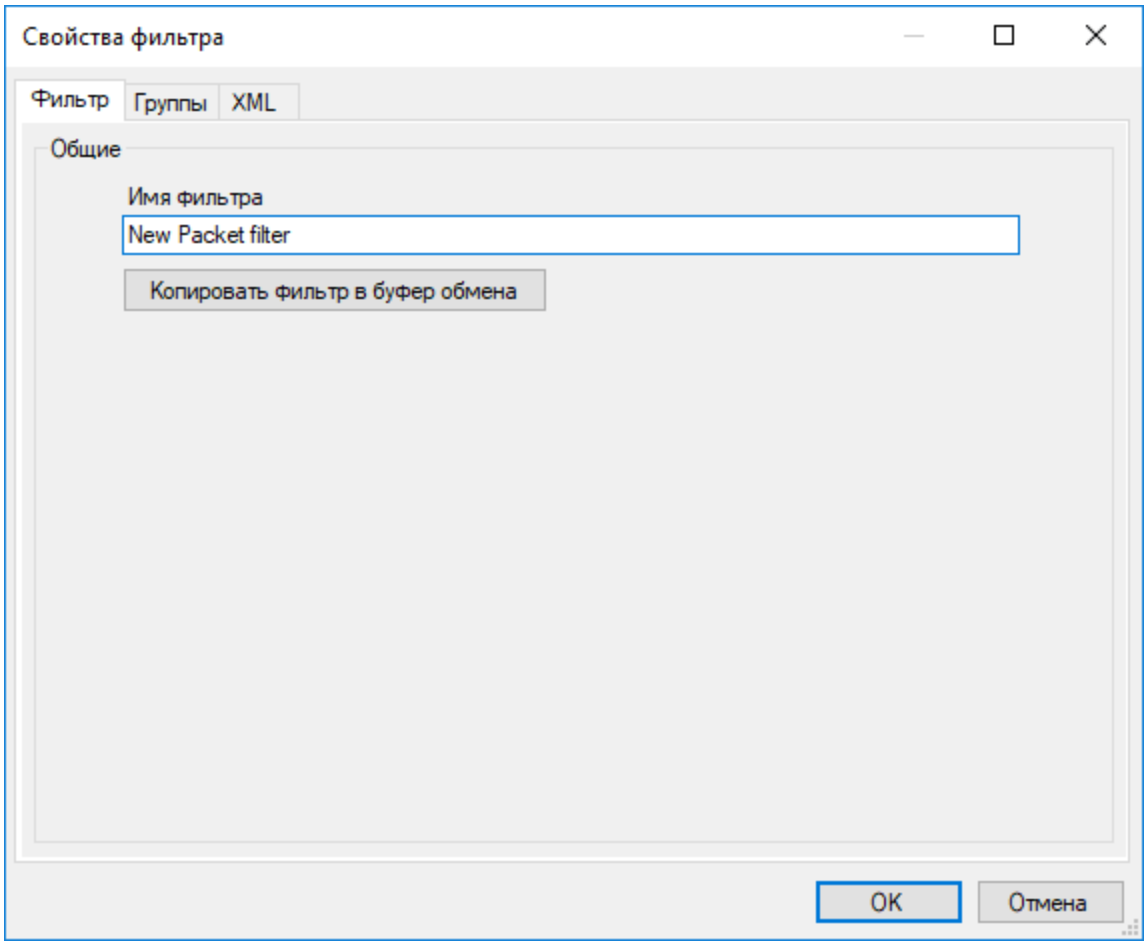

Рис.12. Панель редактирования свойств фильтра.

Имя фильтра:

Название фильтра (на усмотрение администратора).

Вкладка групп правил фильтрации:

BPF-программа не ограничивается по длине и количество правил в фильтре может быть весьма велико. Для удобства работы с правилами используйте возможность объединения их в группы с информативными названиями.

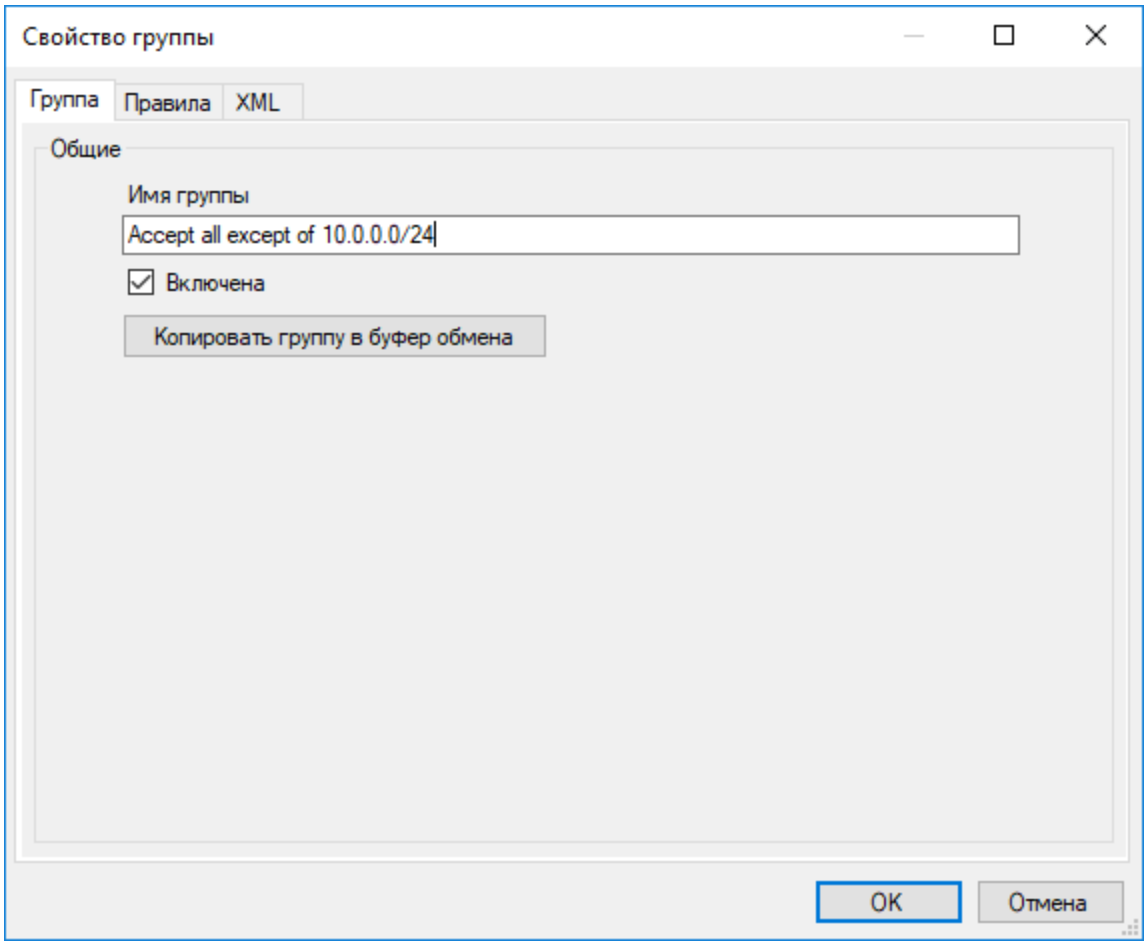

Рис.13. Панель редактирования свойств группы правил.

Имя группы:

Имя группы правил (на усмотрение администратора).

Чекбокс Включена:

Позволяет отключать/подключать группу правил.

Вкладка Правила:

К редактированию правил можно получить доступ как с уровня фильтра, так и из вкладки "Правила" свойств группы правил.

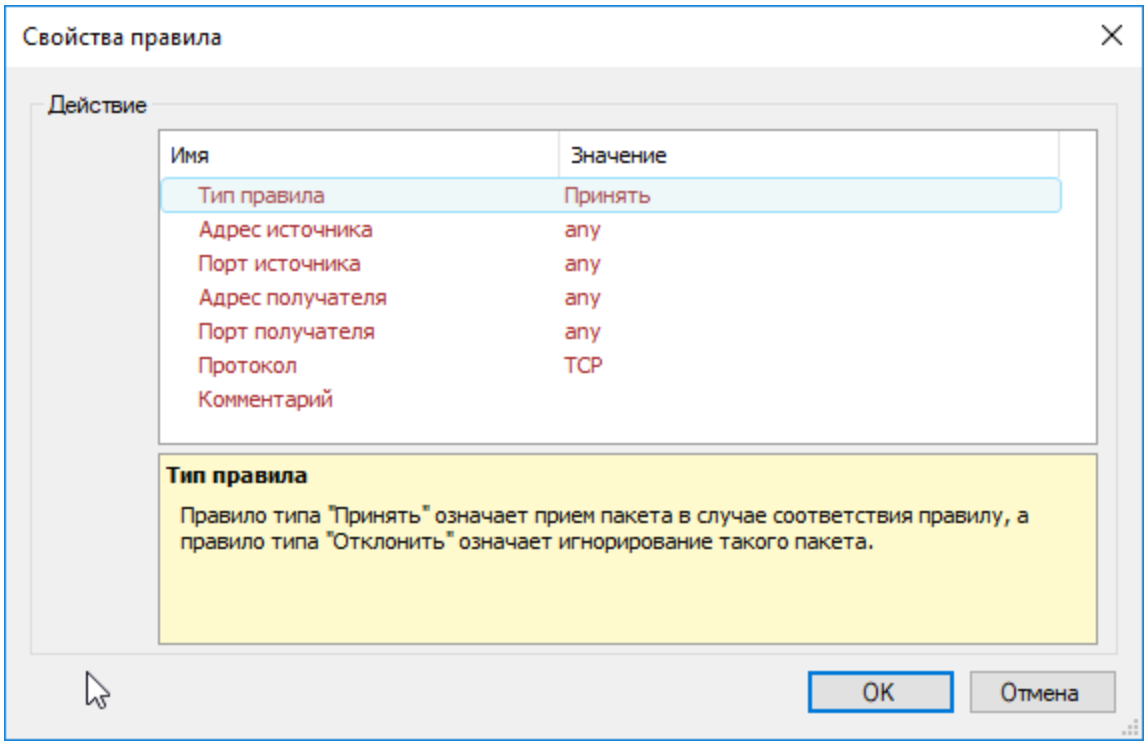

Рис.14. Панель редактирования правила.

Панель редактирования действия и условий правила вызывается двойным щелчком мыши по названию правила.

Тип правила:

Определено два типа правил: "Принять" и "Отклонить." Правило типа "Принять" означает прием пакета в случае соответствия условиям правила, а правило типа "Отклонить" означает отклонение (игнорирование) такого пакета.

> Правила фильтра применяются всегда последовательно, перестановка правил в фильтре может в корне изменить результат его работы.

Адрес источника:

Адрес источника. Например: any или 10.0.0.0/24 или 10.0.0.0-10.0.0.255 или список через запятую вида 100.100.100.1-100.100.100.255, 192.168.0.0/8.

Порт источника:

Порт источника, например: any или 80 или 80-8080 (с 80 по 8080), или 80-8000, 9000-10000 (с 80 по 8000 и с 9000 по 10000).

Адрес получателя:

Адрес получателя. Например: any или 10.0.0.0/24 или 10.0.0.0-10.0.0.255 или список через запятую вида 100.100.100.1-100.100.100.255, 192.168.0.0/8.

Порт получателя:

Порт получателя, например: any или 80 или 80-8080 (с 80 по 8080), или список через запятую 80-8000, 9000-10000 (с 80 по 8000 и с 9000 по 10000).

Протокол:

Один из TCP, UDP, GRE, IP6, либо any.

Комментарий

Ваш комментарий к данному правилу фильтрации.

Подробнее с настройками службы EtherSensor EtherCAP можно ознакомиться в разделе Ручная настройка (файл конфигурации) [37].

3.3. Служба EtherSensor ICAP

Служба EtherSensor ICAP является ICAP-сервером, предназначенным для получения сетевого трафика по протоколу ICAP от любых ICAP-клиентов в режиме REQMOD.

Протокол ICAP (Internet Content Adaptation Protocol) предназначен для работы только с протоколом HTTP и используется для контентной фильтрации и определения вредоносного содержимого (вирусы, spyware/malware).

В качестве клиента ICAP выступает система, через которую передается HTTP-трафик. Такой системой могут быть различные HTTP-прокси, поддерживающие ICAP (например, SQUID, Blue Coat Proxy SG, Cisco IronPort S, Webwasher). После получения данных от клиента некоторые ICAP-серверы выполняют их обработку и, если это необходимо, то и модификацию данных.

Затем данные возвращаются клиенту ICAP, и он их передает дальше серверу или клиенту в зависимости от того, в каком направлении они передавались.

Поскольку ICAP-сервер Microolap EtherSensor использует трафик, получаемый от ICAP-клиентов, только для анализа, трафик всегда возвращается ICAP-клиенту без изменений.

Архитектура системы при использовании протокола ICAP:

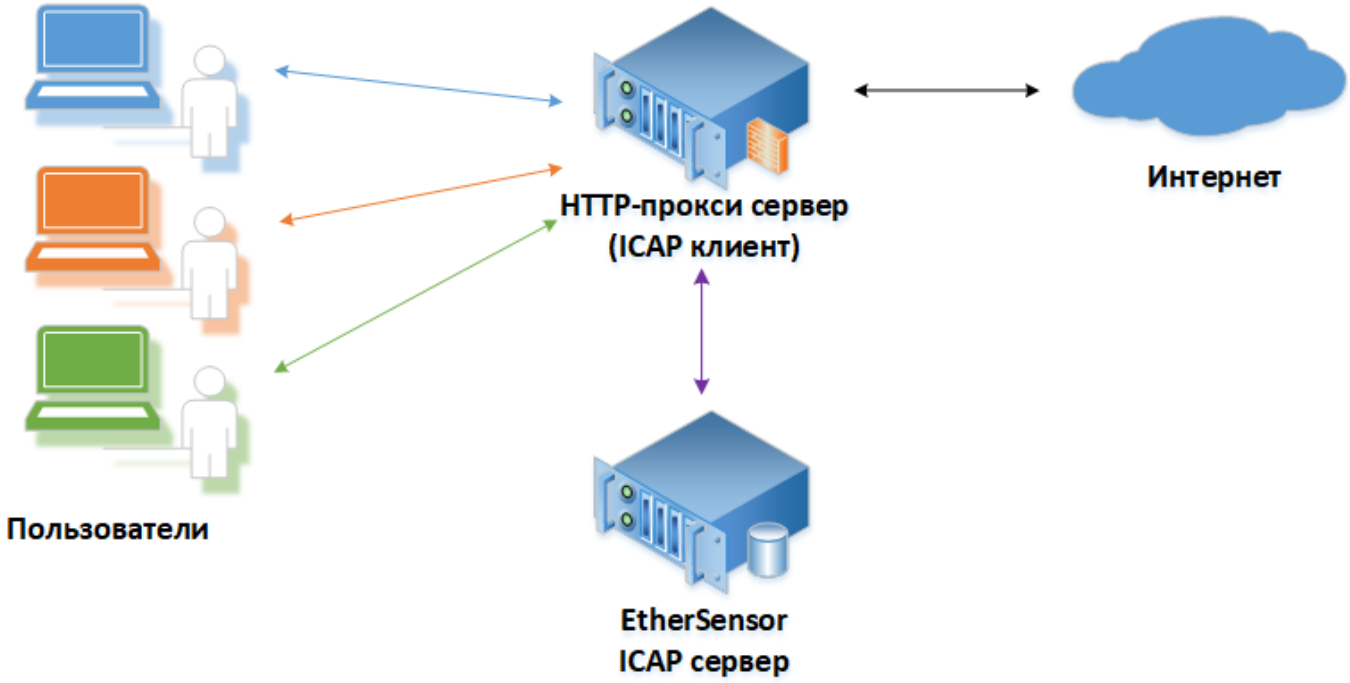

Рис.15. Схема взаимодействия службы EtherSensor ICAP и ICAP-клиента.

Некоторые клиенты ICAP используют расширения заголовков, что позволяет им передавать на ICAP-сервер информацию о пользователе, который авторизовался на прокси-сервере. Эта информация учитывается при дальнейшей обработке сообщений в EtherSensor.

Параметры командной строки

Служба Windows EtherSensor ICAP в ходе инсталляции Microolap EtherSensor устанавливается с автоматическим запуском. Однако, при необходимости процесс sensor_icap.exe можно запустить как приложение Windows со следующими параметрами командной строки:

/process

Запустить процесс sensor_icap.exe как обычный Windows Win32-процесс (возможно использование для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

3.3.1. Настройка EtherSensor ICAP

Служба EtherSensor ICAP является ICAP-сервером, предназначенным для работы с ICAP-клиентами, и имеющим свойственные этому типу серверов настройки:

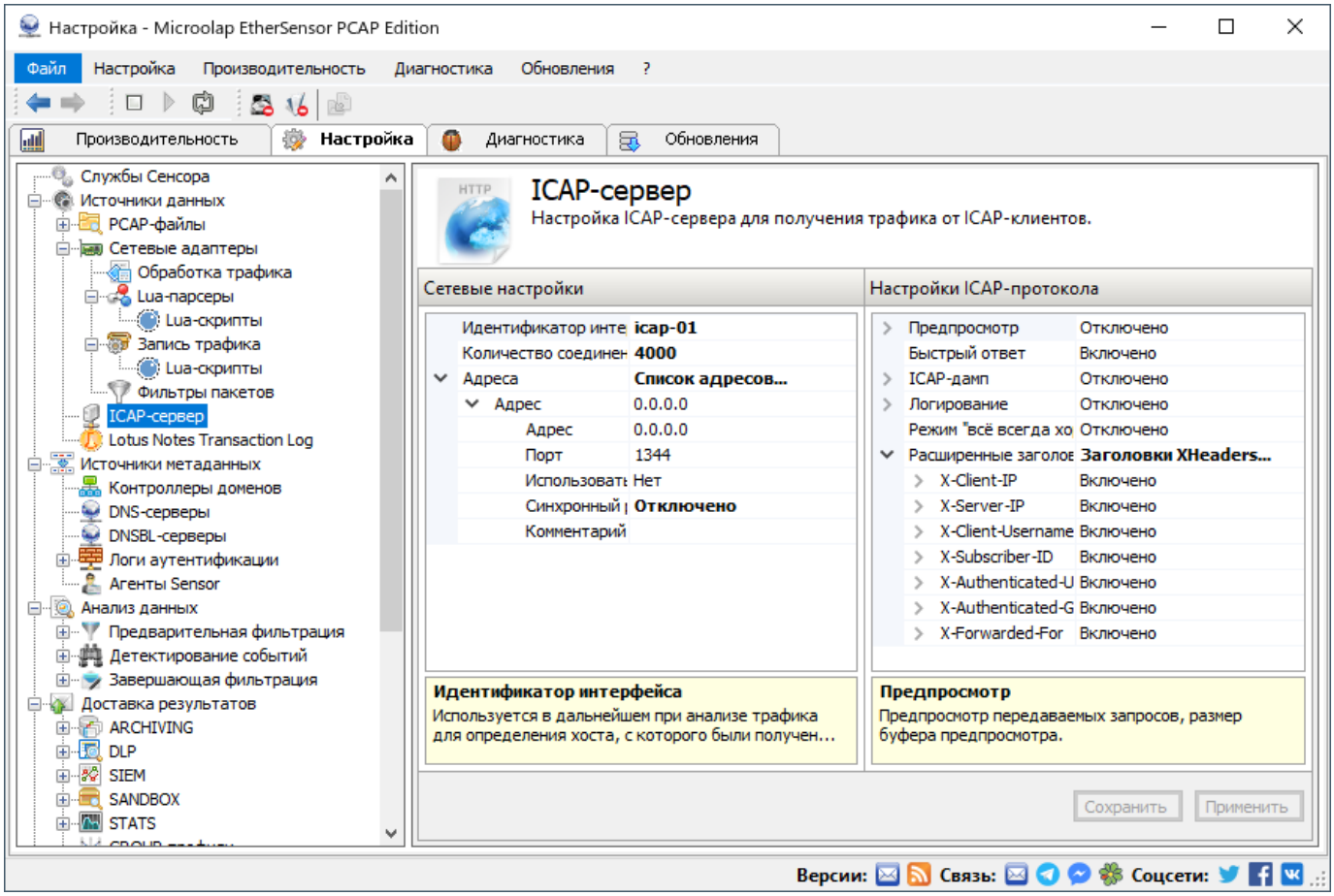

Рис.16. Настройки службы EtherSensor ICAP.

Количество соединений:

Максимальное количество соединений ICAP-сервера с клиентами. Примерное количество потребляемой на одно соединение оперативной памяти составляет около 8КБ.

Адреса:

IP-адреса/порты, прослушиваемые ICAP-сервером.

Предпросмотр:

Разрешение/запрещение предпросмотра, размер буфера.

Быстрый ответ:

Разрешение/запрещение быстрого короткого ответа Allow 204 (No modifications).

Синхронный режим:

Разрешение/запрещение использования синхронного режима работы ICAP-сервера. В синхронном режиме ICAP-сервер сначала получает весь запрос целиком, и только потом его отправляет обратно ICAP-клиенту, даже если такой запрос представляет собой значительный объем данных, например, ISO-образ диска и т.п. Необходимость синхронного режима может быть следствием настроек ICAP-клиента (например, Blue Coat, IronPort и т.п.).

ICAP-дамп:

Дамп всех данных обмена ICAP-клиента и сервера. Применяется только для отладки взаимодействия со сторонним программным обеспечением.

Логирование:

Логирование ICAP-сервером HTTP-запросов для службы EtherSensor Watcher.

Режим "всё всегда хорошо":

В этом режиме ICAP-сервер при обнаружении ошибок в ICAP-протоколе со стороны клиента отвечает ему не соответствующим кодом ошибки, а кодом Allow 204 (No modifications).

Расширенные заголовки ICAP-протокола:

Позволяет уведомлять ICAP-клиентов о том, что сервер поддерживает указанные заголовки.

Поддерживаемые имена расширенных ICAP заголовков:

X-Client-IP

X-Server-IP

X-Client-Username

X-Subscriber-ID

X-Authenticated-User

X-Authenticated-Groups.

Данные заголовки в процессе обработки ICAP-трафика в случае обнаружения сигнатуры сообщения при отправке сообщения системе-потребителю преобразуются в следующие заголовки:

X-Sensor-Icap-Client-Username

X-Sensor-Icap-Subscriber-ID

X-Sensor-Icap-Authenticated-User

X-Sensor-Icap-Authenticated-Groups

Значения заголовков X-Client-IP и X-Server-IP сохраняются в заголовках X-Sensor-Src-Address, X-Sensor-Dst-Address.

Файл конфигурации EtherSensor ICAP

Конфигурация службы EtherSensor ICAP содержится в XML-файле icap.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

3.4. Служба EtherSensor LotusTXN

Служба EtherSensor LotusTXN предназначена для извлечения сообщений Lotus Notes из журнала транзакций (Lotus Notes Transaction Log).

EtherSensor LotusTXN извлекает сообщения из файлов Lotus Notes Transaction Log, и затем передаёт эти сообщения для дальнейшей обработки в службу EtherSensor Analyser.

В текущей версии Microolap EtherSensor (6.1) EtherSensor LotusTXN может отслеживать одновременно сразу несколько директорий Lotus Notes Transaction Log. Это даёт возможность администратору с помощью одной инсталляции EtherSensor отслеживать одновременно несколько систем Lotus Notes, при этом директории Lotus Notes Transaction Log могут быть как локальными, так и удалёнными.

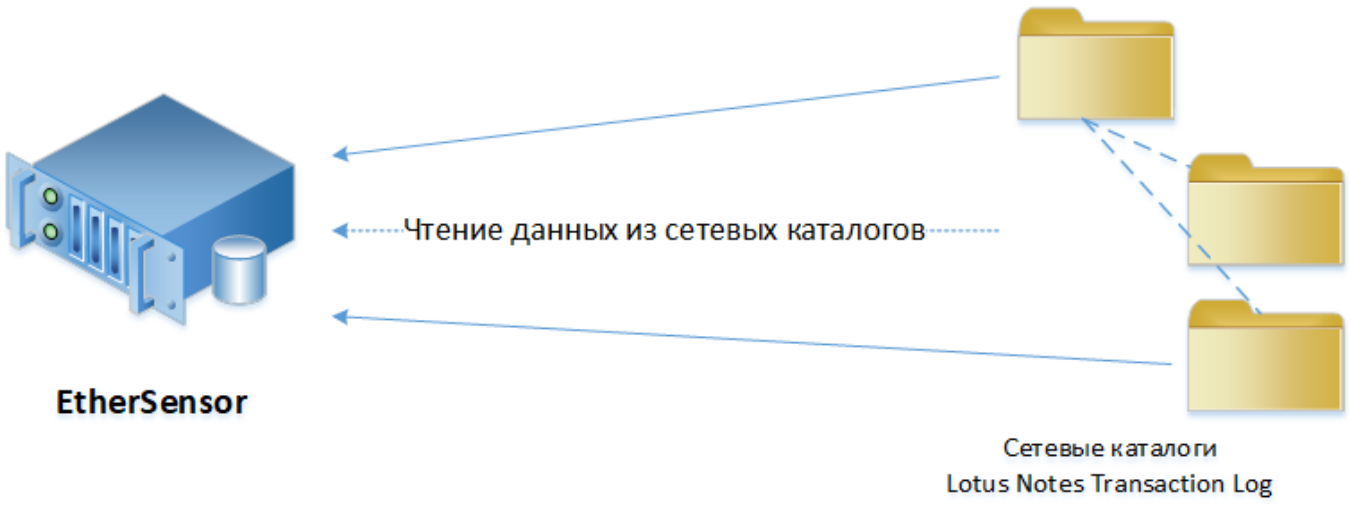

Рис.17. Схема работы службы EtherSensor LotusTXN.

Используйте службу EtherSensor LotusTXN только в случае, если в системе Lotus Notes применяется шифрование или если вам необходимо отслеживать Lotus Notes Transaction Log на удалённых серверах.

Извлечение из трафика нешифрованных сообщений Lotus Notes наравне с SMTP, POP3 и IMAP4 обеспечивает служба EtherSensor EtherCAP.

Параметры командной строки

Служба Windows EtherSensor LotusTXN в ходе инсталляции Microolap EtherSensor устанавливается с автоматическим запуском. Однако, при необходимости процесс sensor_lotustxn.exe можно запустить как приложение Windows со следующими параметрами командной строки:

/process

Запустить процесс sensor_lotustxn.exe как обычный Windows Win32-процесс (возможно использовать для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

3.4.1. Настройка EtherSensor LotusTXN

Служба EtherSensor LotusTXN позволяет вести мониторинг и реконструкцию сообщений системы Lotus Notes методом извлечения их из Lotus Notes Transaction Log (журнала транзакций Lotus Notes).

В случае, если в системе Lotus Notes не применяется шифрование, за извлечение сообщений Lotus Notes из трафика наравне с SMTP, POP3 и IMAP4 отвечает служба EtherSensor EtherCAP.

Для того, чтобы настроить службу EtherSensor LotusTXN для отслеживания сообщений, ей необходимо указать отслеживаемые директории журнала транзакций Lotus Notes.

Информация о настройках директорий Lotus Notes Transaction Log содержится в файле lotustxn.xml, находящемся в директории [INSTALLDIR]\config, и она может быть отредактирована непосредственно в файле конфигурации любым текстовым редактором.

В консоли управления (файл sensor_console.exe) настройка службы EtherSensor LotusTXN происходит следующим образом:

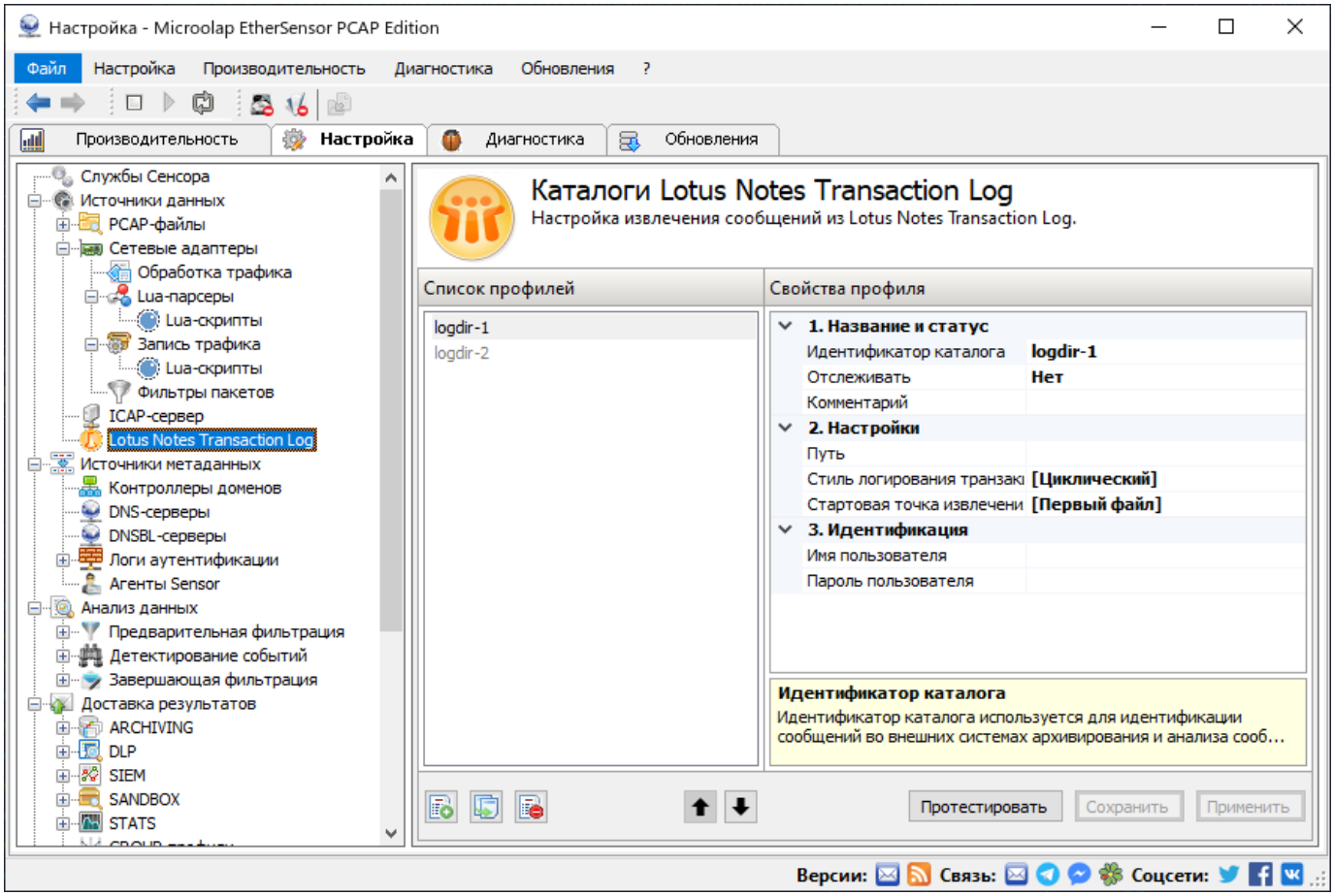

Рис.18. Настройки службы EtherSensor LotusTXN.

Идентификатор каталога:

Идентификатор директории Lotus Notes Transaction Log. Используется для идентификации источника данных в EtherSensor.

Стиль логирования транзакций:

Служит для указания стиля логирования транзакций, который использует система Lotus Notes для данной директории.

Стартовая точка извлечения сообщений:

Стартовая точка извлечения сообщений. Служит для того, чтобы указать EtherSensor, с какого файла в директории Lotus Notes Transaction Log необходимо начинать процесс извлечения сообщений.

Имя пользователя:

Определяет имя пользователя Windows для доступа к директории транзакций. Значение данного параметра должно быть указано в формате UPN (user principal name). Пример: administrator@example.com, где administrator – имя пользователя, example.com – домен пользователя. Если данный параметр не указан, то для доступа к файлам журнала транзакций используются права учётной записи, под которой запущена служба EtherSensor LotusTXN. Следует отметить, что при установке EtherSensor по умолчанию все службы запускаются с правами пользователя SYSTEM.

Пароль пользователя:

Определяет пароль пользователя для доступа к директории транзакций Lotus Notes.

Путь:

Полный путь к директории с журналами транзакций Lotus Notes.

Файл конфигурации EtherSensor LotusTXN

Конфигурация службы EtherSensor LotusTXN указывается в XML-файле lotustxn.xml, расположенном в общей директории конфигураций [INSTALLDIR]\config.

3.5. Служба EtherSensor Identity

Служба EtherSensor Identity отвечает за накопление и обработку метаданных, которые используются в Microolap EtherSensor для решения задачи безагентной привязки перехваченного объекта сетевого трафика к конкретному хосту или пользователю.

В версии 6.1 EtherSensor Identity взаимодействует со следующими источниками метаданных:

Контроллеры доменов [57]

EtherSensor подключается к контроллерам доменов как WMI-клиент и получает данные о пользователях сети, прошедших аутентификацию (для получения данных от контроллера доменов необходима учетная запись с соответствующими правами). Таким образом EtherSensor может видеть активных пользователей корпоративной сети, их IP-адреса и названия компьютеров, а затем использовать эти данные для привязки перехваченного объекта к пользователю или хосту.

> Альтернатива:
>
> Для получения этих данных вместо подключения к контроллеру доменов можно использовать установленный на рабочих станциях/серверах logon script [59], отправляющий данные об аутентификации пользователя на SYSLOG-сервер службы EtherSensor Identity.

DNS-серверы [60]

EtherSensor использует DNS-серверы для связывания IP-адресов перехваченных объектов с именами хостов в сети организации и в сети Интернет.

DNSBL-серверы [62]

EtherSensor использует DNSBL-серверы для детектирования объектов из соединений с небезопасными ресурсами Интернет (как по IP-адресам, так и по DNS-именам).

Логи аутентификации [64]

EtherSensor поддерживает получение сообщений об аутентификации с различных внешних устройств и систем по протоколу SYSLOG. Поддерживаемые транспортные протоколы: UDP, TCP, и TLS over TCP.

Внешними системами, предоставляющими по протоколу SYSLOG данные службе EtherSensor Identity об аутентификации, могут быть:

Межсетевые экраны:

Palo Alto Networks PA, Check Point NGFW и т.д.

Прокси-серверы:

Symantec Proxy SG, Cisco IronPort и т.д.

Рабочие станции/серверы:

Linux, Mac OS и т.д.

Программные агенты:

DLP-агенты, EDR-агенты и т.д.

> Сообщения об аутентификации из разных источников отличаются форматами и наборами данных. Для поддержки конкретного источника требуется дополнительное конфигурирование как самого источника, так и службы EtherSensor Identity.

Агенты Microolap EtherSensor [68]

EtherSensor также может взаимодействовать со своими собственными агентами, установленными на рабочих станциях или серверах. Агенты предоставляют серверу агентов EtherSensor информацию о сетевых соединениях локального пользователя с точностью до имени процесса, участвующего в соединении: данная информация особенно актуальна в случае использования в организации терминальных RDP серверов.

Общая схема работы службы EtherSensor Identity:

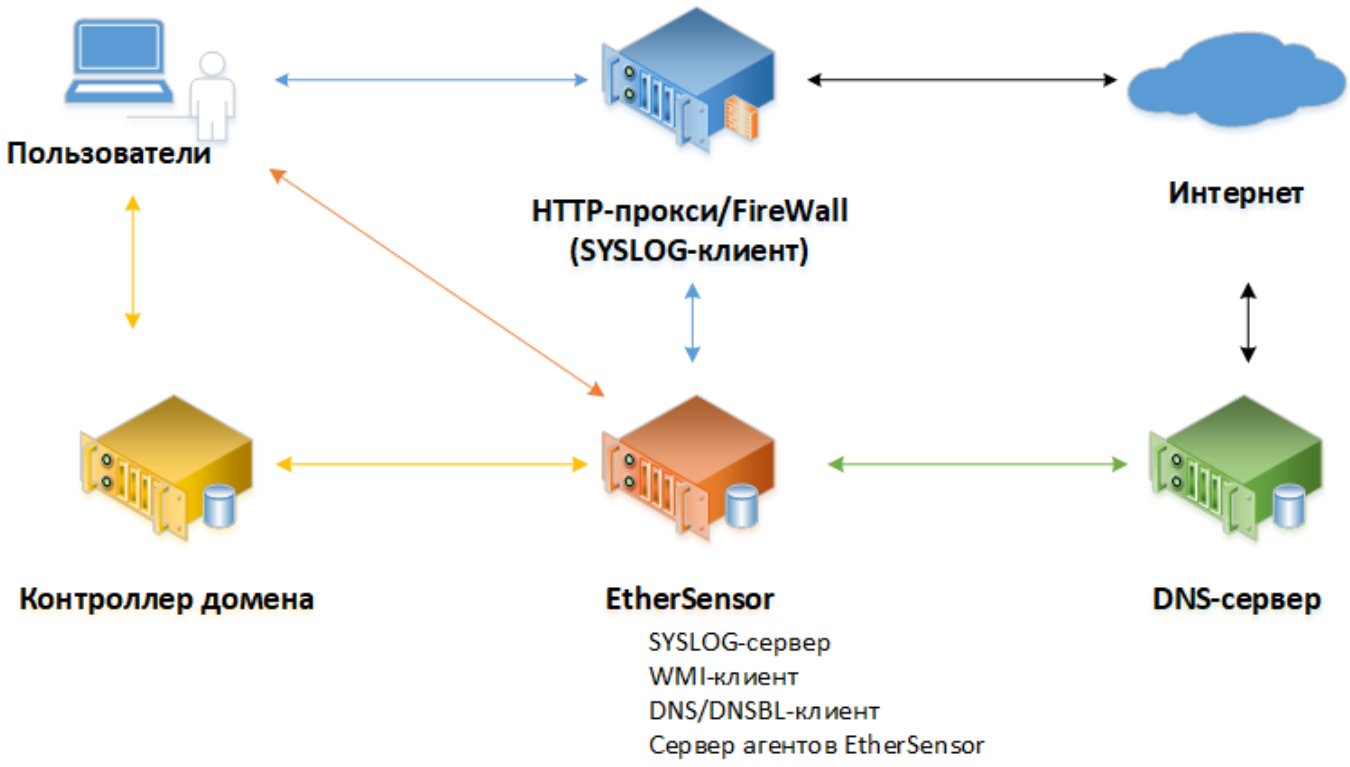

Рис.19. Схема работы службы EtherSensor Identity

Файл конфигурации EtherSensor Identity

Конфигурация службы EtherSensor Identity содержится в XML-файле identity.xml, расположенном в общей директории конфигураций [INSTALLDIR]\config

Параметры командной строки

Служба Windows EtherSensor Identity в ходе инсталляции Microolap EtherSensor устанавливается с автоматическим запуском. Однако, при необходимости процесс sensor_identity.exe можно запустить как приложение Windows со следующими параметрами командной строки:

/process

Запустить процесс sensor_pcap.exe как обычный Windows Win32-процесс (возможно использование для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

3.5.1. Настройка EtherSensor Identity

Настройки службы EtherSensor Identity базируются на понятии заранее заданных профилей. Профили можно активировать по мере возникновения в них надобности, и деактивировать, когда они перестали быть нужными. Удаление неиспользуемых профилей не требуется.

Для того, чтобы подготовить службу EtherSensor Identity к работе, следует настроить профили поставщиков метаданных:

1. Контроллеры доменов [57]
2. DNS-серверы [60]
3. DNSBL-серверы [62]
4. Логи аутентификации [64]
5. Сервер агентов EtherSensor [68]

Кроме того, следует рассмотреть установку на рабочих станциях и серверах logon script [59], отправляющего данные об аутентификации пользователя на сервер агентов EtherSensor [69].

3.5.1.1. Контроллеры доменов

EtherSensor подключается к контроллерам доменов как WMI-клиент и получает данные о пользователях сети, прошедших аутентификацию (для получения данных от контроллера доменов необходима учетная запись с соответствующими правами).

Таким образом EtherSensor может видеть активных пользователей корпоративной сети, их IP-адреса и названия компьютеров, а затем использовать эти данные для привязки перехваченного объекта к пользователю или хосту.

> Альтернатива:
>
> Для получения этих данных вместо подключения к контроллеру доменов можно использовать установленный на рабочих станциях/серверах logon script [59], отправляющий данные об аутентификации пользователя на сервер агентов EtherSensor.

Настройка профиля доступа к контроллеру домена:

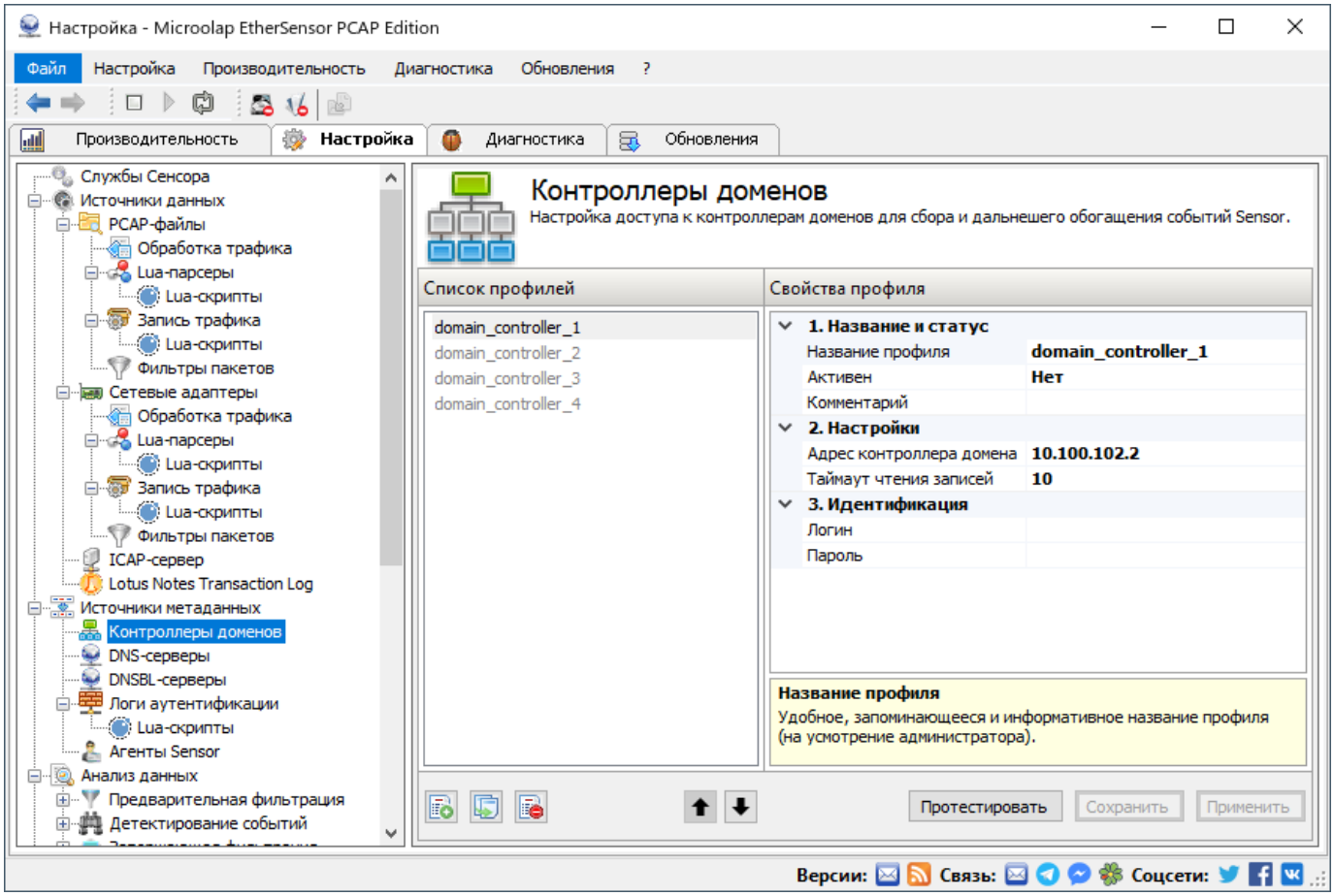

Рис.20. Настройка профиля контроллера домена для службы EtherSensor Identity.

Название профиля

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен

Профиль контроллера домена не используется в анализе сообщений, если он отключен.

Комментарий

Ваш комментарий для данного профиля.

Адрес контроллера домена

IP-адрес или имя контроллера домена для чтения журналов ОС.

Таймаут чтения записей

Таймаут в секундах для чтения записей из журналов ОС.

Логин

Логин на доступ для чтения журналов контроллера домена в формате <domain name>\<user name>.

Пароль

Пароль на доступ для чтения журналов контроллера домена.

3.5.1.1.1. Альтернатива: logon script

Альтернативным способом получения информации об аутентификации пользователя в системе или в домене является использование скрипта, запускающегося по событию входа пользователя в систему.

Пример такого скрипта на языке PowerShell приведен ниже. Данный скрипт сообщает серверу агентов EtherSensor [69] IP-адрес, имя пользователя, имя компьютера и sid пользователя. Эти данные позволяют EtherSensor связать IP-адрес извлечённого из трафика объекта с пользователем или хостом.

> Logon script и его запуск необходимо настроить через средства групповых политик для события входа пользователя в систему.

Пример текста logon script:

```
function GetCUSID
 {
     Param ( $CUIdentity )
     $MyID = new-object System.Security.Principal.NTAccount($CUIdentity)
     return $MyID.Translate([System.Security.Principal.SecurityIdentifier]).toString()
 }

 # EtherSensor address
 $Server = '10.0.1.123&apos

 #0=EMERG, 1=Alert, 2=CRIT, 3=ERR, 4=WARNING, 5=NOTICE, 6=INFO, 7=DEBUG
 $Severity = '5&apos

 #(16-23)=LOCAL0-LOCAL7
 $Facility = '16&apos

 $HostName = $env:ComputerName
 $UserName = $env:UserName
 $UserSID  = GetCUSID([Environment]::UserName)

 # Create a UDP Client Object
 $UDPCLient = New-Object System.Net.Sockets.UdpClient
 $UDPCLient.Connect($Server, 10514)

 # Calculate the priority
 $Priority = ([int]$Facility * 8) + [int]$Severity

 #Time format the SW syslog understands
 $Timestamp = (Get-Date).ToString("MMM dd HH:mm:ss", [CultureInfo]::GetCultureInfo('en-US'))

 # Assemble the full syslog formatted message
 $FullSyslogMessage = "<{0}>{1} computername={2}, username={3}, sid={4}" -f $Priority,
$Timestamp, $HostName, $UserName, $UserSID

 # Create a UTF8 Encoding object
 $Encoding = [System.Text.Encoding]::UTF8

 # Convert into byte array representation
 $ByteSyslogMessage = $Encoding.GetBytes($FullSyslogMessage)

 # Send the Message
 $UDPCLient.Send($ByteSyslogMessage, $ByteSyslogMessage.Length)
```

3.5.1.2. DNS-серверы

EtherSensor использует DNS-серверы для связывания IP-адресов перехваченных объектов с именами хостов в сети организации и в сети Интернет.

При создании правил фильтрации может потребоваться возможность использовать доменные имена, в то время, как в атрибутах перехваченных объектов доступны только IP-адреса.

Для того, чтобы можно было в процессе фильтрации использовать такую возможность в реальном времени (стоит также учитывать динамическое назначение IP-адресов по DHCP), в секции ***DNS*** определите и настройте доступные DNS-серверы.

Пример настройки профиля доступа к DNS-серверу:

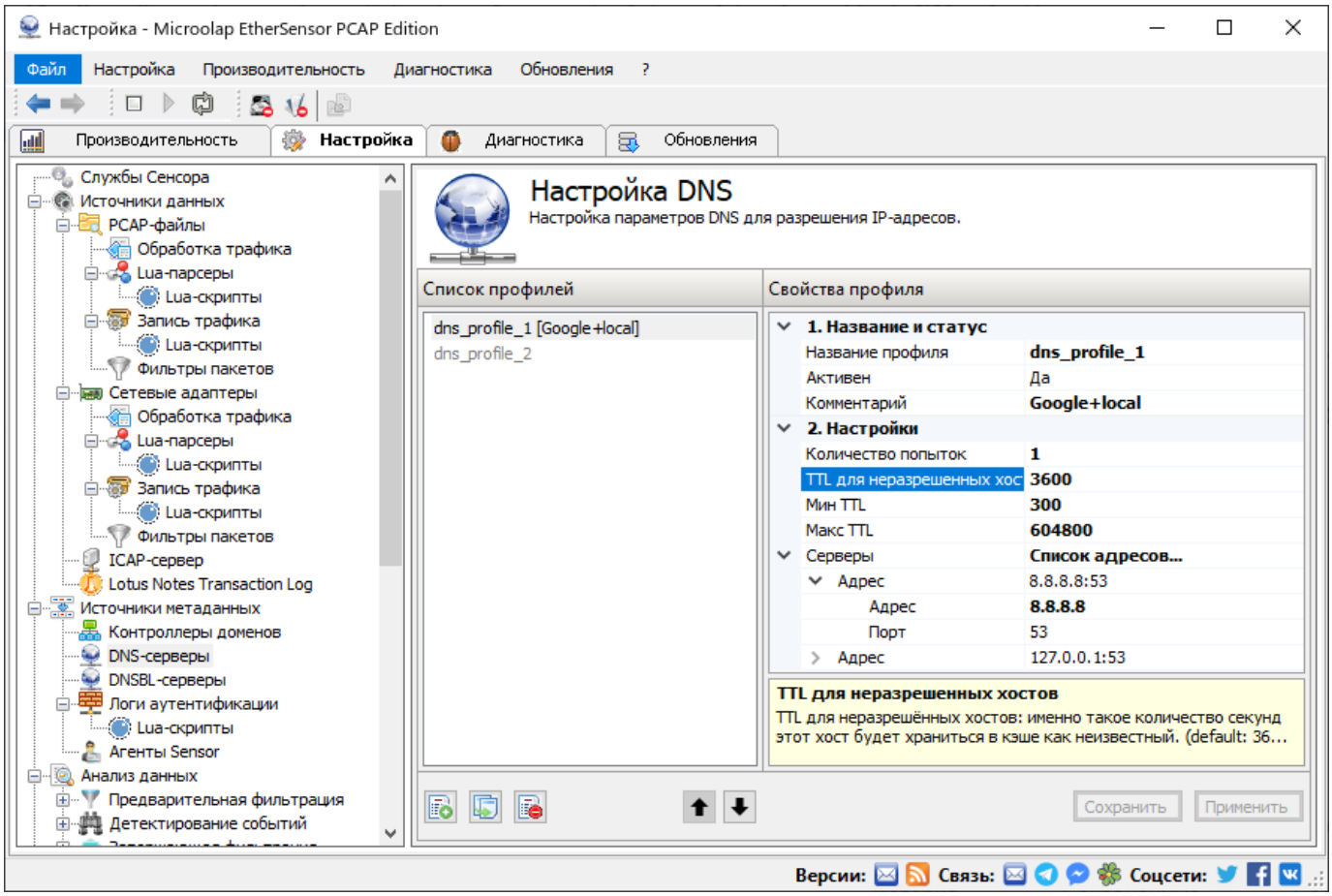

Рис.21. Настройка DNS-профиля для службы EtherSensor Identity.

Название профиля

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен

Профиль DNS-серверов не используется в анализе сообщений, если он отключен.

Комментарий

Ваш комментарий для данного профиля.

Количество попыток

Количество попыток опроса каждого сервера.

TTL для неразрешенных хостов

TTL для неразрешённых хостов: именно такое количество секунд этот хост будет храниться в кэше как неизвестный. (default: 3600 = 1 час) (43200 = 12 часов, 86400 = 24 часа, 172800 = 48 часов).

Мин TTL

Минимальное TTL хранения записи в кэше для имен хостов, полученных от DNS-серверов.

Макс TTL

Максимальное TTL хранения записи в кэше для имён хостов, полученных от DNS-серверов.

Серверы

Список DNS-серверов.

Адрес

Адрес DNS-сервера

Порт

Порт DNS-сервера

3.5.1.3. DNSBL-серверы

В процессе фильтрации сообщений также может потребоваться категоризация сообщений. Для этого в EtherSensor есть функция взаимодействия с DNSBL-серверами для детектирования спам-сообщений в SMTP-потоке. Настройка этой функции полностью идентична настройке работы с DNS-серверами.

EtherSensor использует DNSBL-серверы для детектирования объектов из соединений с небезопасными ресурсами Интернет (как по IP-адресам, так и по DNS-именам).

Пример настройки профиля доступа к DNSBL-серверу:

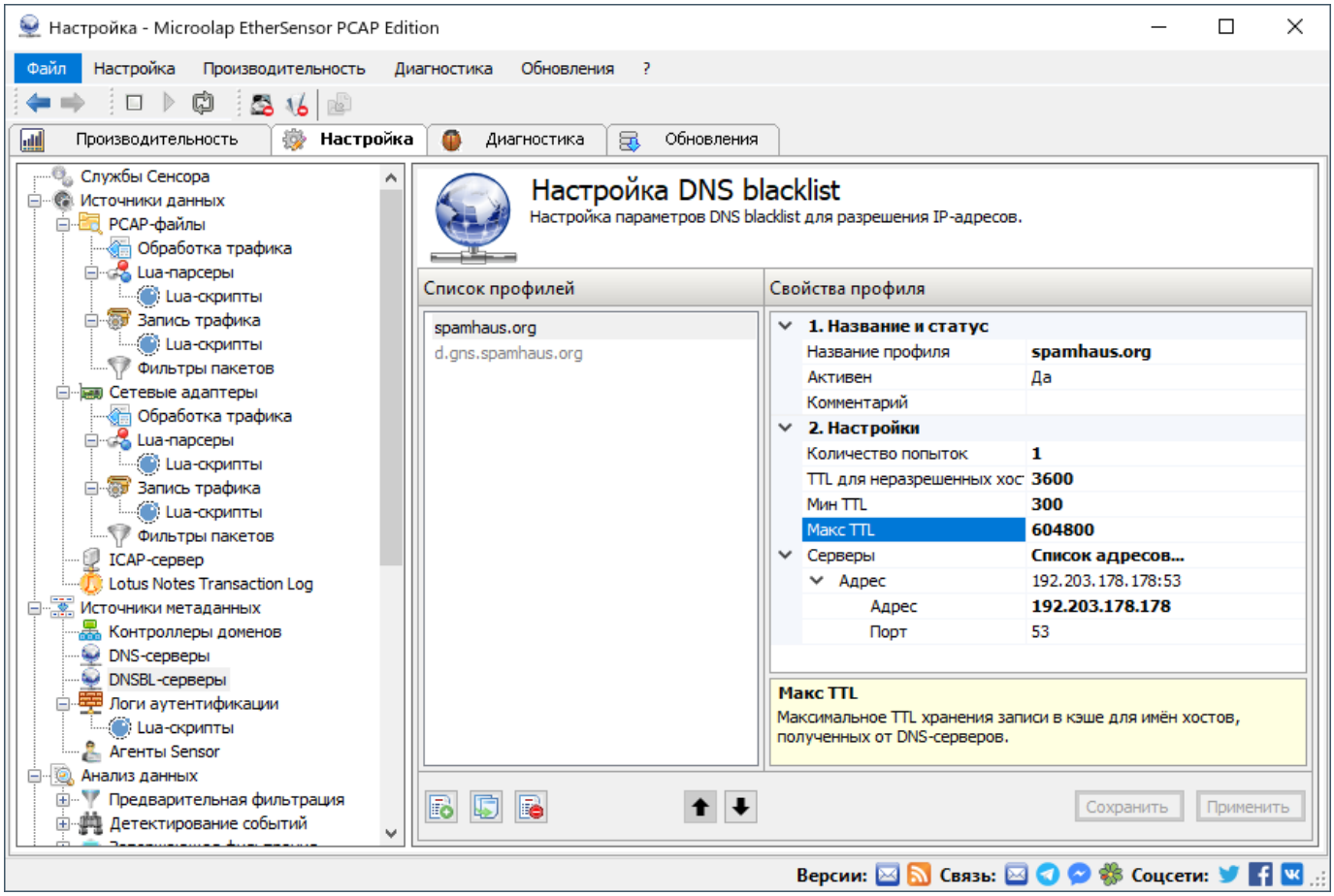

Рис.22. Настройка DNSBL-профиля для службы EtherSensor Identity.

Название профиля

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен

Профиль DNSBL-серверов не используется в анализе сообщений, если он отключен.

Комментарий

Ваш комментарий для данного профиля.

Количество попыток

Количество попыток опроса каждого сервера.

TTL для неразрешенных хостов

TTL для неразрешённых хостов: именно такое количество секунд этот хост будет храниться в кэше как неизвестный. (default: 3600 = 1 час) (43200 = 12 часов, 86400 = 24 часа, 172800 = 48 часов).

Мин TTL

Минимальное TTL хранения записи в кэше для имен хостов, полученных от DNSBL-серверов.

Макс TTL

Максимальное TTL хранения записи в кэше для имён хостов, полученных от DNSBL-серверов.

Серверы

Список DNSBL-серверов.

Адрес

Адрес DNSBL-сервера

Порт

Порт DNSBL-сервера

3.5.1.4. Логи аутентификации

Служба EtherSensor Identity включает в себя SYSLOG-сервер для получения записей логов аутентификации от различных SYSLOG-клиентов: межсетевых экранов, прокси-серверов, программных DLP/EDR-агентов и т.п.

Согласно профилям настройки SYSLOG-сервер может прослушивать различные комбинации IP-адрес:порт, либо 0.0.0.0:порт – в этом случае он будет прослушивать все IP-адреса на заданном порту.

Пример настройки профиля логов аутентификации:

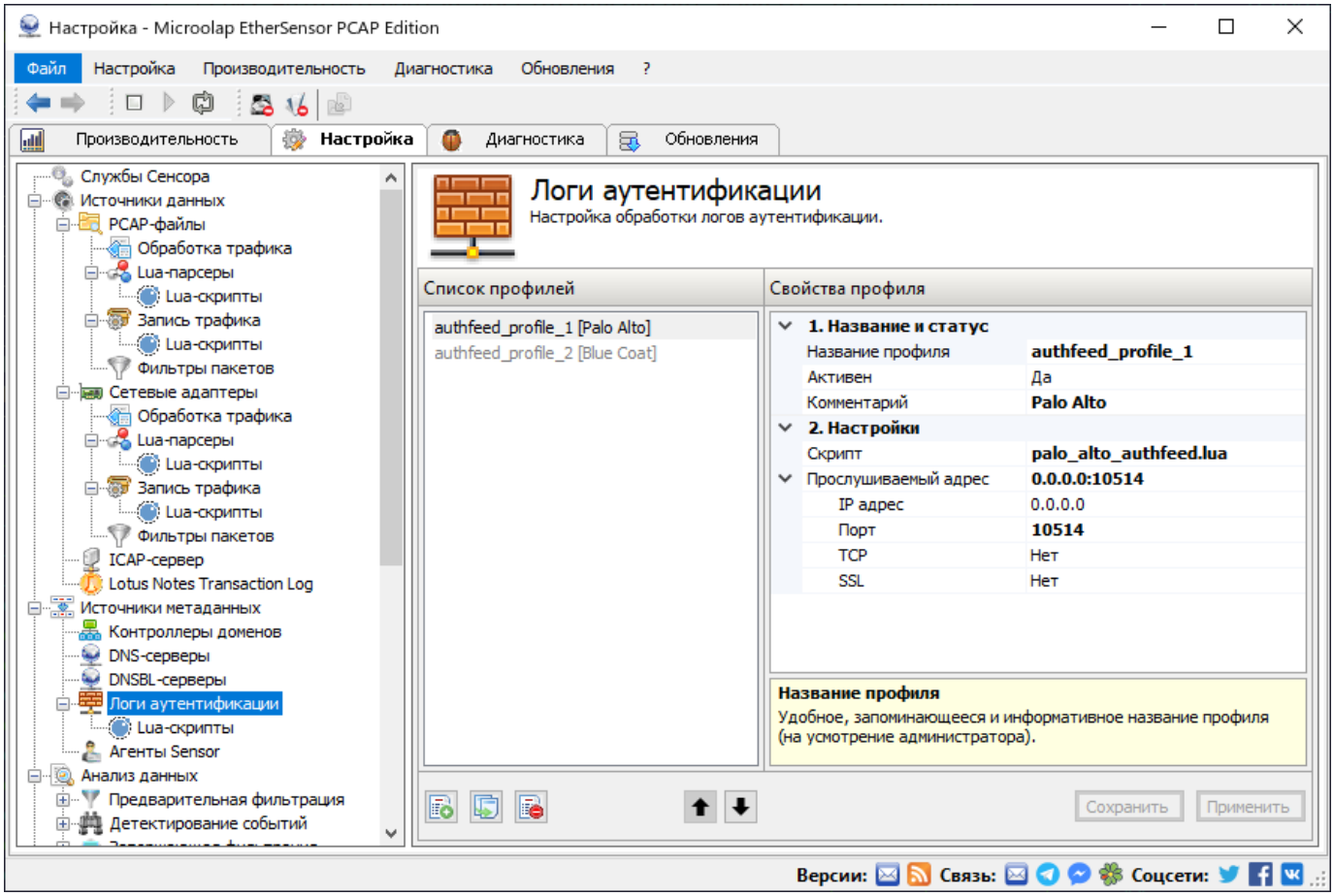

Рис.23. Настройка профиля логов аутентификации для службы EtherSensor Identity.

Название профиля

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен

Профиль обработки логов аутентификации не используется в анализе сообщений, если он отключен.

Комментарий

Ваш комментарий для данного профиля.

Скрипт

Привязанный к профилю скрипт, который выполняется для обработки SYSLOG-сообщения логов аутентификации.

Прослушиваемый адрес

Настройка локального адреса для прослушивания логов аутентификации.

IP адрес

Настройка локального IP адреса для прослушивания логов аутентификации.

Порт

Настройка локального порта для прослушивания логов аутентификации.

TCP

Позволяет использовать TCP-протокол для получения SYSLOG сообщений. Необходимо, если используется SSL-шифрование.

SSL

Включить/выключить использование SSL-шифрования при приёме сообщений.

Если в организации нет возможности обеспечить серверу EtherSensor доступ к контроллерам домена или к Log Collector, существует ещё один способ получения данных для привязки доменных пользователей к извлечённым из трафика объектам.

Установите на контроллеры домена или Log Collector бесплатную утилиту nxlog и настройте в ней отправку событий Security Log на сервер EtherSensor.

В данном случае данные отправляются на сервер EtherSensor по протоколу SYSLOG (***TCP***).

Ниже приведен пример конфигурационного файла nxlog, отправляющего события Kerberos Ticket Authentication (EventID 4768) на сервер EtherSensor (в примере это IP 10.100.0.100).

```
## This is a sample NXLog configuration file created by Loggly. June 2013
## See the nxlog reference manual about the configuration options.
## It should be installed locally and is also available
## online at http://nxlog.org/nxlog-docs/en/nxlog-reference-manual.html

## Please set the ROOT to the folder your nxlog was installed into,
## otherwise it will not start.
#define ROOT C:\\Program Files\\nxlog
#define ROOT_STRING C:\\Program Files\\nxlog
define ROOT C:\\Program Files (x86)\\nxlog
define ROOT_STRING C:\\Program Files (x86)\\nxlog
define CERTDIR %ROOT%\\cert

Moduledir %ROOT%\\modules
CacheDir %ROOT%\\data
Pidfile %ROOT%\\data\\nxlog.pid
SpoolDir %ROOT%\\data
LogFile %ROOT%\\data\\nxlog.log

<Extension json>
 Module xm_json
</Extension>
<Extension syslog>
 Module xm_syslog
</Extension>
<Input internal>
 Module im_internal
 Exec $Message = to_json()
</Input>
# Windows Event Log
<Input eventlog>
# Uncomment im_msvistalog for Windows Vista/2008 and later
 Module im_msvistalog
#Uncomment im_mseventlog for Windows XP/2000/2003
#Module im_mseventlog
#Send only EventID 4768 (Get Kerberos Ticket)
 Exec if $EventID NOT IN (4768) drop()
 Exec $Message = to_json()
</Input>
<Processor buffer>
Module pm_buffer
# 100Mb disk buffer
MaxSize 102400
Type disk
</Processor>
<Output out_ethersensor>
 Module om_tcp
 Host 10.100.0.100
 Port 516
 Exec to_syslog_ietf()
 Exec $raw_event =~ s/(\[.*])//g; $raw_event = replace($raw_event, '{', '[CUSTOMER_TOKEN@41058
tag="windows"] {', 1)
#Use the following line for debugging (uncomment the fileop extension above as well)
#Exec file_write("C:\\Program Files (x86)\\nxlog\\data\\nxlog_output.log", $raw_event)
</Output>
<Route 1>
 Path internal, eventlog => buffer => out_ethersensor
</Route>
```

В консоли управления сервером EtherSensor в разделе ***Настройка -- Источники метаданных -- Логи аутентификации*** сделайте активным профиль ***authfeed_nxlog***. Для этого установите в нём флаг ***Активен*** и убедитесь, что указаны корректные ***Порт*** и протокол (***TCP***).

3.5.1.4.1. Lua-скрипты

Функция разбора SYSLOG-сообщений с помощью Lua-скриптов, назначаемых в профилях логов аутентификации, находится в стадии пре-релиза.

Если вы хотите поэкспериментировать вместе с нами, напишите нам на support@microolap.ru.

3.5.1.5. Сервер агентов EtherSensor

Служба EtherSensor Identity включает в себя сервер агентов EtherSensor, взаимодействующий с агентами [69], устанавливаемыми на рабочих станциях и серверах.

Пример настройки сервера агентов EtherSensor:

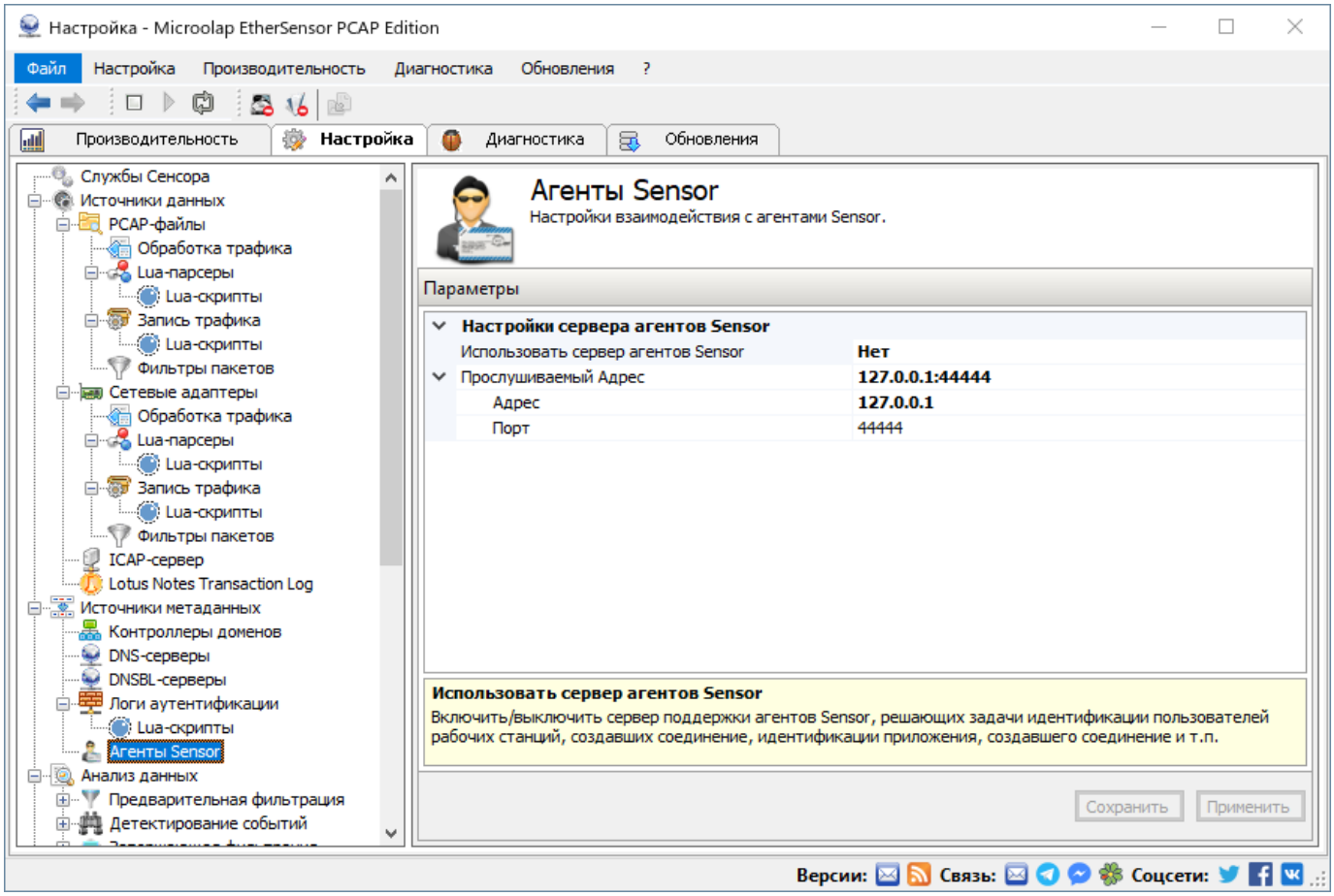

Рис.24. Настройка сервера агентов для службы EtherSensor Identity.

Использовать сервер агентов EtherSensor

Включить/выключить сервер поддержки агентов EtherSensor, решающих задачи идентификации пользователей рабочих станций, создавших соединение, идентификации приложения, создавшего соединение и т.п.

Прослушиваемый Адрес

Используйте данный параметр, чтобы указать серверу локальный адрес для прослушивания сообщений агентов EtherSensor.

Адрес

Используйте данный параметр, чтобы указать серверу локальный IP адрес для прослушивания сообщений агентов EtherSensor.

Порт

Используйте данный параметр, чтобы указать серверу локальный порт для прослушивания сообщений агентов EtherSensor.

3.6. EtherSensor Agent

EtherSensor Agent – служба Windows, устанавливаемая на рабочие станции. EtherSensor Agent решает две задачи:

- На сервер EtherSensor Агент по UDP-протоколу передаёт информацию [77] о процессе, который создаёт внешние TCP-соединения. Это позволяет решить задачу привязки TCP-сессии к конкретной рабочей станции в случае работы пользователей в терминальной сессии, за NAT и других аналогичных случаях.

- Агент по TCP-протоколу передает информацию о событиях [74] на рабочей станции на сервер EtherStat на указанный в конфигурации его адрес и порт. При временной невозможности отправки данных (отсутствие соединения и т.п.) происходит их накапливание в локальной базе данных. Как только Агент обнаружит возможность отправить данные, он пакует пакеты по 64КБ и отправляет на сервер EtherStat.

Агенты предоставляют наиболее полную и точную информацию для EtherSensor. Особую ценность представляет имя процесса, участвующего в соединении.

Если по какой-либо причине вы не можете использовать агентов на рабочих станциях/серверах, рассмотрите возможность использования скрипта на событие входа пользователя в домен или систему (logon script [59]).

3.6.1. Системные требования к Агенту

EtherSensor Agent функционирует на рабочих станциях, удовлетворяющих следующим системным требованиям:

- ОС (Windows 7, Windows 2008, Windows 8, Windows 8.1, Windows 2012, Windows 10) 32/64 бита.
- Размер необходимого свободного пространства на жестком диске не менее 10 МБ.
- Должны быть выполнены требования по работе со сторонними средствами защиты информации.

Работа со сторонними средствами защиты информации

Для обеспечения стабильной работы EtherSensor Agent следует учитывать совместимость со сторонними средствами защиты информации и другими компонентами инфраструктуры:

- В директории, где установлен EtherSensor Agent, расположены рабочие директории. Следует исключать данный путь и все его поддиректории от контроля средств, аналогичных антивирусам, поисковым индексаторам, а также средствам контроля изменения файлов. Подобное программное обеспечение не должно блокировать файлы в данной директории и ее поддиректориях от создания, удаления, перемещения и изменения.
- EtherSensor Agent содержит службу sensor_agent.exe, которая должна быть разрешена к запуску и работать с правами локальной системы.
- Служба EtherSensor Agent требует для нормальной работы возможность обмена информацией по протоколам UDP и TCP с удаленным сервером. Средства защиты информации не должны проверять, модифицировать или ограничивать соединения на сервер, на который EtherSensor Agent отправляет данные. Аналогично, средства защиты информации не должны препятствовать службе EtherSensor Agent открывать соединения на порты, используемые для передачи информации на сервер EtherSensor.
- EtherSensor Agent требует для своей работы регистрации Layered Service Provider модуля sensor_lsp.dll. Сторонние средства защиты информации не должны препятствовать регистрации sensor_lsp.dll и его работе.
- При установке и функционировании процессы EtherSensor Agent используют вызовы, требующие высоких привилегий. Политика безопасности ОС должна позволять операции над драйверами, управление процессами и доступ к сетевым интерфейсам для EtherSensor Agent.

3.6.2. Установка Агента

Установку EtherSensor Agent можно производить как в ручном режиме на каждой рабочей станции, так и при помощи групповых политик Active Directory (GPO).

Для установки Агента в ручном режиме необходимо запустить MSI-инсталлятор (32 или 64 bit), входящий в состав дистрибутива.

> Для корректной установки EtherSensor Agent необходимы права администратора в том виде, в каком они установлены по умолчанию при инсталляции ОС Windows.
>
> Дополнительные ограничения прав администратора могут привести к некорректной работе.

В процессе установки Агента будет установлена служба EtherSensor Agent (процесс sensor_agent.exe), а также в системе будет зарегистрирован Layered Service Provider – модуль sensor_lsp.dll.

По умолчанию установка Агента выполняется в директорию [[INSTALLDIR]] Agent.

> Для корректной работы установленных экземпляров EtherSensor Agent необходимо настроить DNS-сервер сети организации таким образом, чтобы имя сервера, указанное в настройках EtherSensor Agent, соответствовало IP-адресу сервера EtherSensor.

Установку также можно произвести через утилиту Windows Installer msiexec.exe в командной строке с правами администратора, указав следующие параметры:

INSTALLDIR:

Путь к директории, куда будет установлен EtherSensor Agent.

ETHERSTATSERVER:

IP адрес и порт сервера, на котором находится EtherStat в формате "адрес:порт".

KEY:

ключ протокола ZeroMQ для безопасного соединения с сервером EtherStat, должен содержать 40 символов.

ETHERSENSORSERVER:

IP адрес и порт сервера, на котором находится EtherSensor в формате "адрес:порт".

Пример командной строки, с помощью которой будет установлен EtherSensor Agent:

```
msiexec /i [path to EtherSensor Agent MSI package] INSTALLDIR="[[INSTALLDIR]] Agent"
ETHERSTATSERVER="etherstat:44445" KEY="0123456789ABCDEFGabcdefg!@#$%^&*<>?-+=^"
ETHERSENSORSERVER="ethersens:44444"
```

3.6.3. Состав файлов Агента

Файлы, входящие в пакет установки EtherSensor Agent:

[INSTALLDIR]\config\agent.xml

Текстовый файл конфигурации в формате XML для назначения параметров соединения с серверами EtherStat и EtherSensor, периода опроса данных на рабочей станции, а также фильтра приложений, для которых данные о TCP-соединениях не следует отправлять на сервер EtherSensor.

[INSTALLDIR]\syslog.dll

Библиотека, необходимая для создания и ведения всех *.log файлов EtherSensor Agent.

[INSTALLDIR]\sensor_agent.exe

Исполняемый файл EtherSensor Agent, реализует основные функции Агента. При запуске сервиса необходимо указать один из следующих параметров командной строки:

/service

Запуск Агента в режиме службы Windows.

/process

Запуск Агента в режиме процесса Windows.

/install

Установка службы Агента в ОС.

/remove

Удаление службы Агента из ОС.

Если ни один из параметров не был указан, то в файле [INSTALLDIR]\log\svcagent.log будет сделана запись о некорректной командной строке с соответствующей подсказкой, и сервис прекратит свою работу.

Файлы, генерируемые в процессе работы Агента:

[INSTALLDIR]\log\svcagent.log

Текстовый файл в формате XML, в который ведется логирование основных действий и ошибок EtherSensor Agent.

[INSTALLDIR]\log\sensor_agent.exe.log

Текстовый файл для записи информации о событиях, генерируемых внутри подсистемы логирования EtherSensor Agent.

[INSTALLDIR]\log\processinfo.log

Текстовый файл, содержащий информацию о текущих процессах, соединения которых отслеживает EtherSensor Agent.

[INSTALLDIR]\data.db

Файл базы данных сообщений о событиях, которые не удалось отправить по TCP-соединению на сервер EtherStat. При установке соединения с сервером данные будут повторно отправлены, и в случае успешной доставки удалены из базы.

3.6.4. Логические модули Агента

Модуль отслеживания соединений и маркировки HTTP соединений (sensor_lsp.dll).

Данный модуль выполнен как Layered Service Provider. Это означает, что при установке модуля он встраивается в сетевой стек приложений и прозрачно проксирует (при этом отслеживая) все TCP-соединения, создаваемые локальными процессами. Модуль имеет настройки, хранящиеся в реестре Windows:

- UDP порт службы EtherSensor EtherCAP. По умолчанию используется порт 44444.
- Флаг маркировки HTTP-трафика, значение по умолчанию 1. Если флаг выставлен в состояние 1, то каждый HTTP-запрос помечается заголовком вида X-Sensor-UID: 554E4B4E-4F57-4E20-5555-494400000000, где 554E4B4E-4F57-4E20-5555-494400000000 – уникальный идентификатор пользователя, привязанный к конкретной машине, к конкретному пользователю на данной машине, и являющийся глобальным в контексте сети организации.

В процессе своей работы модуль sensor_lsp.dll локально коммуницирует по протоколу UDP со вторым основным модулем EtherSensor Agent – службой EtherSensor Agent (процесс sensor_agent.exe), – и передает ей информацию о процессах, создающих TCP-соединения.

Модуль взаимодействия с сервером EtherSensor (служба EtherSensor Agent).

Данный модуль выполнен как системная служба Windows, в которой реализованы следующие функции:

- Сбор и передача данных о событиях рабочей станции [74] по шифрованному TCP-соединению на сервер мониторинга и статистики EtherStat. Если данные не удалось

отправить, то служба EtherSensor Agent помещает их в локальную базу данных [INSTALLDIR]\data.db.

- Передача по протоколу UDP серверу EtherSensor данных о TCP-соединениях процессов рабочей станции (77). Настройки позволяют учесть список процессов, соединения которых отслеживать не требуется.

- Логирование событий службы Агента и запись их в файл [INSTALLDIR]\log\svcagent.log

Данные о конфигурации модуль sensor_agent.exe берет из [INSTALLDIR]\config\agent.xml. При изменении настроек необходимо произвести перезапуск службы EtherSensor Agent.

3.6.5. Данные, передаваемые на EtherStat

Агент передает данные на сервер EtherStat в двух случаях:

1. Периодически, в соответствии с конфигурацией.

2. При возникновения события, данные о котором EtherSensor Agent должен передать в соответствии с конфигурацией.

Периодически отправляемые данные

Список оборудования
Посылается на сервер EtherStat только при старте службы EtherSensor Agent, далее посылаются только изменения (по событию изменения). Список содержит структуры данных, описывающие установленные на рабочей станции устройства. Данные об устройствах извлекаются их Свойств, предоставленных Диспетчером устройств Windows с помощью соответствующих функций WinAPI, и имеют вид:

- Наименование устройства
- Описание устройства
- Наименование производителя
- Соответствующий Device Id
- GUID Class устройства в формате {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
- Список Hardware Ids устройства
- Наименование службы, с которой взаимодействует устройство.

Список установленных приложений и служб
Посылается на сервер EtherStat только при старте службы EtherSensor Agent, далее посылаются только данные об изменениях. Список содержит структуры данных, описывающие установленное на рабочей станции программное обеспечение. Все данные об установленном программном обеспечении извлекаются из реестра Windows с помощью WinAPI и имеют вид:

- Наименование продукта
- Наименование производителя
- Текущая версия продукта
- Путь инсталляции продукта.

Ветки реестра, из которых берется информация:

- HKML\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
- HKML\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall
- HKEY_USERS\<Просмотр в каждой вкладке>\Software\Microsoft\Windows\CurrentVersion\Uninstall.

Информация об ОС
Посылается на сервер EtherStat в соответствии с тегом конфигурации OSMonitor. Это – данные об ОС Windows на рабочей станции, извлечённые с помощью WinAPI:

- Наименование, текущая версия, тип и состояние операционной системы
- Серийный номер в формате XXXXX-XXXXX-XXXXX-XXXXX
- Тип архитектуры (x86 или x64)
- Имя компьютера в системе и домене
- Раздел физического диска и директория операционной системы
- Дата и время последней перезагрузки системы и дата последнего обновления системы (если обновлений не было, то дата установки ОС). Также посылается текущее время на рабочей станции
- Модель и наименование производителя материнской платы
- Количество запущенных физических и логических процессов
- Размер файла подкачки операционной системы.

Информация о сетевых адаптерах
Посылается на сервер EtherStat в соответствии с тегом конфигурации NETMonitor. Для каждого адаптера создается отдельное сообщение с его описанием и настройками. Данные по сетевому адаптеру и его конфигурации получаются с помощью функций WinAPI и содержат:

- Наименование адаптера
- Наименование производителя
- MAC-адрес, IP-адреса, маска подсети, настройки DNS-адресов и т.д.
- Наименование компьютера в в сети и домене
- Флаг использования DHCP

- GUID сетевого адаптера
- Максимальная скорость передачи данных в сетевом адаптере в битах в секунду.

Данные по текущей загрузке компьютера
Извлекаются с помощью WinAPI и содержат:

- Текущую нагрузку на CPU в процентах
- Текущее использование RAM в процентах
- Текущее заполнение HDD в процентах
- Свободное место на HDD.

Данные, отправляемые по событию

Изменение списка установленного программного обеспечения
Посылается на сервер EtherStat при обнаружении новых или удалении уже установленных приложений на рабочей станции. Все данные об установленном или удаленном программном обеспечении извлекаются из реестра Windows с помощью WinAPI и содержат:

- Наименование продукта
- Наименование производителя
- Текущая версия продукта
- Путь инсталляции продукта.

Изменение списка установленного оборудования
Посылается на сервер EtherStat при обнаружении нового или удаленного устройства на рабочей станции. Все данные по устройствам извлекаются из их Свойств, полученных из Диспетчера устройств Windows с помощью соответствующих функций WinAPI и содержат:

- Наименование устройства
- Описание устройства
- Наименование производителя
- Соответствующий Device Id
- GUID Class устройства в формате {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
- Список Hardware Ids устройства
- Наименование службы, с которой взаимодействует устройство.

Запуск и завершение процесса с привязкой к TCP-сессии
Посылается на сервер EtherStat при обнаружении нового процесса или его завершении. При обнаружении нового процесса будет отправлена структура данных, описывающая процесс и содержащая:

- Название процесса
- Командная строка с аргументами запуска процесса
- Путь к директории запущенного процесса
- Время использования пользователем
- Идентификаторы ProcessID, SessionID и ParentID.

При завершении процесса отправляются идентификаторы ProcessID и SessionID порождённой процессом сессии.

Данные, отправляемые на сервер при совершении пользователем действий в системе

При входе пользователя в систему:

- Наименование домена, в который входит пользователь
- Наименование учетной записи пользователя
- SID – уникальный идентификатор пользователя в ОС
- SessionID – номер сессии на компьютере
- Наименование способа работы пользователя с рабочей станцией: console или rdp
- Дата и время входа пользователя в систему.

При выходе пользователя из системы:

- SID – уникальный идентификатор пользователя в ОС
- SessionID – номер сессии на компьютере
- Дата и время выхода пользователя.

При блокировке или разблокировке учетной записи пользователя:

- SID – уникальный идентификатор пользователя в ОС
- SessionID – номер сессии на компьютере.

При изменении активного окна

- Текущий заголовок окна
- Идентификатор процесса, который владеет этим окном.

3.6.6. Данные, передаваемые на EtherSensor

EtherSensor Agent доставляет полученную информацию на сервер EtherSensor по протоколу UDP.

Агент отправляет на сервер EtherSensor информацию о процессе, который создаёт внешние TCP-соединения:

- Идентификатор процесса
- Имя процесса
- Имя пользователя, под которым выполняется процесс
- Имя компьютера
- Идентификатор пользователя.

Также Агент отправляет информацию о самом TCP-соединении:

- Идентификатор процесса
- Идентификатор пользователя
- Детали соединения.

Основываясь на данных, переданных Агентом, сервер EtherSensor будет помечать реконструированные объекты трафика в зависимости от настроек следующими свойствами:

- Имя пользователя, под которым выполняется процесс
- Имя компьютера
- Идентификатор пользователя в формате X-Sensor-UID: 554E4B4E-4F57-4E20-5555-494400000000.

3.6.7. Работа с Агентом

Перед стартом EtherSensor Agent сделайте следующее:

1. Произведите необходимые настройки конфигурации в файле [INSTALLDIR]\config\agent.xml с помощью любого текстового редактора.
2. Запустите cmd.exe с правами Администратора.
3. Остановите службу EtherSensor Agent командой sc stop "EtherSensor Agent".

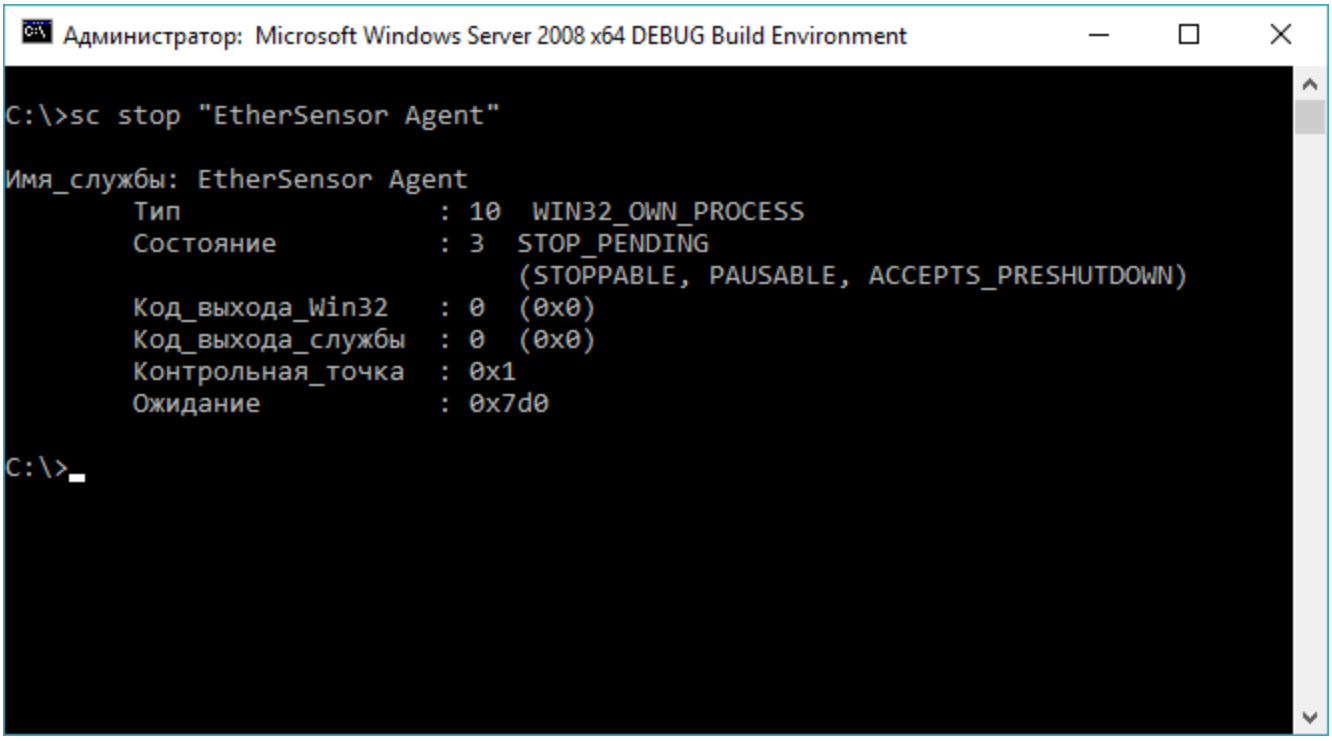

Рис.25. Остановка службы "EtherSensor Agent".

4. Запустите службу EtherSensor Agent командой sc start "EtherSensor Agent".

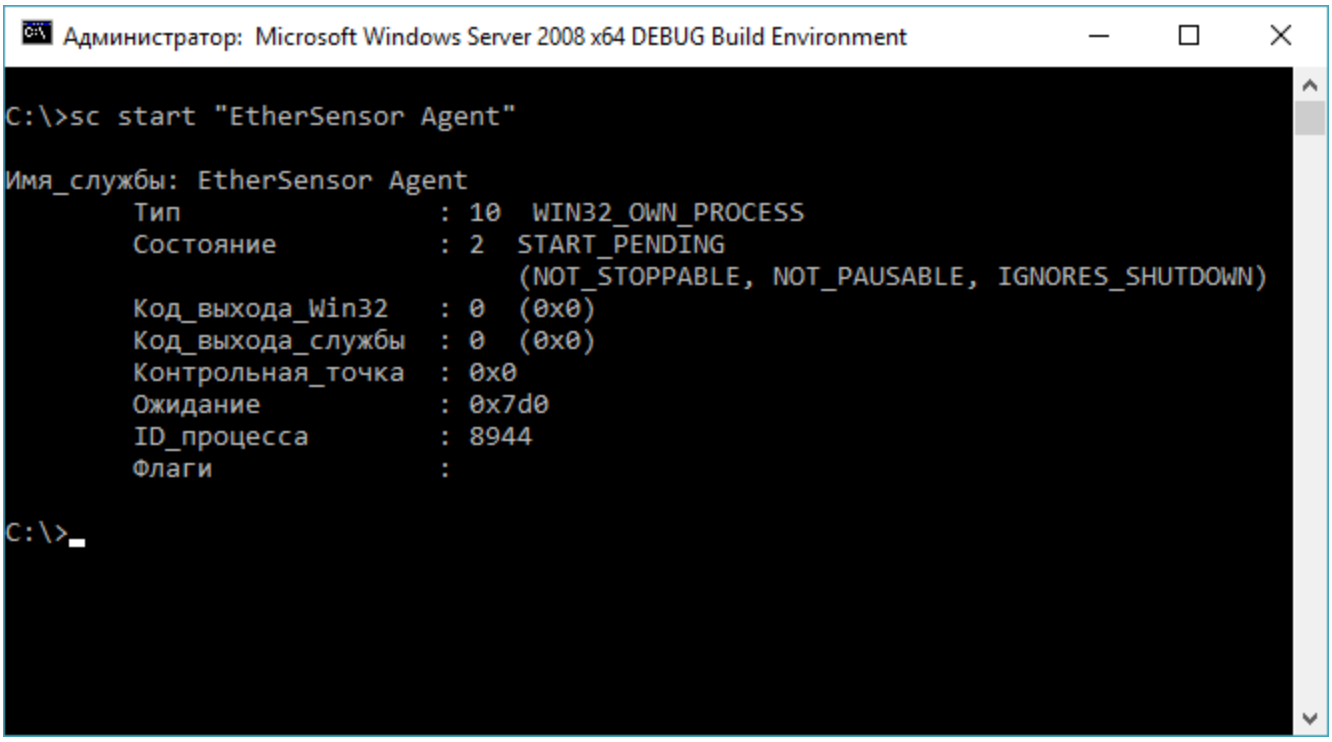

Рис.26. Запуск службы "EtherSensor Agent"

5. Убедитесь, что служба запустилась, набрав команду sc query "EtherSensor Agent". Состояние службы должно быть RUNNING.

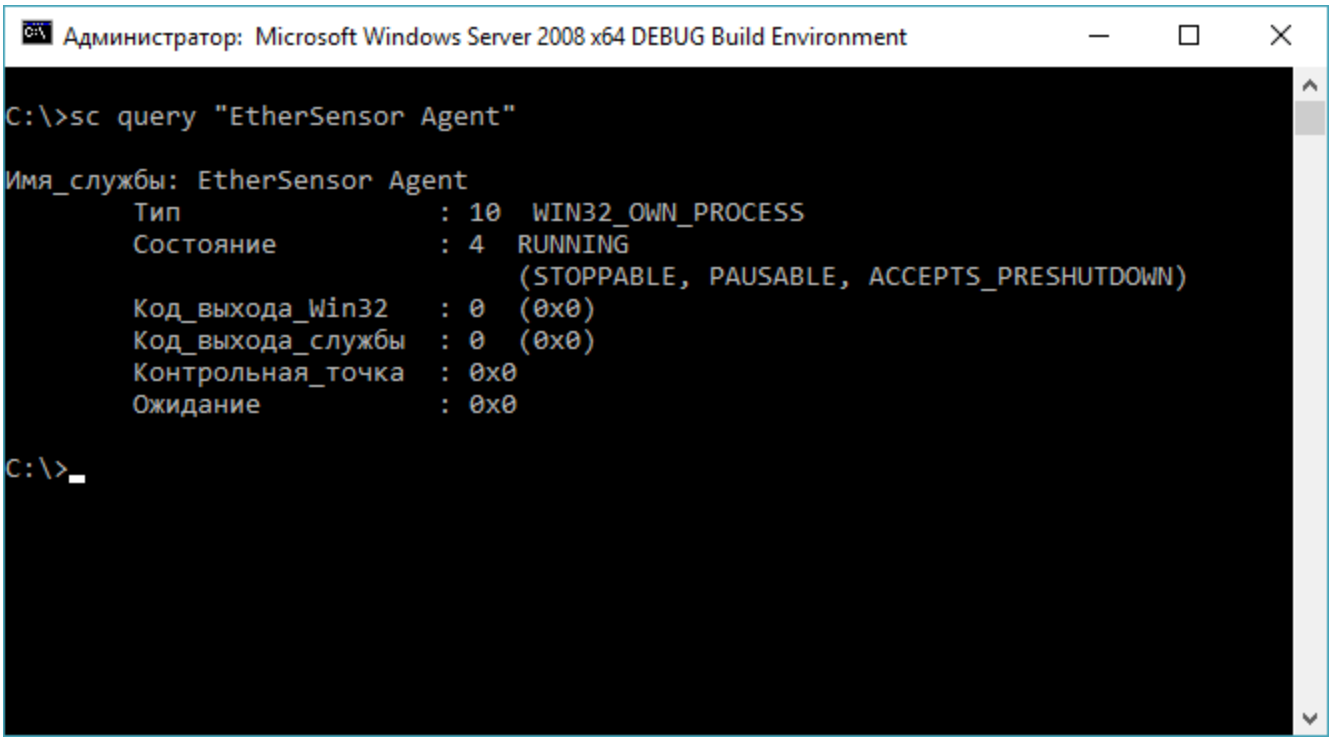

Рис.27. Проверка работы службы "EtherSensor Agent"

3.6.7.1. Возможные варианты работы Агента

Работа Агента с сервером EtherStat

Для работы Агента с сервером EtherStat необходимо, чтобы рабочие станции, на которых установлен EtherSensor Agent, находились в одной с сервером EtherStat локальной сети. EtherStat с помощью шифрованного TCP-соединения собирает информацию от агентов, установленных на рабочих станциях, и анализирует ее. Для идентификации рабочих станций Агент генерирует уникальный UHID и отправляет его в сообщениях серверу EtherStat.

Работа Агента с сервером EtherSensor

Режим прозрачного проксирования без маркировки трафика

В данном режиме Агент прозрачно проксирует соединения приложений, запущенных на компьютере пользователя. При успешной установке приложением соединения Агент отправляет

на сервер EtherSensor информацию об установлении TCP соединения конкретным приложением, выполняющимся под конкретным пользователем сети.

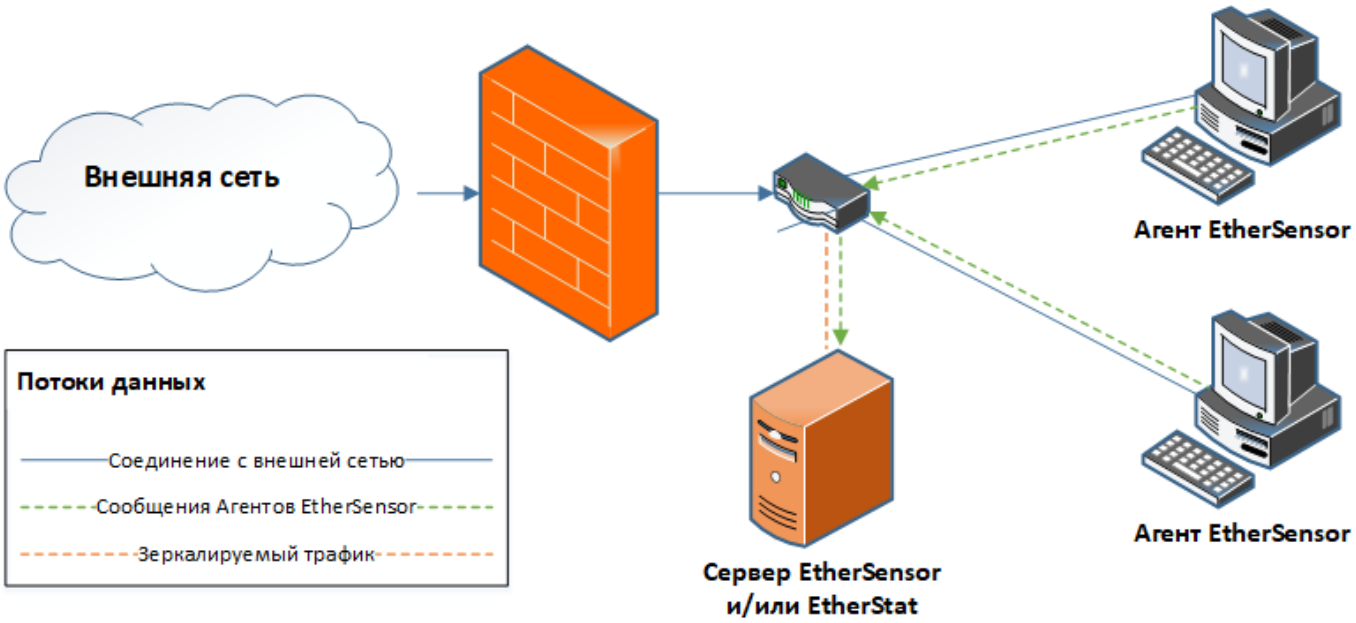

Рис.28. Прозрачное проксирование без маркировки трафика

Таким образом EtherSensor при реконструкции сообщения может полностью идентифицировать пользователя, отправившего данное сообщение посредством любого поддерживаемого на данный момент протокола (ICQ, MSN, MRA, IRC, XMPP, SMTP, POP3, LOTUS, HTTP, FTP и т.п.).

При этом обязательно должно выполняться условие отвода копии анализируемого трафика: копия трафика должна отводиться на EtherSensor до того, как произойдут изменения в параметрах соединений.

Например:

- До прокси-сервера
- До NAT
- До межсетевого экрана.

Режим прозрачного проксирования с маркировкой HTTP трафика

Данный режим работы Агента отличается от режима без маркировки трафика только тем, что Агент модифицирует на клиентской рабочей станции отправляемые приложениями HTTP-запросы, добавляя в них заголовок X-Sensor-UID: <GUID>, где <GUID> – уникальный идентификатор пользователя на конкретном компьютере внутри локальной сети. Эти действия выполняются в полном соответствии с HTTP-протоколом, не нарушая его работу.

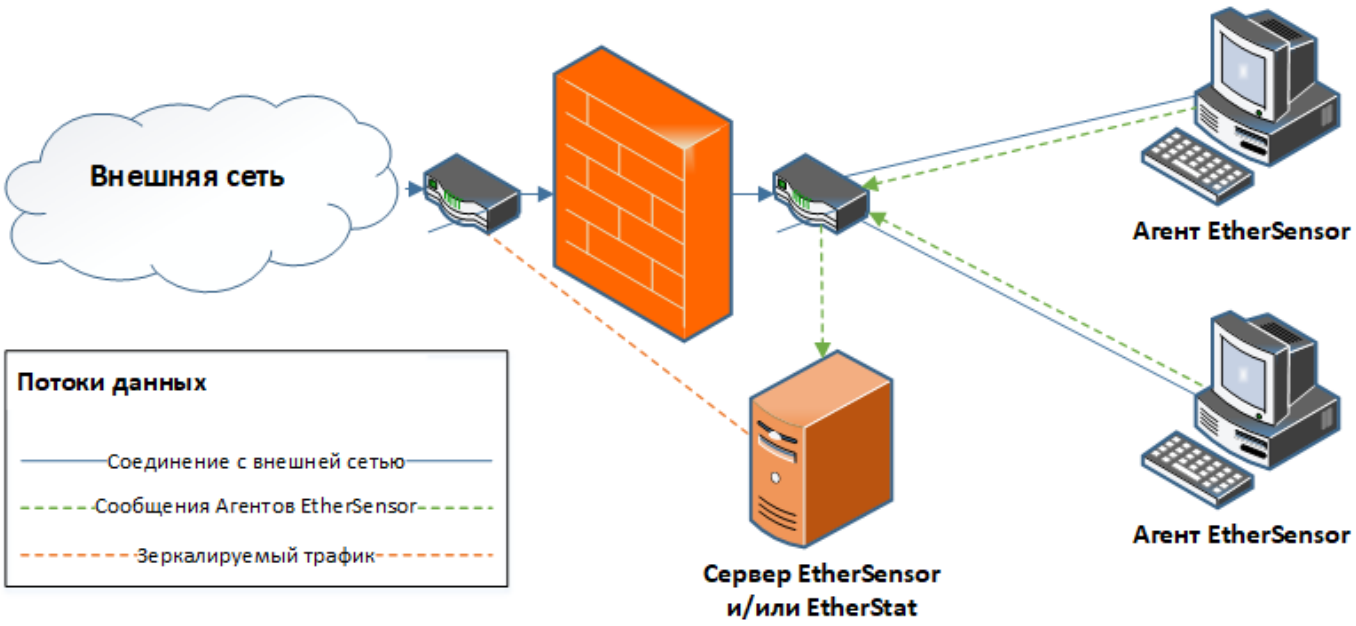

Рис.29. Проксирование с маркировкой трафика

Такой режим работы может использоваться, когда EtherSensor может получать копию трафика для анализа только после модификации параметров соединений. Например, после прохождения соединениями прокси-сервера, NAT или межсетевого экрана.

В этом случае EtherSensor при реконструкции сообщения полностью идентифицирует пользователя, отправившего данное сообщение, по заголовкам X-Sensor-UID из HTTP-протокола.

3.6.7.2. Конфигурирование службы EtherSensor Agent

EtherSensor Agent настраивается через внесение изменений в конфигурационный файл [INSTALLDIR]\config\agent.xml.

Ниже показан пример такого файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<Config version="1.1">
  <Local port="44444" markhttp="true" />

  <EtherSensor protocol="2">
    <server address="ethersensor.server1:44444" transport="udp" />
  </EtherSensor>
  <Filter>
    <Excludes>
      <application name="ethersensor_agent.exe" />
      <application name="mstsc.exe" />
      <application name="wmplayer.exe" />
      <application name="uTorrent.exe" />
      <application name="skype.exe" />
      <application name="wmpnetwk.exe" />
      <application name="winlogon.exe" />
      <application name="svchost.exe" />
      <application name="spoolsv.exe" />
      <application name="nissrv.exe" />
    </Excludes>
  </Filter>

  <EtherStat address="127.0.0.1:44445" ZMQKEY="" />
  <DataCollectionSetup>
    <Hardware duration_ms="10000" />
    <Software duration_ms="30000" />
    <OperatingSystem duration_ms="60000" />
    <Processes duration_ms="1000" />
    <Performance duration_ms="60000" />
    <Network duration_ms="60000" />
    <UserMonitor duration_ms="1000" />
    <DatabaseStore size="1" />
  </DataCollectionSetup>
</Config>
```

Тег Config

Корневой тег конфигурации. Атрибут "version" внутри тега Config определяет версию формата конфигурационных файлов Агента.

Тег Local

Тег Local является вложенным в тег Config и определяет настройки для модуля отслеживания соединений (sensor_lsp.dll). После загрузки конфигурации служба EtherSensor Agent сохраняет данные настройки в реестре Windows.

Атрибут "port" определяет локальный UDP-порт для коммуникации модуля sensor_lsp.dll со службой EtherSensor Agent.

Атрибут "markhttp" разрешает/запрещает маркировку HTTP-трафика модулем sensor_lsp.dll.

Тег EtherSensor

Тег EtherSensor является вложенным в тег Config и определяет список адресов серверов EtherSensor, которым Агент передаёт по протоколу UDP информацию о процессах, создающих TCP-соединения.

Атрибут "protocol" определяет максимальную версию протокола, используемого EtherSensor Agent для отправки сообщений на сервер EtherSensor. На сегодняшний день доступна версия протокола 3. Для поддержки данной версии протокола необходимо использовать EtherSensor версии 4.3.3 или старше. Для совместимости с предыдущими версиями EtherSensor необходимо выставить данное поле в значение 2.

Тег server

Тег "server" является вложенным в тег EtherSensor и определяет адрес и транспортный протокол сервера EtherSensor.

Атрибут "address" определяет адрес и порт для связи с сервером EtherSensor. Возможные варианты адресов – IP:Port или DNSNAME:Port.

Атрибут "transport" определяет тип транспортного протокола для связи с сервером EtherSensor. Возможные варианты – udp.

Тег Filter

Тег Filter является вложенным в тег Config и определяет настройки фильтра сообщений, отправляемых на сервер EtherSensor.

Тег Excludes

Тег Excludes является вложенным в тег Filter и определяет список приложений, информацию о TCP-соединениях которых не следует отправлять на сервер EtherSensor.

Тег application

Тег application является вложенным в тег Excludes и определяет приложение, информация о TCP-соединениях которого не будет отправлена на сервер EtherSensor.

Атрибут "name" определяет точное имя процесса, отслеживание которого требуется исключить.

Таким образом EtherSensor Agent сообщает серверу EtherSensor только о TCP-соединениях, которые создаются для связи с другими рабочими станциями и серверами в локальной сети и сети Интернет, причем настройки позволяют учесть список процессов, соединения которых отслеживать не требуется.

Тег EtherStat

Тег EtherStat является вложенным в тег Config и определяет настройки для соединения к системе мониторинга и статистики EtherStat.

Атрибут "address" определяет адрес сервера в формате "IP-адрес:порт".

Атрибут "ZMQKEY" должен содержать ключ для работы с EtherStat в режиме шифрования соединения.

Тег DataCollectionSetup

Тег DataCollectionSetup является вложенным в тег Config и определяет настройки таймеров опроса данных для службы EtherSensor Agent.

Тег Hardware

Тег Hardware является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для опроса текущего оборудования.

Тег Software

Тег Software является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для опроса текущего установленного программного обеспечения.

Тег OperatingSystem

Тег OperatingSystem является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для опроса данных об операционной системе.

Тег Processes

Тег Processes является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для мониторинга текущих процессов.

Тег Network

Тег Network является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для опроса данных о сетевых адаптерах и их настройках.

Тег Performance

Тег Performance является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для опроса данных о производительности ОС.

Тег Users

Тег Users является вложенным в тег DataCollectionSetup и определяет с помощью атрибута "duration_ms" настройки таймера в секундах для мониторинга действий пользователя.

Тег DatabaseSetup

Тег DatabaseSetup является вложенным в тег DataCollectionSetup и определяет настройки лимита размера базы данных в процентах от свободного места на HDD, на котором установлен EtherSensor Agent.

3.6.7.3. Журналирование работы Агента

EtherSensor Agent логирует свои действия в файлы в директории [INSTALLDIR]\log:

- В файл svcagent.log заносится информация об основных действиях, выполняемых службой EtherSensor Agent.
- В файл sensor_agent.exe.log заносится информация о событиях, генерируемых внутри службы логирования EtherSensor Agent.
- В файл processinfo.log заносится информация о текущих процессах, соединения которых отслеживает EtherSensor Agent.

Файлы журналов (svcagent.log и sensor_agent.exe.log) представляют собой XML-файлы следующего содержания:

```
<Message time="2012-03-23T17:47:48.148+04:00" level="information">
  <Client channelname="MICROOLAPAGENT"
          processname="sensor_agent.exe"
          modulename="sensor_agent.exe" />
  <Text>Start of the application.</Text>
</Message>
```

Тег Message

Тег Message является корневым тегом сообщения, записанного в файл лога. Атрибут "time" – время отправки сообщения, атрибут "level"– с каким приоритетом отправлено сообщение (например, information – информационное сообщение, error – сообщение об ошибке).

Тег Client

Тег Client описывает отправителя сообщения. Атрибут "channelname" содержит имя канала сообщения, атрибут "processname "– имя процесса отправителя, атрибут "modulename" – имя модуля внутри процесса, создавшего сообщение.

Тег Text

Тег Text содержит текст сообщения.

Файл processinfo.log представляет собой XML-файл следующего содержания:

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

  <Process pid="4136" name="chrome.exe">
    <User uuid="32014294-5bbf-11e1-b8f5-005056c00808"
          name="Home-PC\Home"/>
    <Sessions local="0" remote="311"/>
  </Process>

  <Process pid="636" name="svchost.exe">
    <User uuid="3a45de5b-5be6-11e1-b8f5-005056c00808"
          name="HOME\HOME-PC$"/>
    <Sessions local="0" remote="3"/>
  </Process>

  <Process pid="948" name="firefox.exe">
    <User uuid="32014294-5bbf-11e1-b8f5-005056c00808"
          name="Home-PC\Home"/>
    <Sessions local="2" remote="741"/>
  </Process>

  <Process pid="1584" name="googletalk.exe">
    <User uuid="32014294-5bbf-11e1-b8f5-005056c00808"
          name="Home-PC\Home"/>
    <Sessions local="0" remote="89"/>
  </Process>

  <Process pid="2860" name="uTorrent.exe">
    <User uuid="32014294-5bbf-11e1-b8f5-005056c00808"
          name="Home-PC\Home"/>
    <Sessions local="100" remote="27084"/>
  </Process>

  <Process pid="3076" name="vmware.exe">
    <User uuid="32014294-5bbf-11e1-b8f5-005056c00808"
          name="Home-PC\Home"/>
    <Sessions local="4" remote="1"/>
  </Process>

  <Process pid="3908" name="NisSrv.exe">
    <User uuid="fb91c5e7-5eec-11e1-b226-005056c00808"
          name="NT AUTHORITY\LOCAL SERVICE"/>
    <Sessions local="0" remote="1"/>
  </Process>
</Processes>
```

Тег Processes

Тег Processes является корневым тегом списка отслеживаемых процессов.

Тег Process

Тег Process является вложенным в тег Processes и описывает отслеживаемый процесс. Атрибут "pid " содержит идентификатор процесса в ОС, атрибут "name " содержит имя отслеживаемого процесса.

Тег User

Тег User является вложенным в тег Process и описывает пользователя в локальной системе, под чьими правами выполняется отслеживаемый процесс. Атрибут "uuid" содержит идентификатор пользователя, атрибут "name" содержит имя пользователя.

Тег Sessions

Тег Sessions является вложенным в тег Process и описывает отслеживаемые соединения процесса. Атрибут "local " содержит количество локальных соединений, выполненных внутри процесса или между процессами, атрибут "remote" содержит количество удалённых соединений, установленных данным процессом.

3.6.7.4. Проблемы и решения

Не стартует служба EtherSensor Agent.

- Проверьте в конфигурации EtherSensor Agent настройку портов для серверов EtherSensor, EtherStat, а так же настройку локального порта: теги Local, EtherSensor и EtherStat. Порты не должны совпадать. При совпадении портов служба не может корректно работать.
- Проверьте файлы журналов (svcagent.log, sensor_agent.exe.log) на наличие в них сообщений об ошибках работы EtherSensor Agent.
- Проверьте журналы Windows на наличие в них сообщений об ошибках работы EtherSensor Agent.
- Сообщите об инцидентах в службу поддержки.

После старта EtherSensor Agent файл processinfo.log не отображает ни одного отслеживаемого процесса.

- Не подключена сетевая карта или не сконфигурирован стек TCP/IP. Убедитесь, что система создаёт TCP-соединения, например, загрузив браузер и открыв любую страницу. После этих действий в файле processinfo.log должна отобразиться информация о процессе браузера.
- Данный компьютер входит в домен. В таком случае модули EtherSensor Agent, загруженные в процессы, которые создают TCP-соединения, пытаются получить имя пользователя в домене, с чьими правами выполняется процесс. В данном случае очень важно, чтобы были правильно выставлены настройки DNS отслеживаемой ОС, так как системное API, которое предоставляет информацию об имени пользователя в домене, использует настройки DNS.

Ни одно приложение в системе не может создать удалённое соединение, но до инсталляции EtherSensor Agent в систему таких проблем не было.

- Перейдите в директорию инсталляции EtherSensor Agent и выполните команду sensor_instlsp.exe -p > log.txt. Данная команда запишет в файл log.txt список установленных провайдеров сетевого стека ОС.

- Проанализируйте самостоятельно файл log.txt и при необходимости вышлите в службу поддержки.
- Выполните команду sensor_instlsp.exe -f -c b. Данная команда отключит модуль слежения EtherSensor Agent sensor_lsp.dll. Запустите новую копию браузера и откройте удалённую страницу. Если страница открылась, значит проблемы действительно в модуле слежения sensor_lsp.dll. В противном случае проблема не связана с модулем слежения EtherSensor Agent.

EtherSensor не идентифицирует пользователя перехваченного сообщения.

- Проверьте конфигурацию EtherSensor Agent: тег EtherSensor, затем тег server. Атрибут "address" тега server должен содержать корректное DNS-имя сервера EtherSensor, которое правильно определяется на данной рабочей станции. Если указан IP-адрес, проверьте доступность IP-адреса сервера EtherSensor (например, при помощи утилиты ping).

Сервер EtherStat не получает сообщения от EtherSensor Agent:

- Проверьте файл svcagent.log на наличие в нём сообщений об ошибках EtherSensor Agent.
- Проверьте конфигурацию [82] соединения с сервером EtherStat: тег EtherStat. Атрибут "address" должен содержать IP-адрес, по которому происходит соединение. Атрибут "key" содержит открытый ключ для шифрованного соединения. Этот ключ должен совпадать с открытым ключом сервера EtherStat.
- Проверьте доступность IP-адреса сервера EtherStat (например, при помощи утилиты ping).
- Проверьте конфигурацию времени обработки информации в тегах OSMonitor, HWMonitor, SWMonitor, UserMonitor, NetMonitor и ProcMonitor. Если атрибут "timer" установлен в 0, то сообщения соответствующих событий обрабатываться не будут, следовательно, не будут отправляться и на сервер EtherStat.

4. Анализ событий и объектов

Служба EtherSensor Analyser предназначена для детектирования, фильтрации и анализа извлечённых из сетевого трафика объектов.

Служба анализирует объекты протоколов уровня приложений, полученные от служб EtherSensor PCAP, EtherSensor EtherCAP, EtherSensor ICAP и EtherSensor LotusTXN с целью детектирования объектов и событий, распознавая их принадлежность к определённым Интернет или интранет сервисам.

Начиная с версии EtherSensor 6.0 реализована основанная на скриптах Lua открытая подсистема анализа данных и детектирования событий.

Поставка EtherSensor включает обширный набор заранее подготовленных Lua-скриптов, а с помощью EtherSensor IDE возможна самостоятельная разработка фильтров, детекторов и профилей доставки результатов.

Общая схема работы службы EtherSensor Analyser:

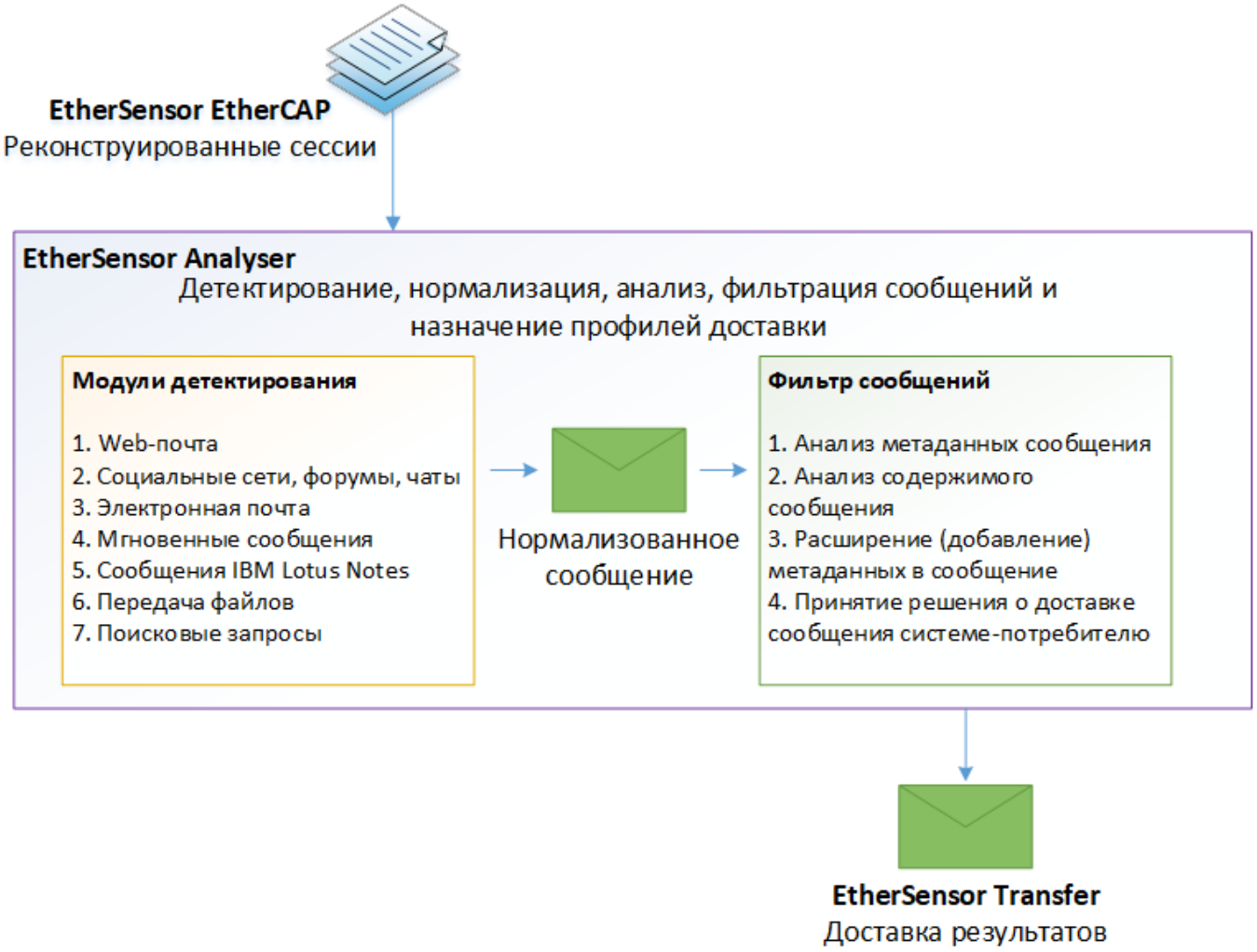

Рис.30. Схема работы службы EtherSensor Analyser.

Файл конфигурации EtherSensor Analyser

Конфигурация службы EtherSensor Analyser хранится в XML-файле analyser.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

Параметры командной строки

Служба Windows EtherSensor Analyser в ходе инсталляции Microolap EtherSensor устанавливается с автоматическим запуском. Однако, при необходимости процесс sensor_analyser.exe можно запустить как приложение Windows со следующими параметрами командной строки:

/process

Запустить процесс sensor_analyser.exe как обычный Windows Win32-процесс (возможно использовать для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

4.1. Настройка EtherSensor Analyser

Анализ извлечённых из сетевого трафика объектов, выполняемый службой EtherSensor Analyser, состоит из следующих этапов:

1. Предварительная фильтрация объектов/событий [93].

Предварительная фильтрация перехваченных объектов применяется в основном для снижения нагрузки на EtherSensor. На этом этапе предполагается использование следующих метаданных объектов, полученных на этапе перехвата трафика службой EtherSensor EtherCAP:

- Сетевые адреса участников коммуникации
- Типы протоколов
- Метаданные перехваченных объектов конкретных протоколов
- Анализ HTTP-объектов (методы запросов, URL, заголовки запроса/ответа)
- Размеры и типы RAW-данных перехвата.

Также на этапе предварительной фильтрации EtherSensor может логировать события и их контент в различных форматах, основанных на SYSLOG: ArcSight CEF, QRadar LEEF, ICSA/CISCO SDEE, DMTF CIM, CEE и т.п.

2. Детектирование и нормализация объектов/событий [96].

На этом этапе объекты/события, полученные из различных источников, трансформируются с помощью скриптов Lua в нормализованные события уровня приложения с одинаковой структурой и кодировкой (UTF-8 по умолчанию).

При нормализации перехваченных объектов доступны следующие функции:

- Анализ URL-encoded объектов
- Анализ содержимого MIME частей
- Анализ объектов JSON/XML/HTML
- Анализ объектов WebSocket: custom binary, json, xml, xmpp
- Распаковка объектов base64, gzip, brotli
- Анализ объектов с помощью hyperscan и pcre-2
- Реконструкция файлов, передаваемых по частям
- Анализ бинарных объектов, например:
  - DNS over HTTPS
  - Protobuf.
- Накопление и хранение метаданных, например:
  - Формирование и хранение профилей пользователей/устройств для дальнейшей привязки событий
  - Хранение Cookie
  - Хранение User Account Information.

3. Завершающая фильтрация [99].

Объекты/события, полученные на этапе детектирования и нормализации, могут проходить этап завершающей фильтрации. На этапе завершающей фильтрации проверяются следующие параметры объекта/события:

- Сетевые адреса участников коммуникации
- Доменные адреса участников коммуникации
- Метаданные объекта/события (атрибуты и заголовки)
- Email-адреса участников коммуникации (from, to, cc, bcc)
- Идентификаторы пользователей клиентов мгновенных сообщений (ICQ, MRA, MSN, IRC, SKYPE, XMPP)
- Идентификаторы пользователей социальных сетей

- Содержимое текстовых полей сообщений (subject, body)
- Имена и типы аттачментов
- Размеры файлов и сообщений.

Объекты/события на данном этапе обработки могут быть обогащены дополнительными метаданными/атрибутами:

- Типы файлов, детектированные с помощью libmagic
- Хэши файлов и различных сегментов объектов/событий (crc32, md5, sha1, sha256)
- Метаданные, накапливаемые службой EtherSensor Identity (пользователи сети, устройства и т.п.)
- Информация о доставке объекта/события системе-потребителю (DLP, SIEM, UEBA, eDiscovery, Enterprise Search, другие архивы и т.п.).

На каждом этапе обработки данных можно задать произвольный набор и порядок вызова Lua-скриптов для обработки перехваченного объекта/события.

Это означает, что пользователь EtherSensor может полностью контролировать порядок действий и логику обработки перехваченных объектов.

Каждый Lua-скрипт может:

- Прекратить обработку текущего объекта/события, после чего он и его данные будут удалены
- Продолжить обработку объекта/события. В этом случае объект будет передан следующему скрипту для дальнейшей обработки
- Создать новый объект/событие и передать его на этап завершающей фильтрации.

4.1.1. Предварительная фильтрация

Для управления настройками предварительной фильтрации используйте окно ***Предварительная фильтрация*** консоли управления EtherSensor:

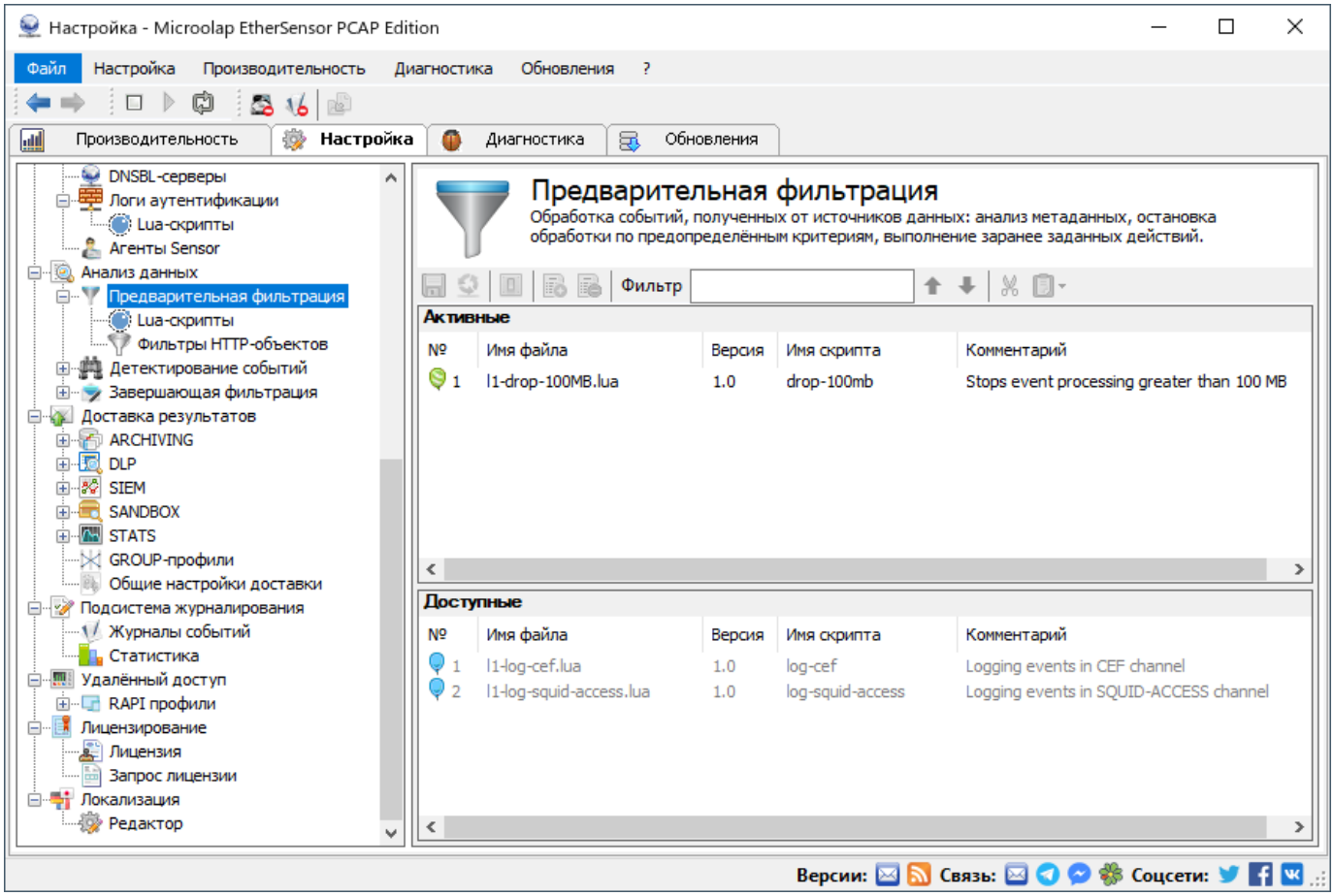

Рис.31. Конфигурирование предварительной фильтрации перехваченных объектов.

В этом окне назначаются Lua-скрипты для предварительной фильтрации объектов. Скрипты вызываются от первого ко второму и так далее. Количество скриптов, участвующих в предварительной фильтрации, может быть сколь угодно большим.

Редактировать скрипты можно с помощью любого текстового редактора (скрипты находятся в каталоге инсталляции [INSTALLDIR]/scripts/an-prefilter) или же прямо в окне консоли управления ***Lua-скрипты***.

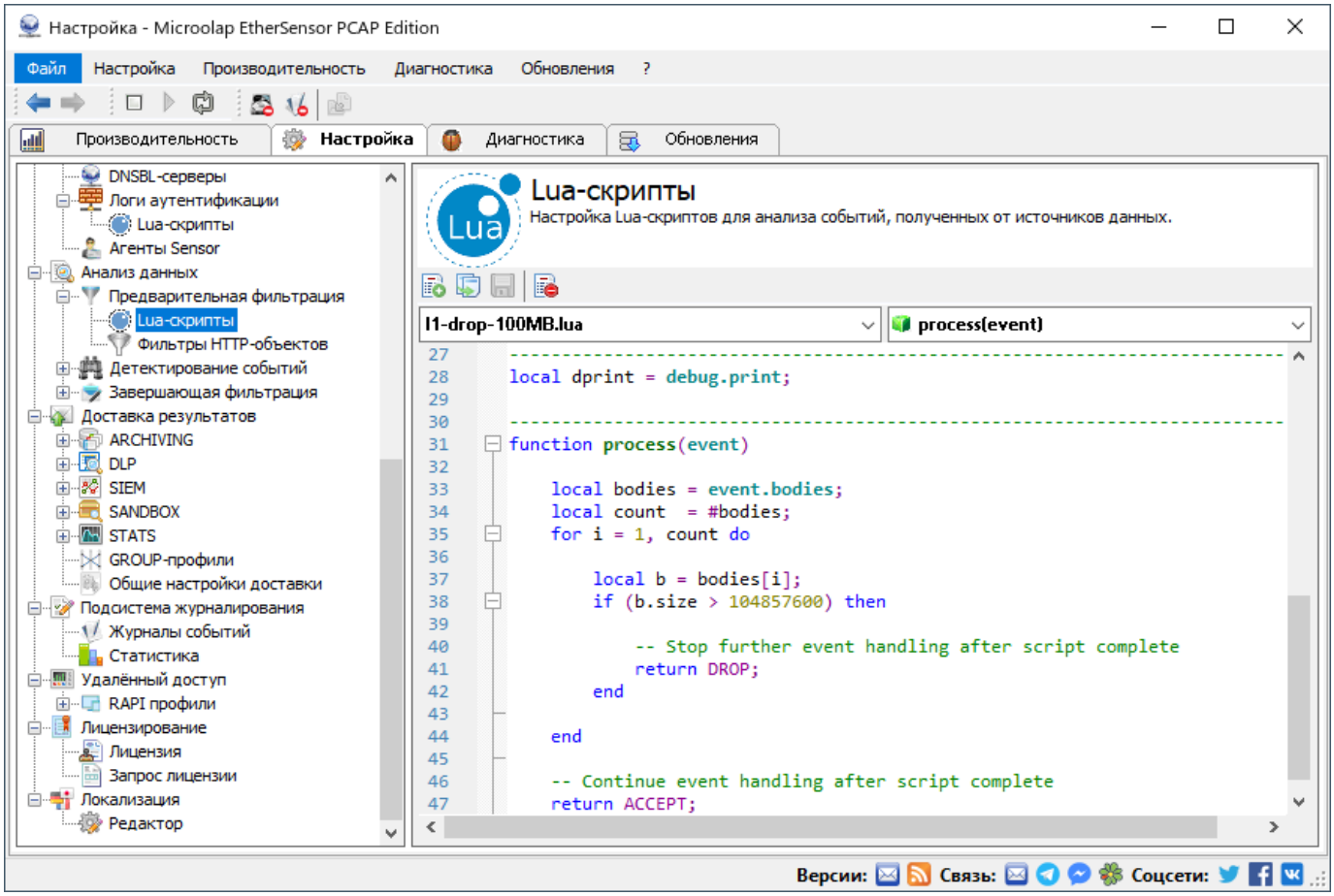

Рис.32. Редактирование скриптов предварительной фильтрации перехваченных объектов.

Фильтрацию перехваченных HTTP-запросов также можно сконфигурировать, используя окно ***Фильтры HTTP-объектов*** консоли управления. Правила фильтрации, создаваемые с помощью фильтра HTTP-объектов хранятся в XML-файлах, находящихся в директории [INSTALLDIR] \config\filter\http.

Во время запуска службы EtherSensor Analyser активный фильтр из формата XML трансформируется в Lua-скрипт и всегда становится первым в цепочке вызова скриптов предварительной фильтрации.

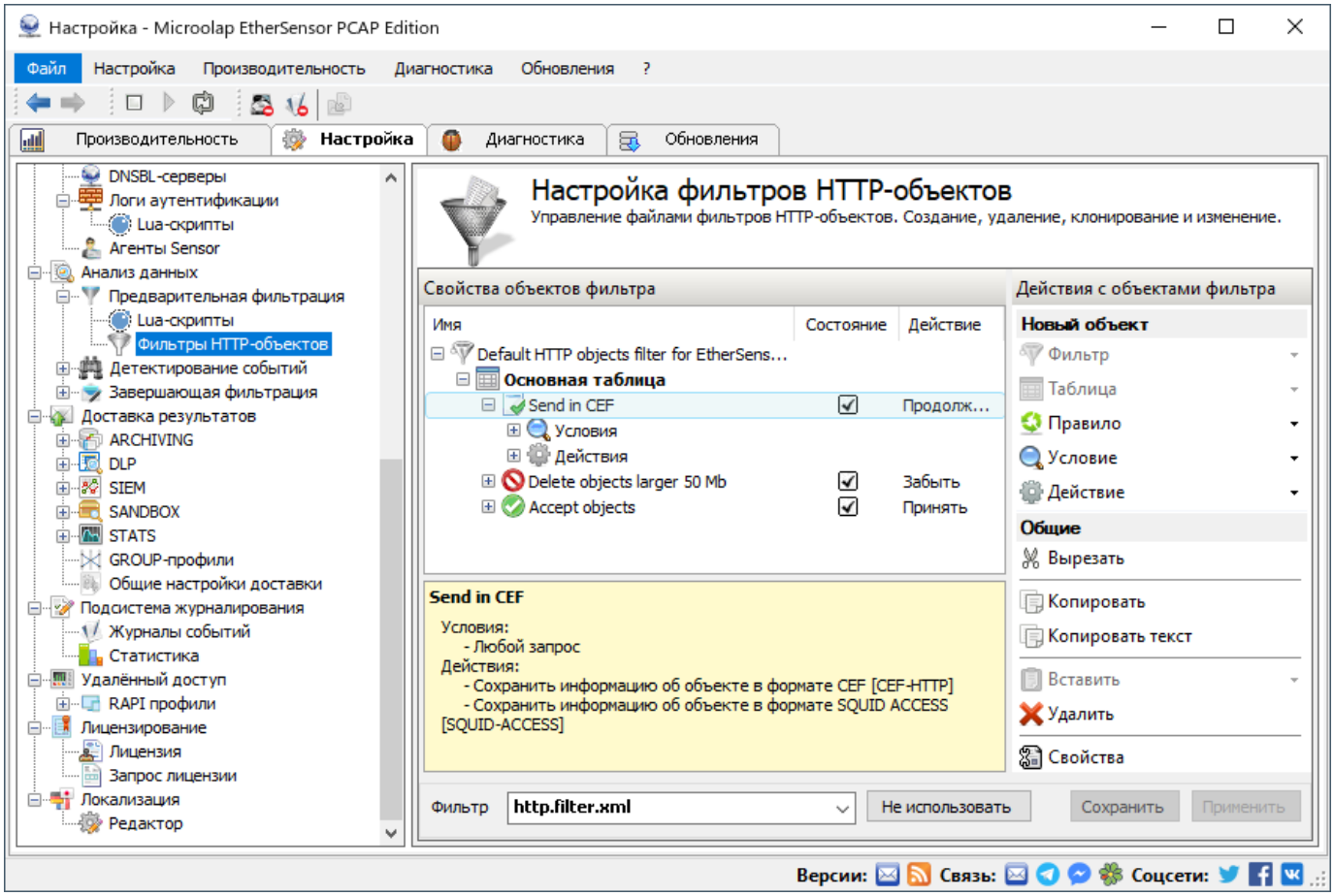

Рис.33. Управление фильтрацией HTTP-объектов.

Используйте правую часть окна для редактирования фильтра, допускается редактирование активного фильтра. Но для того, чтобы служба EtherSensor Analyser начала использовать измененный фильтр, этот фильтр следует сделать активным и перезапустить службу.

Подробнее ознакомиться с работой этой функции можно в разделе Префильтрация HTTP-запросов [166].

4.1.2. Детектирование и нормализация событий

Для управления настройками детектирования и нормализации объектов/событий используйте окно ***Детектирование событий*** консоли управления:

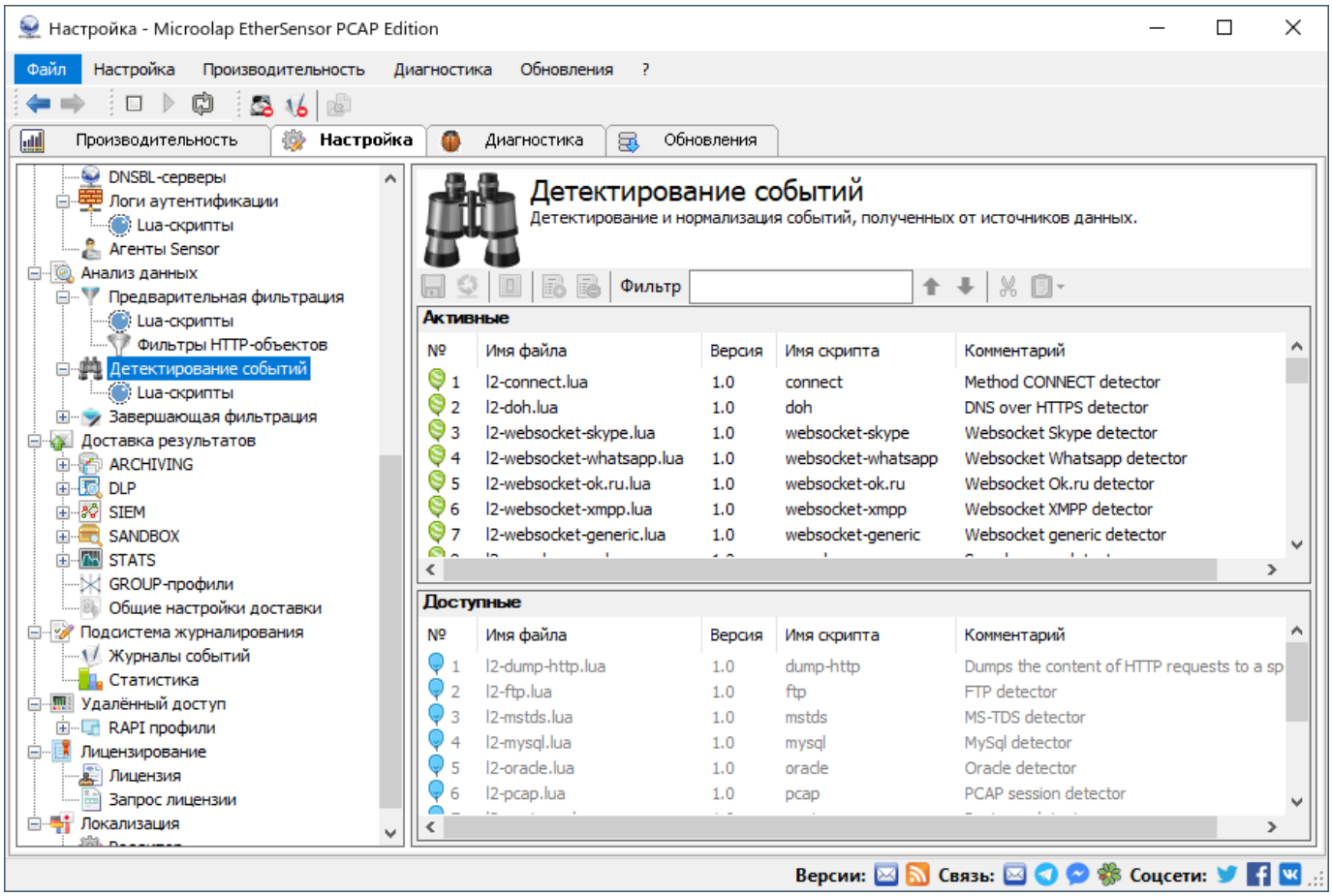

Рис.34. Управление детекторами объектов/событий.

В этом окне назначаются Lua-скрипты для детектирования и нормализации объектов/событий. Скрипты вызываются от первого ко второму и так далее. Количество скриптов, участвующих в детектировании событий, может быть сколь угодно большим.

Редактировать скрипты можно с помощью любого текстового редактора (скрипты находятся в каталоге инсталляции [INSTALLDIR]/scripts/an-detect), или же прямо в окне консоли управления ***Lua-скрипты***.

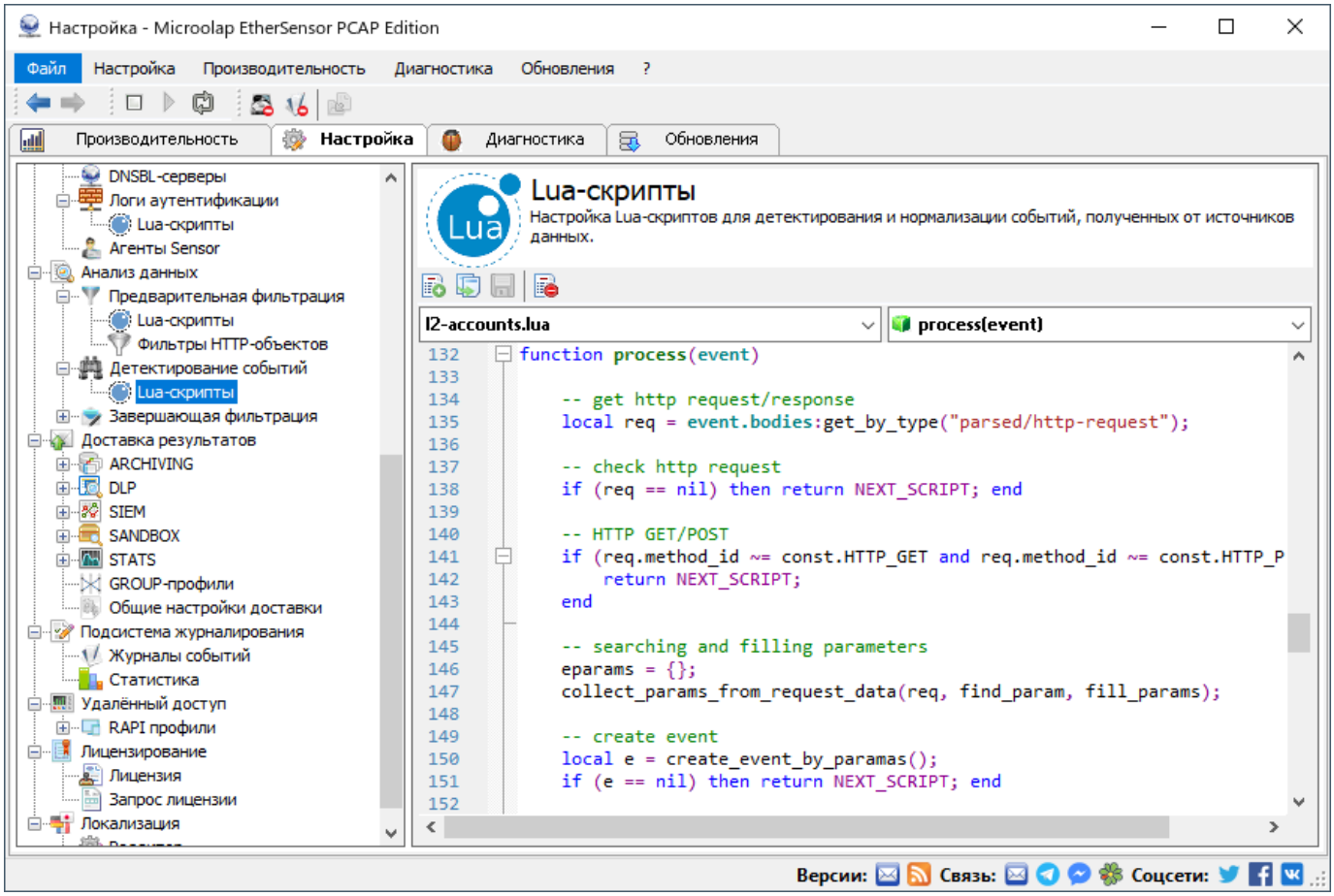

Рис.35. Редактирование скриптов, детектирующих события уровня приложения.

4.1.2.1. INCLUDE: Lua scripts functions

- Анализ URL-encoded объектов
- Анализ содержимого MIME частей
- Анализ объектов JSON/XML/HTML
- Анализ объектов WebSocket: custom binary, json, xml, xmpp
- Распаковка объектов base64, gzip, brotli
- Анализ объектов с помощью hyperscan и pcre-2
- Реконструкция файлов, передаваемых по частям
- Анализ бинарных объектов, например:
  - DNS over HTTPS
  - Protobuf.

- Накопление и хранение метаданных, например:
  - Формирование и хранение профилей пользователей/устройств для дальнейшей привязки событий
  - Хранение Cookie
  - Хранение User Account Information.

4.1.3. Завершающая фильтрация

Для управления настройками завершающей фильтрации перехваченных объектов/событий используйте окно ***Завершающая фильтрация*** консоли управления.

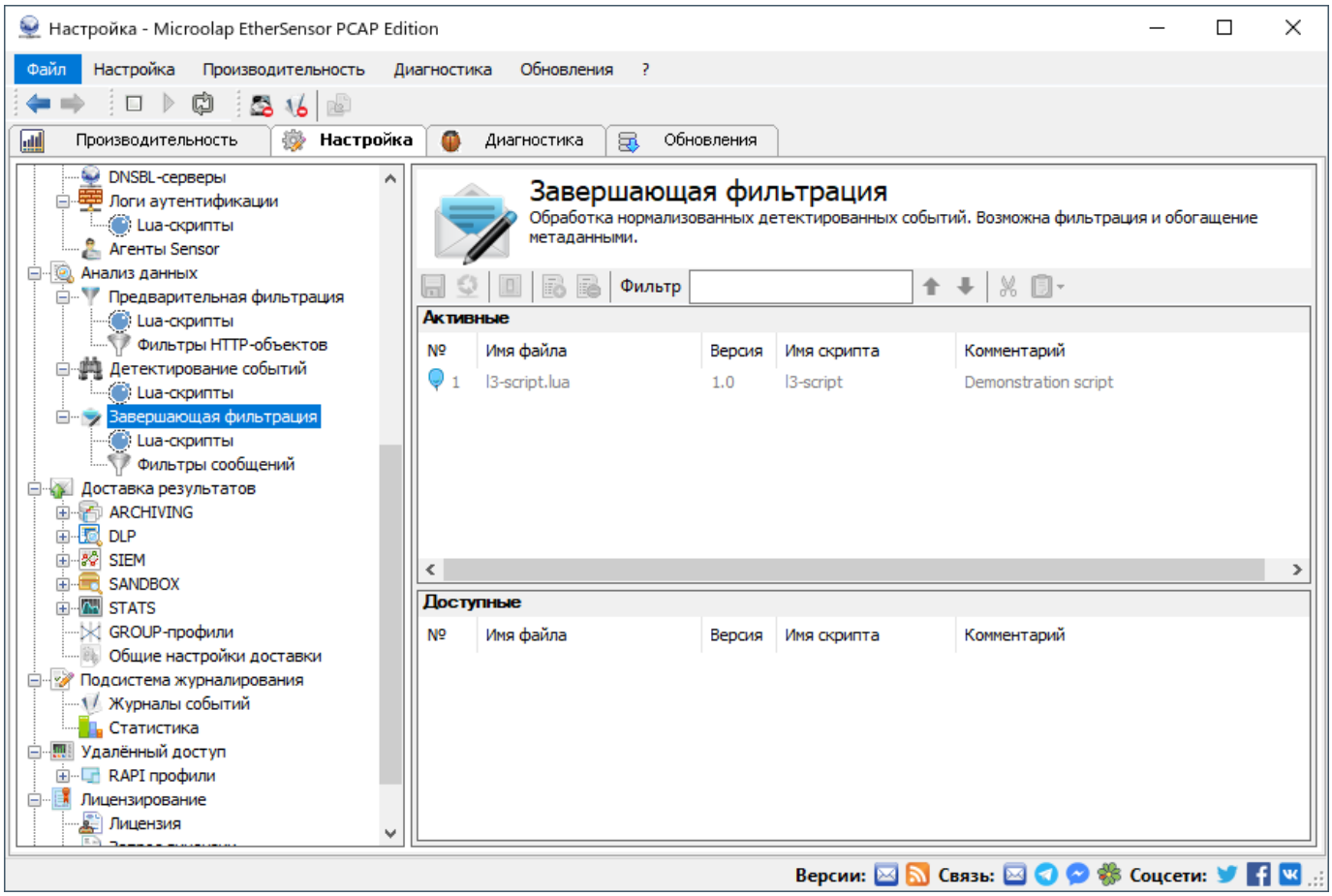

Рис.36. Завершающая фильтрация объектов/событий.

В этом окне назначаются Lua-скрипты для завершающей фильтрации нормализованных объектов/событий. Скрипты вызываются от первого ко второму и так далее. Количество скриптов, участвующих в завершающей фильтрации, может быть сколь угодно большим.

Редактировать скрипты можно с помощью любого текстового редактора (скрипты находятся в каталоге инсталляции [INSTALLDIR]/scripts/an-postfilter), или же прямо в окне консоли управления ***Lua-скрипты***.

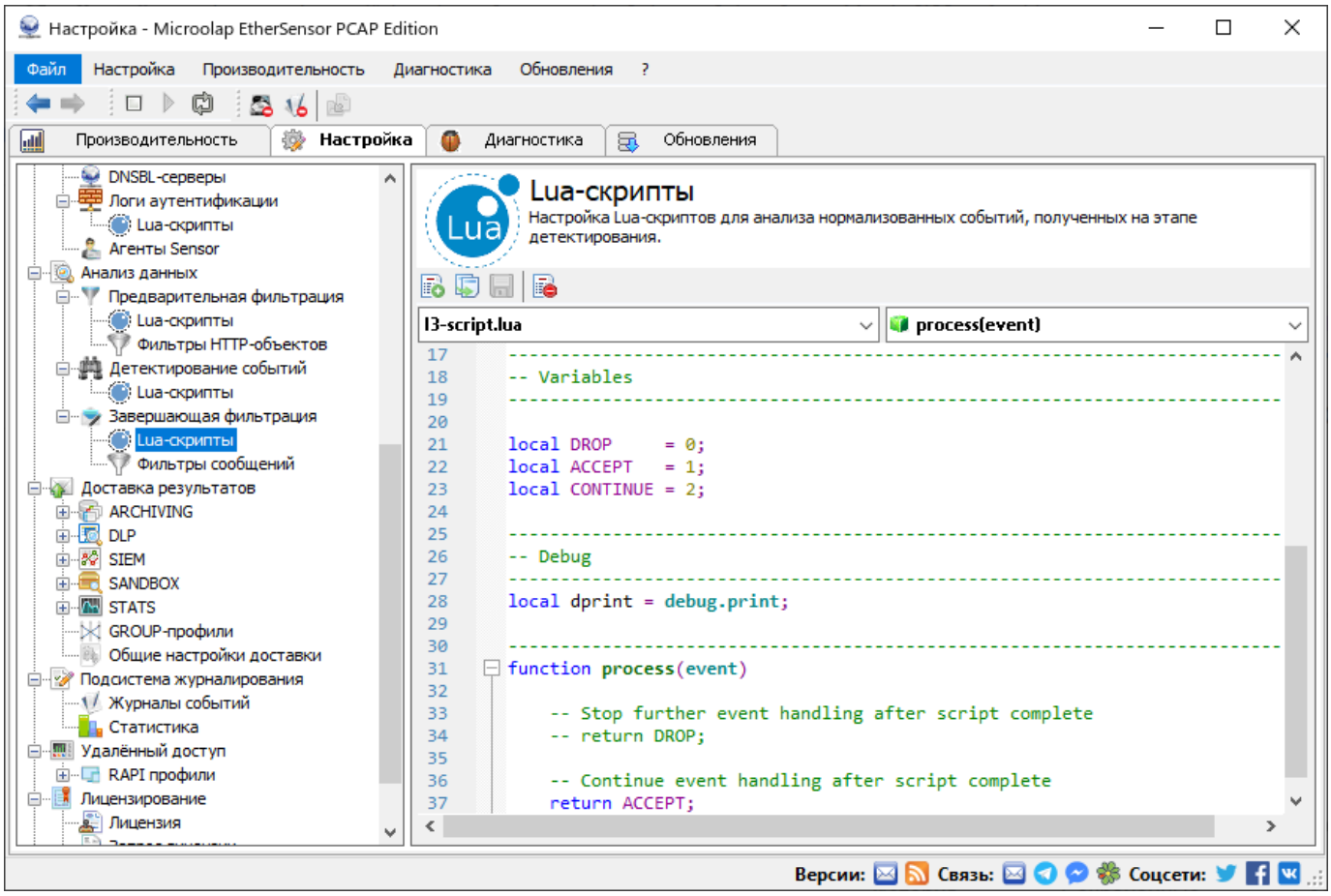

Рис.37. Редактирование скриптов завершающей фильтрации объектов/событий.

Настройка завершающей фильтрации нормализованных событий также возможна с помощью фильтра сообщений в окне ***Фильтры сообщений*** консоли управления. Правила фильтрации, создаваемые с помощью фильтра сообщений хранятся в XML-файлах, находящихся в директории [INSTALLDIR]\config\filter.

Во время запуска службы EtherSensor Analyser активный фильтр из формата XML трансформируется в Lua-скрипт и всегда становится первым в цепочке вызова скриптов завершающей фильтрации.

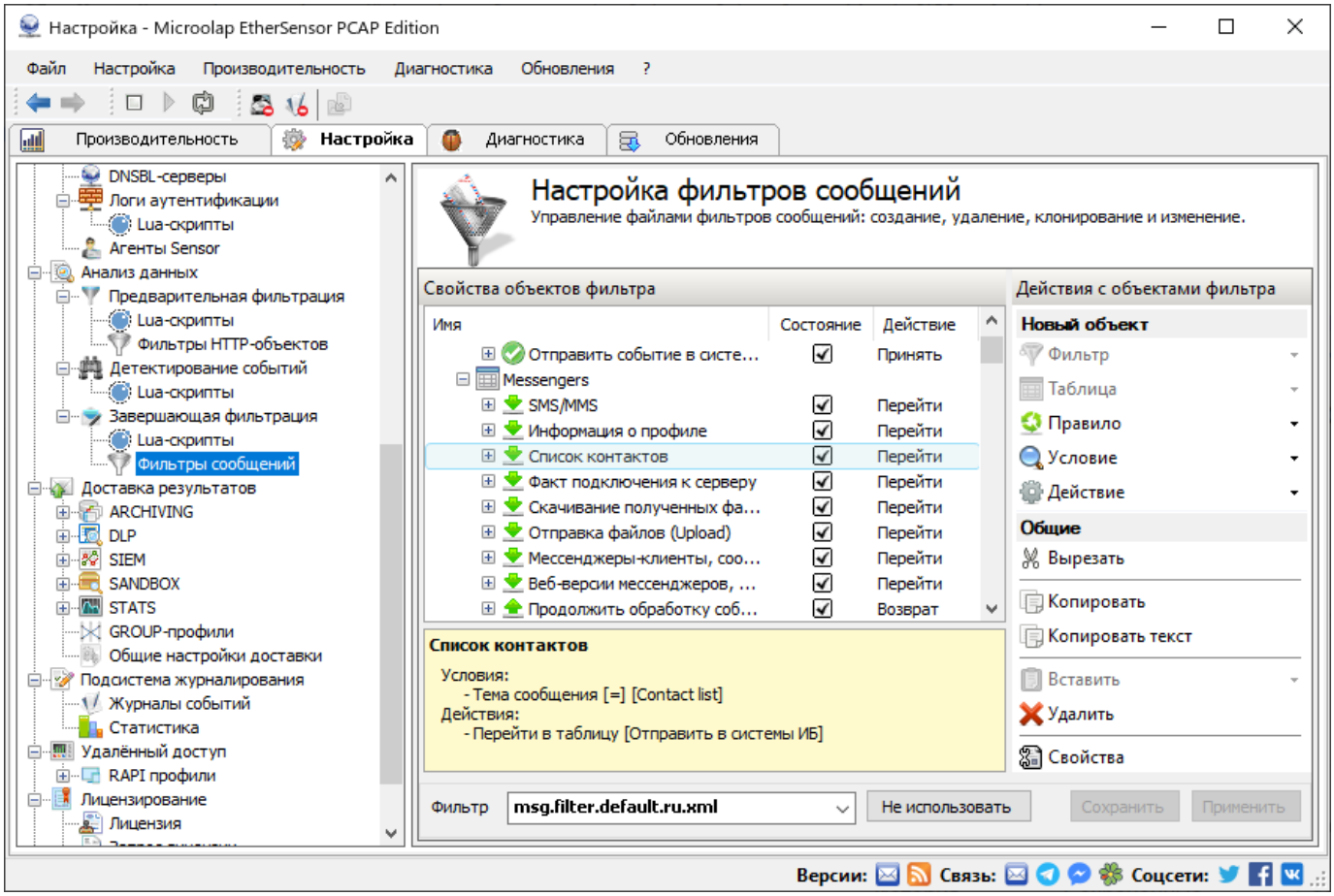

Рис.38. Конфигурирование фильтров объектов/событий.

Используйте правую часть окна для редактирования фильтра, допускается редактирование активного фильтра. Но для того, чтобы служба EtherSensor Analyser начала использовать измененный фильтр, этот фильтр следует сделать активным и перезапустить службу.

Подробнее с созданием фильтров можно ознакомиться в разделе Фильтрация сообщений [109].

4.2. Формируемые события/сообщения

Результатом работы EtherSensor являются события/сообщения системам-потребителям, сформированные из объектов уровня приложения, извлеченных из трафика. Для дальнейшей обработки EtherSensor передает такие сообщения системам-потребителям с помощью службы доставки результатов [204].

Для формирования сообщения системе-потребителю о реконструированной коммуникации применяются следующие механизмы:

1. Перехваченное RAW-событие формируется в бинарный объект, где описаны известные данные о характере перехваченных объектов, отправителях, получателях. В процессе обработки RAW-событие наполняется собираемыми данными.

2. По окончании обработки объекта EtherSensor Analyser передаёт его в виде нормализованного события и других файлов данных в службу доставки результатов.

3. Служба доставки результатов, в зависимости от требуемого типа транспорта и его настроек, формирует из нормализованного события готовый к доставке системе-потребителю объект. Структура этого объекта определяется профилями доставки результатов.

Например, для доставки результатов по протоколу SMTP формируется сообщение электронной почты, где в адреса отправителя и получателя (FROM, TO, CC, BCC) подставляются перехваченные данные об источнике и назначении сообщения, если такие данные есть. Текст сообщения становится текстом email-сообщения, передаваемые файлы – аттачментами к сообщению. Прочие данные, накопленные EtherSensor при обработке соединения, сохраняются в виде MIME-заголовков сообщения.

Формат сообщения для доставки результата

Доставляемые системе-потребителю сообщения должны соответствовать требованиям к сообщениям, передаваемым по протоколу SMTP. Длинные заголовки могут разбиваться и кодироваться, основной кодировкой является UTF-8. EtherSensor детектирует типы данных объектов на основе заголовков протоколов.

1. Для ряда веб-сервисов допускается перехват нескольких сообщений на одну операцию пользователя в веб-интерфейсе – это реакция EtherSensor на отправку в сервис черновиков или аттачментов.

2. Допустима ситуация, когда текст и аттачменты доставляются как отдельные сообщения.

3. Допустима ситуация, когда для нескольких сообщений в систему-потребитель передается один контейнер с содержимым этих сообщений (в основном для IM-протоколов для минимизации трафика и связанных с ним накладных расходов).

Служебные MIME-заголовки EtherSensor, примеры значений

X-Sensor-Version: 6.1
Текстовая строка. Идентифицирует текущую версию EtherSensor.

X-Sensor-Id: sensor-01
Текстовая строка. Идентификатор сенсора, позволяет отличить сообщения от разных экземпляров oEtherSensor или сгруппировать по этому значению. Может быть любой ASCII идентификатор, по умолчанию вставляется UHID.

X-Sensor-Session-Id: 6612456
Целое число. Внутренний идентификатор соединений, обработанных EtherSensor. Выдается источником сообщений (службами EtherSensor ICAP или EtherSensor EtherCAP). Регистрируется в журнале capture.log.

X-Sensor-Net-Interface-Id: 00-21-28-10-58-80
Текстовая строка. В случае, если сообщение перехвачено службой EtherSensor EtherCAP, то значением является MAC-адрес интерфейса или capdrop – виртуальный идентификатор драйвера разбора PCAP-файлов.

В случае ICAP-сервера будет приведен идентификатор из конфигурации ICAP-сервера. Этот заголовок позволяет проследить источник сообщения: с какого именно интерфейса или сервиса были получены данные для обработки.

X-Sensor-Session-Level: 0
Целое число. Показывает, сколько потребовалось разобрать различных протоколов, чтобы добраться до сообщения. Протоколом может быть, например, соединение HTTP, HTTP-proxy соединение через это соединение, а также GRE-инкапсуляция.

X-Sensor-Src-Address: 10.31.90.22:47016

X-Sensor-Dst-Address: 193.203.100.139:8080
IP-адреса и порты соединения: source и destination. Могут быть не определены, если идёт обработка ICAP-трафика, в котором нет заголовков X-Client-IP и X-Server-IP. Пример ICAP - заголовков: X-Client-IP: 192.168.3.67, X-Server-IP: 123.45.67.89.

X-Sensor-Src-Host: pc-test.msk.su

X-Sensor-Dst-Host: nns-team.ru
Текстовые строки. Имена хостов, соответствующие заголовкам X-Sensor-Src-Address и X-Sensor-Dst-Address. Так как многие сети выдают внутренние адреса по DHCP, следует выяснять имя хоста именно в момент перехвата сообщения. EtherSensor делает это, используя обратный DNS-запрос к указанному в конфигурации службы EtherSensor Analyser DNS-серверу. В случае если распознать не удалось, значение будет <not resolved>.

X-Sensor-Protocol: HTTP
Текстовая строка. Название детектора протокола, который был использован для разбора данных. Возможны SMTP, ICQ, MRA и другие.

X-Sensor-Detector: phpbb
Текстовая строка. Название детектора EtherSensor, который определил наличие данных в соединении.

X-Sensor-Attachments-Count: 0
Целое число. Если в сообщении были обнаружены файлы, их количество указывается в этом заголовке.

X-Sensor-Object-Date: Fri, 17 Sep 2010 17:30:24 +0400
Текстовая строка. Время перехвата соединения (время создания объекта в EtherSensor). Таймзона выбирается по настройкам ОС сенсора.

X-Sensor-Object-Size: 2735
Целое число. Размер перехваченного объекта в байтах до обработки.

X-Sensor-Object-MD5Hash: c326230de58279229862b18e818a3912
Текстовая строка, md5 хэш от перехваченного объекта.

> Если у сообщений одинаковый текст, но разные размер, хэш, дата объекта или адреса и порты источника и назначения, то это не дубликаты, а сильно похожие объекты.
>
> Совпадения md5 хэшей у двух разных объектов не должно быть в нормальной ситуации. Если оно происходит, значит одно и то же соединение обрабатывается EtherSensor несколько раз (петля).

X-Sensor-Via: 1.1 off:1080 (squid/2.6.STABLE18)

X-Sensor-Forwarded-For: 10.255.241.31
Текстовые строки. Заголовки из соединения HTTP, для имеющихся заголовков подставляются их значения. По данным значениям можно выяснить, что клиент соединения работает за прокси-сервером, а также иногда узнать его адрес на момент регистрации сообщения.

X-Sensor-Icap-Client-Username: user1

X-Sensor-Icap-Subscriber-Id: mike.smith@mycompany.com

X-Sensor-Icap-Authenticated-User:
TERBUDovLzE5Mi4xNjguMTIuMTAwL289bXljb21wYW55LCBvdT1lbmdpbmVlcmluZywgY249bWlrZS5zbWl0aA==

X-Sensor-Icap-Authenticated-Group:
TERBUDovLzE5Mi4xNjguMTIuMTAwL289bXljb21wYW55LCBvdT1lbmdpbmVlcmluZw==
Значения данных заголовков формируются при обработке ICAP-трафика. Для этого используются соответствующие заголовки ICAP-протокола (X-Client-Username, X-Subscriber-ID, X-Authenticated-User, X-Authenticated-Groups).

X-Sensor-Filter-Name: TEST

X-Sensor-Tags: Filtered=1

X-Sensor-Labels: filter-begin-time="2010-09-17T17:30:24.6318125+04:00",

dns-begin="2010-09-17T17:30:24.6318125+04:00",

dns-end="2010-09-17T17:30:24.6318125+04:00",

Filtered="true",

filter-end-time="2010-09-17T17:30:24.6318125+04:00"

Служебные заголовки службы EtherSensor Analyser. Позволяют определить сработавший фильтр, характер содержимого сообщения, установленные теги и метки, а также отследить обработку сообщения в фильтре. Результирующий состав этих заголовков сильно зависит от политики фильтра.

Date: Fri, 17 Sep 2010 17:30:24 +0400

В данный заголовок дублируется значение X-Sensor-Object-Date.

From: anonymous@nns-team.ru

To: forum@nns-team.ru

CC, BCC и другие заголовки

Subject: Re: World of Tanks

Если это возможно, в заголовки отправителя и получателя подставляются адреса или идентификаторы пользователей, извлеченные из сообщений, заголовков запросов и т.д. Они могут быть не определены, а также зависят от детектора: например, если протокол не использует поле темы сообщения, оно может быть использовано для информации от сенсора.

X-Sensor-RawSource-Type: LotusMail

Все сообщения помечаются заголовком X-Sensor-RawSource-Type для того, чтобы было известно, из какого именно источника данных было получено конечное сообщение.

Значениями этого заголовка в текущей версии EtherSensor (6.1) могут быть:

raw/http-request

Означает, что первичным источником данных был HTTP-запрос

raw/http-response

Означает, что первичным источником данных был HTTP-ответ

raw/ftp-file

Означает, что первичным источником данных был FTP-файл

raw/smtp-eml

Означает, что первичным источником данных было SMTP-сообщение в формате EML

raw/pop3-eml

Означает, что первичным источником данных было POP3-сообщение в формате EML

raw/icq-contact-list

Означает, что первичным источником данных был список контактов ICQ

raw/icq-message-list

Означает, что первичным источником данных был список сообщений ICQ

raw/icq-file

Означает, что первичным источником данных был файл, переданный между ICQ-клиентами

raw/icq-login-info

Означает, что первичным источником данных была ICQ-информация о пользователе

raw/mra-user-info

Означает, что первичным источником данных была MRA-информация о пользователе

raw/mra-contact-list

Означает, что первичным источником данных был список контактов MRA

raw/mra-message-list

Означает, что первичным источником данных был список сообщений MRA

raw/mra-file

Означает, что первичным источником данных был файл, переданный между MRA-клиентами

raw/msn-contact-list

Означает, что первичным источником данных был список контактов MSN

raw/msn-message-list

Означает, что первичным источником данных был список сообщений MSN

raw/msn-file

Означает, что первичным источником данных был файл, переданный между MSN-клиентами

raw/xmpp-contact-list

Означает, что первичным источником данных был список контактов XMPP

raw/xmpp-message-list

Означает, что первичным источником данных был список сообщений XMPP

raw/xmpp-file

Означает, что первичным источником данных был файл, переданный между XMPP-клиентами

raw/irc-message-list

Означает, что первичным источником данных был список сообщений IRC

raw/irc-file

Означает, что первичным источником данных был файл, переданный между IRC-клиентами

raw/ssl-session-list

Означает, что первичным источником данных был список SSL-сессий

raw/lotus-mail

Означает, что первичным источником данных было сообщение протокола LOTUS

raw/lotus-attachment

Означает, что первичным источником данных был файл аттачмента сообщения LOTUS.

X-Sensor-LicOption: Lotus

Все сообщения помечаются заголовком X-Sensor-LicOption для того, чтобы было известно, каким модулем было обработано данное сообщение. Значениями этого заголовка в текущей версии EtherSensor (6.1) могут быть:

webmail

Означает, что сообщение было обработано модулем с лицензионной опцией "Веб-почта"

websocial

Означает, что сообщение было обработано модулем с лицензионной опцией "Социальные сети"

email

Означает, что сообщение было обработано модулем с лицензионной опцией "Электронная почта"

im

Означает, что сообщение было обработано модулем с лицензионной опцией "Мгновенные сообщения"

ft

Означает, что сообщение было обработано модулем с лицензионной опцией "Передача файлов"

webmailread

Означает, что сообщение было обработано модулем с лицензионной опцией "Чтение входящей веб-почты"

lotus

Означает, что сообщение было обработано модулем с лицензионной опцией "Перехват сообщений системы Lotus Notes"

lotustxn

Означает, что сообщение было обработано модулем с лицензионной опцией "Извлечение сообщений из Lotus Notes Transaction Log".

X-Sensor-UID: 0e515c8c-61eb-11e1-a529-000c29ff0707
Данный заголовок формируется при взаимодействии сервера EtherSensor и экземпляров EtherSensor Agent, установленных на рабочих станциях пользователей сети.

Значение заголовка уникально идентифицирует пользователя организации на конкретном компьютере.

X-Sensor-UID-UserName: CN=Administrator,CN=Users,DC=bigbrother,DC=foo
Данный заголовок формируется при взаимодействии сервера EtherSensor и экземпляров EtherSensor Agent, установленных на рабочих станциях пользователей сети.

Значение заголовка уникально идентифицирует пользователя внутри организации.

X-Sensor-UID-UserSID: S-1-5-21-86032015-1269853868-1024056280-1001
Данный заголовок формируется при взаимодействии сервера EtherSensor и экземпляров EtherSensor Agent, установленных на рабочих станциях пользователей сети.

Значение заголовка уникально идентифицирует пользователя внутри организации и содержит идентификатор безопасности текущего пользователя (SID).

X-Sensor-UID-ComputerName: WS325-LOCK.bigbrother.foo
Данный заголовок формируется при взаимодействии сервера EtherSensor и экземпляров EtherSensor Agent, установленных на рабочих станциях пользователей сети.

Значение заголовка уникально идентифицирует компьютер внутри организации.

X-Sensor-UID-AdapterType: if_type_ethernet_csmacd
Данный заголовок формируется при взаимодействии сервера EtherSensor и экземпляров EtherSensor Agent, установленных на рабочих станциях пользователей сети.

Значение заголовка содержит тип сетевого адаптера, через который было отправлено сообщение. Полный список возможных вариантов значений данного поля доступен на сайте компании Microsoft.

X-Sensor-UID-MacAddress: 00-1F-C6-2D-EA-40
Данный заголовок формируется при взаимодействии сервера EtherSensor и экземпляров EtherSensor Agent, установленных на рабочих станциях пользователей сети.

Значение заголовка содержит MAC-адрес сетевого адаптера, через который было отправлено сообщение.

X-Sensor-UHID: UO2D-RNVO-JRN7-R1EN-91C0-61TA-1HP7-YRVF
Содержит уникальный идентификатор для каждого экземпляра EtherSensor и набора оборудования (Unique Hardware IDentifier).

X-Sensor-Lotus-MessageId: <OF4D026078.B21F0C4F-ON44257C15.002E1625-44257C15.002E2225@LocalDomain>
Содержит уникальный идентификатор сообщения, передаваемого по протоколу Lotus Notes.

X-Sensor-Lotus-Form: Reply
Содержит имя формы передаваемого по протоколу LOTUS сообщения.

X-Sensor-Lotus-Mailer: Lotus Notes Release 8.5.2FP2 SHF236 October 24, 2011
Содержит строку, идентифицирующую тип клиента системы Lotus Notes.

X-Sensor-Lotus-INetPrincipal: UserName/OU/O@ServerName.Domain.com
Содержит расширенную информацию об отправителе сообщения.

X-Sensor-Lotus-RouteServers: CN=lotus1/O=Company
Содержит список серверов Lotus Notes, которые участвовали в передаче сообщения.

X-Sensor-Lotus-References: <OF02B9DAC2.33CDF581-ON44257C15.002DF8CD@LocalDomain>
Содержит список идентификаторов сообщений, на которые ссылается текущее сообщение.

4.3. Фильтрация результатов перехвата

Основная идея обработки объектов/событий состоит в формировании цепочек правил, которые объединяются в таблицы. Объект проходит проверку правилами, которые, в зависимости от срабатывания правила, могут изменять содержимое объекта, его метаданных, или ход его дальнейшей обработки.

Эта идея очень схожа с правилами фильтрации, используемыми в iptables.

Использование фильтров позволяет управлять процессом обработки событий, направлять события в различные внешние системы-потребители для дальнейшего анализа, а также удалять заведомо не интересные события с целью снижения нагрузки на EtherSensor.

4.3.1. Основы фильтрации

Ключевыми понятиями фильтрации перехваченных сообщений/объектов являются:

Критерий

Логическое выражение, состоящее из одного или более условий, объединённых логическими операторами OR, AND, XOR и NOT, проверяющее содержимое и/или метаданные сообщения и определяющее, подпадает ли данный конкретный объект под действие текущего правила.

Условие

Элементарное условие проверки сообщения и/или его метаданных. Проверяет одно или несколько полей сообщения или его метаданных единообразным способом на соответствие некому условию (равенству, неравенству, совпадению с паттерном и т.д.).

Действие

Описание действия, которое необходимо выполнить над сообщением в случае срабатывания правила. Более подробно о возможных действиях над сообщениями указано в разделе Действия 140.

Правило

Состоит из условий и действий. Если сообщение соответствует условиям, к нему применяются действия. Условий может и не быть — тогда неявно предполагается критерий "все сообщения" или "любое сообщение".

Таблица

Упорядоченная последовательность правил. Таблица должна иметь уникальное имя. Существует одна системная таблица с именем "main", она является точкой входа для проверки сообщения фильтром. Таблица "main" должна присутствовать в каждом фильтре. Более подробно таблицы рассматриваются в разделе Таблицы 112.

Принципы работы фильтров

Все сообщения проходят через таблицы фильтра, начиная с таблицы "main". При прохождении сообщением таблицы к нему последовательно применяются все правила этой таблицы в порядке их следования.

Под применением правила понимается:

- Проверка сообщения (включая метаданные) на соответствие условиям
- Применение к нему действий правила, если сообщение соответствует этому условию.

Под действием может подразумеваться как базовая операция (встроенное действие, например, ACCEPT, DROP, JUMP или RETURN), так и ориентированное на пользователя действие (это действия установки меток, тегов, изменения сообщения и/или его метаданных и т.п.). Базовые операции могут быть как терминирующими, то есть прекращающими обработку сообщения фильтром (это ACCEPT, DROP), так и нетерминирующими, то есть не прерывающими обработку сообщения (это JUMP и RETURN).

Таблица "main" всегда должна заканчиваться правилом с критерием "все сообщения" и одним из терминирующих действий (ACCEPT или DROP). Остальные таблицы всегда обязаны заканчиваться правилом с условием "все сообщения" и терминальным действием (ACCEPT или DROP), или же действием RETURN.

Для перехода сообщения из текущей таблицы в другую используется действие JUMP. В случае применения в этой таблице к сообщению действия RETURN[144], сообщение вернется в исходную таблицу и продолжит ее прохождение, начиная со следующего правила. Другие таблицы, кроме "main", и переходы в них могут понадобиться для минимизации количества правил, через которые необходимо пройти сообщению, чтобы для него было принято решение ACCEPT или DROP.

Также использование таблиц может быть полезно при объединении групп правил для обработки сообщений, объединяемых некими "глобальными" категориями, которые невозможно описать в контексте критерия одного правила.

> Исходя из этого следует, что таблица main не может заканчиваться правилом с действием RETURN, так как из неё просто некуда возвращаться. Также в таблицу main невозможно сделать переход JUMP.

Следует особо подчеркнуть, что запрещено использовать явные или неявные "циклические" переходы, и это проверяется на этапе компиляции правил. Например, сценарий "таблица "main" -> jump -> таблица "boss-messages" -> таблица "shopping" -> таблица "spam" -> таблица "boss-messages" будет заблокирован на стадии компиляции фильтра.

4.3.1.1. Конфигурация фильтра

Настройки фильтра хранятся в XML-файле, имеющем определённую структуру.

Файл конфигурации фильтра начинается со стандартного тега начала XML-документа с указанием версии XML 1.0 и кодировки документа.

Пример:

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

Далее указывается корневой тег filter, который содержит настройки фильтра. Корневой тег имеет атрибуты "name" – краткое имя фильтра, и "version" – номер версии фильтра. Сейчас это только "1.0".

Пример:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="main filter" version="1.0">
    ...
</filter>
```

Для фильтра можно написать комментарий – текстовое описание для пояснения содержания и назначения фильтра. Комментарии могут содержать любой текст и при работе фильтра игнорируются. Комментарий добавляется в качестве отдельного тега <comment>.

Пример:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="main filter" version="1.0">
    <comment>This is a comment.</comment>
    ...
</filter>
```

Точно так же комментарии можно указывать для таблиц и правил.

4.3.1.2. Таблицы

Фильтр всегда состоит из таблицы "main" и, возможно, других таблиц.

Обработка сообщения фильтром всегда начинается с таблицы "main". Таблица определяется в фильтре тегом "table". Каждая таблица должна иметь уникальное имя, указываемое в атрибуте "name" XML-тега "table". Таблица может иметь в качестве необязательного комментария XML-тег "comment".

Пример:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="main filter" version="1.0">
  <comment>This is a comment.</comment>
    <table name="main">
      <comment>This is a comment for the table "main".</comment>
        ...
    </table>

    <table name="spam">
        ...
    </table>

</filter>
```

В данном фильтре указаны две таблицы – обязательная таблица "main" и дополнительная таблица "spam".

4.3.1.3. Правила

Правила состоят из критерия, состоящего из одного или более условий, и одного или нескольких действий, выполняемых в случае соответствия сообщения данному набору критериев.

Правила определяются внутри таблиц при помощи XML-тега "rule".

Правило может иметь необязательное имя, указываемое в атрибуте "name" XML-тега "rule".

Правило может иметь необязательный комментарий – XML-тег "comment".

Правило может быть активным – будет выполняться в текущей конфигурации фильтра, или неактивным – в ходе выполнения текущей конфигурации фильтра оно будет игнорироваться. Активность правила определяется обязательным атрибутом "enabled" XML-тега "rule". Атрибут "enabled" может принимать следующие значения:

1 или true:

Правило включено, участвует в фильтрации сообщений

0 или false:

Правило выключено, не участвует в фильтрации сообщений, игнорируется.

Критерий правила описывается XML-тегом "match" внутри тега "rule". Действия правила описывается последовательностью XML-тегов "action" внутри тега "rule".

Пример:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="main filter" version="1.0">
  <comment>This is a comment.</comment>

  <table name="main">
    <comment>This is a comment for the table "main".</comment>
    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>

    <rule name="spam" enabled="1">
      <comment>The rule for the messages of the SPAM category.</comment>
      <match ...> ... </match>
      <action ...> ... </action>
      <action ...> ... </action>
      <action ...> ... </action>
    </rule>

    <rule enabled="1">
      <action name="drop" />
    </rule>

  </table>
</filter>
```

В этом примере в таблице "main" есть три правила, все правила включены. Второе правило (в отличие от первого) имеет имя, комментарий и несколько действий. Третье правило – это обязательное терминирующее правило: "отклонить все сообщения, не принятые правилами выше".

Если критерий "match" отсутствует в правиле или является пустым (<match />), то неявно подразумевается критерий "все сообщения": <c name="all"/>.

4.3.1.3.1. Критерии и условия

Критерий в правиле описывается XML-тегом "match" и определяет, будут ли выполняться действия правила для данного сообщения.

Критерий в правиле состоит из одного или нескольких условий, которые связаны логическими операциями AND, OR, XOR или NOT. Это позволяет строить критерий на основе логических выражений, состоящих из условий.

Пример:

```
<rule>
  <match>
    <c name="all" />
  </match>
  <action name="drop" />
</rule>
```

Данный критерий состоит из единственного условия: "все сообщения".

Условия в критериях

Условие – это элементарная проверка сообщения или его метаданных на соответствие чему-либо.

Условие описывается XML-тегом "condition" или "c". Сообщение либо удовлетворяет условию, и в этом случае условие имеет значение TRUE, либо не удовлетворяет условию, и в таком случае результатом проверки условия будет FALSE. Для условия указывается его имя в атрибуте "name", которое определяет, что и как проверять в сообщении. В прочих атрибутах тега указываются параметры для выполнения действия. Имена атрибутов дополнительных параметров зависят от конкретных условий.

Общая структура XML-тега, описывающего условие:

```
<condition name="The name of the condition." [additional parameters] />
```

или

```
<condition name="The name of the condition." [additional parameters] >
</condition>
```

или

```
<c name="The name of the condition." [additional parameters] />
```

или

```
<c name="The name of the condition." [additional parameters] ></c>
```

Большинство условий использует атрибут value="...", в котором указываются значения для проверки на соответствие условию, или атрибут data="...", в котором можно указывать внешний файл для загрузки данных для такой проверки (это могут быть наборы слов, списки доменных имён и другие параметры). Какие атрибуты и как именно проверяются, описано в разделах соответствующих условий.

Логические выражения в критериях

В случае, когда необходимо создать для правила сложный критерий, состоящий из нескольких условий, эти условия можно объединять в логические выражения с использованием логических

операций AND, OR, XOR, NOT. Логические операции в виде XML-тегов выглядят следующим образом:

<and> … </and>:

Соответствует (... & ... & ... & ...) – все условия, находящиеся внутри тега "and", объединяются логической операцией AND.

<or> … </or>:

Соответствует (... | ... | ... | ...) – все условия, находящиеся внутри тега "or", объединяются логической операцией OR.

<xor> … </xor>:

Соответствует (... ^ ... ^ ... ^ ...) – все условия, находящиеся внутри тега "xor", объединяются логической операцией XOR.

<not> … </not>:

Соответствует !(...) – к условию применяется операция отрицания.

Пример:

```
<and>
  <c name="ccc1"/>
  <c name="ccc2"/>
</and>
```

означает результат (ccc1 & ccc2),

а критерий:

```
<not><c name="ccc1"/></not>
```

означает результат "не (ccc1)".

Любые теги логических операций могут быть вложенными.

Пример:

```
<and>
  <c name="ccc1"/>
  <or>
    <c name="ccc2">
    <c name="ccc3">
  </or>
  <or>
    <c name="ccc4">
    <c name="ccc5">
    <not><c name="ccc6"></not>
  </or>
</and>
```

означает результат (ccc1 & (ccc2 | ccc3) & (ccc4 | ccc5 | !ccc6)).

То есть сообщение будет удовлетворять указанному в примере критерию, если: оно удовлетворяет условию ccc1, а также удовлетворяет (условию ccc2 или условию ccc3), а также (удовлетворяет (условию ccc4 или условию ccc5) или не удовлетворяет условию ccc6).

Для наглядности можно разбить данный критерий на блоки условий, которые обязательно должны возвращать TRUE при следующих проверках:

1) сообщение должно удовлетворять условию ccc1

И

2) сообщение должно удовлетворять одному из условий (ccc2 или ccc3)

И

3) сообщение должно удовлетворять одному из условий (ccc4 или ccc5) ИЛИ не удовлетворять условию ccc6.

4.3.1.3.1.1. Условие ALL, *

Специальное условие, которому удовлетворяют все фильтруемые объекты.

Описание

Это условие является истинным для любых объектов, неявно подразумевается, если критерий <match>...</match> является пустым или отсутствует в правиле (правило содержит только теги "action").

Формат

```
<c name="all" />
<c name="*" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="all" или name="*".

Пример:

Правило принимает для дальнейшей обработки все сообщения.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>

  <table name="main">

    <rule enabled="1">
      <comment>This rule accepts all messages for further processing.</comment>
      <match>
        <c name="all" />
      </match>
      <action name="accept" />
    </rule>
  </table>

</filter>
```

Пример (неявное условие all):

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter comment.</comment>
  <table name="main">
    <rule enabled="1">
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.1.3.1.2. Условие DETECTOR

Проверить имя детектора, который идентифицировал сообщение.

Описание

При срабатывании детектора система сохраняет его имя в метаданных сообщения. Условие DETECTOR позволяет выяснить на этапе фильтрации сообщения, какой именно детектор на него сработал (он может быть только один).

Формат

```
<c name="detector" value="[detector-name(s)]" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="detector".

Атрибут "value":

В атрибуте value="..." укажите имя, с которым сравнивается имя детектора, сработавшего на сообщение. Имён детекторов может быть несколько (логическое ИЛИ), в этом случае они перечисляются через запятую ','. Для читаемости позволяется после запятой ставить пробелы.

Пример:

Сообщения от детекторов mail.ru, yandex.ru.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Messages filter.</comment>
  <table name="main">

    <rule enabled="1">
      <comment>
        Messages from detectors mail.ru,
        yandex.ru.
      </comment>
      <match>
        <c name="detector" value="mail.ru, yandex.ru" />
      </match>
      <action name="drop" />
    </rule>

  </table>
```

4.3.1.3.1.3. Условие PROTOCOL

Проверить, по какому протоколу было получено сообщение.

Описание

Это условие проверяет, по какому протоколу было получено сообщение.

Формат

```
<c name="protocol" value="[protocol-name(s)]" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия – name="protocol".

Атрибут "value":

В атрибуте value="..." укажите имя, с которым сравнивается имя протокола, по которому получено сообщение.

Проверяемых протоколов может быть несколько (логическое ИЛИ). В этом случае они перечисляются через запятую ','. Для читаемости допустимо после запятой ставить пробелы.

Пример:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Messages filter.</comment>
  <table name="main">
    <rule enabled="1">
      <comment>
        Drop messages received by SMTP or IMAP.
      </comment>
      <match>
        <c name="protocol" value="smtp, imap" />
      </match>
      <action name="drop" />
    </rule>
    </table>
</filter>
```

4.3.1.3.1.4. Условие MSG-SIZE, TOTAL-SIZE

Проверить размер сообщения.

Описание

Условия MSG-SIZE и TOTAL-SIZE проверяют размер сообщения.

msg-size

Учитывает размер извлечённых текстов и размер извлечённых аттачментов

total-size

Учитывает суммарный размер извлечённых текстов, размер извлечённых аттачментов и размер исходных данных, из которых они получены.

Формат

```
<c name="msg-size" op="<operation>" value="<compare pattern>" />
<c name="total-size" op="<operation>" value="<compare pattern>" />
```

Атрибут "name":

В атрибуте "name" укамите имя условия: name="msg-size" или name="total-size".

Атрибут "value":

В атрибуте value="..." укажите число, с которым сравнивается размер сообщения:

<число> или <число>B

Указывает размер в байтах

<число>K

Указывает размер в килобайтах

<число>M

Указывает размер в мегабайтах

<число>G

Указывает размер в гигабайтах.

Атрибут "op":

Атрибут op="..." указывает на тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если размер РАВЕН указанному числу

ne или != или <>

Условие выполняется, если размер НЕ РАВЕН указанному числу

lt или &lt

Условие выполняется, если размер МЕНЬШЕ указанного числа

gt или >

Условие выполняется, если размер БОЛЬШЕ указанного числа

le или <=

Условие выполняется, если размер МЕНЬШЕ ИЛИ РАВНО указанного числа

ge или >=

Условие выполняется, если размер БОЛЬШЕ ИЛИ РАВНО указанного числа.

Пример:

Правило прекращает обработку сообщений, у которых размер без учёта исходных данных больше 100КБ или размер с учётом исходных данных больше 1МБ.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Messages filter.</comment>

  <table name="main">

      <rule enabled="1">
        <match>
          <or>
            <c name="msg-size" op=">" value="100K"/>
            <c name="total-size" op="gt" value="1M"/>
          </or>
        </match>
        <action name="drop" />
      </rule>

    </table>
</filter>
```

4.3.1.3.1.5. Условие CHECK-MD5

Проверить MD5-хэш сообщения на повторное появление в течение определённого интервала времени (отслеживание дубликатов сообщений).

Описание

Это условие проверяет, не встречалось ли уже в течение указанного периода времени сообщение с таким же MD5 хэшем, как и у текущего проверяемого сообщения.

Формат

```
<c name="check-md5" time="<timeout>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="check-md5".

Атрибут "time":

В атрибуте time="..." укажите время ожидания в миллисекундах.

Время ожидания не может быть меньше 1 миллисекунды и больше 5 минут, т.е. 5*60*1000 = 300000 миллисекунд.

Пример:

Если в течение двух секунд были повторные сообщения с таким же MD5, то удалять их.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
      <rule enabled="1">
        <match>
          <c name="check-md5" time="2000" />
        </match>
        <action name="drop" />
      </rule>

  </table>
</filter>
```

4.3.1.3.1.6. Условие CHECK-MESSAGE-ID

Проверить заголовок Message-ID (для протокола LOTUS проверяется заголовок X-Sensor-Lotus-MessageId) сообщения на повторное появление в течение определённого интервала времени (отслеживание дубликатов сообщений).

Описание

Это условие проверяет, не встречалось ли уже в течение указанного периода времени сообщение с таким же заголовком Message-ID (для протокола LOTUS проверяется заголовок X-Sensor-Lotus-MessageId), как и у текущего проверяемого сообщения.

Формат

```
<c name="check-message-id" time="<timeout>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="check-message-id".

Атрибут "time":

В атрибуте time="..." укажите время ожидания в миллисекундах.

Время ожидания не может быть меньше 1 миллисекунды и больше 5 минут, т.е. 5*60*1000 = 300000 миллисекунд.

Пример:

Если в течение двух секунд были повторные сообщения с таким же заголовком Message-ID, удалять их.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>
  <table name="main">
    <rule enabled="1">
      <match>
        <c name="check-message-id" time="2000" />
      </match>
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.7. Условие HOSTNAME

Проверить имена хостов источника и назначения в сообщении.

Описание

Это условие проверяет имена хостов источника или назначения на соответствие строке или на соответствие шаблону wildcard или regexp. Для корректной полнофункциональной работы условия необходимо произвести разрешения имён хостов (см. раздел Действие DNS[150]).

> Совет: если необходимо проверить только хост назначения и только для протокола HTTP, то разрешение имён через "дорогое" с точки зрения затрат времени и ресурсов действие DNS не имеет смысла, так как имя хоста назначения будет доступно из заголовка Host в HTTP-запросе.

Формат

```
<c name="hostname"
  address="<address type>"
  op="<operation>"
  value="<compare pattern>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="hostname".

Атрибут "address":

В атрибуте address="..." укажите тип адреса для проверки. Возможные значения:

src или client

Проверять только адрес источника

dst или server

Проверять только адрес назначения

both или all или *

Проверять оба адреса (и источника и назначения).

Если этот атрибут отсутствует, то подразумевается "both" – проверять оба адреса.

Атрибут "op":

Атрибут op="..." указывает на тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если проверяемое значение СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если проверяемое значение НЕ СОДЕРЖИТ указанное значение

wc или wildcard

Условие выполняется, если проверяемое значение соответствует указанному wildcard-шаблону

re или regex или regexp

Условие выполняется, если проверяемое значение соответствует указанному regexp-шаблону

Атрибут "value":

В атрибуте value="..." укажите строку, с которой сравнивается значение, или шаблон для проверки.

Пример:

Сообщения, отправляемые на хосты *.yandex.ru, игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match>
        <c name="hostname" address="server" op="wc" value="*.yandex.ru" />
      </match>
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.8. Условие IP

Проверить IP-адреса клиента или сервера на вхождение в диапазон или на принадлежность к подсети.

Описание

Это условие проверяет IP-адреса клиента или сервера на вхождение в диапазон или на принадлежность к подсети.

> Советы:
>
> 1. Нежелательный трафик нужно отсекать как можно раньше. От этого зависит производительность EtherSensor.
>
> 2. Весь трафик с некоторого IP или диапазона лучше всего отсекать в IP-фильтре службы EtherSensor EtherCAP.
>
> 3. Определённый трафик HTTP с некоторого IP (если возможно определить такие критерии) лучше всего отсекать в HTTP-фильтре.
>
> 4. Определённые сообщения с некоторого IP-адреса необходимо обрабатывать в фильтре сообщений.

Формат

```
<c name="ip" address="<address type>" value="<ip-range>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="ip".

Атрибут "address":

В атрибуте address="..." укажите тип адреса для проверки. Возможные значения:

src или client

Проверять адрес источника

dst или server

Проверять адрес назначения

any или *

Соответствие любого из адресов.

Если атрибут пропущен, то подразумевается по умолчанию "*".

Атрибут "value":

В атрибуте value="..." укажите значение для сравнения. Возможные значения:

ipaddress

Проверяет IP-адрес на равенство. Например, value="192.168.0.10"

ip1-ip2

Проверяет на вхождение IP-адреса в диапазон. Например, value="192.168.0.1-192.168.0.10"

ip/netmask

Проверяет на принадлежность IP-адреса указанной подсети. Например, value="192.168.0.1/255.255.255.0"

ip/netmaskbits

Проверяет на принадлежность IP-адреса указанной подсети. Например, value="192.168.0.1/24".

Пример:

Сообщения от клиента с машины 192.168.0.15 игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">

      <rule enabled="1">
        <match>
          <c name="ip" address="client" value="192.168.0.15" />
        </match>
        <action name="drop" />
      </rule>

      <rule enabled="1">
        <action name="accept" />
      </rule>

  </table>
</filter>
```

4.3.1.3.1.9. Условие HEADER

Проверяет значение одного из заголовков сообщения.

Описание

Это условие проверяет значение заголовка сообщения на наличие подстроки в строке или на соответствие шаблону wildcard или regexp. Заголовки сообщения – это любые генерируемые детекторами заголовки метаданных вида "X-Sensor-...".

Также это заголовки из почтовых сообщений, кроме заголовков:

From

To

Cc

Bcc

Subject

Date

Content-Type

Content-Transfer-Encoding

Следует помнить, что для более эффективной фильтрации не стоит фильтровать заголовки метаданных "X-Sensor-...", для которых есть специализированные условия проверки. Например, для заголовка "X-Sensor-Detector" более эффективно проверять не значение этого заголовка через условие HEADER, а использовать специальное условие <c name="detector" value="..." />

Формат

```
<c name="header"
   headername="<header name>"
   op="<operation>"
   value="<compare pattern>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="header".

Атрибут "headername":

В атрибуте headername="..." укажите имя проверяемого заголовка.

Атрибут "op":

Атрибут op="..." указывает на тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если проверяемое значение СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если проверяемое значение НЕ СОДЕРЖИТ указанное значение

wc или wildcard

Условие выполняется, если проверяемое значение соответствует указанному wildcard-шаблону

re или regex или regexp

Условие выполняется, если проверяемое значение соответствует указанному regexp-шаблону

Атрибут "value":

В атрибуте value="..." указывается строка, с которой сравнивается значение, или шаблон для проверки.

Пример:

```
<c name="header" headername="X-Priority" op="eq" value="3" />
```

Условие выполняется, если в сообщении присутствует заголовок X-Priority и его значение содержит строку "3".

```
<c name="header" headername="X-Mailer" op="!=" value="Outlook" />
```

Условие выполняется, если в сообщении присутствует заголовок X-Mailer и его значение не содержит строку "Outlook".

```
<c name="header"
   headername="X-Sensor-Net-Interface-Id"
   op="eq"
   value="01-icap" />
```

Условие выполняется, если в сообщении присутствует заголовок X-Sensor-Net-Interface-Id и его значение содержит строку "01-icap".

```
<c name="header"
   headername="X-Sensor-Net-Interface-Id"
   op="wc"
   value="*-icap" />
```

или

```
<c name="header"
   headername="X-Sensor-Net-Interface-Id"
   op="wildcard"
   value="*-icap" />
```

Условие выполняется, если в сообщении присутствует заголовок X-Sensor-Net-Interface-Id и его значение соответствует wildcard-шаблону "*-icap".

Пример:

Сообщения, отправляемые через Outlook, игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match>
        <c name="header"
           headername="X-Mailer"
           op="=="
           value="Outlook" />
      </match>
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.10. Условие ATTACH-NAME

Проверить имена аттачментов в сообщении.

Описание

Это условие проверяет имена аттачментов на соответствие строке или на соответствие шаблону wildcard или regexp.

Формат

```
<c name="attach-name" op="<operation>" value="<compare pattern>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="attach-name".

Атрибут "op":

В атрибуте op="..." укажите тип операции сравнения. Возможные значения:

eq или = или ==

Условие выполняется, если проверяемое значение СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если проверяемое значение НЕ СОДЕРЖИТ указанное значение

wc или wildcard

Условие выполняется, если проверяемое значение соответствует указанному wildcard-шаблону

re или regex или regexp

Условие выполняется, если проверяемое значение соответствует указанному regexp-шаблону

Атрибут "value":

В атрибуте value="..." укажите строку, с которой сравнивается значение, или шаблон для проверки.

Пример:

```
<c name="attach-name" op="eq" value="instruction.doc" />
```

Условие выполняется, если какое-либо имя аттачмента сообщения содержит "instruction.doc".

```
<c name="attach-name" op="eq" value="instruction.doc" />
```

Условие выполняется если какое-либо имя аттачмента сообщения не содержит "instruction.doc".

```
<c name="attach-name" op="wc" value="*.doc" />
```

или

```
<c name="attach-name" op="wildcard" value="*.doc" />
```

Условие выполняется, если какое-либо имя аттачмента сообщения соответствует wildcard-шаблону "*.doc".

```
<c name="attach-name" op="re" value=".+\.doc" />
```

или

```
<c name="attach-name" op="regexp" value=".+\.doc" />
```

Условие выполняется, если какое-либо имя аттачмента сообщения соответствует regex-шаблону ".+\.doc".

```
<c name="attach-name" op="re" value=".+((\.doc)|(\.exe)|(\.zip))" />
```

Условие выполняется если какое-либо имя аттачмента сообщения соответствует regex-шаблону ".+((\.doc)|(\.exe)|(\.zip))".

Пример:

Сообщения с аттачментами *.exe игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match>
        <c name="attach-name" op="re" value=".+\.exe" />
      </match>
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.11. Условие ATTACH-EXIST

Проверить сообщение на наличие аттачментов.

Описание

Это условие, которому удовлетворяют все сообщения, содержащие любые аттачменты.

Формат

```
<c name="attach-exist" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="attach-exist".

Пример:

Сообщения с любыми аттачментами игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match>
        <c name="attach-exist" />
      </match>
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.12. Условие TAG

Проверить наличие установленного тега и значение его счётчика (см. раздел Действие TAG [147]).

Описание

Это условие проверяет существование тега или значение счётчика тега.

Если тег отсутствует, условие не выполняется.

Формат

```
<c name="tag" op="<operation>" value="<compare value>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="tag".

Атрибут "op":

Атрибут op="..." указывает на критерий проверки:

eq или = или ==

Условие выполняется, если значение счётчика РАВНО указанному числу

ne или != или <>

Условие выполняется, если значение счётчика НЕ РАВНО указанному числу

lt или &lt

Условие выполняется, если значение счётчика МЕНЬШЕ указанного числа

gt или >

Условие выполняется, если значение счётчика БОЛЬШЕ указанного числа

le или <=

Условие выполняется, если значение счётчика МЕНЬШЕ ИЛИ РАВНО указанного числа

ge или >=

Условие выполняется, если значение счётчика БОЛЬШЕ ИЛИ РАВНО указанного числа

exist

Условие выполняется, если тег существует (уже установлен ранее для этого объекта).

По умолчанию (если атрибут "op" отсутствует) принимается значение "exist". Для операции "exist" указывать атрибут "value" не обязательно.

Атрибут "value":

В атрибуте value="..." укажите число, с которым сравнивается значение счётчика тега.

Пример:

```
<c name="tag" tag="SPAM" op="exist" />
```

или

```
<c name="tag" tag="SPAM" />
```

Условие выполняется, если для объекта установлен тег "SPAM".

```
<c name="tag" tag="SPAM" op="eq" value="1" />
```

Условие выполняется, если для сообщения установлен тег "SPAM" и его счётчик равен 1.

```
<c name="tag" tag="SPAM" op=">" value="1" />
```

Условие выполняется, если для сообщения установлен тег "SPAM" и его значение его счётчика больше 1.

```
<c name="tag" tag="SPAM" op=">=" value="3" />
```

Условие выполняется, если для сообщения установлен тег "SPAM" и его значение его счётчика больше или равно 3.

Пример:

Сообщения, помеченные тегом SPAM со значением больше 1, игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match>
        <c name="tag" tag="SPAM" op=">" value="1" />
      </match>
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.13. Условие FROM, TO, CC, BCC, ADDRESS, SUBJECT

Проверить значение одного из полей from, to, cc, bcc, subject, address.

Описание

Это условие проверяет значение поля на содержание указанной подстроки или на соответствие шаблону wildcard или regexp.

from

Проверяет поле FROM (адрес отправителя)

to

Проверяет поле TO (адрес получателя)

cc

Проверяет поле CC

bcc

Проверяет поле BCC

address

Проверяет все поля адресов (from, to, cc, bcc). Если в любом из них найдено соответствие, то условие выполняется

subject

Проверяет поле SUBJECT.

Формат

```
<c name="from" op="<operation>" value="<compare pattern>" />
<c name="to" op="<operation>" value="<compare pattern>" />
<c name="cc" op="<operation>" value="<compare pattern>" />
<c name="bcc" op="<operation>" value="<compare pattern>" />
<c name="subject" op="<operation>" value="<compare pattern>" />
<c name="address" op="<operation>" value="<compare pattern>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="from", name="to", name="cc", name="bcc", name="subject" или name="address"

Атрибут "op":

Атрибут op="..." указывает на тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если значение поля СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если значение поля НЕ СОДЕРЖИТ указанное значение

wc или wildcard

Условие выполняется, если значение поля соответствует указанному wildcard-шаблону

re или regex или regexp

Условие выполняется, если значение поля соответствует указанному regexp-шаблону.

Атрибут "value":

В атрибуте value="..." указывается строка, с которой сравнивается значение, или шаблон для проверки.

Пример:

```
<c name="from" op="eq" value="xxx@mail.ru" />
```

Условие выполняется, если поле FROM сообщения содержит строку "xxx@mail.ru".

```
<c name="to" op="!=" value="xxx@mail.ru" />
```

Условие выполняется, если поле TO сообщения не содержит строку "xxx@mail.ru".

```
<c name="cc" op="wc" value="*@mail.ru" />
```

или

```
<c name="cc" op="wildcard" value="*@mail.ru" />
```

Условие выполняется, если поле CC сообщения соответствует wildcard-шаблону "*@mail.ru".

```
<c name="address" op="re" value=".+@mail.ru" />
```

или

```
<c name="address" op="regexp" value=".+@mail.ru" />
```

Условие выполняется, если любое из адресных полей сообщения (FROM, TO, CC или BCC) соответствует regex-шаблону "+@mail.ru".

```
<c name="subject" op="re" value=".*((badword1)|(badword2)|(badword3)).*" />
```

Условие выполняется, если в теме сообщения найдено соответствие regex-шаблону ".*((badword1)|(badword2)|(badword3)).*".

Проще говоря – условие выполняется, если в теме есть слова badword1, badword2 или badword3.

```
<c name="subject" op="re"
   value="(\+?\d([ \-\(])?\d{3}([ \-\)])?)?([ -])?((\d{7})|
          (\d{3}([ \-])?\d{2}([ \-])?\d{2})|
          (\d{2}([ \-])?\d{3}([ \-])?\d{2}))" />
```

Условие выполняется, если в теме сообщения найден номер телефона (строка, соответствующая regexp).

Телефонный номер в этом случае может соответствовать одному из следующих форматов:

1234567

123 45 67

12 345 67

89031234567

8(903)1234567

8-903-123-45-67

8 903 123 45 67

+79031234567

+7(903)1234567

+7-903-123-45-67

+7 903 123 45 67

Пример:

Письма с адреса или на адрес *@mail.ru продолжить обрабатывать, всё остальное игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">

    <rule enabled="1">
      <match>
        <c name="address" op="wildcard" value="*@mail.ru" />
      </match>
      <action name="accept" />
    </rule>

    <rule enabled="1">
      <action name="drop" />
    </rule>

  </table>
</filter>
```

4.3.1.3.1.14. Условие TEXT

Проверить наличие ключевых слов в тексте сообщения или в его теме.

Описание

Это условие проверяет наличие ключевых слов в тексте сообщения или теме.

Проверка наличия наличия ключевых слов происходит по общему вхождению ключевого слова в текст, а не только как отдельные слова. Другими словами, это работает так, как будто для каждого ключевого слова ищется wildcard-шаблон "*ключевое слово*".

Например, под критерий ключевого слова "secret" будут попадать слова "secret", "sECrEt", "secretary", "insecretory" и т.п.

При поиске ключевых слов регистр символов не учитывается.

Формат

```
<c name="text" op="<operation>" value="<compare pattern>" />
<c name="text" op="<operation>" data="<data source>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="text".

Атрибут "op":

Атрибут op="..." указывает на критерий проверки:

all

В сообщении должны быть найдены все указанные ключевые слова

one

Достаточно, чтобы в сообщении было найдено хотя бы одно ключевое слово.

По умолчанию (если атрибут "op" отсутствует) принимается значение "one".

Атрибут "value":

В атрибуте value="..." перечислите ключевые слова. Если ключевых слов несколько, то перечислите их через запятую ','. Также ключевые слова могут быть указаны в значении самого тега <c>key-word-list</c>

Атрибут "data":

В атрибуте data="..." может указываться другой источник ключевых слов. Это позволяет указывать большие наборы слов, когда это неудобно делать в атрибуте value="...".

Возможные значения:

data="<external data name>"

Загрузить список ключевых слов из внешнего блока в фильтре (тег <data name="extern-data-name">...</data>)

data="extern://<external data name>"

Загрузить список ключевых слов из внешнего блока в фильтре (тег <data name="extern-data-name">...</data>). Ключевые слова перечисляются через запятую

data="file://<full-file-path>"

Загрузить список ключевых слов из указанного файла. Ключевые слова перечисляются каждое на отдельной строке (запятые не нужны).

Пример:

```
<c name="text" op="one" value="secret1, secret2, secret3, secret4" />
```

или

```
<c name="text" op="one">secret1, secret2, secret3, secret4</c>
```

Условие выполняется, если в теме сообщения или в его тексте найдётся хотя бы одно из указанных ключевых слов: как самостоятельных слов, так и как часть другого слова.

Примеры текстов:

1. Тема письма – "RE: secret 1 secret2" – условие ВЫПОЛНЯЕТСЯ.
2. Тема письма – "RE: secret3 secret4" – условие ВЫПОЛНЯЕТСЯ.
3. Текст письма – "посылаю вам наш secret 1 secret2" – условие ВЫПОЛНЯЕТСЯ.
4. Тема письма – "The quick brown fox jumps over the lazy dog" – условие НЕ ВЫПОЛНЯЕТСЯ.

```
<c name="text" op="all" value="secret1, secret2" />
```

или

```
<c name="text" op="all">secret1, secret2</c>
```

Условие выполняется, если в теме сообщения или в его тексте найдутся оба указанных ключевых слова как самостоятельные слова, или же, как части других слов.

Примеры текстов:

1. Тема письма – "RE: secret1 secret2" – условие ВЫПОЛНЯЕТСЯ.
2. Тема письма – "RE: secret1 от буха" – условие НЕ ВЫПОЛНЯЕТСЯ.
3. Текст письма – "посылаю вам наш secret2" – условие НЕ ВЫПОЛНЯЕТСЯ.
4. Текст письма – "посылаю вам наш secret1 от secret2" – условие ВЫПОЛНЯЕТСЯ.

```
<c name="text" op="one" data="dictionary.txt" />
```

Загрузить список ключевых слов для проверки из файла "dictionary.txt".

Пример:

Письма, содержащие ключевые слова из списка keywords, принять для дальнейшей обработки. Всё остальное игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">

    <rule enabled="1">
      <comment></comment>
      <match>
        <c name="text" data="extern://keywords" />
      </match>
      <action name="accept" />
    </rule>

    <rule enabled="1">
      <action name="drop" />
    </rule>

  </table>

  <data name="keywords">
        secret1,
        secret2,
        secret3,
        secret4
  </data>

</filter>
```

4.3.1.3.2. Действия

Все действия описываются в XML-тегах "action". Для каждого действия указывается его имя в атрибуте "name", которое определяет, какая операция будет выполнена для сообщения. В прочих атрибутах тега указываются параметры для выполнения действия. Имена атрибутов дополнительных параметров зависят от конкретных действий.

Общая структура XML-тега, описывающего действие:

```
<action name="action name" [parameters] />
```

или

```
<action name="action name" [parameters]></action>
```

Большинство действий использует атрибут value="...", в котором указываются значения для выполнения действия, или атрибут data="...", в котором можно указывать внешний файл для загрузки данных для выполнения действия (это могут быть наборы слов, списки доменных имён и другие параметры). Описание атрибутов и их действий подробно указано в разделах соответствующих действий.

Базовые действия

Базовые действия – это действия, изменяющие порядок прохождения сообщения через фильтр. К базовым действиям относятся ACCEPT, DROP, JUMP и RETURN.

Пользовательские действия

Пользовательские действия – это действия, изменяющие сообщение или его метаданные, а также выполняющие обработку сообщения через внешние сервисы (песочницы, антивирусы, URL-категоризаторы, DNSBL-серверы и т.п.).

4.3.1.3.2.1. Действие ACCEPT

Принять сообщение к дальнейшей обработке.

Описание

Действие ACCEPT прекращает работу фильтров с текущим сообщением и немедленно пропускает его к дальнейшей обработке (то есть, к доставке его системе-потребителю согласно предопределенному профилю доставки результатов или профилю по умолчанию).

Формат

```
<action name="accept" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="accept".

Пример:

Сообщения, достигшие этого правила, принимаются к дальнейшей обработке.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>

    <rule enabled="1">
      <comment>
        All the messages reaching this rule are accepted for
        further processing.
      </comment>
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.1.3.2.2. Действие DROP

Прекратить обработку сообщения, накопленные данные удалить.

Описание

Это действие прекращает обработку текущего сообщения и сообщает EtherSensor, что все его данные и метаданные необходимо уничтожить.

Формат

```
<action name="drop" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="drop".

Пример:

Данные и метаданные сообщений, достигших этого правила, уничтожить.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <comment>
        Details/metadata of any message reaching this point
        are destroyed (discarded).
      </comment>
      <action name="drop" />
    </rule>
  </table>
</filter>
```

4.3.1.3.2.3. Действие JUMP

Продолжить обработку сообщения в другой таблице.

Описание

Это действие продолжает обработку сообщения в другой таблице.

Формат

```
<action name="jump" value="<table name>" /><action name="jump" value="<table name>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="jump".

Атрибут "value":

В атрибуте value="..." укажите имя таблицы, в которую необходимо перейти для дальнейшей обработки сообщения.

> Следует помнить, что переход по JUMP в таблицу main запрещён. Также запрещены переходы, которые могут привести к явной или неявной цикличности.

Пример:

Из таблицы main переход в таблицу yandex для обработки сообщений от mail.ru и yandex.ru. Таблица yandex: единственное правило уничтожает сообщения больше 100КБ. После этого происходит возврат в таблицу main.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <comment>
        Processing of messages from the mail.ru,
        yandex.ru detectors in the yandex table.
      </comment>
      <match>
        <c name="detector" value="mail.ru, yandex.ru" />
      </match>
      <action name="jump" value="yandex"/>
    </rule>

    <rule enabled="1">
      <action name="drop" />
    </rule>
  </table>

  <table name="yandex">
    <comment>
      The table processes messages from the mail.ru,
      yandex.ru detectors.
    </comment>

    <rule enabled="1">
      <comment>
        Discard messages larger than 100 Kb.
      </comment>
      <match>
        <c name="size" op=">" value="100K"/>
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="1">
      <comment>
        Returning to the main table for further processing.
      </comment>
      <action name="return" />
    </rule>
  </table>

</filter>
```

4.3.1.3.2.4. Действие RETURN

Вернуться в предыдущую (вызвавшую) таблицу и продолжить выполнение в ней со следующего правила.

Описание

Это действие возвращает обработку сообщения в предыдущую (вызвавшую) таблицу и продолжить выполнение в ней со следующего правила.

Формат

```
<action name="return" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="return".

> Действие не может применяться в таблице main.

Пример:

В данном примере в таблице "main" происходит какая-то обработка сообщения. При обработке сообщения вторым правилом стоит правило с именем "spam": если сообщение удовлетворяет критерию этого правила, то оно передаётся на дальнейшую обработку в таблицу "spam".

После прохождения правил в таблице "spam", если обработка сообщения не будет прервана предыдущими правилами, то дойдя до правила с именем "return-to-main", обработка сообщения снова продолжится в таблице "main" с правила, следующего за правилом "spam".

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

  <table name="main">
    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>
    <rule name="spam" enabled="1">
      <match ...> ... </match>
      <action name="jump" value="spam"/>
    </rule>
    <rule enabled="1">
      <action name="drop" />
    </rule>
  </table>

  <table name="spam">
    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>
    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>
    <rule name="return-to-main" enabled="1">
      <action name="return" />
    </rule>
  </table>
</filter>
```

4.3.1.3.2.5. Действие LABEL

Добавляет к метаданным сообщения строковую метку.

Описание

Это действие устанавливает строковую метку в метаданных текущего обрабатываемого сообщения.

Если эта строковая метка уже была установлена ранее для сообщения (или HTTP-объекта, из которого получено сообщение), то его значение заменяется более новым.

Используется для добавления описаний к сообщениям.

Формат

```
<action name="label" label="<label name>" value="<label value>" />
```

или

```
<action name="label" label="<label name>" > label value </action>
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="label".

Атрибут "label":

В атрибуте label="..." укажите имя устанавливаемой строковой метки.

Атрибут "value":

В атрибуте value="..." указывается значение (строка) для метки.

Значение также можно перечислять в самом теге "action".

Пример:

```
<action name="label"
        label="VIRUS-DESCR"
        value="Win.32.BlackHorse.trojan.virus – mail worm, very dangerous!!!" />
```

или

```
<action name="label"
        label="VIRUS-DESCR">
          Win.32.BlackHorse.trojan.virus -
          mail worm, very dangerous!!!
</action>
```

Устанавливает для сообщения строковую метку с именем "VIRUS-DESCR" и записывает в неё строку "Win.32.BlackHorse.trojan.virus – mail worm, very dangerous!!!".

Пример:

Пометить сообщения, обработанные детекторами mail.ru, yandex.ru меткой CONTENT-DESCR.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is the comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Mark messages detected by the mail.ru,
        yandex.ru detectors with the CONTENT-DESCR label.
      </comment>
      <match>
        <c name="detector" value="mail.ru, yandex.ru" />
      </match>
      <action name="label"
              label="CONTENT-DESCR"
              value="Russian mail services"/>
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.1.3.2.6. Действие TAG

Добавляет к метаданным сообщения тег (числовую метку).

Описание

Это действие устанавливает в метаданных сообщения тег (числовую метку). Тегов может быть несколько. В этом случае они перечисляются через запятую ',' или точку с запятой ';'. Если один и тот же тег устанавливать для объекта несколько раз, то у тега повышается "уровень" (его числовое значение). Иначе говоря, у каждого тега есть внутренний счётчик, показывающий, сколько раз он был установлен для этого объекта.

Установлен тег или нет, а также значение его счётчика (сколько раз он был установлен для текущего объекта) можно впоследствии проверять в условиях фильтров для принятия решений.

По умолчанию значение счётчика при установке тега увеличивается на 1. Если счётчик тега необходимо увеличить более чем на 1, то можно задавать значение, на которое необходимо увеличить счётчик после имени тега в круглых скобках: "TAG(...)". Например, SPAM(3) – увеличивает значение счётчика для тега SPAM на 3, а SPAM(1) равносильно просто SPAM. Это может быть необходимо в случаях, когда условия, устанавливающие один и тот же тег, имеют разную значимость, важность или приоритет.

Значение для изменения счётчика может быть отрицательным. SPAM(-3) – уменьшает значение счётчика для тега SPAM на 3.

Формат

```
<action name="tag" value="<tag list>" />
```

или

```
<action name="tag" > tag list </action>
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="tag".

Атрибут "value":

В атрибуте value="..." перечисляются имена тегов.

Имена тегов также можно перечислять в самом теге "action".

Пример:

```
<action name="tag" value="SPAM" />
```

Устанавливает тег с именем "SPAM".

```
<action name="tag" value="SPAM(3)" />
```

Устанавливает тег с именем "SPAM" и увеличивает его счётчик на 3.

```
<action name="tag" value="SPAM, shopping" />
```

Устанавливает теги с именами "SPAM" и "shopping".

```
<action name="tag" value="SPAM(3), shopping(2)" />
```

Устанавливает теги с именами "SPAM" и "shopping" и увеличивает их счётчики на 3 и 2 соответственно.

```
<action name="tag" value="SPAM, shopping">VIP-OFFICE</action>
```

Также устанавливает теги с именами "SPAM" и "shopping".

```
<action name="tag" value="SPAM, shopping">VIP-OFFICE</action>
```

Устанавливает теги с именами "SPAM", "shopping", "VIP-OFFICE".

Пример:

Пометить запросы на популярные российские почтовые сервисы тегом RUS_MAIL.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is the comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Mark requests to popular Russian mail services
        with the RUS_MAIL tag.
      </comment>
      <match>
        <c name="req-header"
           headername="Host"
           op="eq"
           value="win.mail.ru" />
        <c name="req-header"
           headername="Host"
           op="eq"
           value="mail.yandex.ru" />
        <c name="req-header"
           headername="Host" op="eq"
           value="mail.rambler.ru" />
      </match>
      <action name="tag" value="RUS_MAIL"/>
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.1.3.2.7. Действие DATETIME

Установить для сообщения строковую метку со значением текущих даты и времени.

Описание

Это действие устанавливает строковую метку (label) для сообщения со значением текущих даты и времени. Если данная строковая метка уже была ранее установлена для сообщения, то её значение заменяется более новым. Используется для отслеживания времени прохождения сообщения по фильтру.

Формат

```
<action name="datetime" value="label-name" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="datetime".

Атрибут "value":

В атрибуте value="..." укажите имя устанавливаемой строковой метки.

Пример:

Добавить метку начала обработки FILTER-BEGIN и метку конца обработки FILTER-END во все сообщения.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <comment>
        Set the FILTER-BEGIN processing start label
        for all the messages.
      </comment>
      <action name="datetime" value="FILTER-BEGIN" />
    </rule>

    <rule enabled="1">
      <comment>
        Set the FILTER-END processing end label
        for all the messages.
      </comment>
      <action name="datetime" value="FILTER-END" />
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.1.3.2.8. Действие DNS

Проводит разрешение имён DNS в IP-адреса и IP-адреса в имена для неизвестных адресов хостов.

Описание

Это действие производит разрешение имён хостов и IP-адресов сессии.

В каждой сессии есть источник (клиент) и приёмник (сервер) со своими именами или IP-адресами. Возможна ситуация, когда для сообщения известно имя хоста клиента или сервера, но не известен его IP-адрес или наоборот. Действие DNS производит разрешение недостающей информации об адресе клиента или сервера.

Если известен IP-адрес, но не известно имя хоста, соответствующее этому IP, то оно запрашивается у DNS-сервера. То же самое делается и если известно имя хоста, но не известен его IP-адрес. Если в сообщении уже присутствует и имя хоста и его IP, то никаких действий не производится.

Также в случае успешного получения имён в метаданные добавляются заголовки X-Sensor-Src-Host, X-Sensor-Dst-Host и становится возможным их использование в условии "header" (проверка значения заголовка).

Следует также помнить, что условие типа "hostname" (проверка имени хоста) имеет смысл применять только после выполнения действия DNS.

Формат

```
<action name="dns" address="<address type for resolving>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="dns".

Атрибут "address":

В атрибуте address="..." укажите тип адреса для разрешения имен. Возможные значения:

src или client

Проводить разрешение имен только для адреса источника

dst или server

Проводить разрешение имен только для адреса назначения

both или all или *

Проводить разрешение имен для обоих адресов (и источника, и назначения).

Если этот атрибут отсутствует, то подразумевается действие "both" – проводить разрешение имен для обоих адресов.

Пример:

Провести разрешение имен для сообщения.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is the comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Resolve names for the message.
      </comment>
      <action name="dns" address="server"/>
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.1.3.2.9. Действие DNSBL-LH, DNSBL-RH

Проверяет вхождение адресов сообщения в DNSBL списки, и если оно подтверждается, устанавливает указанный тег.

Описание

dnsbl-rh – проверка IP-адреса в DNSBL-RHSBL.

dnsbl-lh – проверка имени хоста в DNSBL-LHSBL.

При вхождении одного из адресов в один из DNSBL это действие увеличивает значение указанного тега на 1.

Список проверяется всегда полностью.

Вхождение каждого адреса (или имени хоста) в каждый DNSBL увеличивает тег на 1. То есть, если в один и тот же DNSBL входят и адрес источника и адрес назначения, то на каждое вхождение происходит увеличение тега.

В дальнейшем значение тега можно проверять в условиях фильтров и принимать решение на основании вхождения адресов сообщения в DNSBL.

Формат

```
<action name="dnsbl-rh"
        address="<address-type>"
        tag="<tag-name>"
        value="<dns-bl domains list>" />
<action name="dnsbl-rh"
        address="<address-type>"
        tag="<tag-name>"
        data="<dns-bl domains list source>" />
<action name="dnsbl-lh"
        address="<address-type>"
        tag="<tag-name>"
        value="<dns-bl domains list>" />
<action name="dnsbl-lh"
        address="<address-type>"
        tag="<tag-name>"
        data="<dns-bl domains list source>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="dns".

Атрибут "address":

В атрибуте address="..." укажите тип адреса для разрешения имен. Возможные значения:

src или client

Проверять только адрес источника

dst или server

Проверять только адрес назначения

both или all или *

Проверять оба адреса (и источника, и назначения).

Если этот атрибут отсутствует, то подразумевается действие "both" – проверять оба адреса.

Атрибут "tag":

В атрибуте tag="..." укажите имя увеличиваемого тега.

Атрибут "value":

В атрибуте value="..." перечисляются домены DNSBL.

Если доменов несколько, они перечисляются через запятую ','.

Также домены могут быть указаны в значении самого тега <action>dns-bl-domains-list</action>

Атрибут "data":

В атрибуте data="..." может указываться другой источник списка DNSBL доменов. Это позволяет указывать длинные списки, когда это неудобно делать в атрибуте value="...".

Возможные значения:

data="<external data name>"

Загрузить список из внешнего блока в фильтре (тег <data name="extern-data-name">...</data>)

data="extern://<external data name>"

Загрузить список из внешнего блока в фильтре (тег <data name="extern-data-name">...</data>). Домены DNSBL перечисляются через запятую.

data="file://<full-file-path>"

Загрузить список из указанного файла. Домены DNSBL перечисляются на отдельных строках (запятые не нужны).

Пример:

```
<action name="dnsbl-rh" tag="SPAM">
  bl.spamcop.net, vote.drbl.sandy.ru,
  sbl.spamhaus.org, cblplus.anti-spam.org.cn
</action>
```

Увеличивает тег "SPAM", если адреса сообщения входят в один из DNSBL. Если какой либо адрес входит только в один список – "SPAM" будет равен 1. Если адрес входит в два списка – "SPAM" будет равен 2 и т.д.

В дальнейшем можно проверить, если ("SPAM" > 3), то это точно спам, а если меньше – то это просто немного подозрительно. Если оба адреса входят в один DNSBL, то тег увеличивается на 2.

```
<action name="dnsbl-rh" address="src" tag="SPAM">
  bl.spamcop.net, vote.drbl.sandy.ru,
  sbl.spamhaus.org, cblplus.anti-spam.org.cn
</action>
```

Проверяет только IP-адрес источника.

```
<action name="dnsbl-lh" address="dst" tag="SPAM">
  bl.spamcop.net, vote.drbl.sandy.ru,
  sbl.spamhaus.org, cblplus.anti-spam.org.cn
</action>
```

Проверяет только hostname назначения.

Пример:

Проверить все адреса сообщения через DNSBL list. Положительные срабатывания пометить тегом SPAM. Сообщения с тегом SPAM удалить, остальные принять.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <comment>
        Checks all message addresses against the DNS-BL list. Hits
        are to be marked with the SPAM tag.
      </comment>
      <action name="dnsbl-rh"
              address="both" tag="SPAM"
              data="extern://dns-bl-list" />
    </rule>

    <rule enabled="1">
      <comment>Delete spam.</comment>
      <match>
        <c name="tag-exist" value="SPAM" />
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="1">
      <comment>Accept the rest.</comment>
      <action name="accept" />
    </rule>

  </table>

  <data name="dns-bl-list">
    bl.spamcop.net,
    vote.drbl.sandy.ru,
    sbl.spamhaus.org,
    cblplus.anti-spam.org.cn
  </data>

</filter>
```

4.3.1.3.2.10. Действие SAVE RAW DATA

Включить/выключить сохранение исходных данных, из которых было получено сообщение.

Описание

Это действие включает/выключает сохранение исходных данных, из которых было получено сообщение. В случае если сообщение получено из перехвата HTTP-трафика, то исходными данными будут два файла, содержащие HTTP-запрос и HTTP-ответ. В случае, если сообщение

получено из перехвата SMTP или POP3 протокола, то исходными данными будет файл, содержащий исходный email в формате EML.

По умолчанию для каждого сообщения сохранение исходных данных отключено. Если его включить, то файлы с исходными данными будут приаттачены к сообщению.

Следует помнить, что включение сохранения исходных данных более чем в два раза увеличивает объем сохраняемых данных, поэтому данной возможностью стоит пользоваться только для отладочных целей, или же когда наличие исходных данных крайне необходимо.

Формат

```
<action name="save-raw-data" value="<true/false/1/0>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="save-raw-data".

Атрибут "value":

В атрибуте value="..." указывается активно действие или нет.

true или 1

Включить сохранение исходных данных для сообщения.

false или 0

Отключить сохранение исходных данных для сообщения (если ранее в фильтре оно было включено).

Пример:

Включить сохранение исходных данных сообщений, полученных по HTTP-протоколу.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">
    <rule enabled="1">
      <comment>
        Enable saving source data for messages
        intercepted over the HTTP protocol.
      </comment>
      <match>
        <c name="protocol" value="http" />
      </match>
      <action name="save-raw-data" value="true" />
    </rule>

  </table>
</filter>
```

4.3.1.3.2.11. Действие TRANSPORT

Установить профиль доставки результатов для сообщения, которым оно должно быть доставлено в случае его успешной обработки фильтром с действием ACCEPT.

Описание

Если такой транспортный профиль уже был ранее установлен для сообщения, то действие ничего не выполняет.

Для сообщения можно указывать несколько транспортных профилей, для этого указываются несколько действий "transport".

> Таким образом вы можете доставлять результаты анализа одного и того же сообщения сразу нескольким системам-потребителям. Например, метаданные отправить в SIEM-систему, а само сообщение - в eDiscovery и в DLP.

Если в ходе фильтрации для сообщения не было применено ни одного транспортного профиля, то сообщение в случае принятия будет доставлено профилем по умолчанию.

Формат

```
<action name="transport" value="<transport-profile-name>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="transport".

Атрибут "value":

В атрибуте value="..." укажите имя устанавливаемого транспортного профиля.

Пример:

Установить транспортные профили "smtp-archive" и "smtp-archive-rezerve" для всех сообщений. Если сообщение будет принято фильтром, то оно будет доставлено с помощью ОБОИХ профилей.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="Message filter" version="1.0">
  <comment>Message filter.</comment>

  <table name="main">

    <rule enabled="1">
      <comment>
        Set the "smtp-archive" and "smtp-archive-rezerve"
        transport profiles for all messages. If the message
        is accepted by the filter, it will be delivered by
        BOTH profiles.
      </comment>
      <action name="transport" value="smtp-archive" />
      <action name="transport" value="smtp-archive-rezerve" />
    </rule>

    </table>
</filter>
```

4.3.1.3.2.12. Действие HEADER

Добавляет к метаданным объекта пользовательский заголовок X-Sensor-... или изменяет значение существующего X-Sensor-... заголовка.

Описание

Это действие добавляет в сообщение пользовательский заголовок X-Sensor-...

Если заголовок с таким именем уже существует в сообщении, то его значение заменяется более новым.

Формат

```
<action name="header" headername="<UserHeader>" value="<user header value>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="header".

Атрибут "headername":

В атрибуте headername="..." укажите имя добавляемой строковой метки.

Атрибут "value":

В атрибуте value="..." указывается значение заголовка.

Пример:

Добавляет к сообщению заголовок: X-Sensor-UserHeader: user header value.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is the comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Adds the following header to the message:
        X-Sensor-UserHeader: user header value.
      </comment>
      <action name="header"
              headername="UserHeader"
              value="user header value" />
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.1.3.2.13. Действие HEADER_EX

Добавляет к метаданным объекта пользовательский заголовок или изменяет значение существующего заголовка, за исключением заголовков From, To, Cc, Bcc, Subject, Date.

Описание

Это действие добавляет в сообщение пользовательский заголовок.

Если заголовок с таким именем (независимо от регистра) уже существует в сообщении, то его значение заменяется более новым.

Имя заголовка не может быть:

From

To

Cc

Bcc

Subject

Date

Формат

```
<action name="header_ex"
        headername="<UserHeader>"
        value="<user header value>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="header_ex".

Атрибут "headername":

В атрибуте headername="..." укажите имя добавляемой/заменяющей строковой метки.

Атрибут "value":

В атрибуте value="..." указывается значение заголовка.

Пример:

Добавляет к сообщению заголовок "X-SomethingElse-UserHeaderEx: user header value".

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is the comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Adds the following header to the message:
        X-SomethingElse-UserHeaderEx: user header value.
      </comment>
      <action name="header_ex"
              headername="X-SomethingElse-UserHeaderEx"
              value="user header value" />
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

   </table>
</filter>
```

4.3.1.3.2.14. Действие LOG

Делает запись в лог в соответствии с назначенными каналом и приемником, используйте его для отладки фильтров.

Описание

Это действие отправляет в указанный приёмник (файл, syslog-сервер) строку текста, значение тегов сообщения, значение меток сообщения или метаданные сообщения.

Формат строки лога:

```
[<timestamp>] <value>
```

В лог можно отправить:

- Строку, указываемую в атрибуте value="..."
- Список меток и их значений, указываемых в атрибуте field="#labels"
- Список тегов и их значений, указываемых в атрибуте field="#tags"
- Значение заголовка метаданных сообщения. Для этого в атрибуте field="X-Sensor-..." укажите имя заголовка.

Действие LOG полезно использовать для отладки фильтров. Расставляя его в нужных правилах фильтра, можно получить детальный отчёт о том, как сообщения проходят обработку, и как при этом меняется значение метаданных, тегов и меток.

Следует помнить, что это действие LOG применяется для целей отладки фильтров при фильтрации сообщений. Постоянное его применение может привести к снижению общей производительности EtherSensor, если будет происходить интенсивная запись на диск.

Формат

```
<action name="log" dst="<log-destination>" value="<user string value>" />
<action name="log" dst="<log-destination>" field="<field type>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="log".

Атрибут "dst":

В атрибуте dst="..." указывается приёмник записи в лог. Это может быть:

dst="syslog://<syslog-server-ip:port>"

Отправляет сообщение на syslog-сервер по протоколу UDP (RFC-3164)

dst="channel://<channel-name>"

Отправляет сообщение в канал, предварительно настроенный в службе EtherSensor Watcher

dst="file://<full-file-path>"

Сохраняет сообщение в файл.

Атрибут "field":

В атрибуте field="..." укажите тип поля сообщения, которое необходимо отправить в лог. Возможные значения:

#labels

Вывод списка меток сообщения и их значений

#tags

Вывод списка тегов сообщения и их значений

#from

Вывод списка адресов FROM

#to

Вывод списка адресов TO

#subject

Вывод темы сообщения

X-Sensor-...

Вывод значения заголовка метаданных сообщения.

Атрибут "value":

В атрибуте value="..." указывается строка сообщения, которая будет отправлена в лог.

Пример:

Действие:

```
<action name="log" dst="file://d:\file\path.log.txt"
        value="user string value" />
```

Сохраняет в файл лога d:\file\path.log.txt строку:

```
"[<timestamp>] user string value".
```

Действие:

```
<action name="log" dst="file://d:\file\path.log.txt" field="#tags" />
```

Сохраняет в файл лога d:\file\path.log.txt строку:

```
"[<timestamp>] Tags:"
"                    <TAG = value>"
"                    <TAG = value>"
"                    <TAG = value>"
```

Каждый тег сохраняется в отдельной строке.

Действие:

```
<action name="log" dst="channel://debug.log.txt" field="#labels" />
```

Отправляет в канал debug.log.txt строку:

```
"[<timestamp>] Labels:"
"                   <LABEL = "value">"
"                   <LABEL = "value">"
"                   <LABEL = "value">"
```

Каждая метка сохраняется в отдельной строке.

Действие:

```
<action name="log"
        dst="syslog://192.168.0.1:514"
        field="X-Sensor-Src-Address" />
```

Отправляет на syslog-сервер с адресом 192.168.0.1:514 строку:

```
"[<timestamp>] X-Sensor-Src-Address: <metadata header value>"
```

Пример:

Отослать данные сообщения на syslog-сервер.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is the comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Sending message details to syslog.
      </comment>
      <action name="log"
              dst="syslog://192.168.0.1:514"
              value="Message info dump:" />
      <action name="log"
              dst="syslog://192.168.0.1:514"
              field="X-Sensor-Src-Address" />
      <action name="log"
              dst="syslog://192.168.0.1:514"
              field="X-Sensor-Dst-Address" />
      <action name="log" dst="syslog://192.168.0.1:514"
              field="#labels" />
      <action name="log"
              dst="syslog://192.168.0.1:514"
              field="#tags" />
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.1.4. Краткие правила написания фильтров

1. Фильтр всегда должен содержать таблицу "main", с неё начинается выполнение фильтра.

2. Таблица не может быть пустой. Каждая таблица фильтра должна заканчиваться правилом с условием "любое сообщение" и одним из действий: ACCEPT, DROP или RETURN. Это правило обязательно должно быть включено.

Пример:

```
<rule name="end" enabled="1">
  <match>
    <c name="all"/>
  </match>
  <action name="drop" />
</rule>
```

или

```
<rule enabled="1">
  <action name="accept" />
</rule>
```

3. Действие RETURN может применяться в любых таблицах, кроме main.

4. Переходы JUMP, образующие циклические ссылки (явные или неявные), запрещены.
5. Переходы JUMP в таблицу "main" делать запрещено во избежание циклических ссылок.

4.3.1.5. Советы

В этом разделе описаны приёмы, которые помогут отлаживать и тестировать фильтры.

Метки времени

Метки времени (<action name="datetime" value="label-name">) можно использовать не только в конце и в начале обработки сообщения, но и в каждом правиле – тогда можно будет узнать, когда оно было обработано относительно начала фильтрации всего сообщения в целом. Для этого действие "datetime" можно указывать непосредственно перед терминирующим действием.

Пример:

```
<rule name="rule2" enabled="1">
  <match>
    <c name="attach-exist"/>
  </match>
  <action name="datetime" value="rule2-end" />
  <action name="accept" />
</rule>
```

В данном примере после обработки сообщения правилом у сообщения будет установлена метка "rule2-end", показывающая время, когда сообщение было принято этим правилом к дальнейшей обработке (ACCEPT).

Выполнение некоторых действий может занимать довольно длительное время (например, DNS-разрешение имён, проверка сообщения внешними антивирусами и т.п.). Так как все действия выполняются последовательно, то можно замерить время выполнения этого длительного действия, добавив к нему действия "datetime".

Пример:

```
<rule name="rule2" enabled="1">
  <match>
    <c name="attach-exist"/>
  </match>
  <action name="datetime" value="rule2-dns-start" />
  <action name="dns" />
  <action name="datetime" value="rule2-dns-end" />
</rule>
```

В этом примере после обработки сообщения правилом можно будет сравнить значение меток "rule2-dns-start" и "rule2-dns-end" и узнать время, которое занимала операция DNS.

Теги правил

Вы можете использовать действие "tag" (установка тега в сообщении) во всех правилах, и при этом имя тега ставить таким же, как имя правила. Тогда после прохождения сообщения через фильтр в нём будет история обработки сообщения правилами. Это может быть полезно для тестирования фильтров и отслеживания "путей" прохождения ими обрабатываемых объектов, когда фильтр сложный и содержит много правил и таблиц.

Пример:

```
<rule name="rule-attach" enabled="1">
  <match>
    <c name="attach-exist"/>
  </match>
  <action name="tag" value="rule-attach" />
</rule>

<rule name="rule-bad-words" enabled="1">
  <match>
    <c name="text" op="all" value="tel, respect" />
  </match>
  <action name="tag" value=" rule-bad-words" />
</rule>

<rule name="rule-only-for-dns" enabled="1">
  <action name="dns" />
  <action name="tag" value=" rule-only-for-dns" />
</rule>

<rule name="accept-all" enabled="1">
  <action name="tag" value="accept-all" />
  <action name="accept" />
</rule>
```

Также можно поступить и с метками времени, в этом случае будет видно не только через какие правила прошло сообщение, но и в какое время.

"От простого к сложному"

Выполнение некоторых действий может занимать довольно длительное время (DNS-разрешение имён, проверка сообщения внешними антивирусами и т.п.). Следует размещать такие действия как можно ближе к концу обработки сообщения, чтобы было как можно меньше ситуаций, когда выполняется длительная операция, а потом применяется для сообщения действие DROP из-за того, что поле FROM не прошло проверку. Лучше сначала проверить FROM и отсеять на нём часть сообщений, и лишь потом выполнять для них DNS или другие длительные операции вроде проверки антивирусами.

4.3.2. Префильтрация HTTP запросов

Механизм предварительной фильтрации HTTP-запросов решает следующие задачи:

1. Отфильтровывание ненужного HTTP-трафика для снижения нагрузки на EtherSensor на этапе анализа сообщений (действия ACCEPT[181] и DROP[182]).

2. Накопление информации о возможных новых тенденциях в HTTP-трафике для предоставления службе поддержки и/или разработчику Microolap EtherSensor (действие COPY[186]).

3. Логирование информации о HTTP-запросах в формате SQUID-ACCESS-LOG (действие ACCESS-LOG[187]).

4. Манипуляции с тегами и метками на этапе препроцессинга с целью использования накопленной в них информации при анализе сообщений (действия TAG[188] и LABEL[191]).

Конфигурация HTTP-фильтра содержится в XML-файле, хранящемся в поддиректории [INSTALLDIR]\config\filter\http.

Для редактирования фильтра воспользуйтесь любым внешним текстовым/XML редактором, либо используйте встроенный в консоль управления EtherSensor редактор фильтров, специально разработанный для этой цели:

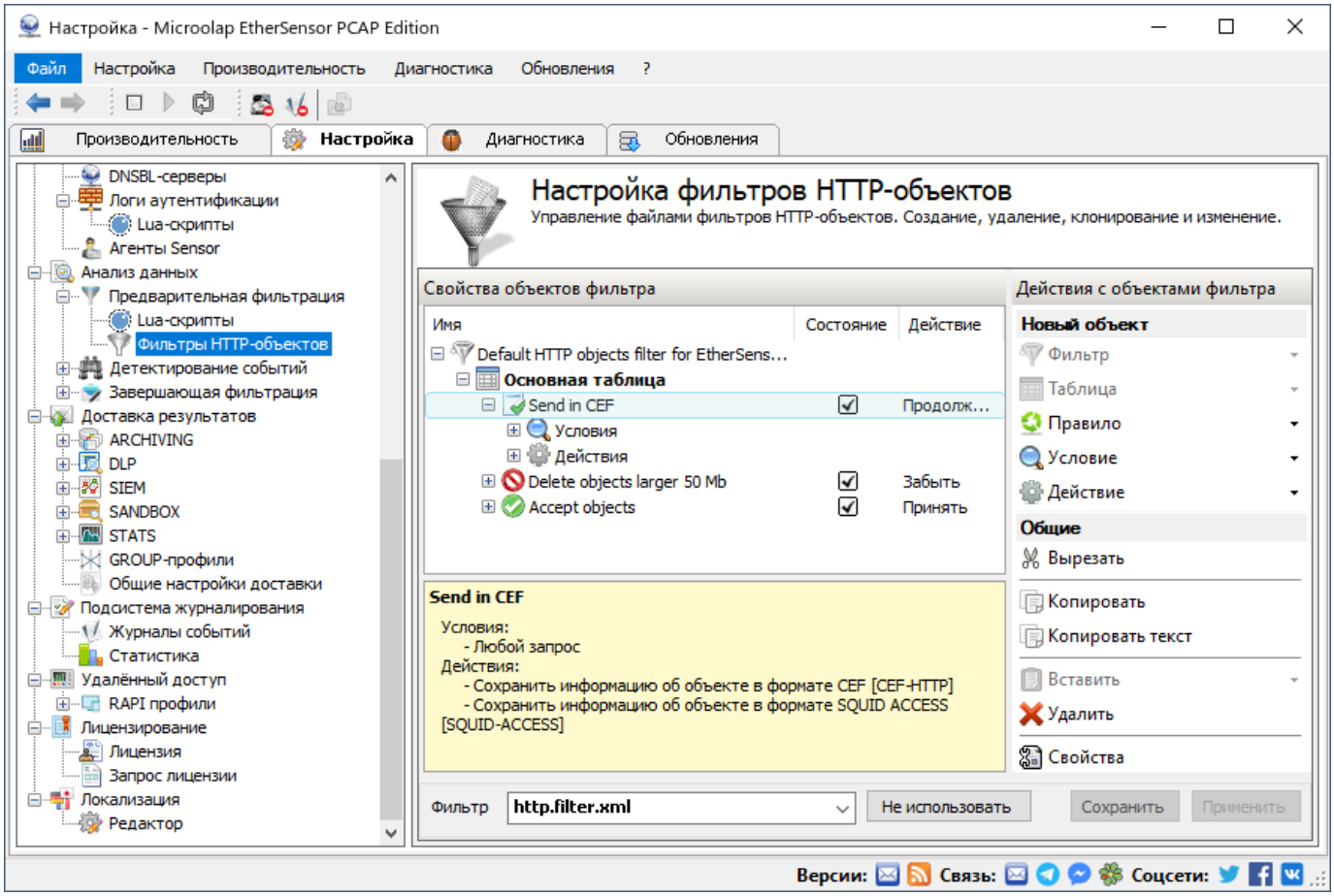

Рис.39. Редактирование HTTP-фильтра.

4.3.2.1. Условия

Ниже приведены условия, использующиеся в правилах префильтрации HTTP запросов.

4.3.2.1.1. Условие ALL, *

Специальное условие, которому удовлетворяют все фильтруемые объекты.

Описание

Это условие является истинным для любых объектов, неявно подразумевается, если критерий <match>...</match> является пустым или отсутствует в правиле (правило содержит только теги "action").

Формат

```
<c name="all" />
<c name="*" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="all" или name="*".

Пример:

Правило принимает все сообщения к дальнейшей обработке.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>

  <table name="main">

    <rule enabled="1">
      <comment>
        This rule accepts all messages for further processing.
      </comment>
      <match>
        <c name="all" />
      </match>
      <action name="accept" />
    </rule>

  </table>

</filter>
```

Пример (неявное условие all):

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.2.1.2. Условие METHOD

Условие проверяет метод HTTP-запроса.

Описание

Это условие проверяет название метода HTTP-запроса. Если имя метода совпадает – условие выполняется.

Формат

```
<c name="method" value="<HTTP method>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="method".

Атрибут "value":

В атрибуте value="..." укажите имя метода для сравнения.

Пример:

Правило прекращает обработку всех GET-запросов.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>

  <table name="main">
    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>

    <rule enabled="1">
      <comment>
        The rule stops further processing of GET requests.
      </comment>
      <match>
        <c name="method" value="GET"/>
      </match>
      <action name="drop" />
    </rule>
  </table>
</filter>
```

4.3.2.1.3. Условие IP

Проверить IP-адреса клиента или сервера на вхождение в диапазон или на принадлежность к подсети.

Описание

Это условие проверяет IP-адреса клиента или сервера на вхождение в диапазон или на принадлежность к подсети.

> Советы:
>
> 1. Нежелательный трафик нужно отсекать как можно раньше. От этого зависит производительность EtherSensor.
>
> 2. Весь трафик с некоторого IP или диапазона лучше всего отсекать в IP-фильтре службы EtherSensor EtherCAP.
>
> 3. Определённый трафик HTTP с некоторого IP (если возможно определить такие критерии) лучше всего отсекать в HTTP-фильтре.
>
> 4. Определённые сообщения с некоторого IP-адреса необходимо обрабатывать в фильтре сообщений.

Формат

```
<c name="ip" address="<address type>" value="<ip-range>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="ip".

Атрибут "address":

В атрибуте address="..." укажите тип адреса для проверки. Возможные значения:

src или client

Проверять адрес источника

dst или server

Проверять адрес назначения.

Атрибут "value":

В атрибуте value="..." укажите значение для сравнения. Возможные значения:

ipaddress

Проверяет IP-адрес на равенство. Например, value="192.168.0.10"

ip1-ip2

Проверяет на вхождение IP-адреса в диапазон. Например, value="192.168.0.1-192.168.0.10"

ip/netmask

Проверяет на принадлежность IP-адреса к указанной подсети. Например, value="192.168.0.1/255.255.255.0"

ip/netmaskbits

Проверяет на принадлежность IP-адреса к указанной подсети. Например, value="192.168.0.1/24".

Пример:

Сообщения от клиента с машины 192.168.0.15 игнорировать.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>

  <table name="main">

    <rule enabled="1">
      <comment>
        Discard messages from 192.168.0.15.
      </comment>
      <match>
        <c name="ip" address="client" value="192.168.0.15" />
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="1">
      <action name="accept" />
    </rule>

    </table>
</filter>
```

4.3.2.1.4. Условие REQ-SIZE, RESP-SIZE, SIZE

Условие проверяет полный (включая заголовки) размер HTTP-объекта.

Описание

Условия группы size проверяют полный (включая заголовки) размер HTTP-запроса/ответа.

req-size

Условие выполняется, если размер HTTP-запроса соответствует условию

resp-size

Условие выполняется, если размер HTTP-ответа соответствует условию

size

Условие выполняется, если размер HTTP-запроса или размер HTTP-ответа соответствуют условию.

Формат

```
<c name="req-size" op="<operation>" value="<compare pattern>" />
<c name="resp-size" op="<operation>" value="<compare pattern>" />
<c name="size" op="<operation>" value="<compare pattern>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="req-size" или name="resp-size" или name="size".

Атрибут "value":

В атрибуте value="..." укажите число, с которым сравнивается размер HTTP-объекта.

<число> или <число>B

Указывает размер в байтах

<число>K

Указывает размер в килобайтах

<число>M

Указывает размер в мегабайтах

<число>G

Указывает размер в гигабайтах.

Атрибут "op":

В атрибуте op="..." указывает тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если размер РАВЕН указанному числу

ne или != или <>

Условие выполняется, если размер НЕ РАВЕН указанному числу

lt или &lt

Условие выполняется, если размер МЕНЬШЕ указанного числа

gt или >

Условие выполняется, если размер БОЛЬШЕ указанного числа

le или <=

Условие выполняется, если размер МЕНЬШЕ ИЛИ РАВНО указанного числа

ge или >=

Условие выполняется, если размер БОЛЬШЕ ИЛИ РАВНО указанного числа.

Пример:

Правило прекращает обработку всех HTTP-объектов запроса размером более 100KB или HTTP-объектов ответа размером более 1MB.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <comment>
        The rule stops processing HTTP objects with the request size
        over 100KB or the response size over 1MB.
      </comment>
      <match>
        <or>
          <c name="req-size" op=">" value="100K"/>
          <c name="resp-size" op="gt" value="1M"/>
        </or>
      </match>
      <action name="drop" />
    </rule>
  </table>
</filter>
```

4.3.2.1.5. Условие REQ-HEADER, RESP-HEADER

Проверяет значение одного из заголовков HTTP-запроса или ответа.

Описание

Это условие проверяет значение заголовка HTTP-запроса на наличие подстроки в строке или на соответствие шаблону wildcard или regexp.

Формат

```
<c name="req-header" headername="..." op="..." value="..." />
```

или

```
<c name="resp-header" headername="..." op="..." value="..." />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="req-header" или name="resp-header".

req-header

Проверяет заголовки HTTP-запроса

resp-header

Проверяет заголовки HTTP-ответа

Атрибут "headername":

В атрибуте headername="..." укажите имя проверяемого заголовка.

Атрибут "headername":

Строка, с которой сравнивается значение или шаблон для проверки, указывается в атрибуте value="...".

Атрибут "op":

В атрибуте op="..." указывает тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если значение поля СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если значение поля НЕ СОДЕРЖИТ указанное значение

wc или wildcard

Условие выполняется, если значение поля соответствует указанному wildcard шаблону

re или regex или regexp

Условие выполняется, если значение поля соответствует указанному regexp шаблону

Для заголовка Content-Length доступны следующие операции:

eq или = или ==

Условие выполняется, если значение поля СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если значение поля НЕ СОДЕРЖИТ указанное значение

lt или &lt

Условие выполняется, если размер МЕНЬШЕ указанного числа

gt или >

Условие выполняется, если размер БОЛЬШЕ указанного числа

le или <=

Условие выполняется, если размер МЕНЬШЕ ИЛИ РАВНО указанного числа

ge или >=

Условие выполняется, если размер БОЛЬШЕ ИЛИ РАВНО указанного числа.

Эти операции выполняются со значением заголовка как с ЧИСЛОМ, а не как со строкой.

Атрибут "value":

В атрибуте value="..." укажите проверяемое значение (строка, wildcard или regexp шаблон).

> Для заголовка Content-Length для сравнения допускается указывать только числовое значение.

Число, с которым сравнивается размер, указывается в виде:

<число> или <число>B

Указывает размер в байтах

<число>K

Указывает размер в килобайтах

<число>M

Указывает размер в мегабайтах

<число>G

Указывает размер в гигабайтах.

Пример:

Игнорировать запросы с Content-Length более 100K. Принимать запросы на win.mail.ru и *.yandex.ru.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <comment>
        Ignore requests where Content-Length is more than 100K.
      </comment>
      <match>
        <c name="req-header"
           headername="Content-Length"
           op=">" value="100K" />
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="1">
      <comment>
        Accept requests to win.mail.ru and *.yandex.ru.
      </comment>
      <match>
        <or>
          <c name="req-header"
             headername="Host" op="eq"
             value="win.mail.ru" />
          <c name="req-header"
             headername="Host"
             op="wc"
             value="*.yandex.ru" />
        </or>
      </match>
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.2.1.6. Условие URL

Проверяет значение URL HTTP-запроса.

Описание

Это условие проверяет значение URL HTTP-запроса на наличие подстроки в строке или на соответствие шаблону wildcard или regexp.

Следует помнить, что проверяется полный URL запроса, т.е. в виде http://www.server.com/script-or-page-path.htm.

> Если необходимо проверять только имя хоста, то следует пользоваться условием "HEADER". Это повысит скорость проверки и сэкономит ресурсы.

Формат

```
<c name="url" op="..." value="..." />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="url".

Атрибут "op":

В атрибуте op="..." указывает тип операции сравнения и может принимать значения:

eq или = или ==

Условие выполняется, если значение поля СОДЕРЖИТ указанное значение

ne или != или <>

Условие выполняется, если значение поля НЕ СОДЕРЖИТ указанное значение

wc или wildcard

Условие выполняется, если значение поля соответствует указанному wildcard шаблону

re или regex или regexp

Условие выполняется, если значение поля соответствует указанному regexp шаблону

Атрибут "value":

В атрибуте value="..." укажите проверяемое значение (строка, wildcard или regexp шаблон).

Пример:

```
<c name="url" op="eq" value="game" />
```

Условие выполняется, если в URL присутствует подстрока "game".

```
<c name="url" op="wc" value="http://*.mail.ru/*send*" />
```

Условие выполняется, если URL соответствует wildcard шаблону "http://*.mail.ru/*send*".

```
<c name="url" op="re" value="satan|shopping|dating|movies|hexogen" />
```

Условие выполняется, если URL соответствует regexp шаблону "satan|shopping|dating|movies|hexogen".

Пример:

Детектировать запросы с нехорошими словами (satan|shopping|dating|movies|hexogen), метить тегом shopping. Игнорировать запросы, помеченные тегом shopping.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <comment>
        Detect requests possibly related to bad content.
      </comment>
      <match>
        <c name="url"
           op="re"
           value="satan|shopping|dating|movies|hexogen" />
      </match>
      <action name="tag" value="shopping"/>
    </rule>

    <rule enabled="1">
      <comment>
        Ignore requests tagged as "shopping".
      </comment>
      <match>
        <c name="tag" tag="shopping"/>
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="1">
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.2.1.7. Условие TAG

Проверяет наличие установленного тега и значение его счётчика (см. раздел Действие TAG[188]).

Описание

Это условие проверяет активность указанного тега для объекта, а также его уровень.

Формат

```
<c name="tag" tag="<tag name>" op="<operation>" value="<compare value>" />
```

Атрибут "name":

В атрибуте "name" укажите имя условия: name="tag".

Атрибут "tag":

В атрибуте tag="..." укажите имя проверяемого тега.

Атрибут "op":

В атрибуте op="..." укажите тип операции сравнения. Возможные значения:

eq или = или ==

Условие выполняется, если значение РАВНО указанному числу

ne или != или <>

Условие выполняется, если значение НЕ РАВНО указанному числу

lt или &lt

Условие выполняется, если значение МЕНЬШЕ указанного числа

gt или >

Условие выполняется, если значение БОЛЬШЕ указанного числа

le или <=

Условие выполняется, если значение МЕНЬШЕ ИЛИ РАВНО указанного числа

ge или >=

Условие выполняется, если значение БОЛЬШЕ ИЛИ РАВНО указанного числа

exist

Условие выполняется, если тег существует (уже установлен ранее для этого HTTP-объекта).

По умолчанию (если атрибут "op" отсутствует) принимается значение "exist". Для операции "exist" указывать атрибут "value" не обязательно.

Атрибут "value":

В атрибуте value="..." укажите проверяемое значение, если проверяется значение счётчика тега.

Пример:

```
<c name="tag" tag="SPAM" op="exist" />
```

или

```
<c name="tag" tag="SPAM" />
```

Условие выполняется, если для объекта установлен тег SPAM.

```
<c name="tag" tag="SPAM" op=">=" value="3" />
```

Условие выполняется, если для объекта установлен тег SPAM и значение его счётчика больше или равно 3.

Пример:

Детектировать запросы с нехорошими словами (satan|shopping|dating|movies|hexogen), метить тегом shopping. Игнорировать запросы, помеченные тегом shopping.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Detect requests possibly related to bad things.
      </comment>
      <match>
        <c name="url"
           op="re"
           value="satan|shopping|dating|movies|hexogen" />
      </match>
      <action name="tag" value="shopping"/>
    </rule>

    <rule enabled="true">
      <comment>
        Ignore a request tagged as "shopping".
      </comment>
      <match>
        <c name="tag" tag="shopping"/>
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="1">
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.2.2. Действия

Ниже приведён список действий, исполняемых правилами HTTP префильтрации.

4.3.2.2.1. Действие ACCEPT

Принять HTTP-объект к дальнейшей обработке.

Описание

Действие ACCEPT прекращает работу фильтров с текущим HTTP-объектом и пропускает его к дальнейшей обработке детекторами.

Формат

```
<action name="accept" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="accept".

Пример:

Все сообщения, достигшие этого правила, принимать к дальнейшей обработке.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <match ...> ... </match>
      <action ...> ... </action>
    </rule>

    <rule enabled="1">
      <comment>
        All messages that reach this rule are accepted for
        further processing.
      </comment>
      <action name="accept" />
    </rule>
  </table>
</filter>
```

4.3.2.2.2. Действие DROP

Проигнорировать HTTP-объект без продолжения дальнейшей обработки.

Описание

Это действие прекращает обработку текущего HTTP-объекта в EtherSensor и сообщает ему, что сообщение необходимо проигнорировать ("забыть") и уничтожить все накопленные о нем данные.

Формат

```
<action name="drop" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="drop".

Пример:

Все сообщения, достигшие этого правила, проигнорировать, их накопленные метаданные уничтожить.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <comment>
        All messages that reach this rule will be discarded.
      </comment>
      <action name="drop" />
    </rule>
  </table>
</filter>
```

4.3.2.2.3. Действие JUMP

Продолжить обработку HTTP-объекта в другой таблице.

Описание

Это действие продолжает обработку HTTP-объекта в другой таблице.

Формат

```
<action name="jump" value="<table name>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="jump".

Атрибут "value":

В атрибуте value="..." укажите имя таблицы, в которую необходимо перейти для дальнейшей обработки HTTP-объекта.

> Следует помнить, что переход в таблицу "main" запрещён для исключения зацикливания. Также запрещены переходы, которые могут привести к явной или неявной цикличности.

Пример:

GET-запросы передаются на обработку в таблицу get-process, обрабатываются в ней (ACCEPT для запросов на win.mail.ru), затем возврат в таблицу "main" для дальнейшей обработки.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">
    <rule enabled="1">
      <comment>Processing of GET requests is passed to the "get-process" table.</comment>
      <match>
        <c name="method" value="GET"/>
      </match>
      <action name="jump" value="get-process"/>
    </rule>
    <rule enabled="1">
      <action name="drop" />
    </rule>
  </table>

  <table name="get-process">
    <comment>GET requests are processed in the table get-process.</comment>
    <rule enabled="1">
      <comment>ACCEPT for requests for win.mail.ru.</comment>
      <match>
        <c name="req-header"
           headername="Host"
           op="eq"
           value="win.mail.ru" />
      </match>
      <action name="accept" />
    </rule>
    <rule enabled="1">
      <comment>Return to "main" table for further processing.</comment>
      <action name="return" />
    </rule>
  </table>
</filter>
```

4.3.2.2.4. Действие RETURN

Вернуться в предыдущую (вызвавшую) таблицу и продолжить выполнение в ней со следующего правила.

Описание

Это действие возвращает обработку HTTP-объекта в предыдущую (вызвавшую) таблицу и продолжает выполнение в ней со следующего правила.

Формат

```
<action name="return" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="return".

Действие не может применяться в таблице "main".

Пример:

GET-запросы передаются на обработку в таблицу get-process, затем GET-запросы обрабатываются в таблице get-process, затем ACCEPT для запросов на win.mail.ru, затем возврат в таблицу "main" для дальнейшей обработки.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <comment>HTTP filter.</comment>
  <table name="main">

    <rule enabled="1">
      <comment>
        Processing of GET requests is passed to the "get-process" table.
      </comment>
      <action name="jump" value="get-process"/>
    </rule>

    <rule enabled="1">
      <action name="drop" />
    </rule>
  </table>

  <table name="get-process">
    <comment>
      GET requests are processed in the table.
    </comment>

    <rule enabled="1">
      <comment>
        ACCEPT for requests for win.mail.ru.
      </comment>
      <match>
        <c name="req-header"
           headername="Host"
           op="eq"
           value="win.mail.ru" />
      </match>
      <action name="accept" />
    </rule>

    <rule enabled="1">
      <comment>
        Return to "main" table for further processing.
      </comment>
      <action name="return" />
    </rule>
  </table>

</filter>
```

4.3.2.2.5. Действие COPY

Копировать текущий обрабатываемый HTTP-объект (запрос и ответ, если есть) и его метаданные в указанную директорию.

Описание

Это действие копирует текущий обрабатываемый HTTP-объект (запрос и ответ, если есть) и его метаданные в указанную директорию.

Формат

```
<action name="copy" value="<folder path>"/>
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="copy".

Атрибут "value":

В атрибуте value="..." укажите путь к директории, в которую следует копировать данные.

Если директория не существует, она будет создана.

Пример:

Копировать запросы на win.mail.ru в папку d:\data_from_mail.ru.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is a comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Copy requests for win.mail.ru to d:\data_from_mail.ru.
      </comment>
      <match>
        <c name="req-header"
           headername="Host"
           op="eq"
           value="win.mail.ru" />
      </match>
        <action name="copy" value="d:\data_from_mail.ru"/>
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.2.2.6. Действие ACCESS-LOG

Сохранить информацию о текущем HTTP-объекте в канал логирования в формате SQUID-ACCESS-LOG.

Описание

Действие access-log сохраняет информацию о текущем HTTP-объекте в канал логирования в формате SQUID-ACCESS-LOG.

Формат

```
<action name="access-log" value="<log-channel-name>" />
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="access-log".

Атрибут "value":

В атрибуте value="..." укажите имя канала логирования. Этот канал должен быть предварительно настроен в службе EtherSensor Watcher.

Пример:

Логировать все запросы в канал all-log-channel, а запросы на win.mail.ru ещё и в mail-ru-log-channel.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is a comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Log requests for win.mail.ru to mail-ru-log-channel.
      </comment>
      <match>
        <c name="req-header"
           headername="Host"
           op="eq"
           value="win.mail.ru" />
      </match>
      <action name="access-log" value="mail-ru-log-channel"/>
    </rule>

    <rule enabled="true">
      <comment>
        Log all requests to all-log-channel.
      </comment>
      <action name="access-log" value="all-log-channel"/>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.2.2.7. Действие TAG

Добавляет к метаданным объекта тег (числовую метку).

Описание

Это действие устанавливает в метаданных объекта тег (числовую метку). Тегов может быть несколько. В этом случае они перечисляются через запятую ',' или точку с запятой ';'. Если один и тот же тег устанавливать для объекта несколько раз, то у тега повышается "уровень" (его числовое значение). У каждого тега есть внутренний счётчик, показывающий, сколько раз он был установлен для этого объекта.

Существование тега, а также значение его счётчика (сколько раз он был установлен для текущего объекта) можно впоследствии проверять в условиях фильтров для принятия решений.

По умолчанию значение счётчика при установке тега увеличивается на 1. Если счётчик тега необходимо увеличить более чем на 1, то можно задавать значение, на которое необходимо

увеличить счётчик, после имени тега в круглых скобках: "TAG(...)". Например, SPAM(3) – увеличивает значение счётчика для тега SPAM на 3, а SPAM(1) равносильно просто SPAM. Это может быть необходимо в случаях, когда условия, устанавливающие один и тот же тег, имеют разную значимость / важность / приоритет.

Значение для изменения счётчика может быть отрицательным. SPAM(-3) – уменьшает значение счётчика для тега SPAM на 3.

> Установленные в этом действии теги в дальнейшем будут доступны в условиях фильтра сообщений , если из данного HTTP-объекта будет извлечено сообщение.

Формат

```
<action name="tag" value="<tag list>" />
```

или

```
<action name="tag" > tag list </action>
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="tag".

Атрибут "value":

В атрибуте value="..." перечисляются имена тегов.

Имена тегов также можно перечислять в самом теге <action>.

Пример:

```
<action name="tag" value="SPAM" />
```

Устанавливает тег с именем "SPAM".

```
<action name="tag" value="SPAM(3)" />
```

Устанавливает тег с именем "SPAM" и увеличивает его счётчик на 3.

```
<action name="tag" value="SPAM, shopping" />
```

Устанавливает теги с именами "SPAM" и "shopping".

```
<action name="tag" value="SPAM(3), shopping(2)" />
```

Устанавливает теги с именами "SPAM" и "shopping" и увеличивает их счётчики на 3 и 2.

```
<action name="tag">SPAM, shopping</action>
```

Также устанавливает теги с именами "SPAM" и "shopping".

```
<action name="tag" value="SPAM, shopping">VIP-OFFICE</action>
```

Устанавливает теги с именами "SPAM", "shopping", "VIP-OFFICE".

Пример:

Пометить запросы на популярные российские почтовые сервисы тегом RUS_MAIL.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>Filter comment.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Mark requests to popular Russian mail services
        with RUS_MAIL tag.
      </comment>
      <match>
        <c name="req-header"
           headername="Host"
           op="eq"
           value="win.mail.ru" />
        <c name="req-header"
           headername="Host"
           op="eq"
           value="mail.yandex.ru" />
        <c name="req-header"
           headername="Host"
           op="eq"
           value="mail.rambler.ru" />
      </match>
      <action name="tag" value="RUS_MAIL"/>
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.2.2.8. Действие LABEL

Добавляет к метаданным объекта строковую метку.

Описание

Это действие устанавливает строковую метку в метаданных текущего обрабатываемого HTTP-объекта. Если данный ярлык уже был ранее установлен для HTTP-объекта, то его значение заменяется более новым. Используется для добавления описаний к HTTP-объектам.

> Установленные строковые метки в дальнейшем будут доступны в фильтре сообщений (если из данного HTTP-объекта было извлечено сообщение).

Формат

```
<action name="label" label="<label name>" value="<label value>" />
```

или:

```
<action name="label" label="<label name>" > label value </action>
```

Атрибут "name":

В атрибуте "name" укажите имя действия: name="label".

Атрибут "label":

В атрибуте label="..." укажите имя устанавливаемой строковой метки.

Атрибут "value":

В атрибуте value="..." укажите строковое значение для метки.

Значение также можно перечислять в самом теге <action>.

Пример:

```
<action name="label"
        label="VIRUS-DESCR"
        value="Win.32.BlackHorse.trojan.virus -
               mail worm, extremely dangerous!!!" />
```

Это действие устанавливает для сообщения строковую метку с именем "VIRUS-DESCR" и записывает в неё строку "Win.32.BlackHorse.trojan.virus – mail worm, extremely dangerous!!!".

Пример:

Помечать запросы на популярные российские почтовые сервисы меткой CONTENT-DESCR.

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="TEST" version="1.0">
  <comment>This is a comment for the filter.</comment>
  <table name="main">

    <rule enabled="true">
      <comment>
        Mark requests for popular Russian mail services
        with CONTENT-DESCR label.
      </comment>
      <match>
        <or>
          <c name="req-header"
             headername="Host"
             op="eq"
             value="win.mail.ru" />
          <c name="req-header"
             headername="Host"
             op="eq"
             value="mail.yandex.ru" />
          <c name="req-header"
             headername="Host"
             op="eq"
             value="mail.rambler.ru" />
        </or>
      </match>
      <action name="label"
              label="CONTENT-DESCR"
              value="Russian mail services"/>
    </rule>

    <rule enabled="true">
      <match><c name="all"/></match>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

4.3.3. Примеры применения фильтров

Ниже описаны примеры применения фильтров для обработки результатов перехвата.

4.3.3.1. Добавление имени хоста

Задача

Необходимо, чтобы в реконструированных сообщениях вместе с IP-адресами содержались доменные имена хостов отправителя и получателя.

Описание логики решения

Для этого необходимо, чтобы EtherSensor производил разрешение доменных имён через службу DNS.

Сделайте следующее:

1. В конфигурации службы EtherSensor Analyser укажите, через какие DNS-серверы необходимо производить разрешение имён.

2. Создайте фильтр фильтр сообщений, содержащий правило с действием, означающим, что для текущего обрабатываемого сообщения необходимо произвести DNS-разрешение имён хостов.

Решение

1. Настройка в конфигурации службы EtherSensor Analyser параметров DNS-серверов для разрешения имён.

Настройку можно производить в консоли управления или в конфигурационном файле.

В файле конфигурации службы EtherSensor Analyser:

Корневой тег <AnalyserConfig>, далее вложенный тег настроек фильтров <Filter> (включается фильтр сообщений), далее вложенный тег <Dns> – настройки DNS-серверов для разрешения имён в фильтре сообщений.

```
<?xml version="1.0" encoding="utf-8"?>
<AnalyserConfig version="3.2">

<!-- specify other settings for the service here -->

  <Filter enabled="true" filename="msg_filter.xml">

<!-- specify other filter settings here -->

    <Dns>
      <AttemptsCount>3</AttemptsCount>
      <TtlForUnknown>3600</TtlForUnknown>
      <MinTtl>300</MinTtl>
      <MaxTtl>604800</MaxTtl>
      <Server ipaddress="127.0.0.1" port="53" />
    </Dns>

  </Filter>
</AnalyserConfig>
```

В данном случае предполагается, что DNS-сервер располагается по адресу 127.0.0.1:53.

2. Настройка фильтра сообщений

Например, файл msg_filter.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="DNS resolve" version="1.0">

  <table name="main">

    <rule enabled="true">
      <comment>
        Resolve the sender and the recipient host names
        for the message.
      </comment>
      <action name="dns" address="both"/>
    </rule>

    <rule enabled="true">
      <comment>
        Accept all message that reached this point.
      </comment>
      <action name="accept" />
    </rule>

  </table>
</filter>
```

Комментарии и общие рекомендации

1. После выполнения действия DNS в метаданные сообщения будут добавлены заголовки X-Sensor-Src-Host, X-Sensor-Dst-Host со значениями доменных имён или со значением "not resolved", если получить какое-то из доменных имён не удалось.

2. Для более быстрого разрешения имён DNS в конфигурации службы EtherSensor Analyser желательно указывать как можно более быстрые DNS-серверы. Это могут быть как серверы Интернет-провайдера, так и собственные DNS-серверы.

3. Действие DNS (разрешение имён DNS) является относительно длительной операцией, в связи с этим старайтесь выполнять его в фильтре только для тех сообщений, для которых это необходимо.

4. Старайтесь применять действие DNS в конце фильтра и непосредственно в том месте, где понадобятся результаты его работы.

Например, если необходимо, чтобы на выходе в сообщении адреса хостов отправителя и получателя просто содержали доменные имена, достаточно применять действие DNS прямо перед окончанием фильтрации сообщения и действием ACCEPT. Нет необходимости делать это в начале фильтра, так как в ходе фильтрации часть сообщений может быть отклонена действием DROP и для этих сообщений действие DNS (если оно установлено в начале фильтра) будет производить лишнюю нагрузку на EtherSensor.

Если же требуется полученные DNS-имена отправителя и получателя использовать далее в условиях фильтра для отсеивания сообщений по именам хостов, то разрешение имён DNS

следует делать в правиле, которое расположено непосредственно перед правилом, в условиях которого будут использоваться полученные DNS-имена.

Поиск неисправностей

Если разрешение DNS-имён для IP-адресов отправителя или получателя не происходит:

1. Проверьте доступность DNS-сервера непосредственно с машины-сенсора при помощи утилиты ping (например, ping <IP-адрес-dns-сервера>) и telnet (например, telnet <IP-адрес-dns-сервера>).

2. Убедитесь, что используемый DNS-сервер может производить разрешение имён для машины-сенсора. Это можно сделать утилитой nslookup.

- В командной строке запустите утилиту nslookup.
- В запущенной утилите выполните команду server <IP-адрес-dns-сервер>, чтобы установить DNS-сервер, через который будет проводиться разрешение имён.
- Введите IP-адрес, имя которого не удаётся разрешить.

Если nslookup может разрешить имя хоста, то необходимо проверить ход выполнения фильтрации:

1. Убедитесь, что правило в фильтре, в котором находится действие DNS, включено и условия этого правила соответствуют сообщениям.

2. Убедитесь, что правило в фильтре, в котором находится действие DNS, вообще будет выполняться, то есть не возникает ситуации, когда сообщения принимаются правилами выше и процесс фильтрации не доходит до этого правила (например, выше этого правила нет правил с действием ACCEPT или DROP).

3. Убедитесь, что при старте EtherSensor в журналах службы EtherSensor Analyser отсутствуют сообщения о неправильной загрузке фильтра сообщений и ошибках.

Если ничего не помогло, отправьте разработчику Microolap EtherSensor:

- Несколько примеров сообщений, для которых разрешение имён не происходит.
- Используемый фильтр сообщений.
- Диагностический отчет работе EtherSensor, генерируемый утилитой sensor_console.exe из поставки Microolap EtherSensor.
- Свои комментарии и соображения по вышеперечисленным действиям.

4.3.3.2. Фильтрация по хостам

Задача

Необходимо в случае выполнения запросов на сайты <*baender*.com, benderlog.ru, banderlog.biz> прекратить обрабатывать такое сообщение и удалить накопленные данные о нем.

Описание логики решения

Возможны два варианта решения задачи:

1. Полностью игнорировать трафик на хосты с указанными именами в фильтре протокола HTTP. Для этого необходимо написать условие фильтрации по заголовку HTTP-запроса "Host".

2. В фильтре сообщений удалять сообщения, отправляемые на указанные хосты. Для этого необходимо произвести разрешение DNS-имён и написать условие фильтрации "hostname".

Вариант 1 является более предпочтительным, так как позволяет отфильтровывать ненужные данные на более раннем этапе, что значительно снижает нагрузку на EtherSensor.

Решение

1. Вариант 1 – игнорирование трафика на ранней стадии при помощи фильтра протокола HTTP.

Например, файл фильтра HTTP-протокола может выглядеть так:

```
<?xml version="1.0" encoding="utf-8"?>
<filter name="HTTP filter" version="1.0">
  <table name="main">

    <rule enabled="true">
      <match>
        <or>
          <c name="req-header"
             headername="Host"
             op="wc"
             value="*baender*.com*"/>
          <c name="req-header"
             headername="Host"
             op="eq"
             value="benderlog.ru"/>
          <c name="req-header"
             headername="Host"
             op="eq"
             value="banderlog.biz"/>
        </or>
      </match>
      <action name="drop" />
    </rule>

    <rule enabled="true">
      <action name="accept" />
    </rule>

  </table>
</filter>
```

Подробное описание условия фильтрации "req-header" смотрите в разделе Условие REQ-HEADER, RESP-HEADER [174].

2. Вариант 2 – удаление сообщений в фильтре сообщений.

Необходимо включить разрешение DNS-имён, файл фильтра сообщений может выглядеть так:

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

  <table name="main">

    <rule enabled="1">
      <match>
        <or>
          <c name="hostname"
             address="server"
             op="wc"
             value="*baender*.com*" />
          <c name="hostname"
             address="server"
             op="eq"
             value="benderlog.ru" />
          <c name="hostname"
             address="server"
             op="eq"
             value="banderlog.biz" />
        </or>
      </match>
      <action name="drop"/>
    </rule>

    <rule enabled="1">
      <action name="accept" />
    </rule>

  </table>
</filter>
```

Подробное описание условия фильтрации "hostname" смотрите в разделе Условие HOSTNAME [124].

Комментарии и общие рекомендации

1. В варианте 1 обратите внимание, что проверка имени хоста из заголовка "Host" по wildcard-маске "*baender*.com*" содержит "*" в начале и в конце маски. В начале маски это необходимо потому, что в начале имени хоста могут идти имена доменов верхнего уровня (например, www.baender.com или 123.baender.com). В конце маски это необходимо потому, что в конце имени хоста в заголовке "Host" может быть указан порт назначения (например, baender.com:80).Проверка на равенство (операция в условии "eq") подразумевает просто поиск подстроки в строке. Поэтому "eq" со значением "benderlog.ru" будет срабатывать и для www.benderlog.ru и для benderlog.ru:80.

2. Для варианта 2 в общем случае для работы условия "hostname" необходимо предварительно произвести разрешение имён DNS. Однако, в указанном случае (мы проверяем только хост назначения и только для протокола HTTP) этого делать не обязательно, так как значение хоста будет доступно из заголовка "Host" HTTP-протокола.

4.3.3.3. Фильтрация по URL

Задача

Необходимо в случае присутствия в URL слов <forum, phorum, post, submit> для метода get установить на сообщение пометку "user reads forums".

Описание логики решения

Проверьте URL и метод HTTP-запроса, используя фильтр HTTP-протокола. Для проверки URL следует использовать условие "url", для проверки метода HTTP-протокола следует использовать условие "method".

Решение

Например, файл фильтра HTTP-протокола может выглядеть так:

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

  <table name="main">

    <rule enabled="true">
      <comment>
        Detect requests possibly related to forums.
      </comment>
      <match>
        <and>
          <c name="method"
             value="GET"/>
          <c name="url"
             op="re"
             value="http://.*/.*(forum|phorum|post|submit).*" />
        </and>
      </match>
        <action name="tag" value="USER_READS_FORUMS"/>
    </rule>

    <rule enabled="true">
      <action name="accept" />
    </rule>
  </table>
</filter>
```

Подробное описание условий фильтрации "url" и "method" смотрите в разделах **Условие URL**[177] и **Условие METHOD**[169].

Комментарии и общие рекомендации

1. Следует помнить, что на этапе фильтрации HTTP-протокола ещё не существует сообщений, проверка на их наличие в трафике будет выполнена позже. Однако, метки и теги, установленные

для запросов на этом этапе будут сохранены и в сообщении (если оно будет извлечено из этих запросов). Эти метки и теги будут доступны для проверки в фильтре сообщений и в дальнейшем в виде заголовков X-Sensor-Tags и X-Sensor-Labels.

2. Следует помнить, что в условие "url" попадает полный URL запроса, включая имя хоста (т.е. в виде http://www.mail.ru/mail/read.php?some=parameter&param2), это необходимо учитывать в условии проверки.

4.3.3.4. Фильтрация по HTTP+DNSBL

Задача

Если сообщение получено или передано методом GET HTTP-протокола и адрес сервера есть в списке dnsbl (например, dnsbl.sorbs.net), то необходимо пометить сообщение меткой "possible malware or proxy".

Описание логики решения

Проверьте метод HTTP-запроса, используя фильтр HTTP-протокола. Чтобы убедиться, что адрес сервера присутствует в некотором DNSBL-списке, необходимо использовать фильтр сообщений. Таким образом для решения общей задачи будут использованы два фильтра: фильтр HTTP-протокола и фильтр сообщений. В фильтре HTTP-протокола будет проверен метод GET и поставлена некоторая метка на все GET-запросы (например "HTTP_GET").

Эта метка останется и в сообщениях, которые будут извлечены из этих запросов и будет доступна далее в фильтре сообщений. В фильтре сообщений, для сообщений, содержащих метку "HTTP_GET", будет проверено наличие адреса сервера в DNSBL-списке через выполнение действия "dnsbl".

Если адрес сервера будет входить в DNSBL-список, то на это сообщение будет установлена метка (например, "DNSBL_EXIST"). Далее в фильтре сообщений необходимо будет проверить для сообщения уже две метки – "HTTP_GET" и "DNSBL_EXIST" и уже для сообщений с обеими метками установить метку "possible malware or proxy".

Также для корректной работы действия "dnsbl" необходимо в конфигурации службы EtherSensor Analyser настроить DNS-серверы для проверки DNSBL.

Решение

1. Настройка DNS-серверов в конфигурации службы EtherSensor Analyser для проверки DNSBL.

Настройку можно производить в консоли управления или в конфигурационном файле.

В файле конфигурации службы EtherSensor Analyser:

Корневой тег <AnalyserConfig>, далее вложенный тег <RawHttpFilter> – включаем фильтр HTTP-протокола. Тег настроек фильтров <Filter> (включаем фильтр сообщений). Далее вложенный тег <DnsBl> – настройки DNS-серверов для проверок адресов в DNSBL-списках в фильтре сообщений.

```
<?xml version="1.0" encoding="utf-8"?>
<AnalyserConfig version="3.2">

<!-- other service settings -->
  <RawHttpFilter enabled="true" filename="http-filter.xml" />
  <Filter enabled="true" filename="msg_filter.xml">

<!-- other filter settings -->

    <DnsBl>
      <AttemptsCount>3</AttemptsCount>
      <TtlForUnknown>3600</TtlForUnknown>
      <MinTtl>300</MinTtl>
      <MaxTtl>604800</MaxTtl>
      <Server ipaddress="127.0.0.1" port="53" />
    </DnsBl>

  </Filter>

</AnalyserConfig>
```

В данном случае предполагается, что DNS-сервер располагается по адресу 127.0.0.1:53.

2. Файл фильтра HTTP-протокола может выглядеть так (http-filter.xml):

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

  <table name="main">

    <rule enabled="true">
      <match>
        <c name="method" value="GET"/>
      </match>
      <action name="tag" value="HTTP_GET"/>
    </rule>

    <rule enabled="true">
      <action name="accept" />
    </rule>
  </table>
</filter>
```

Подробное описание условия фильтрации "method" и действия "tag" смотрите в разделах Условие METHOD [169] и Действие TAG [188].

3. Файл фильтра сообщений может выглядеть так (msg_filter.xml):

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

  <table name="main">

    <rule enabled="true">
      <match>
        <c name="tag" tag="HTTP_GET"/>
      </match>
      <action name="dnsbl-rh"
              address="both"
              tag="DNSBL_EXIST"
              value="dnsbl.sorbs.net" />
    </rule>

    <rule enabled="true">
      <match>
        <and>
          <c name="tag" tag="HTTP_GET"/>
          <c name="tag" tag="DNSBL_EXIST"/>
        </and>
      </match>
      <action name="tag" value="MALWARE_OR_PROXY"/>
    </rule>

  </table>
</filter>
```

Подробное описание условия "tag", действия "tag" и действия "dnsbl" смотрите в разделах Условие TAG[179], Действие TAG[188] и Действие DNSBL-LH, DNSBL-RH[152].

Комментарии и общие рекомендации

1. Для более быстрого разрешения имён DNS в конфигурации службы EtherSensor Analyser желательно указывать как можно более быстрые DNS-сервера. Это могут быть как серверы Интернет-провайдера, так и собственные DNS-серверы.

2. Действие "dnsbl" (разрешение имён DNS для DNSBL) является относительно длительной операцией (особенно если указано использование нескольких DNSBL-сервисов), в связи с этим старайтесь выполнять его в фильтре только для тех сообщений, для которых это необходимо.

3. Старайтесь применять действие "dnsbl" в конце фильтра, в месте, где понадобятся результаты его работы.

4. Следует помнить, что на этапе фильтрации HTTP-протокола ещё не существует сообщений, проверка на их наличие в трафике будет выполнена позже. Однако метки и теги, установленные для запросов, на этом этапе будут сохранены в сообщении (если оно будет извлечено из этих запросов).

4.3.3.5. Фильтрация больших объектов HTTP

Задача

Иногда по протоколу HTTP происходит передача очень больших объектов (закачка файлов, просмотр фильмов онлайн). При большом количестве таких объектов, передаваемых одновременно, EtherSensor может потреблять большое количество оперативной памяти. Такая ситуация может приводить к общей деградации производительности EtherSensor.

Необходимо с помощью фильтров отсекать большие HTTP-объекты и удалять их до начала анализа.

Описание логики решения

Для удаления больших HTTP-объектов до их полного анализа (и загрузки для этого в память целиком) необходимо применить HTTP-фильтр. Для проверки размера HTTP-запроса или ответа следует использовать условие "size", которое проверяет размер и запроса и ответа.

Решение

Например, файл фильтра HTTP-протокола может выглядеть так:

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

  <table name="main">

    <rule enabled="1">
      <comment>
        The rule stops processing HTTP objects for which
        the request or response size is greater than 1MB.
      </comment>
      <match>
        <c name="size" op="gt" value="1M"/>
      </match>
        <action name="drop" />
    </rule>

    <rule enabled="true">
      <action name="accept" />
    </rule>
  </table>
</filter>
```

Подробное описание условий фильтрации HTTP-запросов "size" смотрите в разделе Условие REQ-SIZE, RESP-SIZE, SIZE [172].

Комментарии и общие рекомендации

1. Вместо условия "size", которое проверяет и размер запроса и размер ответа, можно применять условия "req-size" (для проверки размера только HTTP-запроса) или "resp-size" (для проверки размера только HTTP-ответа).

5. Доставка результатов системам-потребителям

Служба EtherSensor Transfer отвечает за доставку результатов работы EtherSensor Analyser внешним системам-потребителям.

Основной идеей службы EtherSensor Transfer является понятие заранее определенного транспортного профиля (профиля доставки результатов) для доставки извлечённого из трафика объекта.

Транспортный профиль содержит данные о сервере и аутентификации на нем, если система-потребитель перехваченных объектов является сервером, или же путь к локальной директории, а также требования к сохраняемым объектам, если речь идет о профилях группы ARCHIVING[208].

В правилах фильтрации[113] сообщений службы EtherSensor Analyser один профиль в случае необходимости может быть мгновенно заменен другим, а так как профили созданы администратором заранее и тщательно проверены, вероятность ошибок в этой части настроек минимальна.

Результаты анализа одного и того же перехваченного объекта могут быть одновременно доставлены многими способами многим потребителям с использованием многих транспортных профилей.

> Например:
>
> Контент и метаданные сообщения доставляются одновременно в DLP-систему и в eDiscovery-систему, метаданные – в SIEM-систему, работающую в SOC внешнего MSSP, а файл аттачмента – в sandbox.

Типы транспортных профилей:

ARCHIVING[208]

Профили группы ARCHIVING служат для доставки контента перехваченных объектов и их метаданных системам типа eDiscovery, Enterprise Archiving, Enterprise Search и многим DLP-системам. При этом используются методы доставки SMTP/SMTPS[223], FTP/FTPS[211], SFTP[217], IMAP[214], FILEDROP[208] (локальная файловая система) и SMB/CIFS[220] (сетевой каталог).

DLP [226]

Профили группы DLP служат для доставки DLP-системам контента перехваченных объектов и их метаданных. Все поддерживаемые DLP-системы имеют свой архив, и именно в него EtherSensor напрямую доставляет перехваченные объекты с использованием их проприетарных протоколов передачи данных.

SIEM [235]

Профили группы SIEM служат для доставки в SIEM-системы по протоколу SYSLOG данных о перехваченном объекте. Если необходимо, доставляется также и контент объекта.

SANDBOX [238]

Профили группы SANDBOX служат для доставки в решения класса sandbox подозрительных на вредоносность перехваченных объектов для дальнейшего анализа.

STATS [243]

Профили группы STATS служат для доставки статистических данных на NetFlow-коллекторы, в частности, на NetFlow-коллекторы SIEM-систем.

GROUP [246]

Групповые профили служат для балансировки нагрузки между системами-потребителями, включают в себя перечисленные выше транспортные профили с заранее установленными весами.

Транспортный профиль перехваченному объекту назначается в фильтре сообщений [109]. Если в процессе анализа объекта выяснилось, что ему не был назначен ни один транспортный профиль, то он доставляется профилем по умолчанию. После успешной доставки объект удаляется из кэша перехваченных объектов, вся имеющаяся информация о нём уничтожается.

Основной формат, в котором EtherSensor поставляет контент реконструированных объектов системам-потребителям – EML-конверт. Также служба EtherSensor Transfer способна передавать данные в собственном внутреннем XML и/или JSON форматах в тех случаях, когда EML-конверт не является обязательным форматом доставки (копирование данных в директории или использование FTP-протокола).

Архитектура системы при использовании незащищённых протоколов передачи данных:

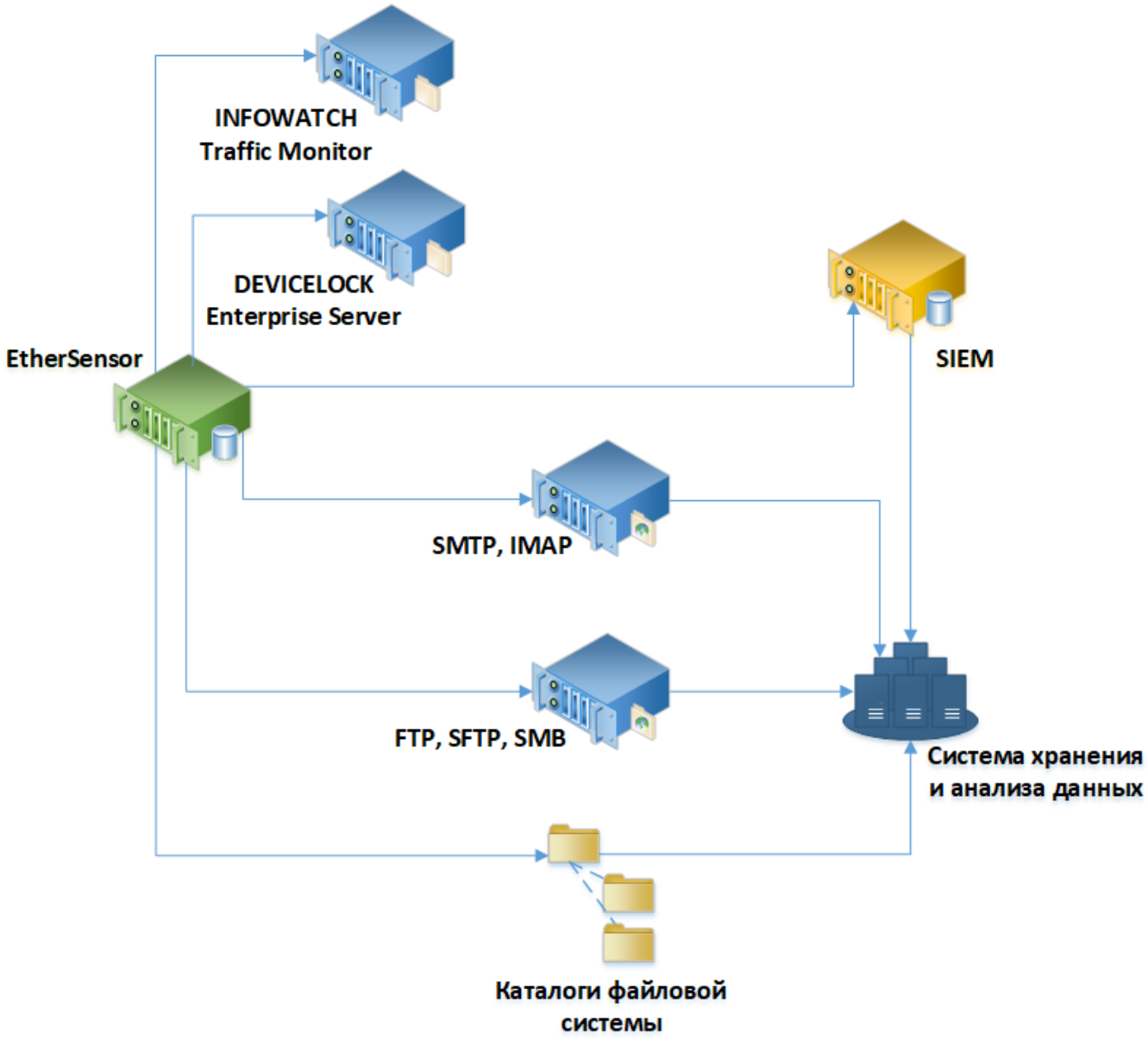

Рис.40. Схема работы службы EtherSensor Transfer.

Архитектура системы при использовании защищённых протоколов передачи данных:

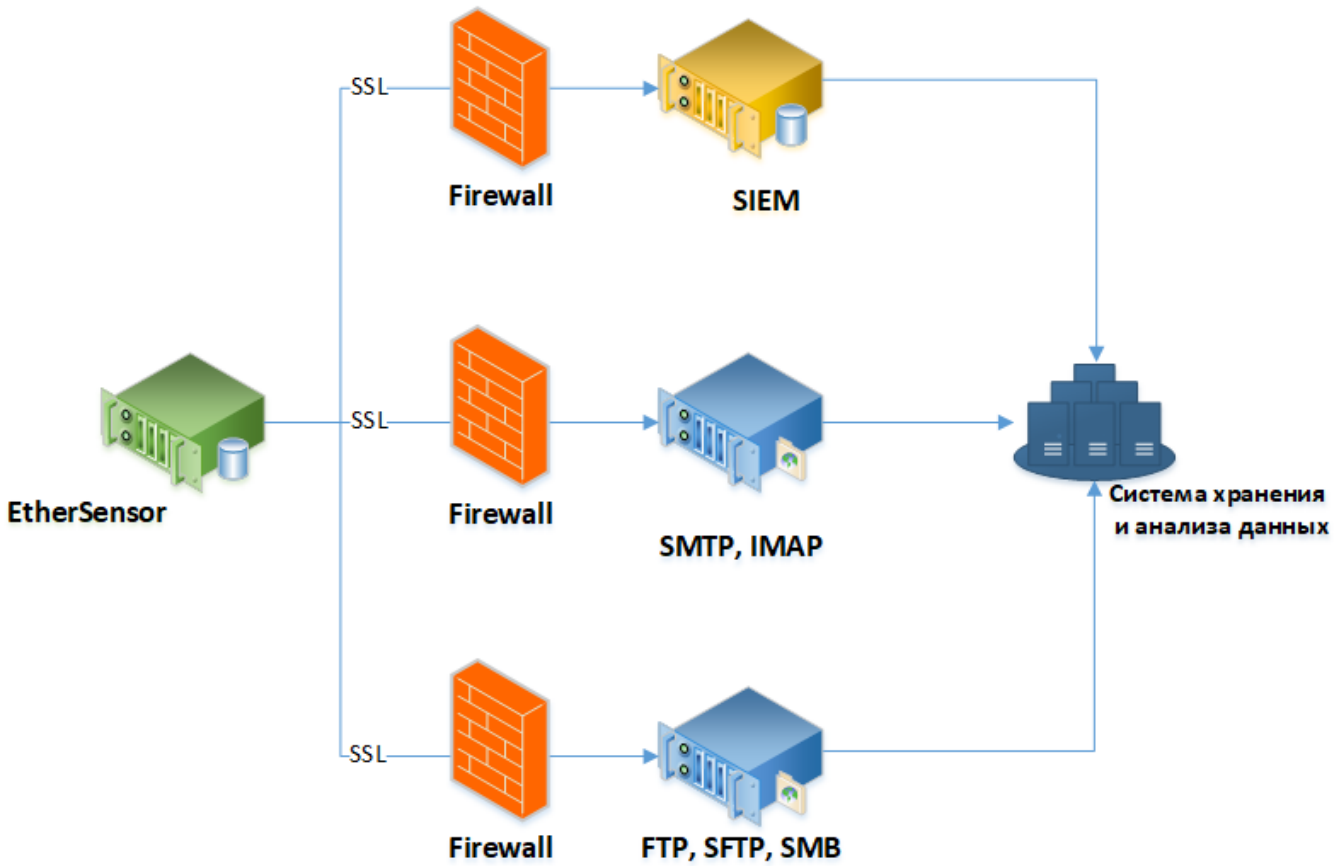

Рис.41. Работа службы EtherSensor Transfer с защищенными протоколами

Конфигурационный файл службы EtherSensor Transfer

Конфигурация службы EtherSensor Transfer содержится в XML-файле transfer.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

Параметры командной строки

Служба EtherSensor Transfer в ходе процедуры установки Microolap EtherSensor устанавливается как служба Windows, настроенная на автоматический запуск. Однако, она также может быть запущена как приложение Windows sensor_transfer.exe и имеет следующие параметры командной строки:

/process

Запустить процесс sensor_transfer.exe как обычный Windows Win32-процесс (возможно использовать для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

5.1. ARCHVING-профили

ARCHIVING-профили решают задачу доставки перехваченных объектов большинству eDiscovery, Enterprise Archiving, Enterprise Search и DLP-систем.

FILEDROP [208]

Профили FILEDROP позволяют сохранять перехваченные объекты на локальную файловую систему, что бывает очень удобно для целей отладки и тестирования. Кроме того, доступ к сохранённым объектам может быть легко предоставлен внешним системам-потребителям.

SMB/CIFS [220]

Профили SMB/CIFS позволяют сохранять перехваченные объекты непосредственно на сетевые папки систем-потребителей.

SMTP/SMTPS [223]**, FTP/FTPS** [211]**, SFTP** [217]**, IMAP** [214]

Данные профили доставки результатов анализа удобны для доставки перехваченных объектов удалённым/облачным системам-потребителям.

При обработке сервером EtherSensor больших потоков данных внешние системы-потребители могут не справиться с получением результатов анализа от EtherSensor из-за создаваемой им большой нагрузки.

В этом случае используйте групповые профили для балансировки нагрузки на системы-потребители.

5.1.1. FILEDROP-профили

Профили FILEDROP позволяют сохранять перехваченные объекты на локальную файловую систему, что бывает очень удобно для целей отладки и тестирования. Кроме того, доступ к сохранённым объектам может быть легко предоставлен внешним системам-потребителям.

Подводные камни FILEDROP:

1. При большом потоке трафика EtherSensor создаёт большой поток сообщений. Если вы эти сообщения сохраняете на диск с помощью FILEDROP, то дисковая подсистема может испытывать чрезмерную нагрузку. Таким образом она может стать узким местом сервера EtherSensor в целом. Вполне возможно, что рано или поздно возникнет дефицит системных ресурсов и пострадает основная функция EtherSensor – перехват и анализ трафика.

2. Абсолютно неприемлемо позволять большому количеству объектов бесконтрольно накапливаться на файловой системе: обязательно устанавливайте параметр ***Время жизни результатов*** в настройках FILEDROP-профилей.

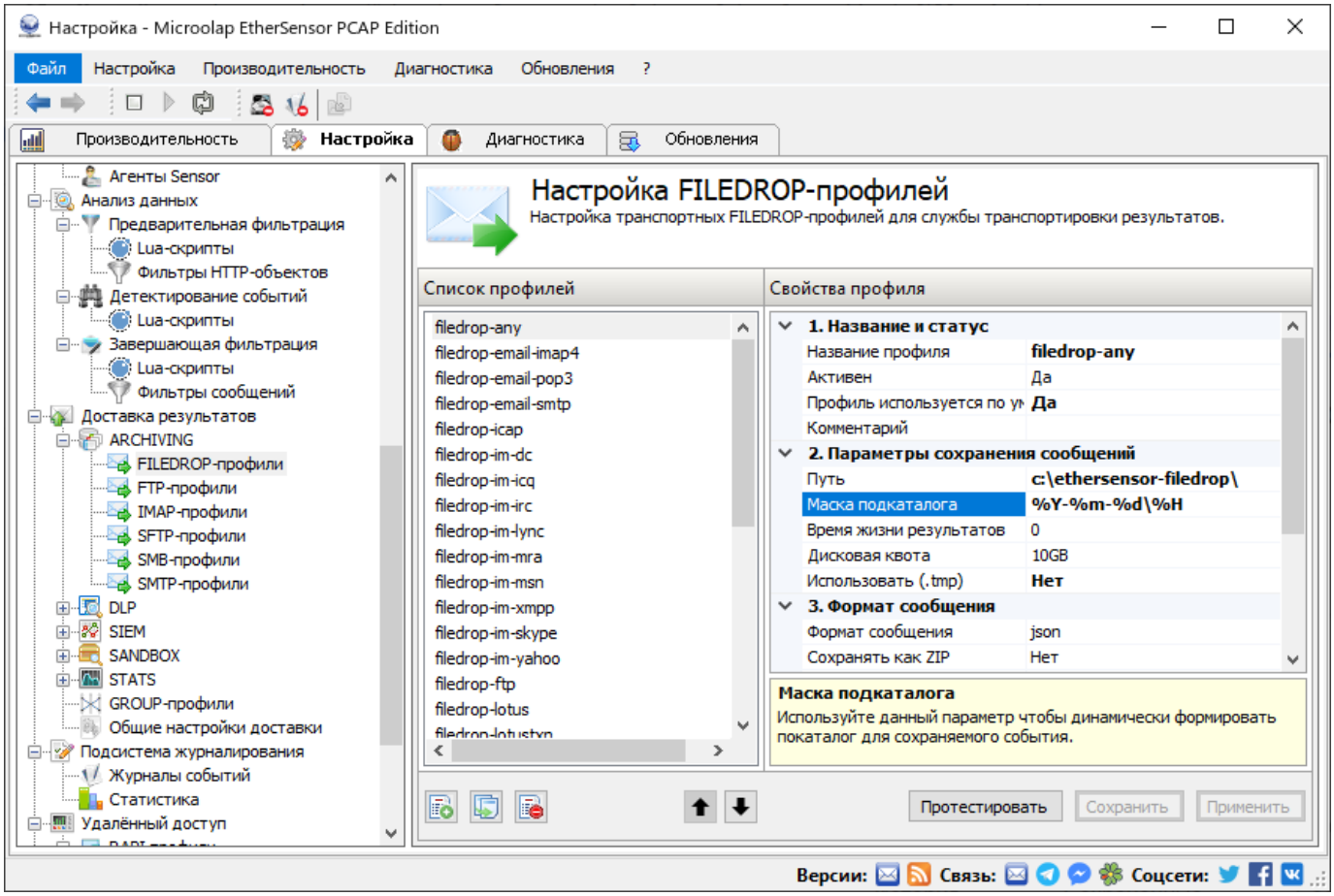

Рис.42. Настройки FILEDROP-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Параметры сохранения сообщений

Путь:

Путь к директории, в которой будут сохраняться перехваченные сообщения.

Дисковая квота:

Размер квоты дискового пространства для хранения сообщений. Пример: 10GB или 500MB или 100KB или 10000. При исчерпании квоты сохранение файлов будет остановлено до тех пор, пока квота вновь позволит это делать. Чтобы возобновить сохранение файлов, следует либо освободить место, либо увеличить квоту.

Использовать (.tmp):

Разрешить/запретить использовать двухэтапное перемещение файлов во избежание коллизий: 1) Файл перемещается с временным расширением, например 2012-01-08-15-29-48-586.7.m.zip.tmp, затем 2) После завершения перемещения из имени файла удаляется ".tmp", файл переименовывается в 2012-01-08-15-29-48-586.7.m.zip. Бывает полезно в случае отслеживания каким-либо процессом появления в директории новых файлов (переименование - атомарная операция).

3. Формат сообщения

Формат сообщения:

Формат сохраняемого или доставляемого сообщения (EML, XML, JSON...)

Сохранять как ZIP:

Упаковывать ли сохраняемые сообщения в ZIP-архив (файл с расширением ZIP). Если включена данная настройка совместно с настройкой "Сохранять как EML", то EML-конверты сообщений будут упаковываться в ZIP-архив. Если включена данная настройка, и при этом настройка "Сохранять как EML" отключена, то в ZIP-архив будут упаковываться сообщения во внутреннем формате.

Степень сжатия ZIP-архива:

Степень сжатия ZIP-архива от [0-9], где 0 - без сжатия, 1 - наилучшая скорость сжатия, 9 - наилучший алгоритм сжатия.

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

4. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

5. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.1.2. FTP-профили

FTP/FTPS-профили используются для доставки перехваченных объектов удалённым/облачным системам-потребителям.

При обработке сервером EtherSensor больших потоков данных внешние системы-потребители могут не справиться с получением результатов анализа от EtherSensor из-за создаваемой им большой нагрузки.

В этом случае используйте групповые профили для балансировки нагрузки на системы-потребители.

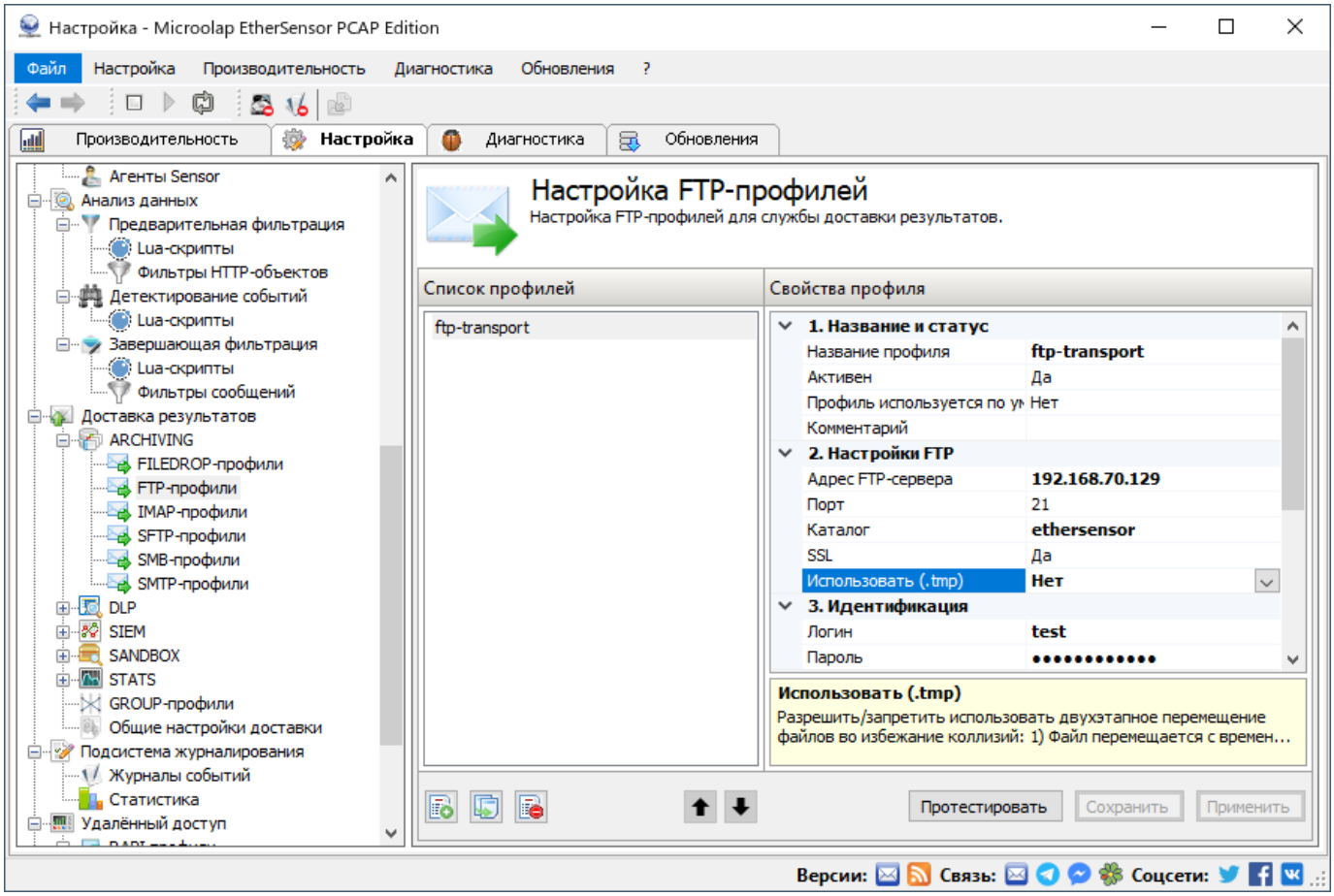

Рис.43. Настройки FTP-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки FTP

Адрес FTP-сервера:

IP-адрес или имя FTP-сервера для отправки сообщений.

Порт:

Порт FTP-сервера для отправки сообщений.

Каталог:

Каталог для сохранения сообщений.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Использовать (.tmp):

Разрешить/запретить использовать двухэтапное перемещение файлов во избежание коллизий: 1) Файл перемещается с временным расширением, например 2012-01-08-15-29-48-586.7.m.zip.tmp, затем 2) После завершения перемещения из имени файла удаляется ".tmp", файл переименовывается в 2012-01-08-15-29-48-586.7.m.zip. Бывает полезно в случае отслеживания каким-либо процессом появления в директории новых файлов (переименование - атомарная операция).

3. Идентификация

Логин:

Логин на доступ к FTP-серверу.

Пароль:

Пароль на доступ к FTP-серверу.

4. Формат сообщения

Формат сообщения:

Формат сохраняемого или доставляемого сообщения (EML, XML, JSON...)

Сохранять как ZIP:

Упаковывать ли сохраняемые сообщения в ZIP-архив (файл с расширением ZIP). Если включена данная настройка совместно с настройкой "Сохранять как EML", то EML-конверты сообщений будут упаковываться в ZIP-архив. Если включена данная настройка, и при этом настройка "Сохранять как EML" отключена, то в ZIP-архив будут упаковываться сообщения во внутреннем формате.

Степень сжатия ZIP-архива:

Степень сжатия ZIP-архива от [0-9], где 0 - без сжатия, 1 - наилучшая скорость сжатия, 9 - наилучший алгоритм сжатия.

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

5. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

6. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.1.3. IMAP-профили

IMAP-профили используются для доставки перехваченных объектов удалённым/облачным системам-потребителям.

При обработке сервером EtherSensor больших потоков данных внешние системы-потребители могут не справиться с получением результатов анализа от EtherSensor из-за создаваемой им большой нагрузки.

В этом случае используйте групповые профили для балансировки нагрузки на системы-потребители.

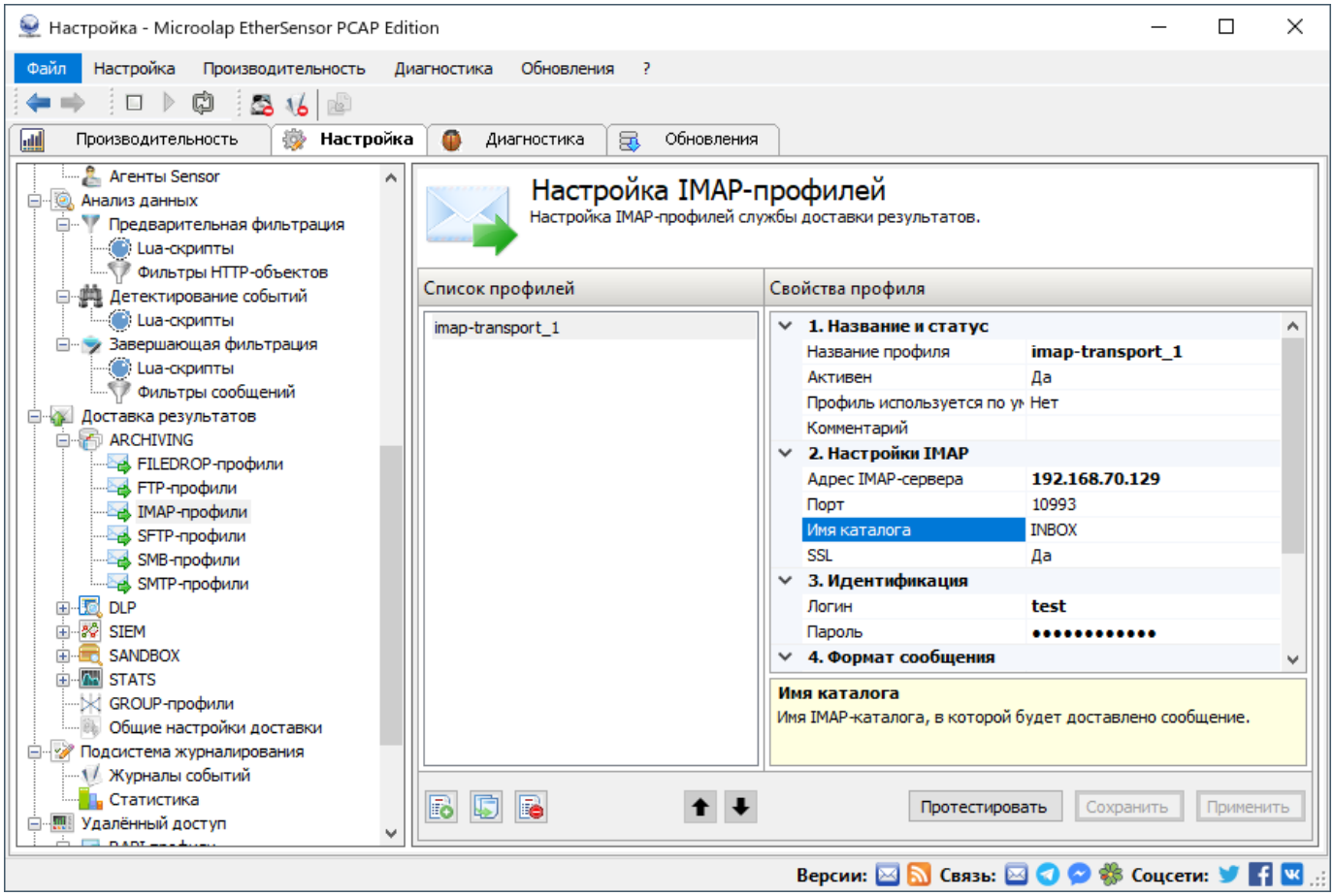

Рис.44. Настройки IMAP-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки IMAP

Адрес IMAP-сервера:

IP-адрес или имя IMAP-сервера для доставки результатов.

Порт:

Порт IMAP-сервера для доставки сообщений.

Имя каталога:

Имя IMAP-каталога, в которой будет доставлено сообщение.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

3. Идентификация

Логин:

Логин на доступ к IMAP-серверу.

Пароль:

Пароль на доступ к IMAP-серверу.

4. Формат сообщения

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

5. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

6. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.1.4. SFTP-профили

SFTP-профили используются для доставки перехваченных объектов удалённым/облачным системам-потребителям.

При обработке сервером EtherSensor больших потоков данных внешние системы-потребители могут не справиться с получением результатов анализа от EtherSensor из-за создаваемой им большой нагрузки.

В этом случае используйте групповые профили для балансировки нагрузки на системы-потребители.

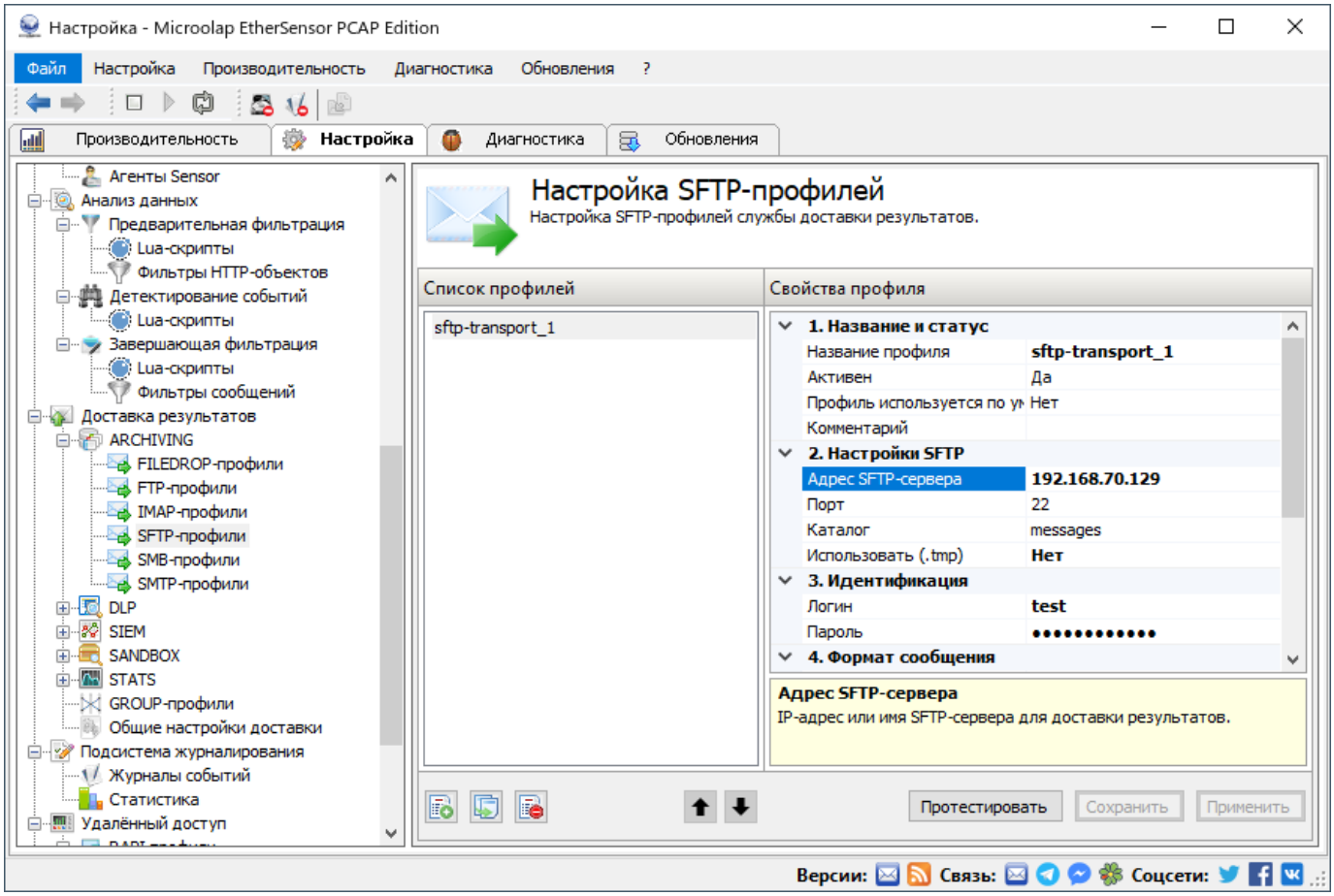

Рис.45. Настройки SFTP-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки SFTP

Адрес SFTP-сервера:

IP-адрес или имя SFTP-сервера для доставки результатов.

Порт:

Порт SFTP-сервера для отправки сообщений.

Каталог:

Каталог для сохранения сообщений.

Использовать (.tmp):

Разрешить/запретить использовать двухэтапное перемещение файлов во избежание коллизий: 1) Файл перемещается с временным расширением, например 2012-01-08-15-29-48-586.7.m.zip.tmp, затем 2) После завершения перемещения из имени файла удаляется ".tmp", файл переименовывается в 2012-01-08-15-29-48-586.7.m.zip. Бывает полезно в случае отслеживания каким-либо процессом появления в директории новых файлов (переименование - атомарная операция).

3. Идентификация

Логин:

Логин на доступ к SFTP-серверу.

Пароль:

Пароль на доступ к SFTP-серверу.

4. Формат сообщения

Формат сообщения:

Формат сохраняемого или доставляемого сообщения (EML, XML, JSON...)

Сохранять как ZIP:

Упаковывать ли сохраняемые сообщения в ZIP-архив (файл с расширением ZIP). Если включена данная настройка совместно с настройкой "Сохранять как EML", то EML-конверты сообщений будут упаковываться в ZIP-архив. Если включена данная настройка, и при этом настройка "Сохранять как EML" отключена, то в ZIP-архив будут упаковываться сообщения во внутреннем формате.

Степень сжатия ZIP-архива:

Степень сжатия ZIP-архива от [0-9], где 0 - без сжатия, 1 - наилучшая скорость сжатия, 9 - наилучший алгоритм сжатия.

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

5. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

6. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.1.5. SMB-профили

Профили SMB/CIFS позволяют сохранять перехваченные объекты непосредственно на сетевые папки систем-потребителей.

> Рекомендации касательно производительности дисковой подсистемы и настроек времени жизни сохранённых объектов те же, что и для FILEDROP-профилей.

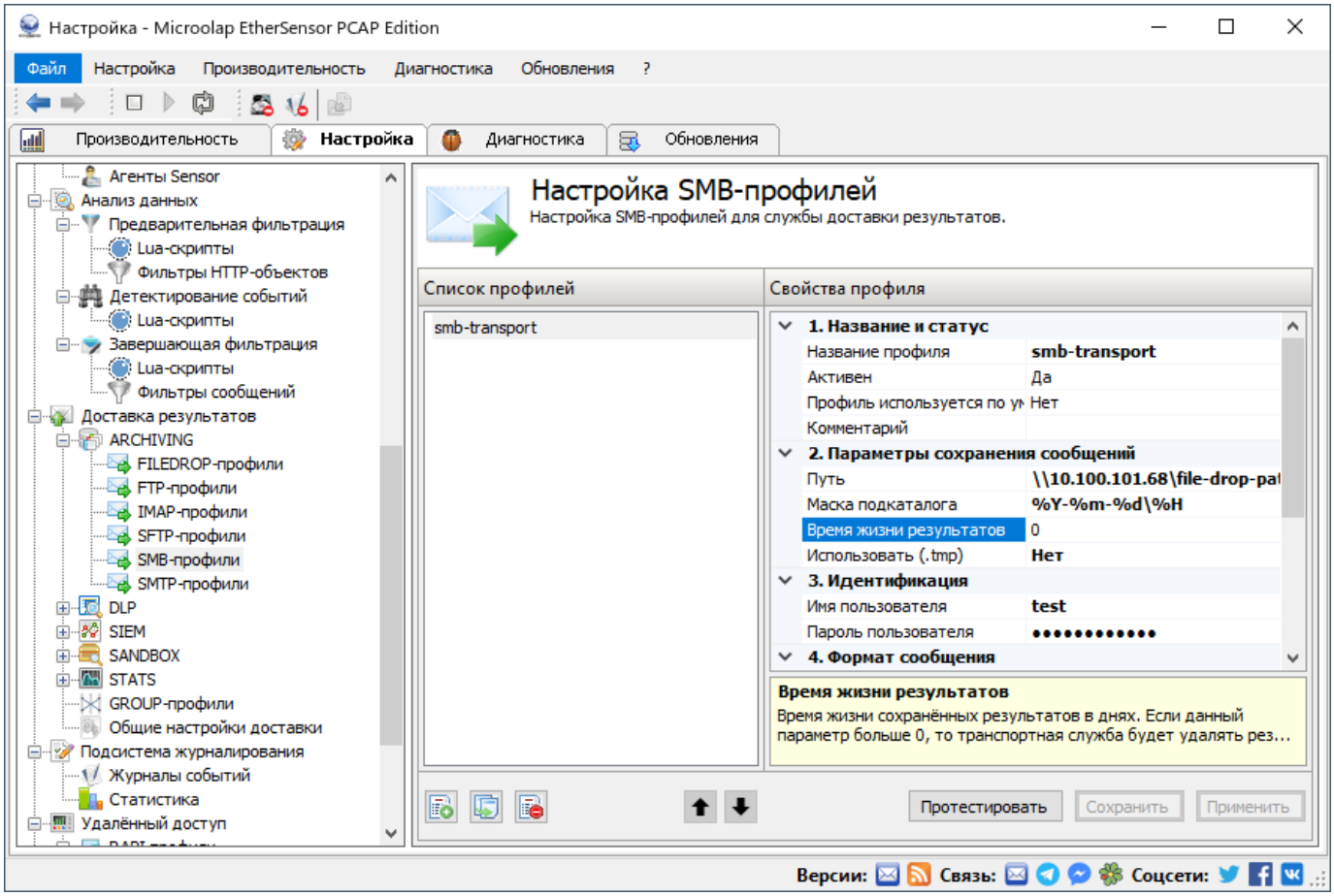

Рис.46. Настройки SMB-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Параметры сохранения сообщений

Путь:

Путь к SMB/CIFS каталогу для записи сообщения. Например: \\ARCHSERVER1\Messages

Использовать (.tmp):

Разрешить/запретить использовать двухэтапное перемещение файлов во избежание коллизий: 1) Файл перемещается с временным расширением, например 2012-01-08-15-29-48-586.7.m.zip.tmp, затем 2) После завершения перемещения из имени файла удаляется ".tmp", файл переименовывается в 2012-01-08-15-29-48-586.7.m.zip. Бывает полезно в случае отслеживания каким-либо процессом появления в директории новых файлов (переименование - атомарная операция).

3. Идентификация

Имя пользователя:

Имя пользователя для доступа к SMB/CIFS каталогу для записи сообщения.

Пароль пользователя:

Пароль пользователя для доступа к SMB/CIFS каталогу для записи сообщения.

4. Формат сообщения

Формат сообщения:

Формат сохраняемого или доставляемого сообщения (EML, XML, JSON...)

Сохранять как ZIP:

Упаковывать ли сохраняемые сообщения в ZIP-архив (файл с расширением ZIP). Если включена данная настройка совместно с настройкой "Сохранять как EML", то EML-конверты сообщений будут упаковываться в ZIP-архив. Если включена данная настройка, и при этом настройка "Сохранять как EML" отключена, то в ZIP-архив будут упаковываться сообщения во внутреннем формате.

Степень сжатия ZIP-архива:

Степень сжатия ZIP-архива от [0-9], где 0 - без сжатия, 1 - наилучшая скорость сжатия, 9 - наилучший алгоритм сжатия.

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки

сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

5. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

6. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.1.6. SMTP-профили

SMTP-профили используются для доставки перехваченных объектов удалённым/облачным системам-потребителям.

При обработке сервером EtherSensor больших потоков данных внешние системы-потребители могут не справиться с получением результатов анализа от EtherSensor из-за создаваемой им большой нагрузки.

В этом случае используйте групповые профили для балансировки нагрузки на системы-потребители.

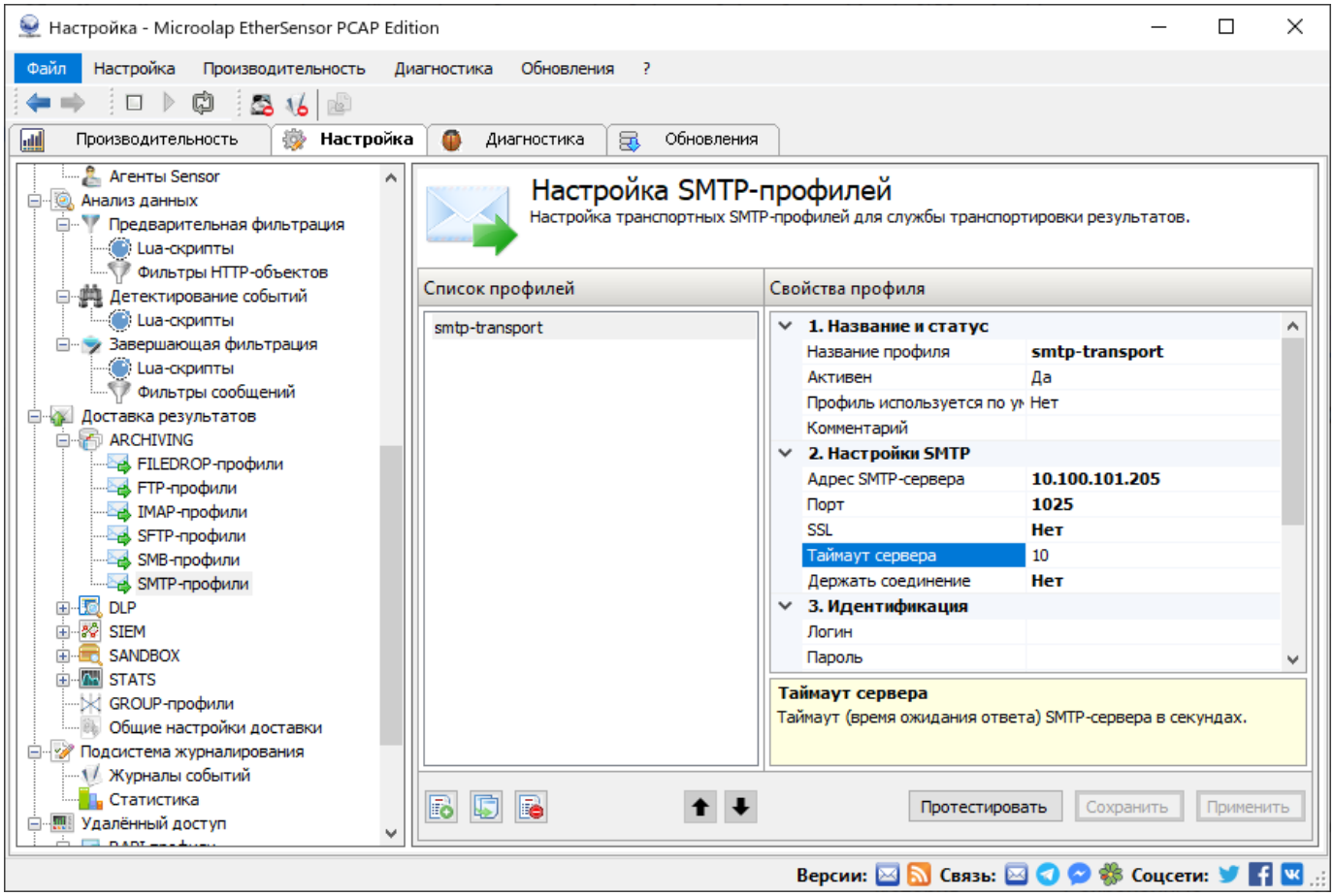

Рис.47. Настройки SMTP-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки SMTP

Адрес SMTP-сервера:

IP-адрес или имя SMTP-сервера для доставки результатов.

Порт:

Порт SMTP-сервера для отправки сообщений.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Таймаут сервера:

Таймаут (время ожидания ответа) SMTP-сервера в секундах.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

3. Идентификация

Логин:

Логин на доступ к SMTP-серверу.

Пароль:

Пароль на доступ к SMTP-серверу.

Адрес отправителя:

Адрес отправителя сообщения для системы-потребителя.

Ваш комментарий к данному профилю.

Адрес получателя:

Работающий email-адрес системы-потребителя (например, системы архивирования сообщений).

4. Формат сообщения

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

5. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

6. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.2. DLP-профили

Интеграция EtherSensor с DLP-системами решает проблему получения данных от хостов, на которые невозможно или нежелательно устанавливать развитое и ресурсоёмкое программное обеспечение endpoint-DLP.

Большинство DLP-систем поддерживает получение перехваченных сообщений универсальными способами из раздела ARCHIVING[208], но доставка перехваченных объектов некоторым из них предполагает использование их собственных проприетарных протоколов.

DeviceLock Enterprise Server[227] (DLES[227])

DeviceLock Enterprise Server – центральный архив решения DeviceLock DLP Suite. Идея интеграции EtherSensor и DLES состоит в том, что с точки зрения обмена информацией с DLES сенсор ведёт себя как endpoint-агент DeviceLock, используя для непосредственной доставки перехваченных объектов в архив DLES проприетарный протокол DeviceLock. Дальнейшая

работа над прикладными задачами происходит в парадигме DeviceLock. Для доставки объектов в DLES используются DEVICELOCK-профили [227].

Falcongaze Secure Tower [230]

Сервер Falcongaze Secure Tower – основной компонент DLP-решения от компании Falcongaze. Для доставки перехваченных объектов в архив Falcongaze Secure Tower используются FALCONGAZE-профили [230].

InfoWatch Traffic Monitor [233]

Архив InfoWatch Traffic Monitor аккумулирует важные с точки зрения задач DLP объекты для дальнейшего анализа. Перехваченные объекты EtherSensor доставляет напрямую в архив, используя INFOWATCH-профили [233].

5.2.1. DEVICELOCK-профили

DeviceLock Enterprise Server (DLES) – центральный архив решения DeviceLock DLP Suite.

Идея интеграции EtherSensor и DLES состоит в том, что с точки зрения обмена информацией с DLES EtherSensor ведёт себя как endpoint-агент DeviceLock, используя для непосредственной доставки перехваченных объектов в архив DLES проприетарный протокол DeviceLock.

Дальнейшая работа над прикладными задачами происходит в парадигме DeviceLock.

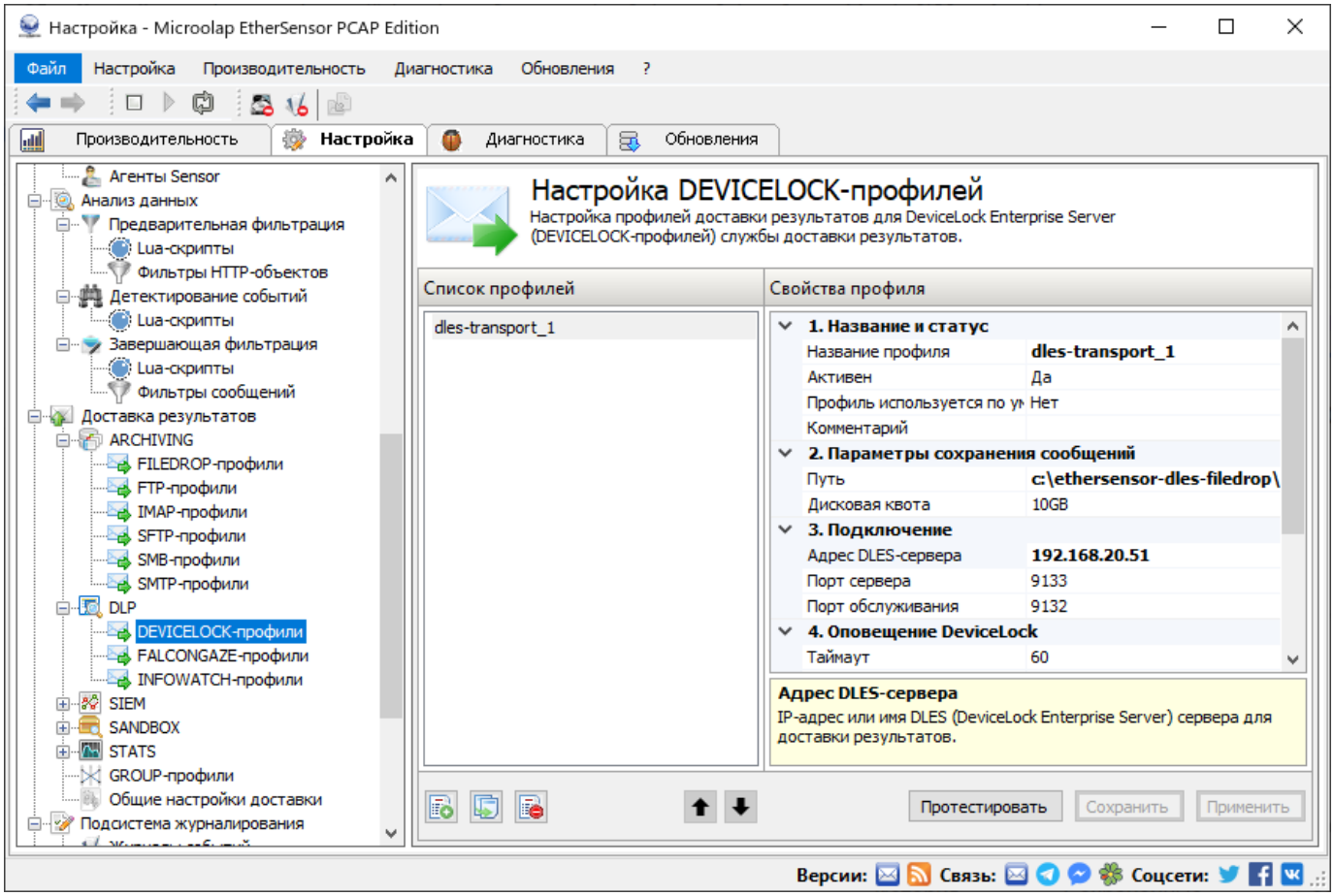

Рис.48. Настройки DEVICELOCK-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Параметры сохранения сообщений

Путь:

Путь к директории, в которой будут сохраняться перехваченные сообщения.

Используется для промежуточного хранения результатов, отправляемых в DEVICELOCK Enterprise Server.

Дисковая квота:

Размер квоты дискового пространства для хранения сообщений. Пример: 10GB или 500MB или 100KB или 10000. При исчерпании квоты сохранение файлов будет остановлено до тех пор, пока квота вновь позволит это делать. Чтобы возобновить сохранение файлов, следует либо освободить место, либо увеличить квоту.

3. Подключение

Адрес DLES-сервера:

IP-адрес или имя DLES (DeviceLock Enterprise Server) сервера для доставки результатов.

Порт сервера:

Порт, через который происходит оповещение DEVICELOCK Enterprise Server о наличии сообщений (событий).

Порт обслуживания:

Порт, через который DEVICELOCK Enterprise Server "забирает" результаты (сообщения/события).

4. Оповещение DeviceLock

Таймаут:

Устанавливает таймаут оповещения DLES (DeviceLock Enterprise Server) в секундах. При наличии неотправленных сообщений по истечении таймаута будет выполнено оповещение сервера о наличии результатов.

Количество:

Устанавливает количество накапливаемых сообщений, по достижении которого будет выполнено оповещение DEVICELOCK Enterprise сервера.

5. Формат сообщения

Дублировать заголовки:

Разрешить/запретить сохранение стандартных заголовков сообщения, а также заголовков "X-Sensor" дополнительно в отдельном аттачменте сообщения - файле "microolap_msis_headers.txt". Возможные варианты: "all" - сохраняются все заголовки сообщения; "xsensor" - сохраняются заголовки с префиксом "X-Sensor"; "other" - сохраняются стандартные заголовки "From", "To", "Cc", "Bcc" и другие, не подпадающие под действие флага "xsensor"; "none" - запрещает сохранение заголовков сообщения в отдельном вложении.

6. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

7. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.2.2. FALCONGAZE-профили

Сервер Falcongaze Secure Tower – основной компонент DLP-решения от компании Falcongaze.

Идея интеграции EtherSensor и Falcongaze Secure Tower состоит в том, что с точки зрения обмена информацией с Falcongaze сенсор ведёт себя как endpoint-агент Falcongaze, используя для непосредственной доставки перехваченных объектов в архив проприетарный протокол Falcongaze.

Дальнейшая работа над прикладными задачами происходит в парадигме Falcongaze Secure Tower.

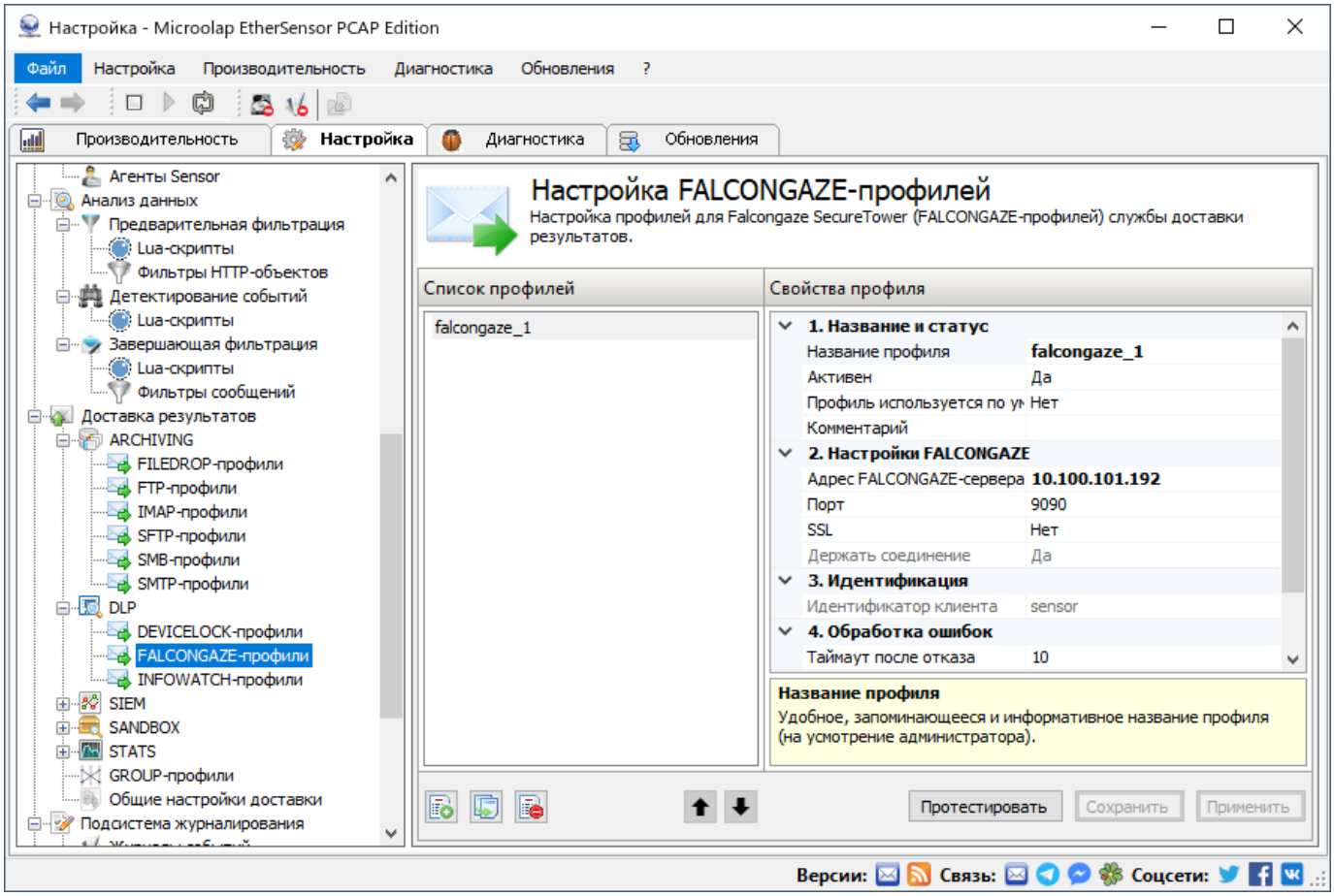

Рис.49. Настройки FALCONGAZE-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки FALCONGAZE

Адрес FALCONGAZE-сервера:

IP-адрес или имя сервера Falcongaze SecureTower для доставки результатов.

Порт:

Порт сервера Falcongaze SecureTower для отправки сообщений..

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

3. Идентификация

Идентификатор клиента:

Уникальный идентификатор клиента. Служит для регистрации клиента в Falcongaze SecureTower для отправки сообщений.

4. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

5. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.2.3. INFOWATCH-профили

Архив InfoWatch Traffic Monitor аккумулирует важные с точки зрения задач DLP объекты для дальнейшего анализа.

Идея интеграции EtherSensor и InfoWatch Traffic Monitor состоит в том, что с точки зрения обмена информацией с архивом InfoWatch Traffic Monitor сенсор ведёт себя как endpoint-агент InfoWatch, используя для непосредственной доставки перехваченных объектов в архив проприетарный протокол InfoWatch.

Дальнейшая работа над прикладными задачами происходит в парадигме InfoWatch Traffic Monitor.

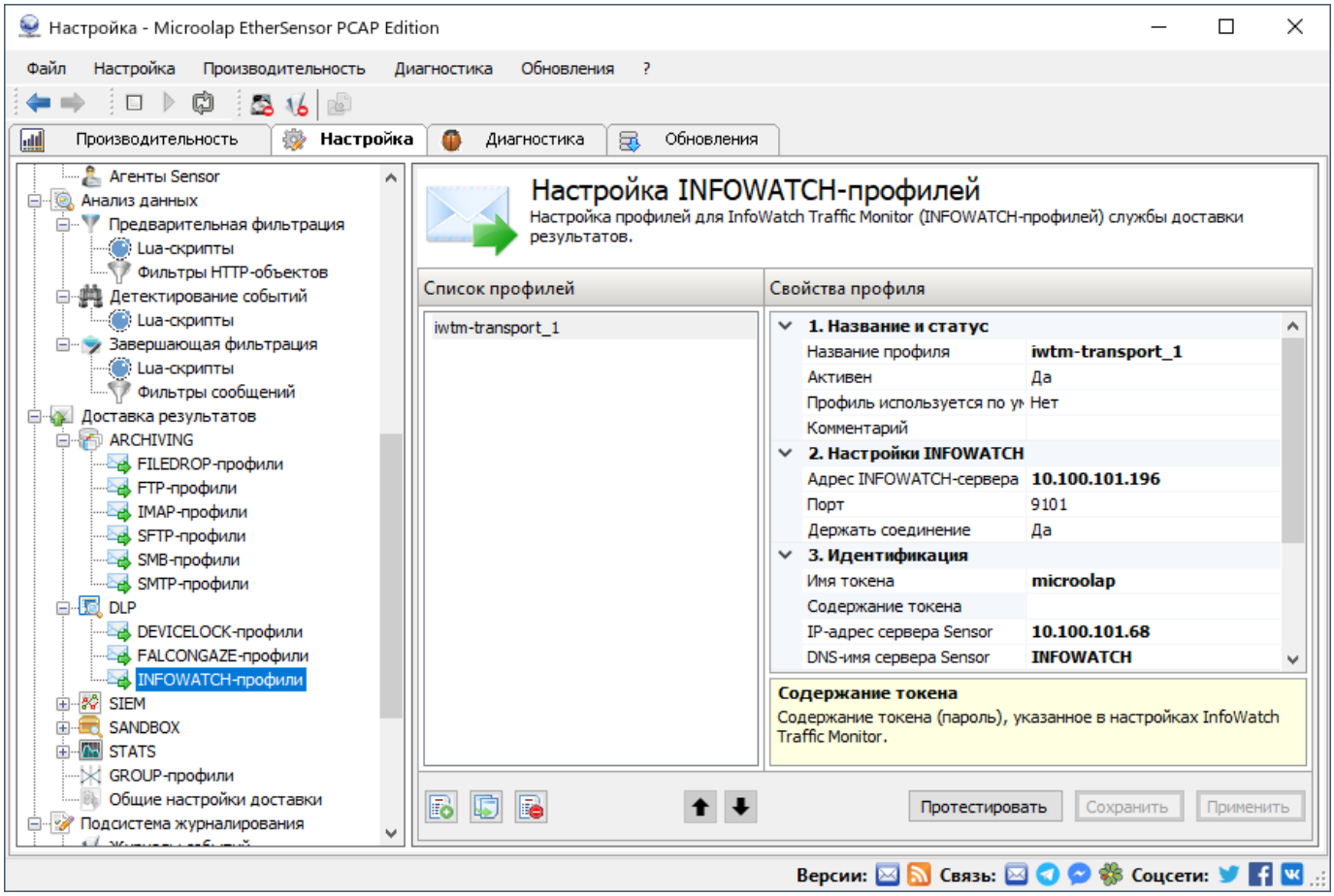

Рис.50. Настройки INFOWATCH-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки INFOWATCH

Адрес INFOWATCH-сервера:

IP-адрес или имя сервера InfoWatch Traffic Monitor для доставки результатов.

Порт:

Порт сервера ITM (InfoWatch Traffic Monitor) для отправки сообщений, по умолчанию 9101.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

3. Идентификация

Имя токена:

Имя токена, указанное в настройках InfoWatch Traffic Monitor, по умолчанию "microolap".

Содержание токена:

Содержание токена (пароль), указанное в настройках InfoWatch Traffic Monitor.

IP-адрес сервера EtherSensor:

IP-адрес сервера EtherSensor, позволяет InfoWatch Traffic Monitor различать различные экземпляры одного источника данных.

DNS-имя сервера EtherSensor:

DNS-имя сервера EtherSensor, позволяет серверу InfoWatch Traffic Monitor различать различные источники данных.

4. Формат сообщения

Формат сообщения:

Формат сохраняемого или доставляемого сообщения (EML, XML, JSON...)

5. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

6. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.3. SIEM-профили

SIEM-профили служат для доставки результатов анализа перехваченных объектов системам-потребителям, принимающим данные о событиях через SYSLOG-сервер.

Как правило, это SIEM-системы, а любая SIEM-система имеет в своём составе SYSLOG-сервер. EtherSensor может динамически формировать произвольную SYSLOG-строку с помощью пользовательского Lua-скрипта[238], адаптируя её формат к требованиям конкретной SIEM-системы.

5.3.1. SYSLOG-профили

SYSLOG-профили служат для доставки результатов анализа перехваченных объектов системам-потребителям, принимающим данные о событиях через SYSLOG-сервер (как правило, это SIEM-системы).

SYSLOG-строка может быть сформирована как в результате работы фильтра службы EtherSensor Analyser[110], так и посредством обработки перехваченного объекта заранее подготовленным Lua-скриптом[238].

Использование Lua-скриптов позволяет в реальном времени подготовить данные в специфическом для любой SIEM-системы формате без так называемых "коннекторов".

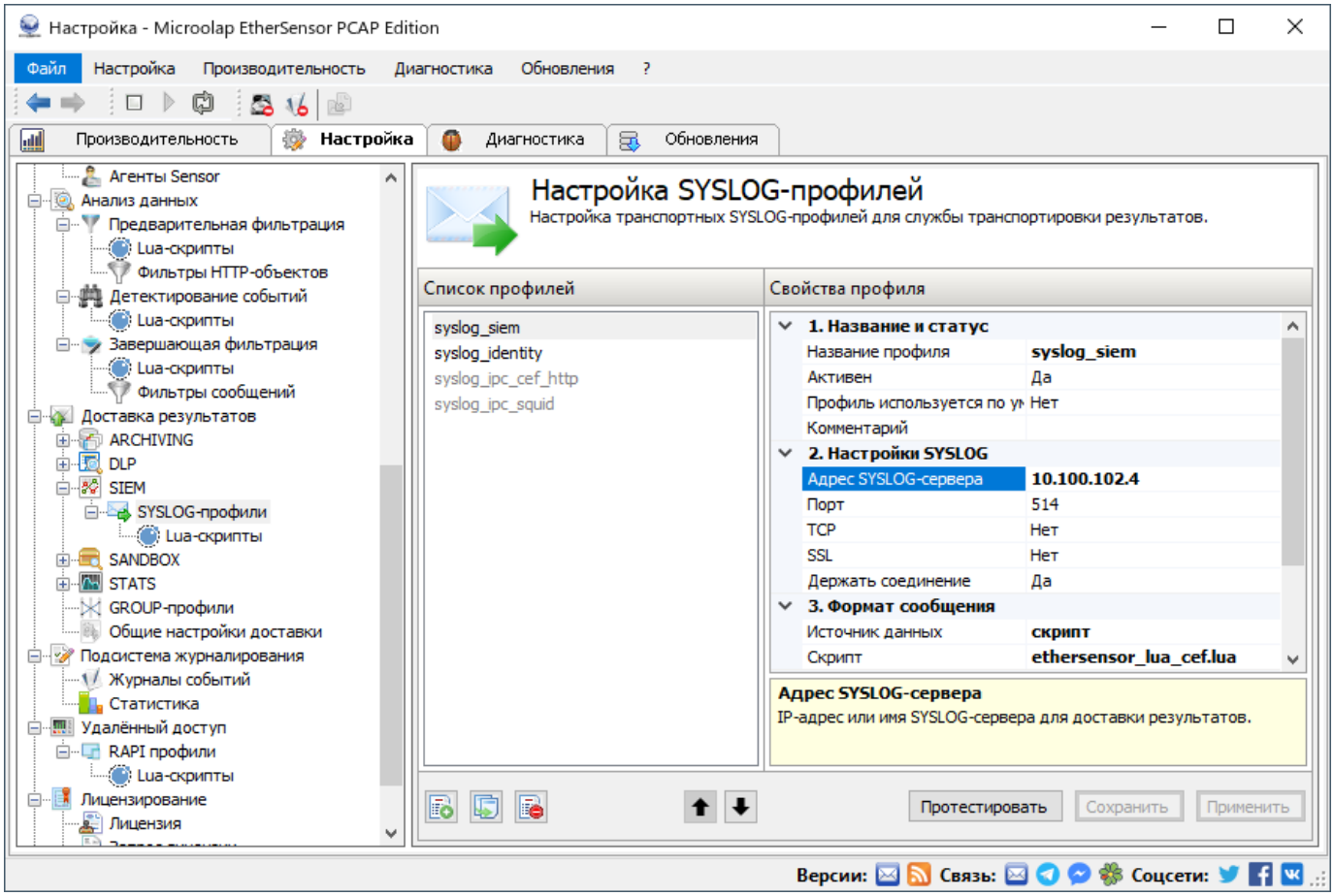

Рис.51. Настройки SYSLOG-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки SYSLOG

Адрес SYSLOG-сервера:

IP-адрес или имя SYSLOG-сервера для доставки результатов.

Порт:

Порт SYSLOG-сервера для отправки сообщений.

TCP:

Позволяет использовать TCP-протокол для отправки сообщений на SYSLOG-сервер. Необходимо, если используется SSL-шифрование при передаче сообщений системе-потребителю.

> Отправка на SYSLOG-сервер через SSL работает только при включенном TCP.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

3. Формат сообщения

Скрипт:

Имя файла Lua-скрипта, который будет использоваться для формирования сообщения SYSLOG-серверу. Файл скрипта должен находиться в подпапке \scripts.

4. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

5. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.3.1.1. Lua-скрипты

> Функция подготовки SYSLOG-сообщений с помощью назначаемых в профиле доставки Lua-скриптов находится в стадии пре-релиза.
>
> Если вы хотите поэкспериментировать вместе с нами, напишите нам на support@microolap.ru.

5.4. SANDBOX-профили

VIRUSTOTAL[238]
VIRUSTOTAL-профили служат для доставки перехваченных объектов на сервис VirusTotal для последующего анализа.

ATHENA[241]

ATHENA-профили служат для доставки перехваченных объектов на сервер ATHENA для проверки как локально установленными антивирусами, так и внешними аналитическими ресурсами: VirusTotal, Spamhaus и др.

5.4.1. VIRUSTOTAL-профили

VIRUSTOTAL-профили служат для доставки перехваченных объектов на сервис VirusTotal для последующего анализа.

> Согласно лицензионной политике VirusTotal вы не можете использовать публичный сервис VirusTotal в коммерческих целях. Пожалуйста, изучите условия VirusTotal прежде, чем получить ключ доступа: https://developers.virustotal.com/reference.

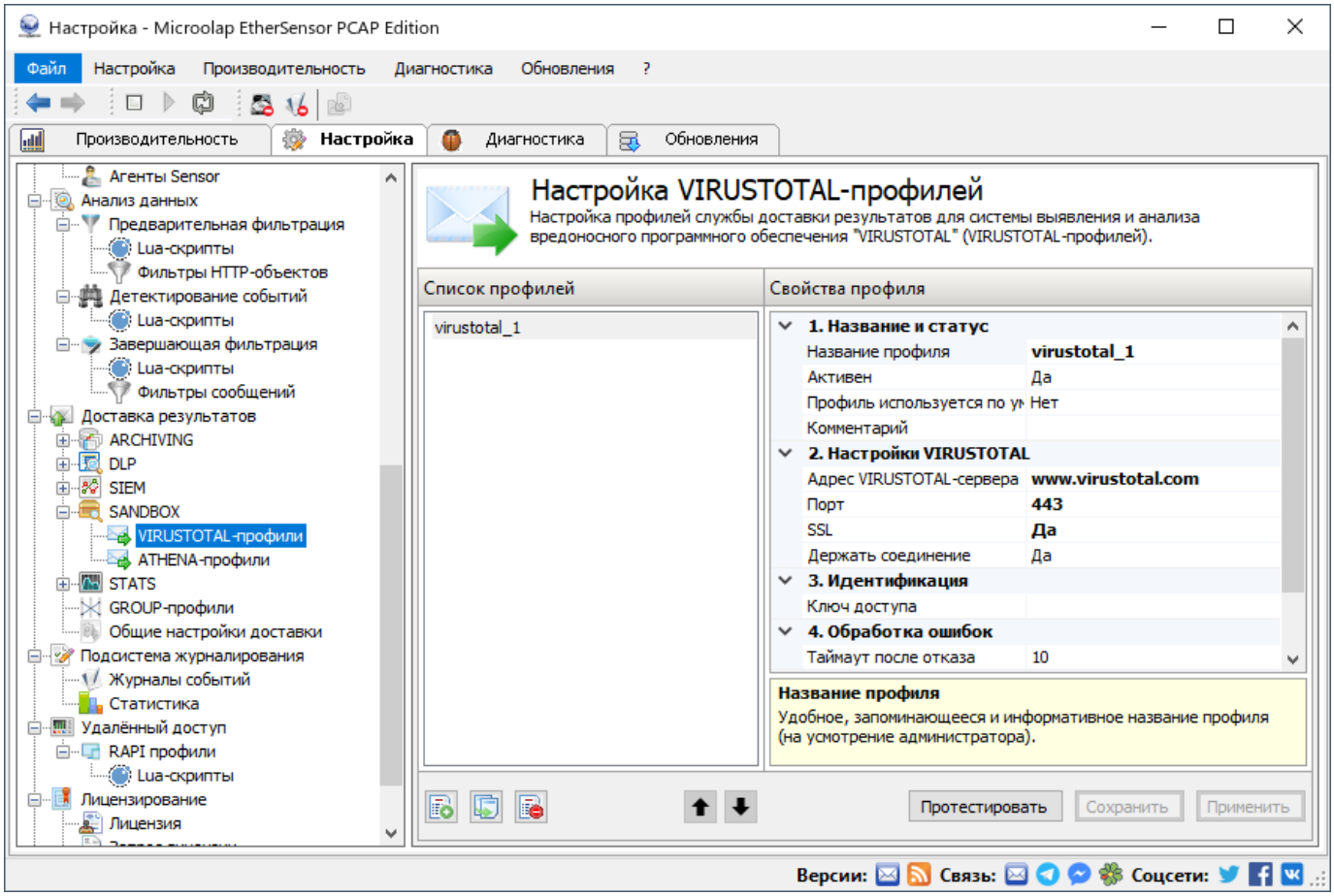

Рис.52. Настройки VIRUSTOTAL-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки VIRUSTOTAL

Адрес VIRUSTOTAL-сервера:

IP-адрес или имя сервера VIRUSTOTAL для доставки результатов.

Порт:

Порт сервера VIRUSTOTAL для отправки сообщений.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

3. Идентификация

Ключ доступа:

Ключ доступа к ресурсам VIRUSTOTAL сервера.

4. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

5. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.4.2. ATHENA-профили

ATHENA-профили служат для доставки в перехваченных объектов на сервер ATHENA для проверки как локально установленными антивирусами, так и внешними аналитическими ресурсами: VirusTotal, Spamhaus и др.

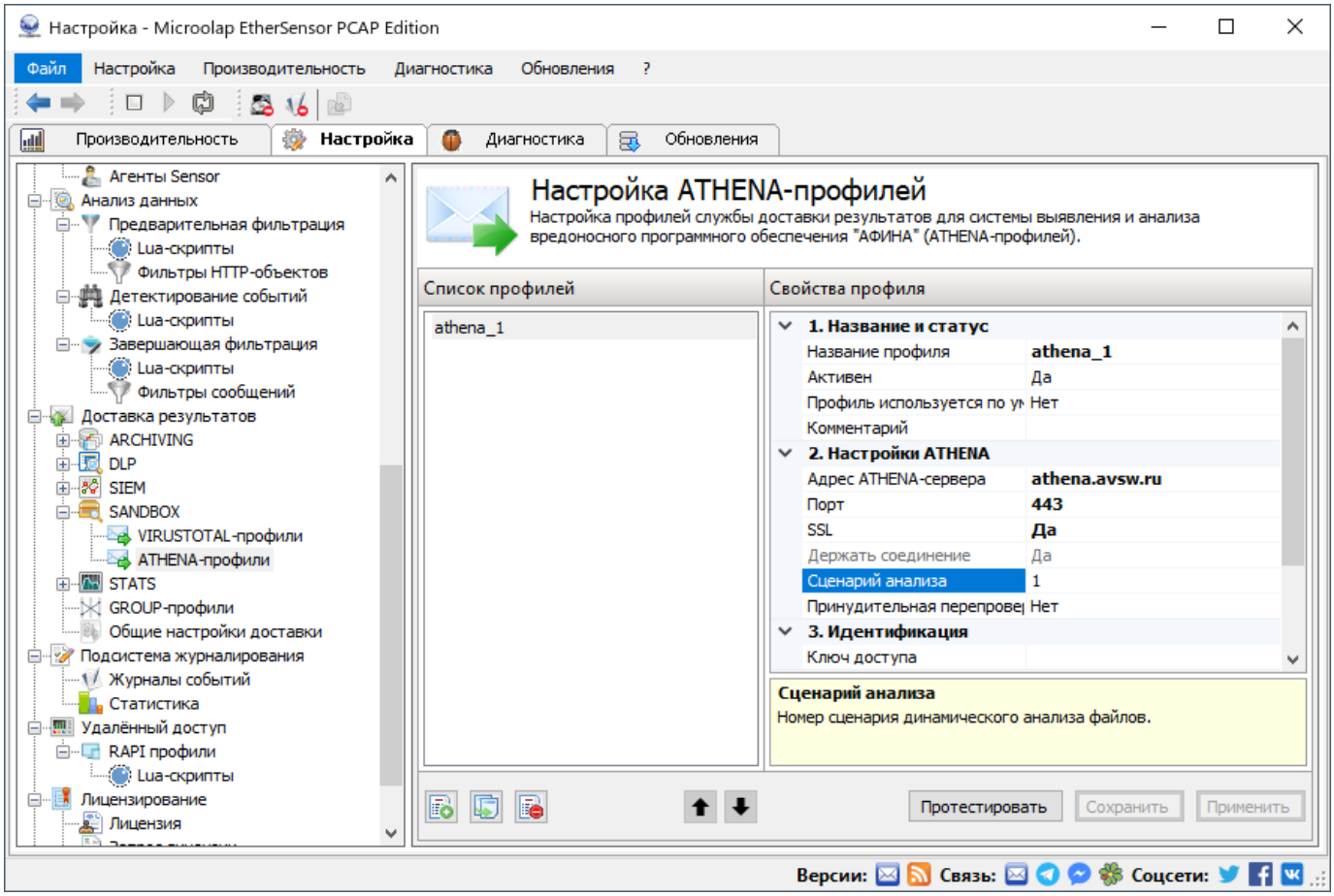

Рис.53. Настройки ATHENA-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки ATHENA

Адрес ATHENA-сервера:

IP-адрес или имя сервера ATHENA для доставки результатов.

Порт:

Порт сервера ATHENA для отправки сообщений.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

Сценарий анализа:

Номер сценария динамического анализа файлов.

Принудительная перепроверка:

Флаг запуска принудительной перепроверки.

3. Идентификация

Ключ доступа:

Ключ доступа к ресурсам ATHENA сервера.

4. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

5. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.5. STATS-профили

NETFLOW [243]

Профили группы STATS служат для доставки статистических данных на NetFlow-коллекторы, в частности, на NetFlow-коллекторы SIEM-систем.

5.5.1. NETFLOW-профили

NETFLOW-профили служат для доставки статистических данных на NetFlow-коллекторы, обычно на на NetFlow-коллекторы SIEM-систем.

NetFlow-коллекторы могут получать статистические данные и по срезам активных TCP-сессий от сетевого оборудования, и это - часто используемый подход. Но есть известная проблема: чем меньше временной интервал получения таких срезов, тем большие затраты ресурсов потребуются от сетевого оборудования – это не полезно для его основной функции. Кроме того, уменьшение интервала получения срезов повышает затраты ресурсов на стороне самого NetFlow-коллектора для сборки полученных данных.

EtherSensor для извлечения объекта из TCP-сессии в любом случае должен её реконструировать, и после этого у него есть все необходимые данные о самой TCP-сессии. В этот момент он вполне может передать данные о сессии на NetFlow-коллектор, назначенный в NETFLOW-профиле.

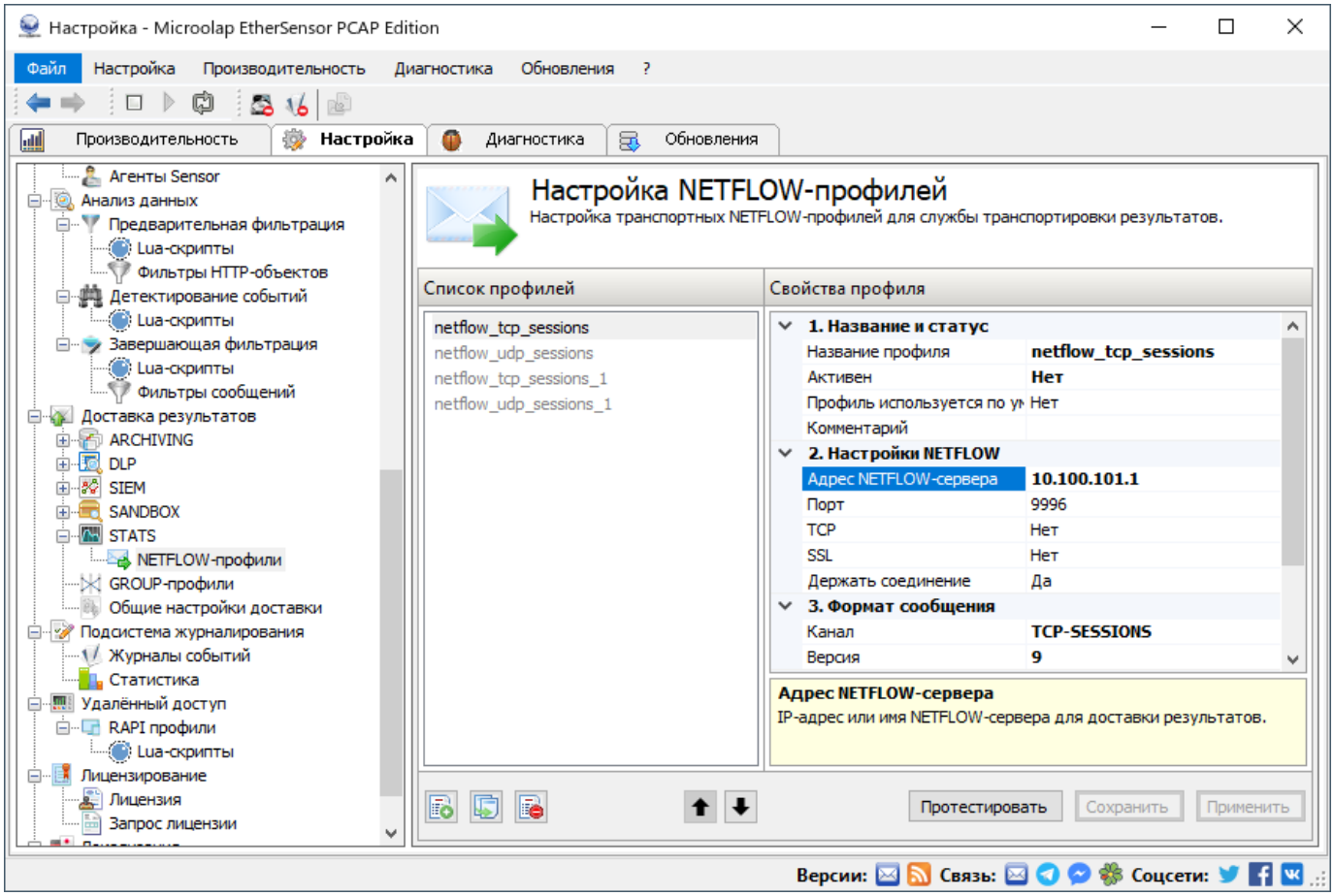

Рис.54. Настройки NETFLOW-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен:

Транспортный профиль не используется в передаче сообщений, если он отключен.

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Настройки NETFLOW

Адрес NETFLOW-сервера:

IP-адрес или имя NETFLOW-сервера для доставки результатов.

Порт:

Порт NETFLOW-сервера для отправки сообщений.

TCP:

Позволяет использовать TCP-протокол для отправки сообщений на NETFLOW-сервер. Необходимо, если используется SSL-шифрование при передаче сообщений системе-потребителю.

SSL:

Включить/выключить использование SSL-шифрования при передаче сообщений.

Держать соединение:

Отправлять все сообщения в одном соединении с сервером. Если данный параметр выключен, то каждое отправляемое сообщение будет отправлено в отдельном TCP-соединении.

3. Формат сообщения

Канал:

Имя IPC-канала, который будет использоваться как источник сообщений для NETFLOW-сервера.

Версия:

Версия протокола NETFLOW.

4. Обработка ошибок

Таймаут после отказа:

Таймаут в секундах для повторной попытки доставки сообщений в случае отказа приёмника.

5. Для GROUP-профилей

Вес:

Вес профиля (от 1 до 10) задаёт пропорцию распределения сообщений между профилями и используется только если данный профиль включен в GROUP-профиль.

Резервный профиль:

Включить или выключить использование профиля как резервного. Если этот флаг включён, то доставка сообщений с использованием данного профиля будет задействована в случае отказа основных (не резервных) транспортных профилей. Параметр действителен только в составе GROUP-профилей.

5.6. GROUP-профили

В случае больших входных потоков трафика EtherSensor может создать такие потоки перехваченных объектов для систем-потребителей, что они перестанут справляться с их получением. Как правило, в этом случае требуется установка нескольких систем-потребителей данного типа и распределение результатов работы EtherSensor между ними.

Групповые профили служат для балансировки нагрузки между системами-потребителями и включают в себя профили доставки с заранее установленными весами.

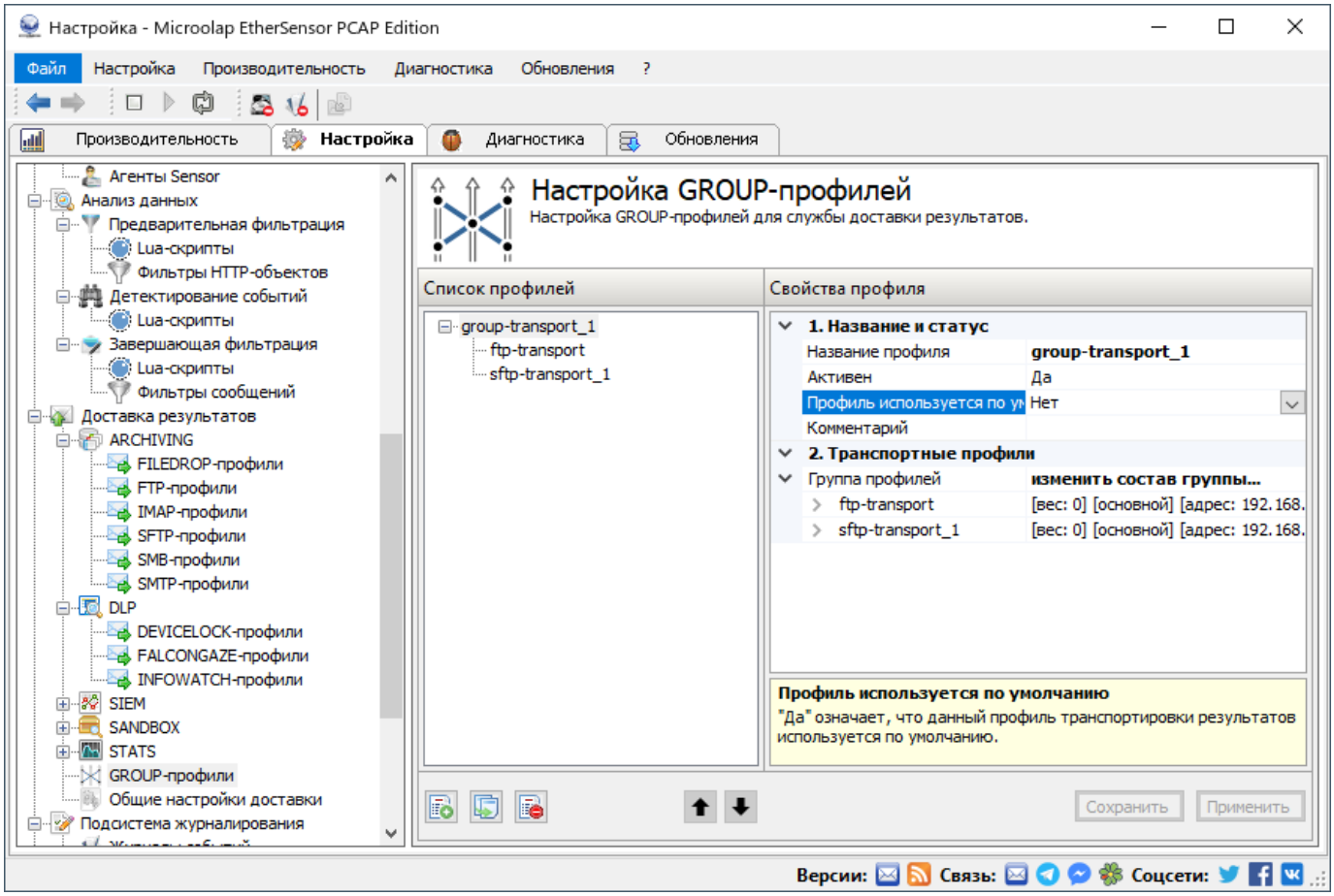

Рис.55. Настройки GROUP-профилей.

1. Название и статус

Название профиля:

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Профиль используется по умолчанию:

"Да" означает, что данный профиль транспортировки результатов используется по умолчанию.

Комментарий:

Описание транспортного профиля.

2. Транспортные профили

Группа профилей:

Список уже существующих профилей доставки, используемых в составе группового профиля.

Как работает групповой транспортный профиль

Наравне с обычным транспортным профилем групповой транспортный профиль может быть назначен как профиль по умолчанию. Также он может быть напрямую назначен для использования в правилах фильтрации сообщений.

В отличие от стандартного транспортного профиля, где содержатся тонкие настройки доставки результатов EtherSensor согласно используемому способу передачи их системе-потребителю, групповой профиль содержит только имена заранее созданных стандартных транспортных профилей.

В групповом профиле стандартные профили могут иметь статус основных и резервных. Основные профили используются для доставки сообщений в нормальном режиме работы. Резервные профили используются для доставки сообщений, когда ни один из основных профилей доставить сообщение не может.

Также в групповом профиле стандартным профилям могут быть назначены веса для доставки сообщений к системам-потребителям в необходимой пропорции. Веса могут назначаться как основным, так и резервным профилям.

Доставка результатов с использованием группового профиля

Рассмотрим пример доставки результатов с использованием группового профиля, в котором три стандартных транспортных профиля назначены как основные (SMTP 1, SMTP 2, SMTP 3) с равными весами, и один стандартный транспортный профиль (FILEDROP 1) назначен как резервный.

В нормальном режиме работы сообщения в равной пропорции доставляются к приёмникам сообщений.

Если один из приёмников сообщений, описанный стандартным транспортным профилем, не может принять сообщения, то нагрузка по приёму сообщений ложится на остальные основные транспортные профили.

Если все основные транспортные профили (SMTP 1, SMTP 2, SMTP 3) не могут принять сообщения, то для доставки сообщений будет использоваться резервный транспортный профиль (FILEDROP 1) до тех пор, пока хотя бы один основной транспортный профиль не станет способен доставлять сообщения.

5.7. Общие настройки доставки

Общие настройки доставки результатов работы EtherSensor системам-потребителям.

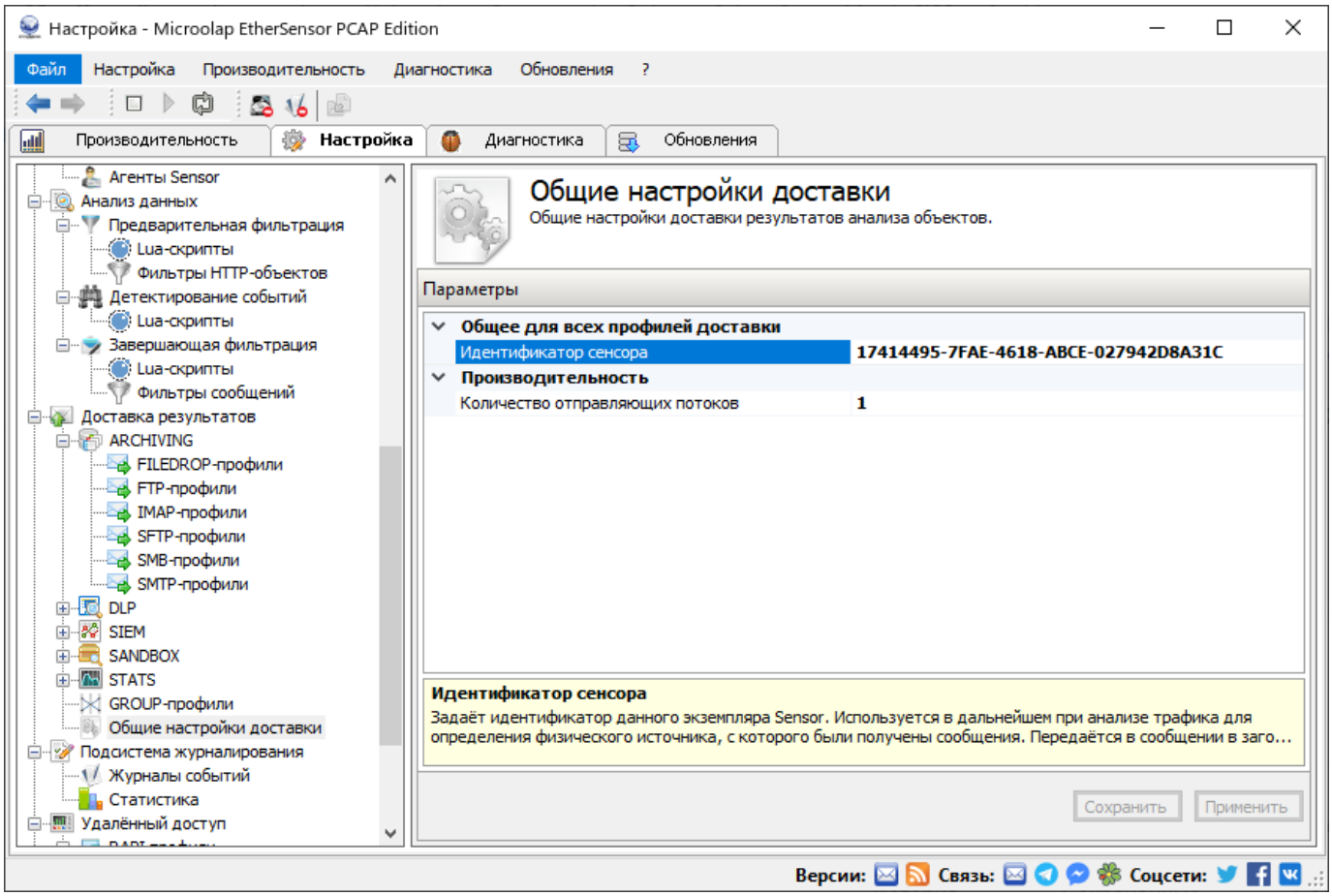

Рис.56. Общие настройки службы EtherSensor Transfer.

Общее для всех профилей доставки

Идентификатор сенсора:

Задаёт идентификатор данного экземпляра EtherSensor. Используется в дальнейшем при анализе трафика для определения физического источника, с которого были получены сообщения. Передаётся в сообщении в заголовке X-Sensor-Id.

Производительность

Количество отправляющих потоков:

Определяет количество отправляющих сообщения потоков. Максимальное количество отправляющих потоков не может быть больше чем текущее количество CPU * 2 и не может быть меньше 1.

6. Журналирование работы EtherSensor Watcher

EtherSensor начинает свою работу с запуска подсистемы логирования EtherSensor Watcher. Подсистема логирования предназначена для логирования сообщений служб EtherSensor, работы с логами, отслеживания текущего состояния EtherSensor, а также действующих лицензий.

Служба EtherSensor Watcher собирает системные сообщения от других служб EtherSensor и записывает их в зависимости от именованного канала логирования, уровня логирования и других критериев в назначенные администратором файлы или syslog-серверы.

Конфигурационный файл службы EtherSensor Watcher

Конфигурация службы EtherSensor Watcher содержится в XML-файле watcher.xml, расположенном в общей директории конфигураций EtherSensor [INSTALLDIR]\config.

Параметры командной строки

Служба EtherSensor Watcher в ходе процедуры установки Microolap EtherSensor устанавливается в качестве службы Windows, настроенной на автоматический запуск. Однако, при необходимости она может быть запущена как приложение Windows sensor_watcher.exe со следующими параметрами командной строки:

/process

Запустить процесс sensor_watcher.exe как обычный Windows Win32-процесс (возможно использовать для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

6.1. Настройка логирования EtherSensor Watcher

В окне, показанном ниже, можно создать или отредактировать существующие правила логирования событий EtherSensor.

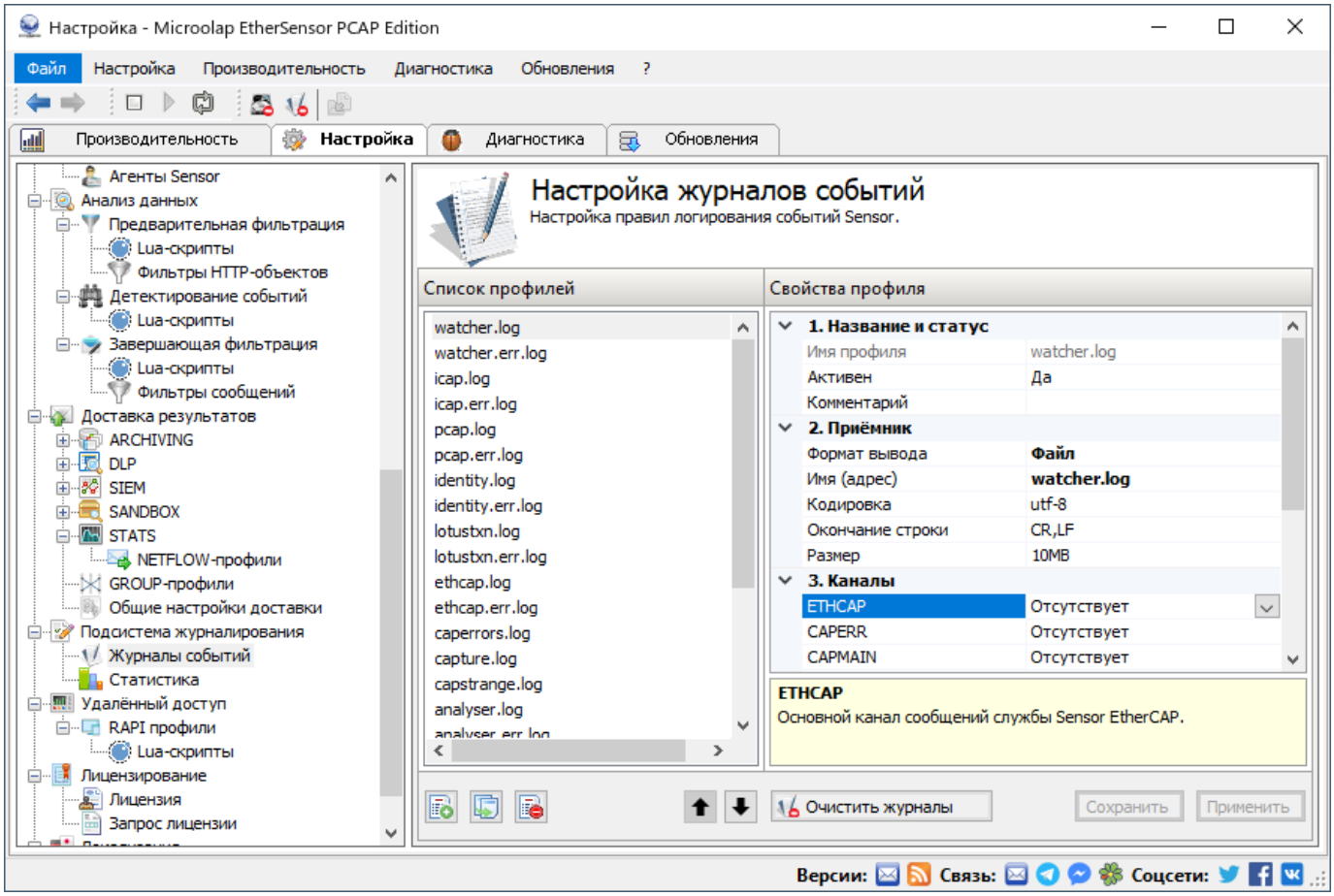

Рис.57. Настройки профилей логирования службы EtherSensor Watcher.

Именованные каналы:

Подсистема логирования "слушает" так называемые именованные каналы (ETHERCAP, ANALYSER ...), по которым другие службы EtherSensor передают сообщения различных уровней важности – info, warning, error и criterror. Названия таких именованных каналов в общем случае произвольны и могут быть заданы администратором по своему усмотрению в настройках соответствующих служб.

В соответствии с настройками полученные из именованных каналов сообщения либо игнорируются, либо пересылаются потребителям: это могут быть указанные в настройках SYSLOG-серверы или локальные/сетевые файлы.

В настройках также могут быть указаны предельные размеры файлов логов.

Настройка параметров профилей логирования:

В случае необходимости удаления имеющихся профилей или определения дополнительных профилей логирования используйте кнопки ***Новый, Клон*** и ***Удалить***:

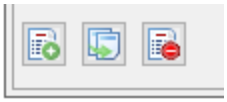

Рис.58. Кнопки создания, удаления и клонирования профилей логирования.

1. Название и статус

Имя профиля

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен

Профиль журнала событий не используется, если он отключен.

Комментарий

Описание профиля журнала событий.

2. Приёмник

Формат вывода

Выберите формат выводимых данных. file, win-file - вывод данных в формате utf-8, unix-file - вывод данных в формате ASCII, syslog - вывод данных в формате ASCII на удалённый SYSLOG-сервер.

Имя (адрес)

Имя файла или адрес удалённого SYSLOG-сервера. Например, файл: events.log, размещается в поддиректории log директрии установки EtherSensor, SYSLOG-сервер: 192.0.0.168:514.

Кодировка

Кодировка вывода данных.

Окончание строки

Варианты символов, завершающих сохраняемое сообщение в журнал событий.

Размер

Максимальный размер журнала событий. Варианты записи (1000B, 100KB, 50MB, 1GB).

3. Каналы

ETHCAP

Основной канал сообщений службы EtherSensor EtherCAP.

CAPERR

Канал логирования ошибок анализа трафика службы EtherSensor EtherCAP.

CAPMAIN

Канал логирования обрабатываемых соединений службы EtherSensor EtherCAP.

ICAP

Основной канал логирования службы EtherSensor ICAP.

ICAP-REQUEST

Канал логирования HTTP-запросов службы ICAP.

IDENTITY

Канал логирования службы Identity.

PCAP

Канал логирования службы PCAP.

LOTUSTXN

Основной канал логирования службы EtherSensor LotusTXN.

ANALYSER

Основной канал логирования службы EtherSensor Analyser.

FILTER

Канал логирования системы фильтрации данных службы EtherSensor Analyser.

TRANSFER

Основной канал логирования службы доставки результатов EtherSensor Transfer.

WATCHER

Основной канал логирования службы EtherSensor Watcher.

SQUID-ACCESS

Канал логирования обработанных HTTP-запросов в формате SQUID-ACCESS-LOG.

CEF-HTTP

Канал логирования обработанных HTTP-запросов в формате CEF.

6.2. Настройка статистики EtherSensor Watcher

В процессе детектирования сообщений EtherSensor позволяет накапливать различную статистику соединений: MAC-адрес интерфейса, на котором было перехвачено соединение, время создания, время завершения соединения, IP-адреса и порты соединения, объём переданных данных от клиента к серверу и обратно, используемый протокол уровня приложений (HTTP, ICQ, SMTP, POP3...) и т.д.

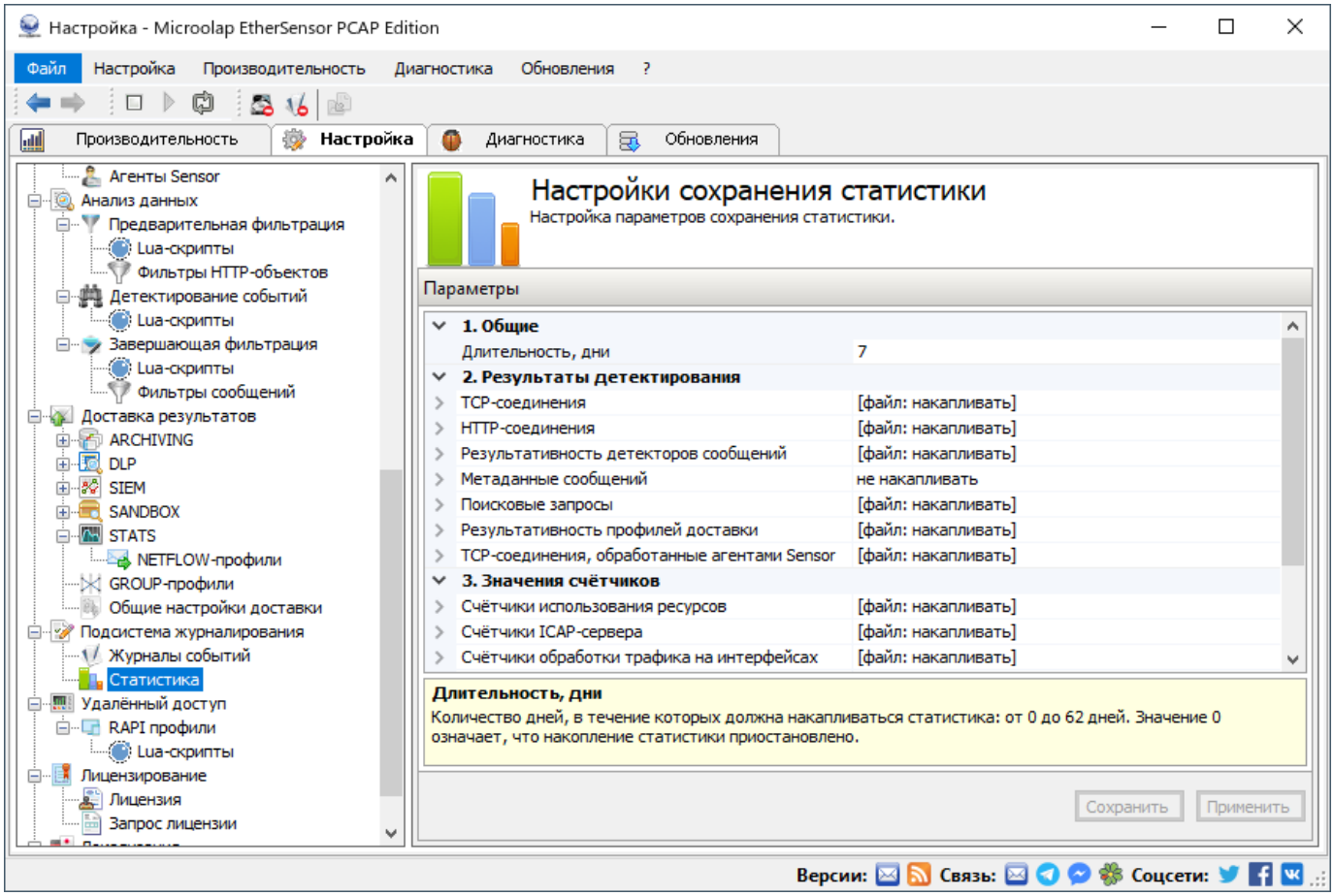

Рис.59. Настройка параметров сохранения статистики.

Полученная статистика накапливается либо в предназначенных для этого локальных директориях, либо может быть отправлена на syslog-сервер.

1. Общие

Длительность, дни

Количество дней, в течение которых должна накапливаться статистика: от 0 до 62 дней. Значение 0 означает, что накопление статистики приостановлено.

2. Результаты детектирования

TCP-соединения

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\sessions и включает в себя следущие параметры отслеживаемых соединений:"- MAC-адрес интерфейса, на котором было обнаружено соединение;"- Время создания, время завершения соединения;"- IP-адреса и порты соединения;"- Количество данных, переданных от клиента к серверу и обратно;"-

Протокол, используемый поверх TCP/IP (HTTP, ICQ, SMTP, POP3...);"- Другие параметры (см. документацию).

Накапливать в файл

Включает/отключает накопление статистики EtherSensor.

Использовать SYSLOG-сервер

Включить/выключить использование SYSLOG-сервера для журналирования событий работы EtherSensor.

Адрес SYSLOG-сервера

Адрес и порт SYSLOG-сервера. Пример: "192.168.0.1:514".

Имя источника

Используется для разбора сообщений на стороне SYSLOG-сервера (только латинские буквы и цифры). Пример: sensor_syslog_sessions

HTTP-соединения

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\hosts и включает в себя следущие параметры отслеживаемых соединений:"- MAC-адрес интерфейса, на котором было обнаружено соединение;"- IP-адреса и порты соединения;"- DNS-имя сервера, к которому было выполнено соединение.

Накапливать в файл

Включает/отключает накопление статистики EtherSensor.

Использовать SYSLOG-сервер

Включить/выключить использование SYSLOG-сервера для журналирования событий работы EtherSensor.

Адрес SYSLOG-сервера

Адрес и порт SYSLOG-сервера. Пример: "192.168.0.1:514".

Имя источника

Используется для разбора сообщений на стороне SYSLOG-сервера (только латинские буквы и цифры). Пример: sensor_syslog_sessions

Далее однотипные настройки получателей статистики опущены.

Результативность детекторов сообщений

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\detectors и включает в себя следущие параметры:"- Временная метка события;"- Имя детектора;"- Статус детектирования;"- Количество детектированных сообщений.

Метаданные сообщений

Накапливается в формате XML в директориях data\statistics\YYYY-MM-DD\messages и включает в себя метаданные реконструированных сообщений:"- Служебные заголовки протокола, по которому передано сообщение;"- Метаданные, сформированные EtherSensor при обработке сообщений (заголовки X-Sensor-...);"- Заголовки From, To, Cc, Bcc, Subject.

Поисковые запросы

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\squeries и включает в себя следущие параметры:"- Временная метка события;"- IP-адрес и порт отправителя поискового запроса;"- DNS-имя поискового сервиса;"- Фраза поискового запроса.

Результативность профилей доставки

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\transports и включает в себя следущие параметры:"- Временная метка события;"- Имя транспортного профиля;"- Используемый протокол для отправки сообщения;"- Статус отправки сообщения.

TCP-соединения, обработанные агентами EtherSensor

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\agents и включает в себя следущие параметры отслеживаемых соединений:"- MAC-адрес интерфейса, на котором было обнаружено соединение;"- Время создания;"- IP-адреса и порты соединения;"- Имя процесса, создавшего соединение;"- Имя пользователя, с правами которого был запущен процесс, создавший соединение;"- Другие параметры (см. документацию).

3. Значения счётчиков

Счётчики использования ресурсов

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\performance и включает в себя следущие параметры:" - Временная метка события;" - Использование CPU (текущее, среднее, пиковое);" - Использование памяти (текущее, среднее, пиковое);" - Использование системных потоков;" - Использование дескрипторов системных объектов (файлы, события и т.д.);" - Общие показания загрузки ОС (CPU, RAM).

Счётчики ICAP-сервера

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\icap и включает в себя показания счётчиков ICAP-службы:"- Временная метка события;"- Количество соединений с ICAP-сервером;"- Количество обрабатываемых запросов (GET, POST, PUT).

Счётчики обработки трафика на интерфейсах

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\interfaces и включает в себя следущие параметры:"- Временная метка события;"- Счётчики обработанных пакетов;"- Счётчики обрабатываемых TCP-соединений.

Счётчики детектирования соединений по протоколам

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\parsers и включает в себя следущие параметры:"- Временная метка события;"- Счётчики обрабатываемых TCP-соединений по протоколам (SMTP, POP3, HTTP, FTP...).

Счётчики кэша службы анализа

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\cache и включает в себя следущие параметры:"- Временная метка события;"- Счётчики обрабатываемых объектов кэша службы анализа.

Счётчики детектирования сообщений

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\analysers и включает в себя следущие параметры:"- Временная метка события;"- Счётчики детектирования сообщений службой анализа.

Счётчики фильтров

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\filter и включает в себя следущие параметры:"- Временная метка события;"- Счётчики фильтров службы анализа (RAW-фильтр HTTP-запросов, фильтр сообщений).

Счётчики дисковых квот

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\quotas и включает в себя следущие параметры:"- Временная метка события;"- Счётчики использования дисковых квот.

Счётчики доставки сообщений

Накапливается в формате CSV в директориях data\statistics\YYYY-MM-DD\quotas и включает в себя следущие параметры:"- Временная метка события;"- Счётчики использования дисковых квот.

7. Удалённое управление и мониторинг EtherSensor

Служба EtherSensor RAPI отвечает за удалённый мониторинг и управление сервером EtherSensor. Для этого она использует предварительно настроенные профили [258].

Профили службы EtherSensor RAPI обеспечивают работу независимых и изолированных друг от друга http2 web-сервисов.

Каждый профиль связан с определённой учётной записью пользователя локальной системы, на которой установлен EtherSensor и имеет в точности тот объём прав, который имеет данный пользователь.

Точкой входа для каждого профиля EtherSensor RAPI является Lua-скрипт, в который передаётся HTTP запрос, а результатом работы профиля является HTTP ответ, сформированный в соответствии с замыслом разработчика скрипта.

Функциональные возможности Lua-скрипта конкретного профиля EtherSensor RAPI ограничиваются только лишь возможностями текущей реализации языка Lua.

Также Lua-скрипт любого профиля EtherSensor RAPI имеет возможность использовать весь API, реализованный для этапа детектирования и анализа событий [89].

Конфигурационный файл службы EtherSensor RAPI

Конфигурация службы EtherSensor RAPI содержится в XML-файле rapi.xml, расположенном в общей директории конфигураций Microolap EtherSensor [INSTALLDIR]\config.

Параметры командной строки

Служба EtherSensor RAPI в ходе процедуры установки Microolap EtherSensor устанавливается в качестве службы Windows, настроенной на автоматический запуск. Однако, при необходимости она может быть запущена как приложение Windows sensor_rapi.exe со следующими параметрами командной строки:

/process

Запустить процесс sensor_rapi.exe как Windows Win32-процесс (возможно использовать для отладки).

/service

Запустить как службу Windows.

/config

Сохранить конфигурацию службы по умолчанию.

7.1. Настройка профилей EtherSensor RAPI

Каждый профиль EtherSensor RAPI обеспечивает работу независимого http2 web-сервиса. Количество профилей EtherSensor RAPI может быть сколь угодно большим.

Все http2 web-сервисы, определяемые профилями, изолированы друг от друга и выполняют исключительно те функции, которые определяют связанные с ними Lua-скрипты. Таким образом

администратор может организовать удалённый доступ к серверу EtherSensor на основе любой собственной ролевой модели.

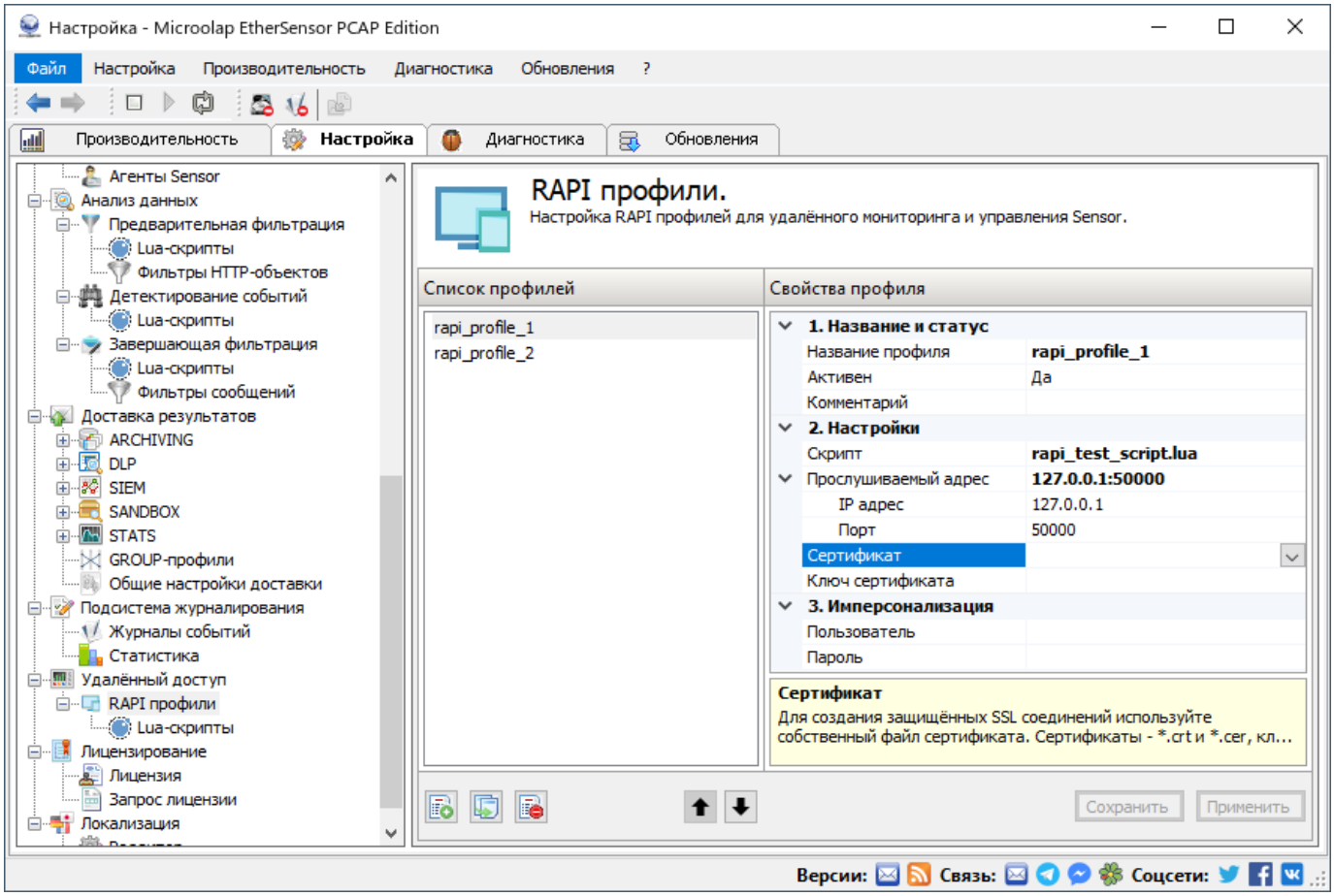

Рис.60. Настройки профилей службы EtherSensor RAPI.

В случае необходимости удаления или создания профилей EtherSensor RAPI используйте кнопки ***Новый, Клон*** и ***Удалить***:

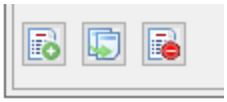

Рис.61. Кнопки создания, удаления и клонирования профилей EtherSensor RAPI.

Настройка параметров профиля EtherSensor RAPI:

1. Название и статус

Название профиля

Удобное, запоминающееся и информативное название профиля (на усмотрение администратора).

Активен

Профиль RAPI не используется, если он отключен.

Комментарий

Описание профиля RAPI.

2. Настройки

Скрипт

Привязанный к профилю скрипт, который является точкой входа для обработки HTTP запросов на прослушиваемом адресе.

Прослушиваемый адрес

Настройка локального адреса для обработки HTTP2 соединений.

IP адрес

Настройка локального IP адреса для прослушивания HTTP2 соединений.

Порт

Настройка локального порта для прослушивания HTTP2 соединений.

Сертификат

Для создания защищённых SSL соединений используйте собственный файл сертификата. Сертификаты - *.crt и *.cer, ключи - - *.key и .pem.

Ключ сертификата

Для создания защищённых SSL соединений используйте собственный файл ключа сертификата. Сертификаты - *.crt или *.cer, ключи - - *.key или .pem.

3. Имперсонализация

Пользователь

Имя локального пользователя ОС для имперсонализации потока обработки HTTP запросов.

Пароль

Пароль локального пользователя ОС для осуществления имперсонализации потока обработки HTTP запросов.

Редактировать Lua-скрипты профиля можно с помощью любого текстового редактора (скрипты находятся в каталоге инсталляции [INSTALLDIR]\scripts\rapi) или же прямо в окне консоли управления ***Lua-скрипты***.

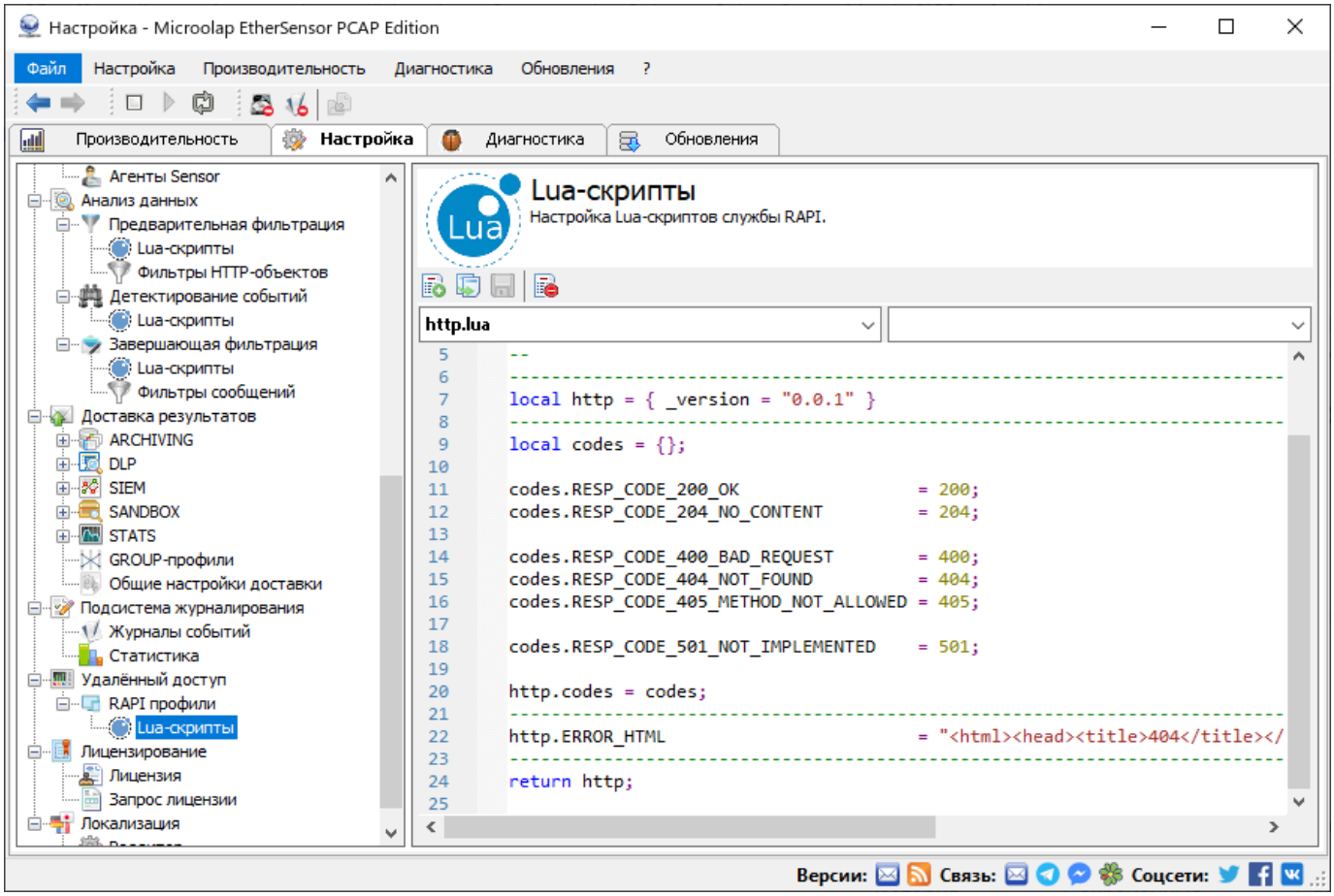

Рис.62. Редактирование скриптов EtherSensor RAPI.

8. Служба обновления Microolap EtherSensor

Служба обновления EtherSensor Updater (процесс sensor_updater.exe) предназначена для автоматизированной загрузки и установки файлов обновлений и патчей. EtherSensor Updater является частью сервиса разработчика по поддержке работы Microolap EtherSensor в штатном режиме и обновлению его функций.

Функции EtherSensor Updater:

- Проверка наличия обновлений и патчей для Microolap EtherSensor, их загрузка и установка.
- Оперативное обновление файла лицензии Microolap EtherSensor при его изменении.
- Создание архивной копии предыдущей версии продукта и помещение в неё всех необходимых файлов.
- Восстановление предыдущей версии из архива при возникновении ошибки установки обновления.

Вы можете указать интервал проверки обновлений или использовать настройки по умолчанию. EtherSensor Updater также позволяет гибко настраивать сетевое подключение, поддерживает работу с прокси сервером.

Как работает EtherSensor Updater:

Служба обновлений проверяет как наличие новых версий EtherSensor, так и наличие обновлённых версий лицензий. Для этого она через заданные в настройках интервалы времени устанавливает соединение с сервером обновлений и передаёт ему информацию о версии и лицензии Microolap EtherSensor.

Если для установленного программного обеспечения доступны более новые версии, сервер отправляет службе ответный файл-манифест, в котором указаны файлы, которые необходимо загрузить и установить.

EtherSensor Updater загружает с сервера обновлений эти файлы и помещает их в очередь на установку. Во время, указанное в конфигурации службы, эти файлы запускаются, и происходит обновление программного обеспечения.

Установка EtherSensor Updater:

Первоначальная установка EtherSensor Updater выполняется запуском файла ethersensor_updater_X.X.X.XXXX_x64.msi из состава дистрибутива. В дальнейшем служба будет обновляться самостоятельно при выходе новых версий.

Для установки в директорию, отличную от директории по умолчанию следует применить утилиту msiexec.exe ОС Windows.

Пример:

```
msiexec.exe /i ethersensor_updater_4.6.3.12232_x64.msi INSTALLDIR="[[INSTALLDIR]]\updater"
```

Технические требования:

- Служба EtherSensor Updater должна иметь доступ в Интернет через порт 80 или 443 к серверам серверами license.microolap.com и kpps-downloads.microolap.com
- Если используется прокси-сервер, он должен поддерживать корректную работу SOAP-протокола.
- Операционная система Windows Server 2012, Windows Server 2016 или Windows Server 2019.

EtherSensor Updater состоит из следующих частей:

Служба EtherSensor Updater (процесс sensor_updater.exe)

Запускается в фоновом режиме и выполняет все действия, необходимые для проверки наличия доступных обновлений и их дальнейшей установки. В процессе работы создаётся статус-файл status.xml, в котором содержится информация о текущем состоянии службы.

Раздел "Обновления" консоли управления EtherSensor (приложение sensor_console.exe)

Служит для настройки конфигурации и отображения статуса службы, в отдельной закладке отображается основной файл журнала обновлений.

> Примечание:
>
> 1. Консоль управления EtherSensor соединяется со службой sensor_updater.exe по TCP/IP-протоколу, порт 52076.
>
> 2. Если консоль управления EtherSensor обнаруживает, что служба sensor_updater.exe не запущена, попытка установить соединение не происходит. Если необходимо установить соединение со службой, работающей в консольном режиме, можно воспользоваться параметром командной строки /ignore-service-status
>
> 3. Также можно использовать параметр командной строки /no-connect, который запрещает любые попытки связаться со службой. В этом случае консоли управления EtherSensor получает информацию о состоянии службы только через файл status.xml.

Директория [INSTALLDIR]\config

Содержит конфигурационные файлы службы (настройка службы).

Директория [INSTALLDIR]\downloads

При настройке по умолчанию в эту директорию выполняется загрузка файлов, необходимых для обновления EtherSensor.

Директория [INSTALLDIR]\log

Содержит файлы журналов и статус-файл:

status.xml

Статус-файл, содержащий информацию о текущем состоянии службы.

log.txt

Основной журнал работы службы.

uupdater_log.txt

Журналы специальной программы sensor_uupdater.exe, которая вызывается при необходимости установить новую версию EtherSensor Updater.

Файл конфигурации EtherSensor Updater

Конфигурация EtherSensor Updater хранится в файле system.xml, расположенном в директории [INSTALLDIR]\config.

8.1. Настройка службы обновления Microolap EtherSensor

Текущее состояние
В разделе Текущее состояние отображается информация из статус-файла EtherSensor Updater, касающаяся работы самой службы обновления.

Также вэтом разделе отображается список обнаруженного на сенсоре программного обеспечения EtherSensor, для которого возможна проверка обновлений, список загружаемых файлов (если что-то загружается прямо сейчас) и список готовых к установке обновлений.

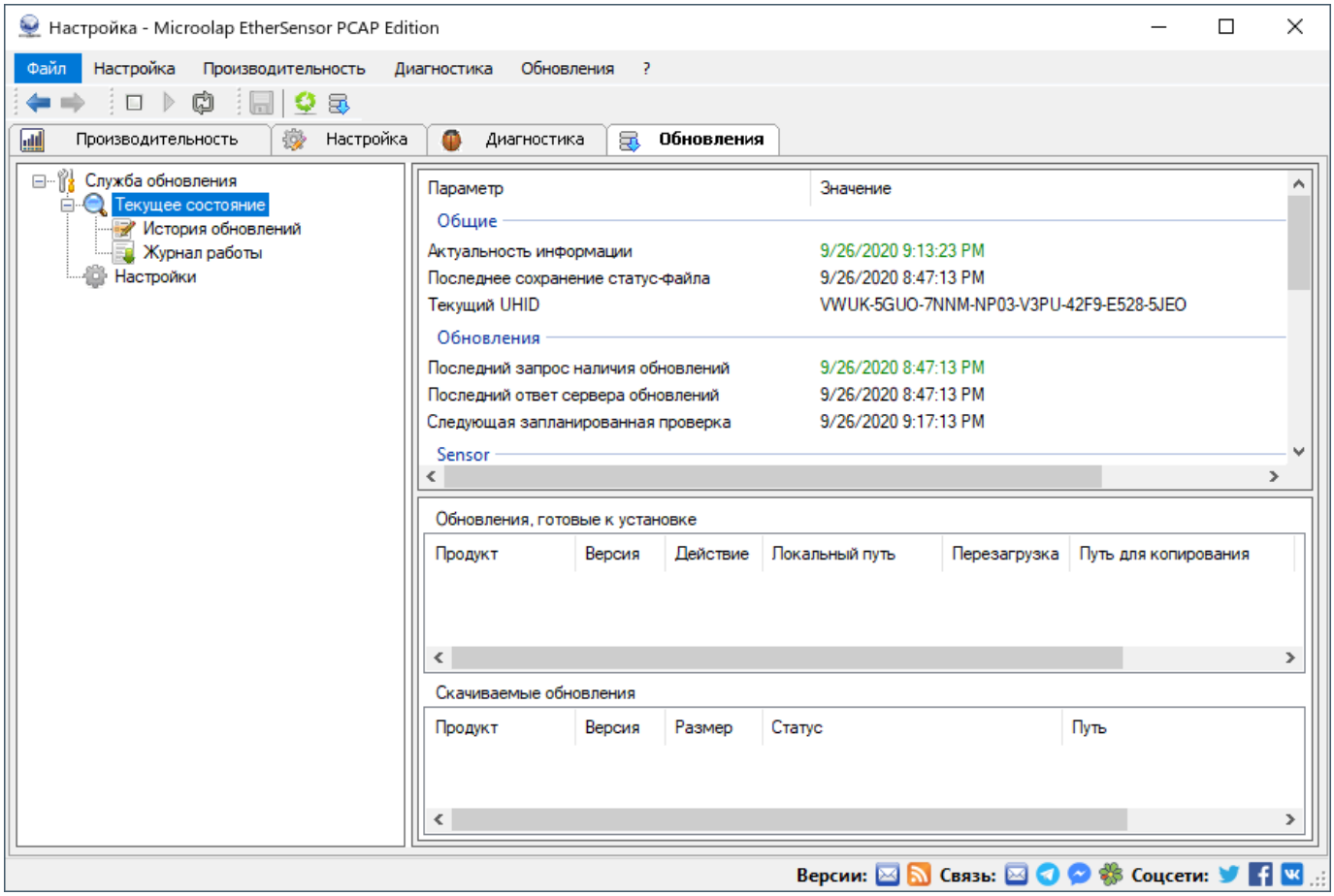

Рис.63. Раздел "Текущее состояние"

Настройки, вкладка Загрузка обновлений

В разделе Загрузка обновлений устанавливаются режимы работы EtherSensor Updater и частота проверок наличия обновлений. Также в этом разделе можно включить тестовый режим, в котором в файлы журналов добавляются более подробные сообщения о работе службы.

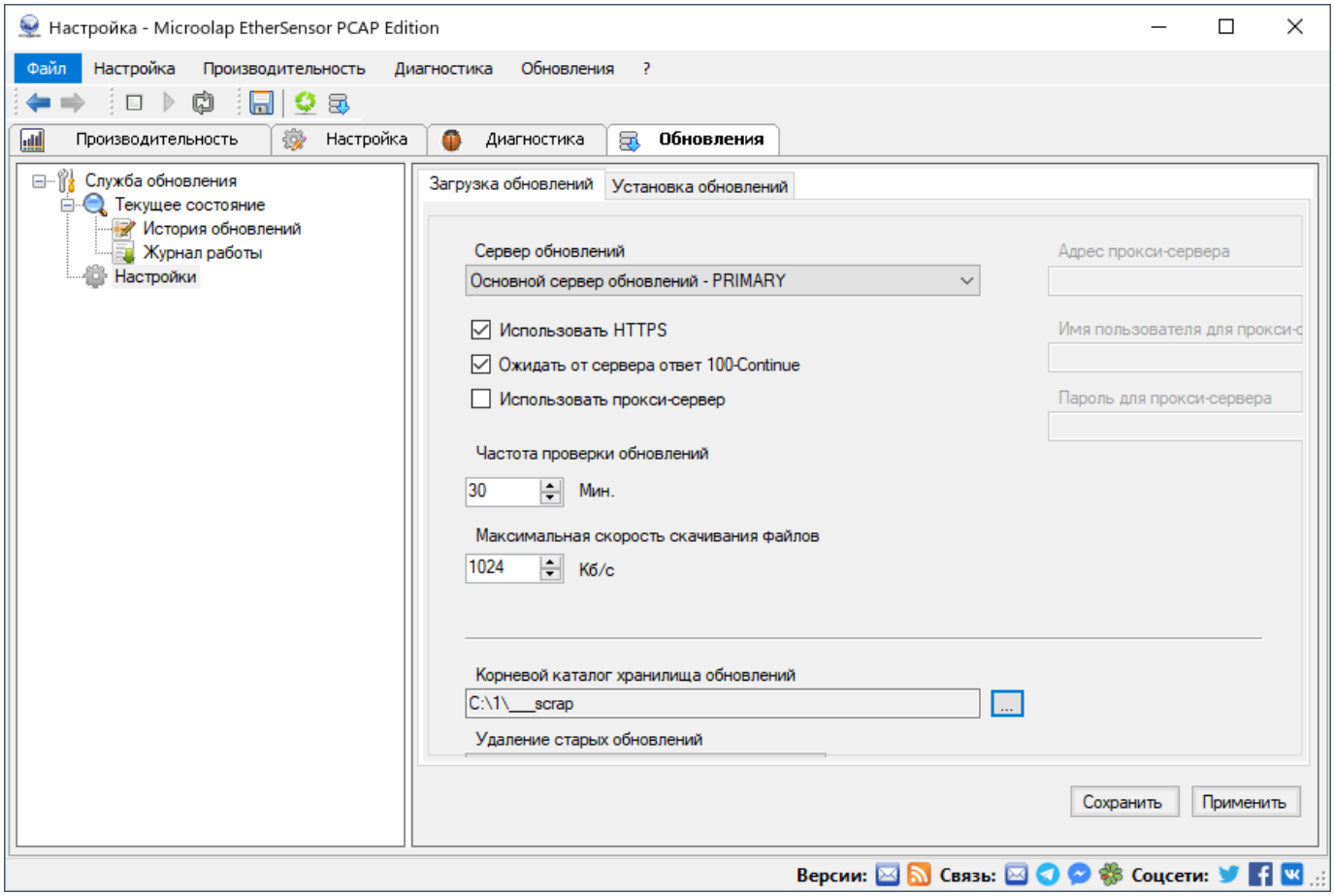

Рис.64. Раздел "Загрузка обновлений"

Настройки, вкладка Установка обновлений

В разделе Установка обновлений настраивается время автоматической установки обновлений. После установки обновлений происходит перезагрузка ОС, если таковая потребовалась.

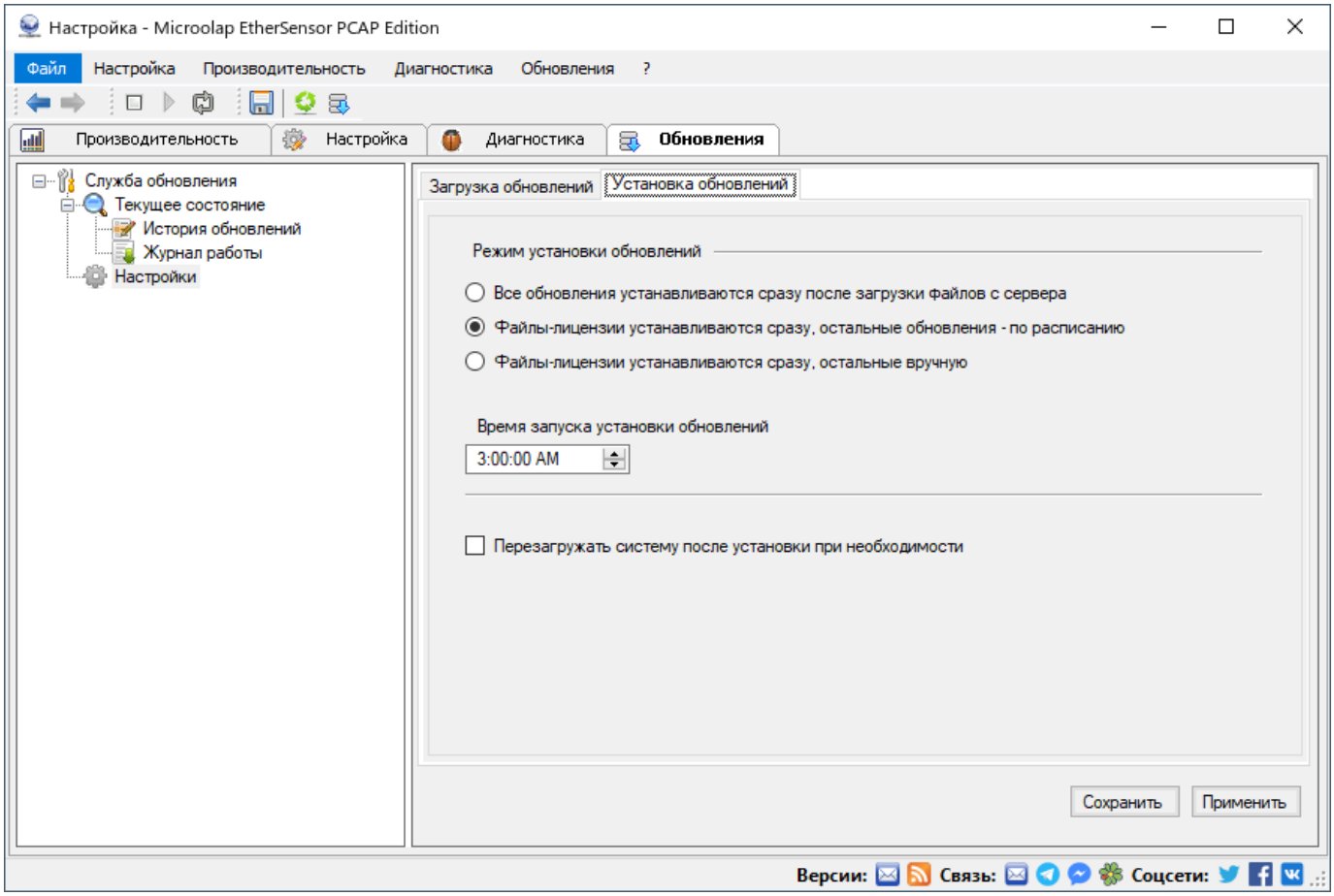

Рис.65. Раздел "Установка обновлений"

Журнал работы

В разделе Журнал работы отображаются последние записи файлов журналов EtherSensor Updater.

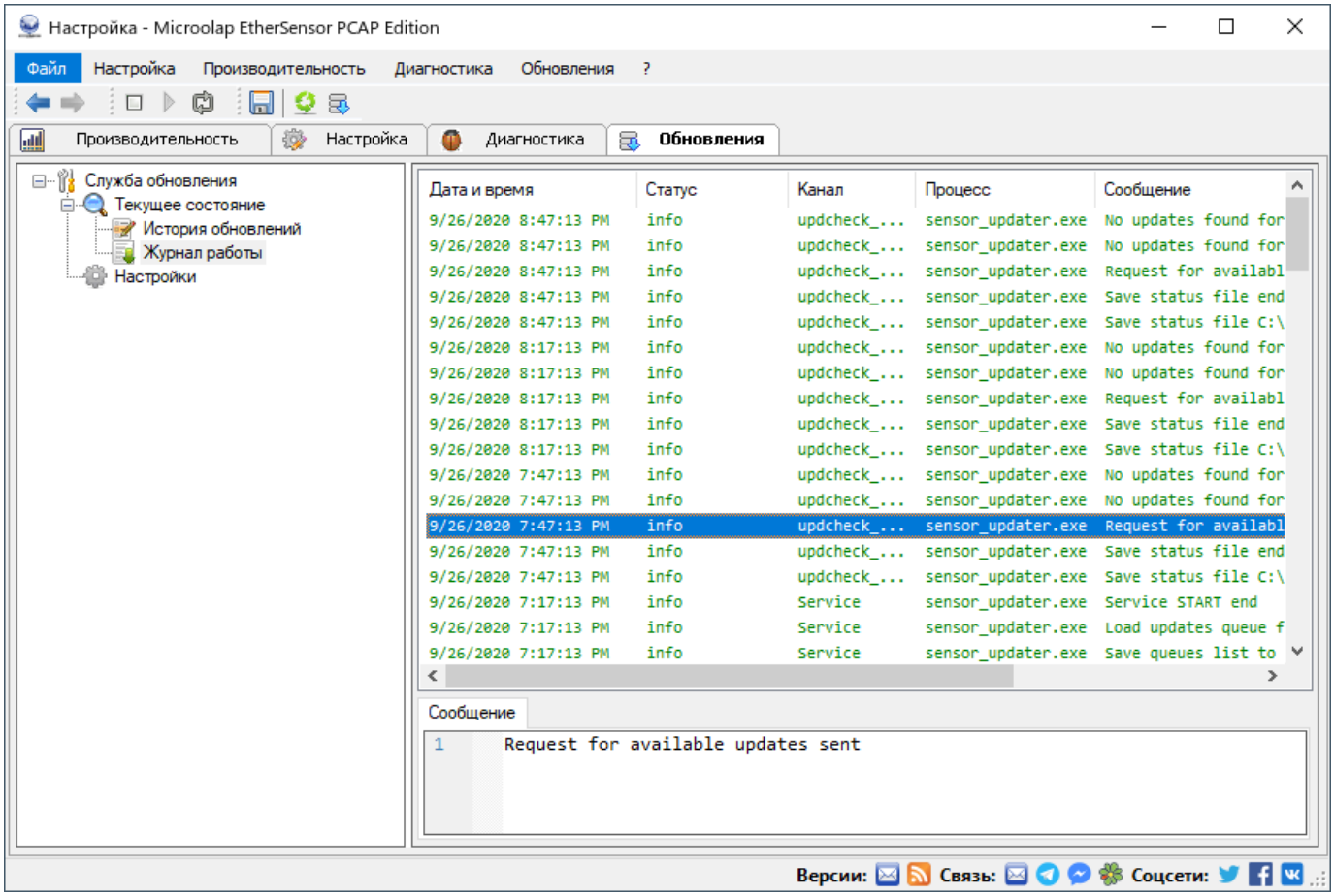

Рис.66. Раздел "Журнал работы"

История обновлений

В разделе История обновлений отображается список установленных ранее обновлений.

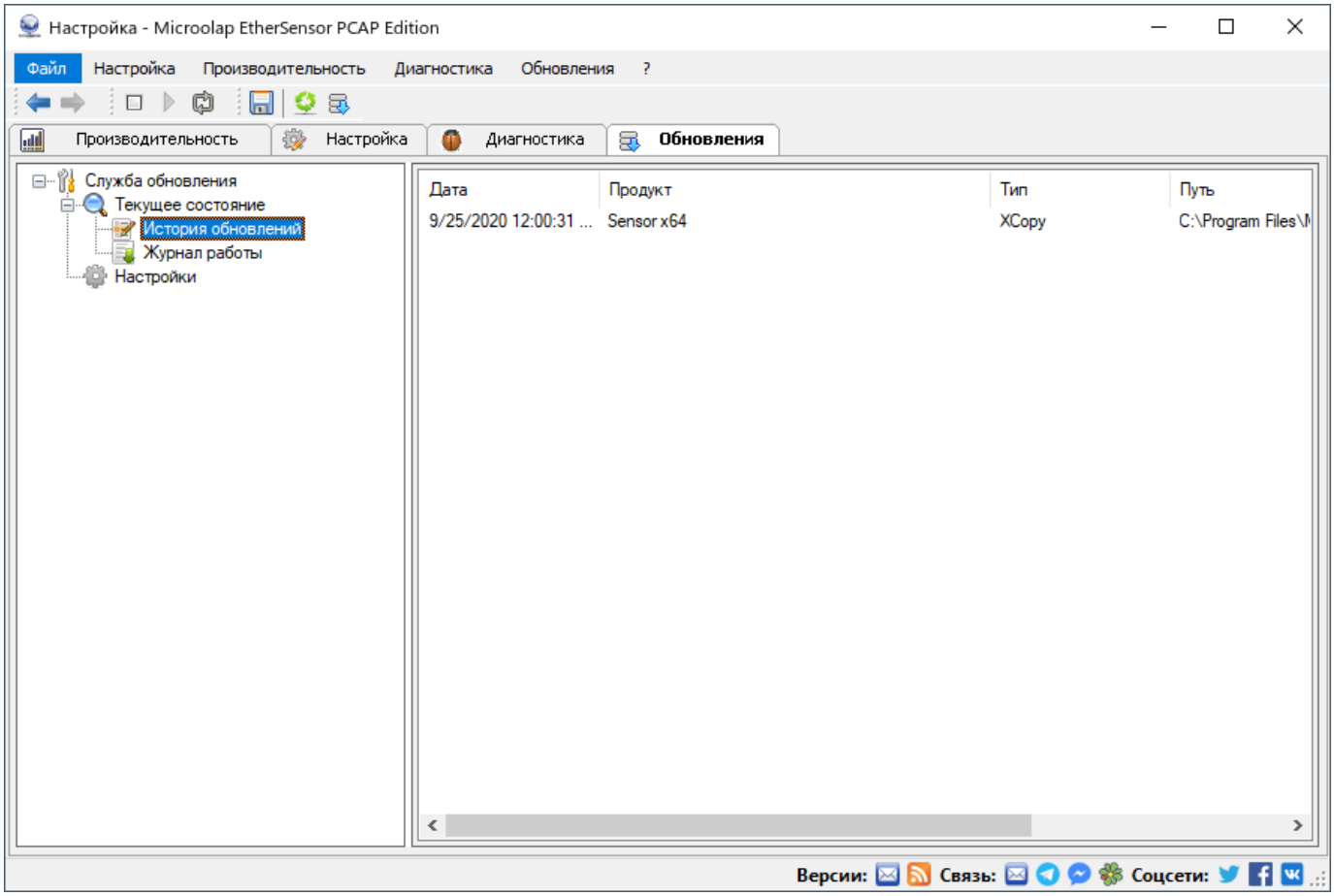

Рис.67. Раздел "История обновлений"

9. Регламентное обслуживание сенсора

В целях поддержания оптимального рабочего состояния сенсора и обнаружения проблем на самых ранних этапах, рекомендуется регулярно выполнять следующие операции.

1. Проверка журналов событий сенсора на предмет наличия предупреждений и ошибок служб.

2. Выполнение резервного копирования. Необходимо убедиться в том, что конфигурационный файл сенсора регулярно резервируется на внешний носитель информации.

3. Проверка индикаторов производительности сервера, на котором функционирует EtherSensor, таких как свободное место на диске, утилизация процессора, использование оперативной памяти. Для этой цели может быть использовано средство Windows Performance Monitor и приложение sensor_console.exe, установленное в директории установки EtherSensor. Использование механизма уведомлений позволяет уведомлять сотрудников группы

поддержки EtherSensor о внезапных изменениях или проблемах с производительностью сервера.

4. Проверка журнальных файлов на сетевом устройстве. Следует проводить анализ журнальных файлов на предмет корректности работы транспортной службы, а также оценить объемы ответвляемого трафика с целью недопущения перегрузки его потребителей.

5. Проверка доступности потребителей перехваченных сообщений со стороны сенсора. Следует убедиться, что сообщения, формируемые службами сенсора, доставляются на серверы потребителей сообщений. В случае невозможности доставки сообщений потребителям возможно переполнение дискового пространства на сенсоре (зависит от установленных администратором дисковых квот).

9.1. Вопросы по обслуживанию сенсора

- Как удалить сервисы Microolap EtherSensor вручную?

1. В командной строке выполнить следующие команды:

```
sc delete "EtherSensorEtherCAP"
sc delete "EtherSensorICAP"
sc delete "EtherSensorLotusTXN"
sc delete "EtherSensorAnalyser"
sc delete "EtherSensorTransfer"
sc delete "EtherSensorWatcher"
```

2. Запустить утилиту regedit и удалить ветки реестра:

```
HKLM/System/CurrentControlSet/EtherSensorAnalyser
HKLM/System/CurrentControlSet/EtherSensorEtherCAP
HKLM/System/CurrentControlSet/EtherSensorICAP
HKLM/System/CurrentControlSet/EtherSensorLotusTXN
HKLM/System/CurrentControlSet/EtherSensorTransfer
HKLM/System/CurrentControlSet/EtherSensorWatcher
```

3. Удалить файлы старой версии.

4. Перезагрузить сервер.

- Почему много дубликатов в сообщениях?

1. Веб сервисы могут отправлять формы более одного раза, например при сохранении черновиков или добавлении аттачментов.

2. В некоторых сложных сетевых условиях (например, при использовании цепочки прокси-серверов или балансировщиков нагрузки) возможна ситуация, когда сенсор видит одно и тоже соединение несколькими интерфейсами одновременно. В таком случае следует использовать фильтр check-md5, это позволит принять решение – обрабатывать или не обрабатывать данное сообщение еще раз.

- Я вижу трафик из mirror порта другим сниффером, а ваш сенсор ничего не ловит.

- Счетчики трафика увеличиваются, но сенсор ничего не перехватывает.

- В записи перехвата не видно обратных пакетов от HTTP-сервисов, т.е. пакеты от клиентского IP на удаленный сервер/порт видны, а ответы от удаленного сервера/порта – нет. От клиента при этом летят пакеты ACK, FIN/ACK, т.е. это означает, что это рабочие сессии с реальным трафиком, который доходит до клиента.

Проверка производится в следующем порядке:

1. Убедиться, что службы запущены и работают.

2. Проверить правила IP-фильтрации, помня о том, что для применения правил необходимо перезапустить службы.

3. Проверить счетчики трафика. Не должно быть количество Получено равно количеству Отклонено.

4. Проверить наличие перехваченных данных в поддиректории data директории установки EtherSensor. Если используются фильтры, проверить также содержимое [INSTALLDIR]\data\filter.

5. Проверить директорию [INSTALLDIR]\data\result – есть ли перехваченные сообщения, которые не отправлены?

6. Проверить журналы и счетчики на предмет ошибок. Если их нет, то службы работают нормально.

7. Проверить настройки mirror-порта. Например, на устройствах компании Cisco по умолчанию зеркалируются пакеты либо RX, либо TX. Нужно указать в настройке ключевое слово "both":

```
monitor session 1 source interface <interface-id> both
```

8. Проверить, настроен ли профиль в настройках службы транспорта, правильный ли профиль указан в качестве профиля по умолчанию.

9. В случае если выяснить причину самостоятельно не удается, обратиться в техническую поддержку.

10. Действия в аварийных ситуациях

Отказ технических средств

Ниже перечислены возможные виды отказа технических средств, на которых работает Microolap EtherSensor, действия по их устранению и последующему восстановлению работы сенсора.

Невозможность запуска служб Microolap EtherSensor в случае некорректной перезагрузки или остановки сервера.
Симптомом данной аварийной ситуации является системное сообщение (Error 1053) о невозможности запуска службы.

В случае если не удается запустить какую-либо службу EtherSensor, необходимо выполнить следующие действия:

- Попробовать запустить службы повторно, при этом необходимо помнить, что для успешного запуска служб EtherSensor необходим запуск службы EtherSensor Watcher
- Если повторный запуск служб не был выполнен успешно, необходимо обратиться в службу технической поддержки.

Переполнение дискового пространства, выделенного под спулы перехвата сообщений.

Симптомами данной аварийной ситуации являются:

- Оповещения операционной системы об отсутствии свободного места на диске
- Записи в системном журнале операционной системы об отсутствии свободного места на диске.

Системный администратор должен установить причину отсутствия свободного пространства. Для этого воспользуйтесь оснасткой Disk Management. Если причиной является переполнение спулов (директория [INSTALLDIR]\data), удалите или перенесите данные спулов на свободный дисковый раздел. Также причинами переполнения могут быть:

- Недоступность сервера, на который сенсор должен передавать сообщения – в таком случае требуется устранить неполадки связи
- Слишком большой объем трафика, обрабатываемый сенсором – в таком случае необходима установка дополнительных сенсоров и разделением нагрузки между ними.

Аварийное выключение питания сервера, на котором работает EtherSensor.
После возобновления питания и включения сервера, на котором работает EtherSensor, системный администратор должен убедиться в том, что все службы успешно запущены.

При возникновении отклонений от нормального запуска следует просмотреть журналы служб. Если там обнаруживаются сообщения об ошибках, необходимо обратиться в службу технической поддержки.

Обрыв сетевого подключения сервера, на котором функционирует EtherSensor.

Симптомами данной аварийной ситуации являются:

- Сетевая карта сервера, на котором функционирует сенсор, сигнализирует о механическом нарушении канала связи
- Возврат пакетов, отправленных по протоколу ICMP на серверы систем-потребителей результатов EtherSensor, не происходит или происходит частично

Системный администратор должен проверить и восстановить канал связи сервера, на котором функционирует EtherSensor. Затем следует проверить настройки сети в операционной системе. После устранения проблем с сетевым подключением необходимо убедиться в том, что все службы EtherSensor работают.

Несанкционированный доступ к Microolap EtherSensor или ОС

Несанкционированное вмешательство в Microolap EtherSensor или ОС можно определить по следующим признакам:

- В журнале безопасности Windows присутствуют записи о попытках несанкционированного доступа в систему
- В системе происходит самопроизвольное удаление или создание файлов, запуск произвольных служб.

В случае обнаружения подобных признаков необходимо выполнить следующие действия:

1. Отключить доступ сенсора к сети.
2. Остановить функционирование Microolap EtherSensor посредством остановки всех служб.
3. Устранить причины несанкционированного вмешательства в систему.
4. Восстановить работоспособность Microolap EtherSensor.

11. Что нового

```
Что нового в версии 6.1 по сравнению с версией 6.0

Среда выполнения:
   Windows Server 2012, Windows Server 2016, Windows Server 2019.
Источники данных и реконструкция объектов:
  Обработка PCAP-файлов:
    [+] Добавлена возможность детектирования и обработки протоколов с помощью Lua-
       парсеров. Теперь обработка новых протоколов уровня приложения полностью
       задаётся в конфигурации EtherSensor.
    [+] В качестве примеров добавлены следующие парсеры протоколов на Lua:
        - TCP: MySQL, Postgres, MSSQL (TDS), Oracle
        - UDP: DNS
        - IP:  ICMP.
    [+] Добавлена возможность записи трафика в PCAP-файлы. Записанный трафик имеет все
       метаданные, полученные в процессе обработки сессии:
        - Тип протокола: атрибуты NDPI, атрибуты EtherSensor
        - Трафик привязан к пользователю.
Анализ перехваченных объектов:
    [+] Детекторы Lua стали событийно-ориентированными. В настройках детектора можно
       указать в поле Settings.EventFilter, какой именно тип событий обрабатывает
       данный детектор.
       Например: EventFilter = "raw/mysql" или EventFilter = "raw/xmpp".
    [+] Реализованы следующие Lua-скрипты для этапа детектирования событий:
```

```
    - Мессенджеры:
      - l2-xmpp.lua
      - l2-slack.com.lua
      - l2-websocket-slack.lua
    - Операции с файлами:
      - l2-ftp.lua
    - Детекторы событий обмена между клиентом и сервером СУБД:
      - l2-mysql.lua
      - l2-mstds.lua
      - l2-postgres.lua
      - l2-oracle.lua
    - Обработка PCAP-файлов:
      - l2-pcap.lua
    - Обработка SIP-запросов:
      - l2-sip.lua
    - Обработка DNS-запросов:
      - l2-dns.lua
    - Обработка ICMP-событий:
      - l2-icmp.lua.
```

Доставка результатов анализа системам-потребителям:

```
  [+] Добавлена интеграция с аналитической платформой Dataplan (Angara).
  [+] Обновлён модуль libcurl.dll до версии 7.73.0.0.
  [+] Обновлён модуль libssh2.dll до версии 1.9.0.0.
```

Консоль управления:

```
  [+] Настройка перехвата трафика и обработки PCAP-файлов стала более детальной:
      - Настройка встроенных парсеров
      - Настройка Lua-парсеров
      - Настройка записи трафика в PCAP-файлы.
  [+] Фильтры пакетов теперь используют синтаксис фильтров tcpdump и библиотеки
     libpcap для генерации BPF (Berkeley Packet Filters).
      ВНИМАНИЕ:
      В версии 6.1 изменился формат пакетных фильтров. Если вы используете пакетные
     фильтры, их следует преобразовать в новый формат (tcpdump/libpcap) в ручном
     режиме. Старые фильтры сохранены в каталоге \backup\DD.MM.YYYY\config.
  [+] Lua-детекторы группируются по типам обрабатываемых событий.
```

12. Локализация GUI

Пользовательский интерфейс может быть за очень короткое время локализован на ваш язык прямо в консоли управления EtherSensor PCAP Edition.

Если кликнуть по узлу ***Локализация--Редактор***, то откроется окно локализации элементов GUI:

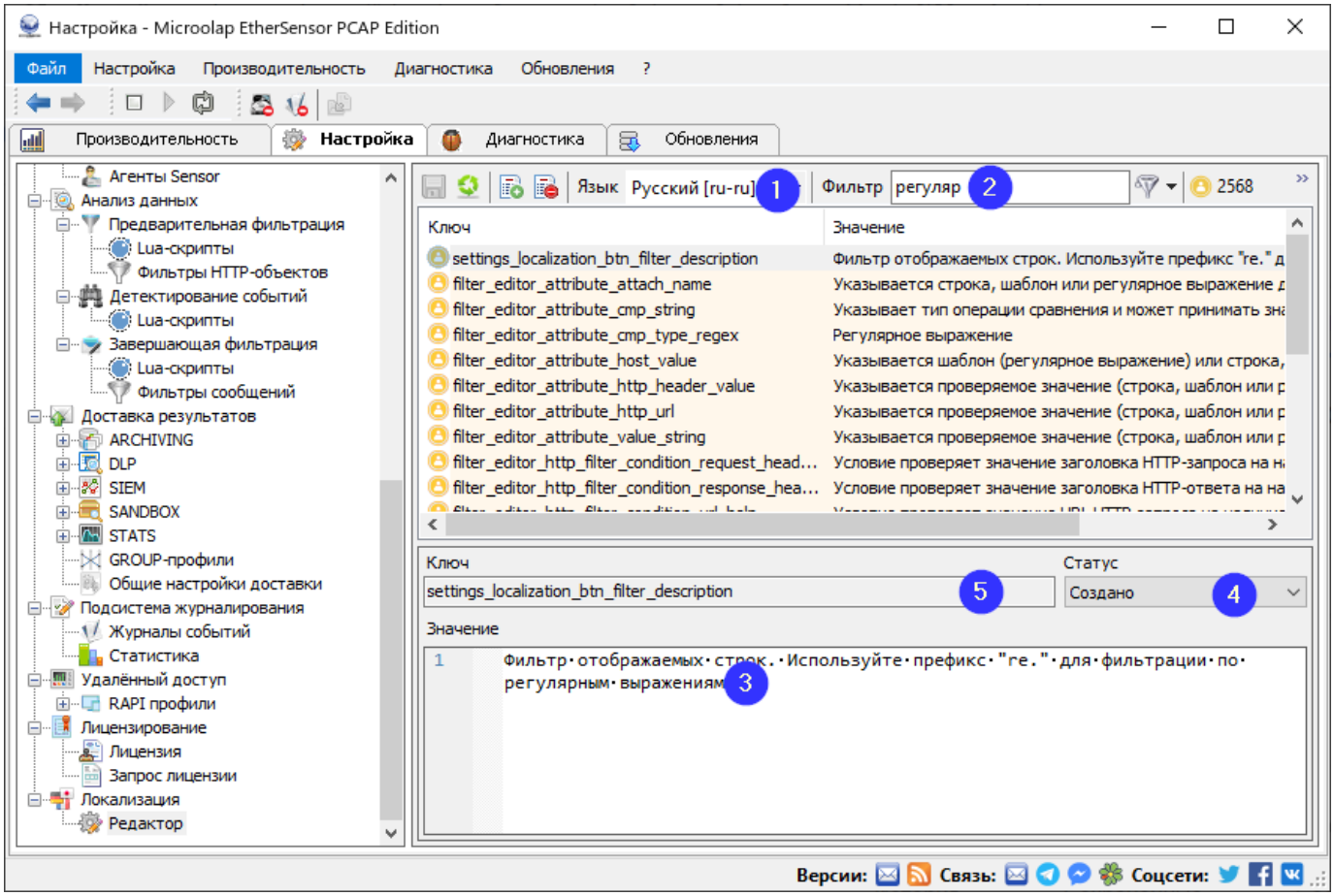

Рис.68. Окно узла Локализация--Редактор

Текстовые строки элементов GUI хранятся в языковых XML-файлах, вы можете отредактировать их непосредственно в окне узла ***Редактор***.

Для того, чтобы сохранить изменённый текст в языковом XML-файле, нажмите Ctrl-S. Чтобы увидеть обновлённые элементы GUI, нажмите Ctrl-R. Эти же действия можно произвести кнопками на тулбаре слева от выпадушки ***Язык*** (1).

Основные элементы управления окна локализации:

1. Язык:

Переключатель редактируемого языкового XML-файла (не путать с языком GUI, переключаемым по Настройка -- Изменить язык). Надписи на этом элементе определяются названием и содержимым языковых XML-файлов (см. ниже).

2. Фильтр:

Фильтр для поиска строковых значений. Поиск может происходить в колонках ***Ключ, Значение*** и ***Статус*** или же во всех сразу. Область поиска устанавливается в выпадушке около значка воронки справа от элемента ***Фильтр***.

3. Значение:

Окно редактирования поля *Значение*. Содержимое именно этого окна используется при отображении элемента интерфейса.

4. Статус:

Переключатель статуса элемента интерфейса. Названия статусов могут быть изменены на ваше усмотрение здесь же. Каждому статусу соответствует свой фоновый цвет строки элемента интерфейса. Справа вверху окна интерфейса локализации расположены счетчики элементов с различными статусами, что позволяет оценить объем оставшейся работы.

5. Ключ:

Текстовые элементы GUI хранятся в XML-файлах в виде пар "ключ-значение", где поле ***Ключ*** является уникальным ключом, а поле ***Значение*** содержит отображаемый в элементе GUI текст.

Языковые XML-файлы

Языковые файлы находятся в поддиректории lang директории установки Microolap EtherSensor и поименованы по принципу <язык>-<культура>.xml, например en-us.xml, pt-br.xml и т.п.

Внутри языковой XML-файл выглядит так:

en-us.xml

```
<?xml version="1.0" encoding="utf-8"?>
<root lang="English">
  <data name="application_name" status="created">
    <value>EtherSensor</value>
  </data>
  <data name="about_form_check_box_dont_show_again" status="created">
    <value>Don't show this window again</value>
  </data>
  <data name="application_caption_performance" status="created">
    <value>Performance</value>
  </data>
  <data name="application_caption_settings" status="created">
    <value>Settings</value>
  </data>
  <data name="application_caption_bug_report" status="created">
    <value>Diagnostics</value>
  </data>
  <data name="application_caption_updater" status="created">
    <value>Updates</value>
  </data>
  <data name="application_version" status="created">
    <value>Version</value>
  </data>
```

Рис.69. Содержание языкового XML-файла GUI.

Значение атрибута "lang" тега "root" вместе с именем файла используются при отображении текущего языкового файла в элементе ***Язык***.

Для того, чтобы начать локализацию GUI на новый язык, нужно сделать следующее:

1. Скопировать существующий языковой файл, например, en-us.xml в новый, например, pt-br.xml

2. Открыть только что созданный файл pt-br.xml любым текстовым редактором и исправить English на, например, Português brasileiro

3. Закрыть файл и заново запустить консоль управления sensor_console.exe. В выпадушке ***Язык*** появится новый язык и новый файл.

Теперь можно начинать редактировать текстовые элементы GUI в файле pt-br.xml.

Редактирование элементов GUI

Для того, чтобы отредактировать текст элемента GUI, кликните по окошку ***Значение***, измените текст, нажмите Ctrl-S – новое значение элемента сохранится в языковом файле, затем Ctrl-R –

новое значение элемента прочитается из языкового файла и отобразится в соответствующем элементе. Эти же действия можно произвести кнопками на тулбаре слева от выпадушки ***Язык***.

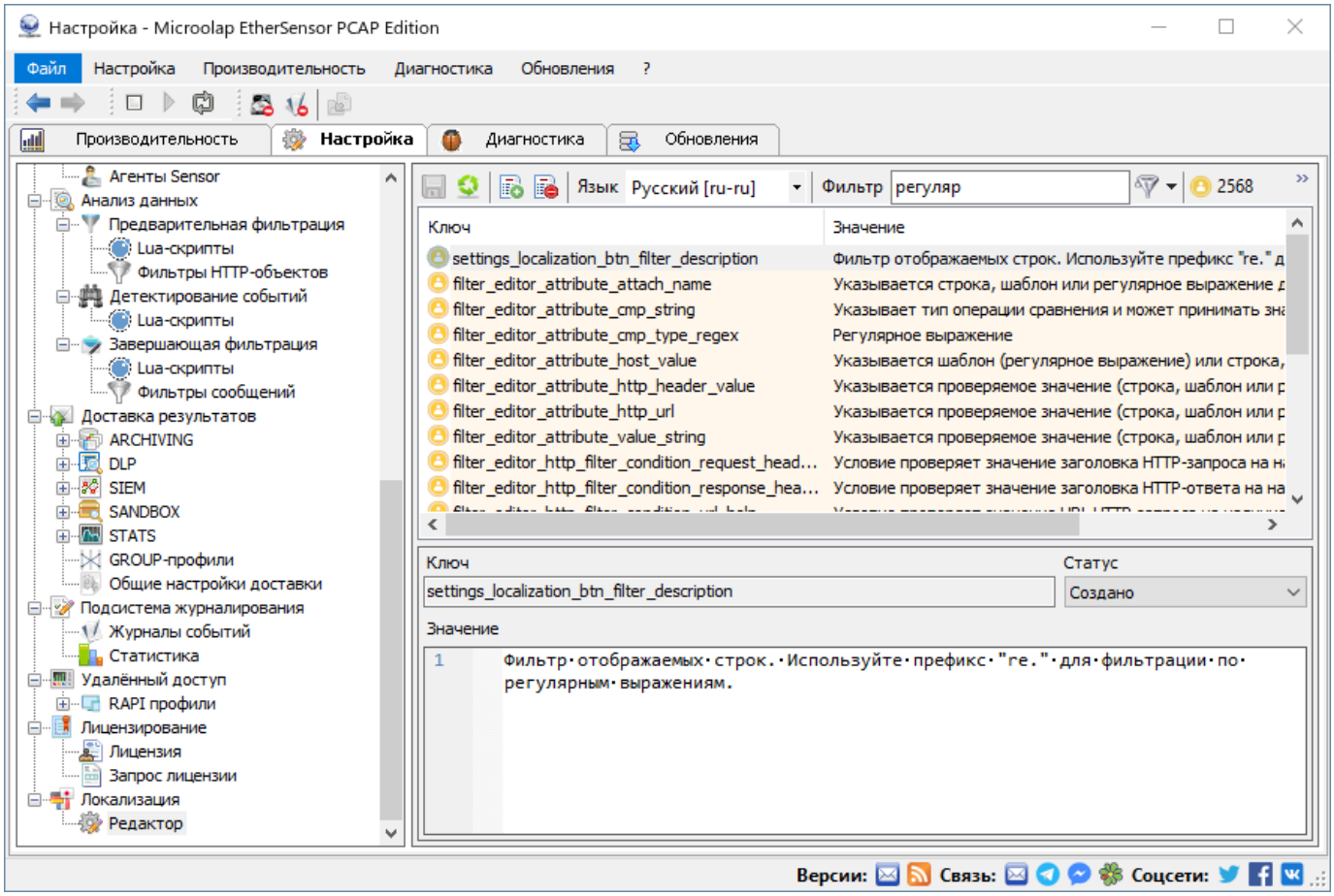

Рис.70. Редактирование текста элемента GUI.

Не забудьте сразу же поменять статус элемента, если это необходимо.

Одновременная работа с двумя языками GUI.

Чтобы не переключаться постоянно между двумя языками языковой пары GUI, лучше иметь открытыми одновременно два окна консоли управления. Для этого проделайте следующее:

1. Создайте папку для работы с первым языком, например, en-us.xml. Назовём её C:\EtherSensor Agent-en-us

2. Создайте папку для работы со вторым языком, например, pt-br.xml. Назовём её C:\EtherSensor Agent-pt-br

3. Скопируйте в эти папки все файлы из директории установки EtherSensor, оставив в первой папке в поддиректории lang только файл en-us.xml, а во второй – только pt-br.xml.

Теперь можно открыть одновременно два окна приложения консоли управления, одно из C:\EtherSensor Agent-en-us, а другое – из C:\EtherSensor Agent-pt-br и редактировать файлы, избежав возможных ошибок и путаницы.

Когда закончите, переместите файл pt-br.xml в поддиректорию lang директрии установки EtherSensor, а эти два новых каталога просто удалите.

Использование регулярных выражений при поиске в полях элементов (Фильтр).

Если строку с искомым текстом в элементе ***Фильтр*** предварить комбинацией символов re. ("r", "e" и "."), то она будет интерпретироваться как регулярное выражение.

Пример:

re.[0-9]$ – Найти все элементы, в которых значение оканчивается на цифру

re.: $ – Найти все элементы, в которых значение оканчивается на "двоеточие пробел"

re.^.{4,6}$ – Найти все элементы, состоящие из строк длиной от 4 до 6 символов.

Если вы пришлёте нам свой языковой файл, то мы его добавим к дистрибутиву ближайшего релиза EtherSensor. Кроме того, этот файл будет добавлен системой обновлений ко всем работающим на данный момент экземплярам программы.

13. Лицензирование Microolap EtherSensor

Microolap EtherSensor PCAP Edition текущей версии распространяется бесплатно, лицензионный файл также не требуется.

Приобретение лицензии Microolap EtherSensor

Поставщик: ООО Микроолап Текнолоджис

Email: sales@microolap.ru

Телефон: +7 495 748 8105

WWW: https://www.microolap.ru/company/contacts

Как приобрести Microolap EtherSensor

Для получения коммерческого предложения от ООО Микроолап Текнолоджис напишите на sales@microolap.ru.

Приобрести у партнёров: https://www.microolap.ru/partners/.

Купить онлайн на сайте www.microolap.ru.

13.1. Лицензионный файл

Информация о лицензиях Microolap EtherSensor содержится в файле license.licx, который должен находиться в корневой директории инсталляции.

Лицензия содержит информацию о модулях, лицензированных для данной инсталляции, сроках действия лицензии для каждого модуля, дату истечения подписки на обновления и другие данные. Содержимое файла лицензии открыто для просмотра, но защищено контрольной суммой и цифровой подписью. Лицензия выдается для инсталляции EtherSensor на конкретном оборудовании, для чего предусмотрена специальная проверка.

Состояние лицензии можно просмотреть с помощью консоли управления (секция ***Лицензирование***).

В зависимости от наличия в директории инсталляции файла license.licx и типов лицензий на EtherSensor, все модули работают в одном из режимов:

Демо-режим

В демо-режиме детектируются все сообщения, но только каждое пятое сообщение передается службе доставки результатов в полном оригинальном виде.

Срок работы EtherSensor в демо-режиме не ограничен: вы можете работать над развёртыванием его в сети вашей организации столько времени, сколько потребуется.

Чтобы переключить EtherSensor на полный режим, поместите в директорию инсталляции EtherSensor корректный файл license.licx.

Полный режим

В полном режиме EtherSensor работает в течение указанного явно в файле лицензии периода времени, получая и устанавливая доступные обновления.

По истечении указанного периода времени:

- Если была приобретена перманентная лицензия, то установка обновлений прекращается до тех пор, пока не будет получен файл лицензии license.licx с обновленными лицензионными данными. При этом EtherSensor продолжает работать в полном режиме.
- Если лицензия на EtherSensor была приобретёна по подписке на фиксированный срок, происходит переход в демо-режим.

При запуске в демо-режиме версии EtherSensor, выпущенной после даты окончания подписки на обновления, в лог будет записано сообщение о последней версии, работа с которой допускается текущей лицензией.

Microolap EtherSensor делает запись о факте окончания лицензии в лог. Файл лицензии передаётся на работающий сенсор автоматически через систему обновлений.

Запрос лицензии

Файл лицензии license.licx генерируется поставщиком Microolap EtherSensor на основе файла запроса на лицензию request.licx, созданного на конкретном экземпляре оборудования сенсора.

При переносе EtherSensor на другой сервер или замене вышедшего из строя оборудования EtherSensor будет распознавать лицензию как невалидную. В этом случае следует отправить новый запрос на лицензию и получить новый файл лицензии license.licx.

> Каждая лицензия привязывается к оборудованию сервера (UHID, HardwareID), для которого она создавалась.
>
> Поэтому для исключения проблем с изменением UHID в процессе эксплуатации перед созданием файла request.licx необходимо привести всё оборудование сервера в то состояние, в котором оно будет эксплуатироваться, то есть отключить все временные устройства (флеш-карты, USB-HDD и т.п.), отключить все временные, виртуальные или неиспользуемые сетевые карты, настроить MAC-адреса и т.п.
>
> Более подробно с понятием UHID можно ознакомиться в разделе "UHID (HardwareID) сервера".

Для создания файла request.licx следует в консоли управления открыть секцию ***Запрос лицензии*** и заполнить в появившемся окне соответствующие поля, затем нажать кнопку ***Сохранить***.

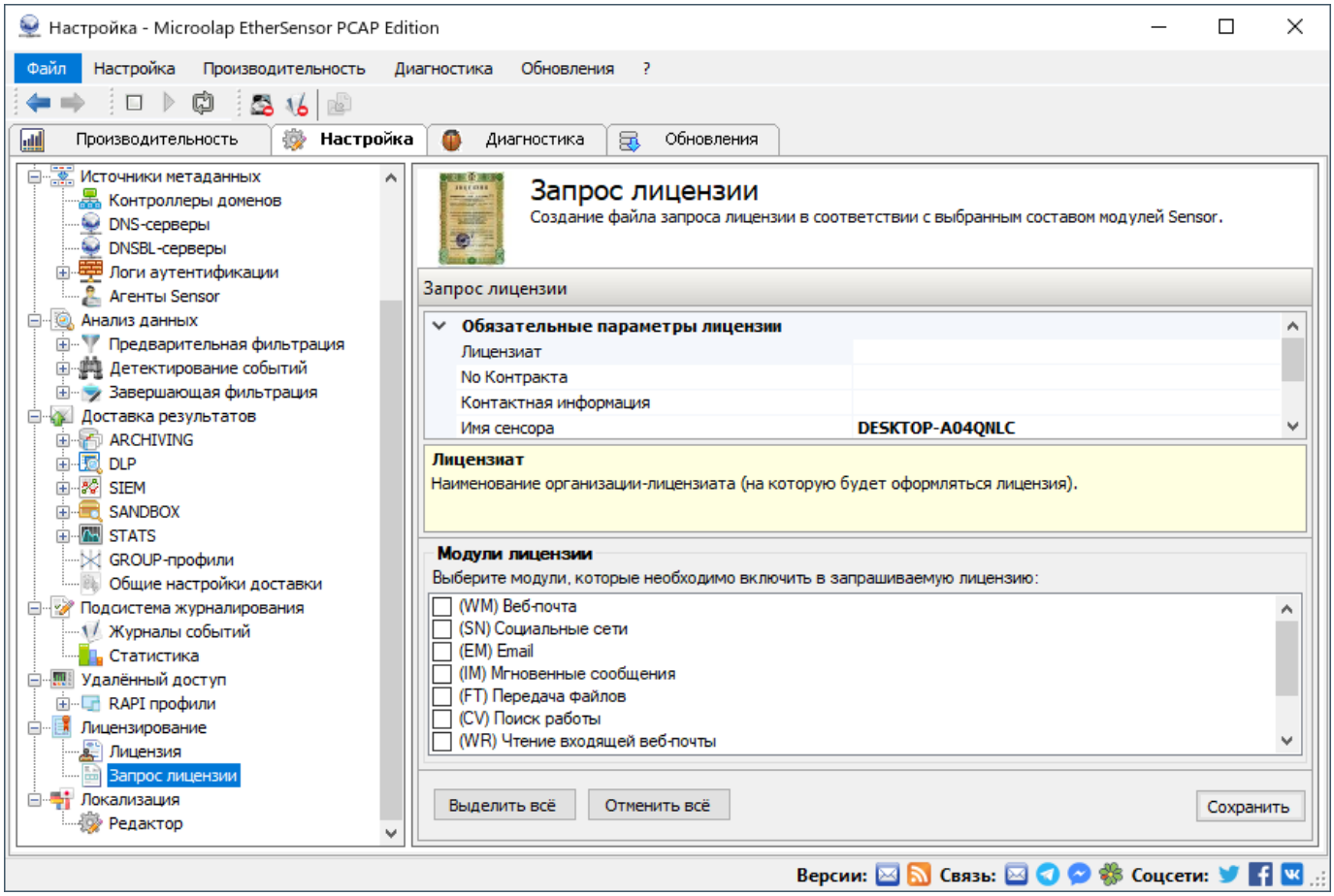

Рис.71. Форма запроса лицензии.

Файл запроса лицензии будет сохранен в директорию инсталляции с именем request.licx.

Обязательные параметры лицензии

Лицензиат

Наименование организации-лицензиата (на которую будет оформляться лицензия).

Контактная информация

Контактная информация для связи с представителем лицензиата (ФИО, e-mail, телефон).

No Контракта

Основание для предоставления лицензии: ссылка на номер контракта или договора. Если лицензия предоставляется без договора, например для тестирования, необходимо указать дату завершения тестирования (например, "Тестирование EtherSensor до 01.12.2011").

Дата окончания подписки на обновления

Обычно это дата, до которой произведена оплата по договору или дополнительному соглашению (время суток не учитывается).

Бета-лицензия

Выберите "Да", если в процессе обновления лицензия должна разрешать установку новейших бета-версий EtherSensor. Если выбрано "Нет", то обновления включают только стабильные версии.

Имя сенсора

Наименование сенсора (физической станции). Используется, чтобы отличать разные экземпляры EtherSensor в пределах одной организации. Сюда можно ввести имя хоста или какое-либо условное наименование сенсора, однозначно его идентифицирующее в пределах организации-лицензиата.

Модули лицензии

Модули EtherSensor, подлежащие лицензированию.

После нажатия кнопки ***Сохранить*** в корневой директории установки EtherSensor будет создан файл request.licx. Его следует отправить поставщику EtherSensor для создания на его основе нового файла лицензии license.licx.

Как приобрести Microolap EtherSensor
Для получения коммерческого предложения от ООО Микроолап Текнолоджис напишите на sales@microolap.ru.

Приобрести у партнёров: https://www.microolap.ru/partners/.

Купить онлайн на сайте www.microolap.ru.

13.2. UHID (HardwareID) среды выполнения

Каждая лицензия Microolap EtherSensor (файл license.licx) привязывается к оборудованию сервера EtherSensor, для которой она предназначена. Эта привязка делается через т.н. UHID – специальный код, уникальный для каждой системы и набора оборудования (Unique Hardware IDentifier). UHID добавляется в файл-лицензию, и затем EtherSensor проверяет соответствие UHID системы и UHID в файле лицензии.

Если UHID в файле лицензии не соответствует текущему UHID системы, то EtherSensor автоматически переходит в демо-режим.

При вычислении текущего UHID учитывается список сетевых карт системы и список устройств хранения системы. Это означает, что при изменении списка сетевых карт или списка устройств хранения системы текущий UHID также может измениться.

Для исключения проблем с изменением UHID в процессе эксплуатации перед созданием файла request.licx необходимо привести всё оборудование сервера EtherSensor в то состояние, в котором оно будет эксплуатироваться: отключить все временные устройства (флеш-карты, USB-HDD и т.п.), отключить все временные, виртуальные или неиспользуемые сетевые карты, настроить MAC-адреса и т.п.

В процессе эксплуатации EtherSensor вы можете временно подключать устройства хранения (флеш-карта, переносной HDD и т.п.). EtherSensor отслеживает такие ситуации и продолжает работать корректно с лицензионным файлом даже несмотря на временную смену UHID.

Однако при запланированной замене или добавлении оборудования изменения в UHID могут стать постоянными. В этой ситуации необходимо заново сгенерировать файл request.licx, и отправить его на адрес support@microolap.ru с объяснением того, какое именно оборудование изменилось. Оператор поставщика EtherSensor проверит данные и изменит лицензию, чтобы она соответствовала новому UHID сервера EtherSensor.

Проверить соответствие UHID сервера и UHID из лицензии можно в консоли управления (утилита sensor_console.exe).

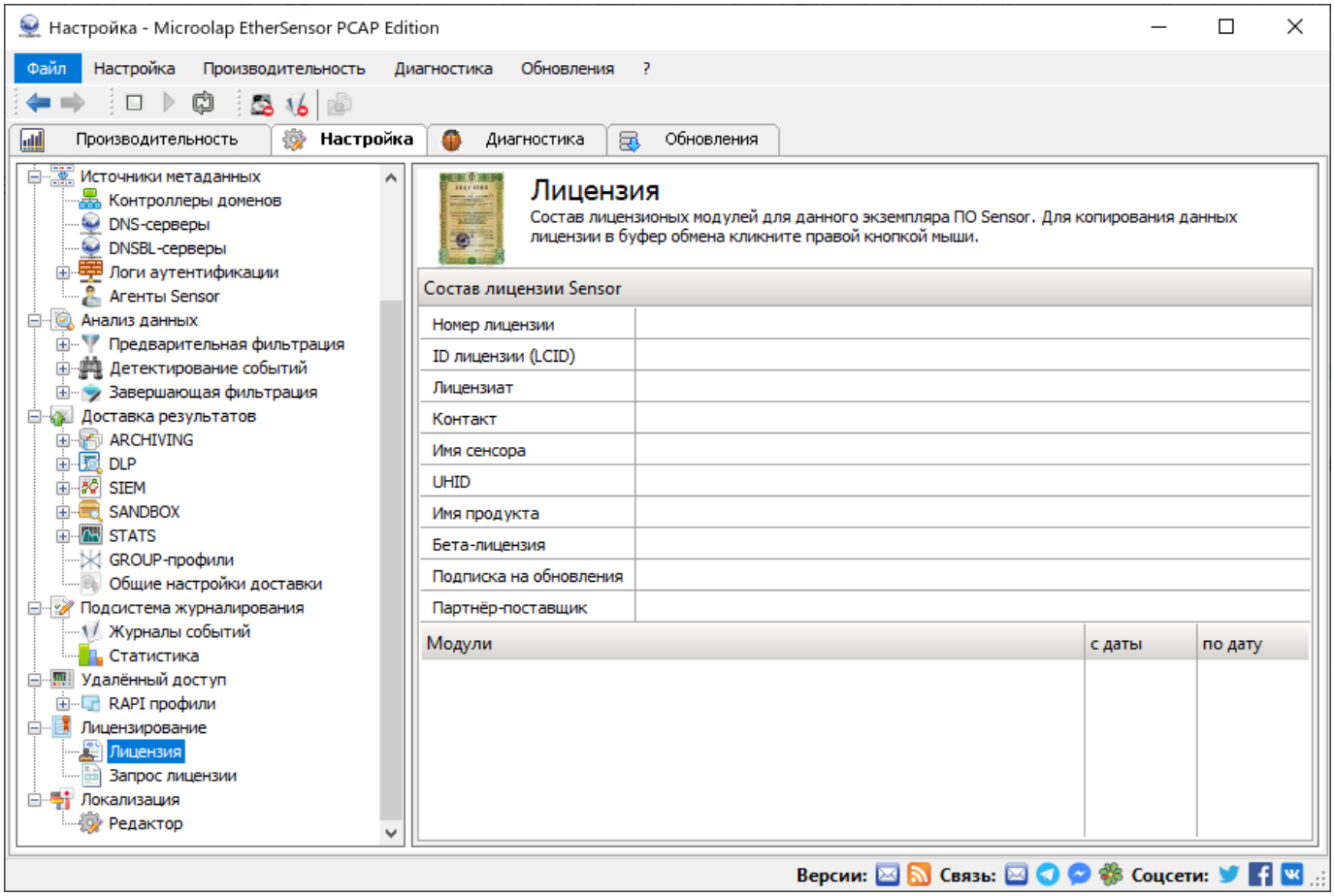

Рис.72. Информация об UHID сервера и UHID лицензии в консоли управления.

13.3. Работа с системой лицензирования

Невозможно выписать лицензию до момента установки EtherSensor на сервер, так как для генерации лицензии системе лицензирования необходимы данные о среде выполнения. Лицензии привязаны к оборудованию сервера и на других платформах работать не должны.

После инсталляции создайте запрос на лицензию в консоли управления. Запрос сохраняется в виде файла request.licx в корневой директории инсталляции и содержит данные о сервере, на который установлен EtherSensor. Более подробно о генерации файла request.licx можно узнать в разделе "Лицензионный файл".

На основании данного запроса поставщик EtherSensor выдает тот или иной тип лицензии. Этот файл может быть передан по email или на съемном носителе.

Генерация лицензии требует подтверждения поставщика EtherSensor для того, чтобы конечный

пользователь смог получить ее. Получение подтверждения занимает не менее трёх часов.

> Следует помнить, что от состояния лицензии напрямую зависит работа системы обновления EtherSensor. С истекшей лицензией обновления получить невозможно. От данных лицензии зависит, когда и какие обновления получит EtherSensor.
>
> Для штатной работы Microolap EtherSensor требуется доступ его сервера установки к Службе обновлений. Если такого доступа нет, то корректная работа EtherSensor не гарантируется и поддержка не предоставляется.

Бета-лицензии

Система лицензирования позволяет определить некоторые лицензии как бета-лицензии. Инсталляции с бета-лицензиями получают обновления до того, как они будут выпущены для всех прочих инсталляций. Это позволяет иметь тестовые инсталляции продукта для изучения новых версий.

13.4. После приобретения лицензии

Вы приобрели лицензии на Microolap EtherSensor.

Каждая лицензия позволяет использовать EtherSensor на одном компьютере или виртуальной машине.

Каждая лицензия имеет привязку к конкретному аппаратному обеспечению и не может быть использована на другом компьютере.

Пожалуйста, выполните следующие инструкции для подготовки к запуску EtherSensor:

1. Скачайте подходящий (платформа x86 или x64) дистрибутив EtherSensor по ссылке, предоставленной ООО Микроолап Текнолоджис.
2. Поместите скачанный архив на компьютер, предназначенный для запуска EtherSensor.
3. Кликните правой кнопкой мышки на архиве и выберите Properties.
4. Если на вкладке General присутствует раздел Security, поставьте галочку в поле Unblock и нажмите OK. Если раздел Security и поле Unblock отсутствуют, никаких действий не требуется, закройте окно Properties.
5. Распакуйте архив и запустите файл ethersensor_pcap_setup_x64_v6.1_ru-ru.exe.
6. Следуйте инструкциям по установке.

Для получения файла лицензии следующие действия должны быть выполнены на каждом компьютере, где планируется установить EtherSensor:

1. Из меню Start запустите консоль управления EtherSensor (sensor_console.exe).
2. Откройте вкладку ***Настройка***, раздел ***Лицензирование***, подраздел ***Запрос лицензии***.
3. Заполните форму в правой части окна и нажмите ***Сохранить***.
4. Файл request.licx будет создан и сохранён в директории инсталляции EtherSensor.
5. Пожалуйста, пришлите нам этот файл по электронной почте на адрес support@microolap.ru, чтобы мы создать для вас файл лицензии.
6. Мы вышлем вам файл license.licx, который необходимо будет положить в директорию инсталляции EtherSensor.
7. Готово. Теперь вы можете использовать зарегистрированную версию EtherSensor!

Это означает, что в течение периода подписки на обновления версий вы можете скачивать и устанвливать последние версии Microolap EtherSensor с нашего сайта, и полученный файл лицензии будет работать с новыми версиями продукта.

Кроме того, если вы не забудете включить автоматическое обновление, новые версии будут устанавливаться без необходимости вашего участия.

> ПОЖАЛУЙСТА, НЕ ПЫТАЙТЕСЬ ОТКРЫТЬ ИЛИ ПЕРЕИМЕНОВАТЬ ФАЙЛ ЛИЦЕНЗИИ!

Последняя информация и обновления: https://www.microolap.ru/support/.

13.5. Лицензионное соглашение

ВНИМАНИЕ! Внимательно ознакомьтесь с условиями лицензионного соглашения перед началом работы с программным обеспечением.

Нажатие Вами кнопки подтверждения согласия в окне согласия с лицензионным соглашением при установке программного обеспечения или использование устанавливаемого программного обеспечения означает Ваше безоговорочное согласие с условиями настоящего лицензионного соглашения. Если Вы не согласны с условиями настоящего лицензионного соглашения, Вы должны прервать установку и/или использование программного обеспечения.

Лицензионное соглашение

Настоящее пользовательское лицензионное соглашение (далее - Соглашение) является Договором присоединения между **Общество с ограниченной ответственностью Микроолап Текнолоджис** (далее - Правообладатель) и Вами, физическим или юридическим лицом (далее - Пользователь), правомерно владеющим экземпляром программного комплекса **Microolap EtherSensor** (далее - Система).

1. Определения

1.1. ПО - обозначает, как вместе, так и по отдельности, Систему, сопроводительные материалы, Документация к Системе, дополнительно заказанные обновления Системы, правообладателем которых является **Общество с ограниченной ответственностью Микроолап Текнолоджис**.

1.2. Пользователь (Вы) - лицо, которое устанавливает или использует ПО от своего лица или правомерно владеет копией ПО. Если ПО было загружено или приобретено от имени юридического лица, то под термином Пользователь (Вы) далее подразумевается юридическое лицо, для которого ПО было загружено или приобретено и которое поручило отдельному физическому лицу принять данное соглашение от своего лица.

1.3. Партнеры - организации, осуществляющие распространение ПО на основании договора с Правообладателем.

1.4. Документация к Системе - сопроводительные печатные и иные материалы, Руководство Пользователя, Руководство Администратора, справочник, файл справки и аналогичные им сопроводительные печатные и электронные документы в отношении Системы, правообладателем которых является **Общество с ограниченной ответственностью Микроолап Текнолоджис**.

2. Предоставление лицензии

2.1. Правообладатель предоставляет Вам неисключительную лицензию на использование ПО в рамках функциональности, описанной в Документации к Системе, при условии соблюдения Вами всех технических требований, описанных в Документации к Системе, а также всех ограничений и условий использования Системы, указанных в настоящем Соглашении.

2.2. В случае если Вы получили, загрузили и/или установили ПО, предназначенное для ознакомительных целей, Вы имеете право использовать ПО только в целях ознакомления и только в течение одного ознакомительного периода, если не прописано иначе, начиная с даты начальной установки ПО. Любое использование ПО для других целей или по завершении ознакомительного периода запрещено.

2.3. Если Вы используете ПО разных версий или версии ПО для разных языков, если Вы получили ПО на нескольких носителях, если Вы иным способом получили несколько копий ПО или получили ПО в составе пакета другого программного обеспечения, то общее количество Ваших компьютеров, на которых установлены и/или используются все версии Системы, должно соответствовать количеству лицензий, полученных от Правообладателя.

2.4. Вы имеете право изготовить копию ПО при условии, что эта копия предназначена только для архивных целей и для замены правомерно приобретенного экземпляра в случаях, когда оригинал утерян, уничтожен или стал непригоден для использования. Такая копия не может быть использована для иных целей и должна быть уничтожена в случае, если владение экземпляром ПО перестанет быть правомерным.

2.5. Любая правомерная передача прав в отношении пользования ПО возможна только при

условии присоединения к настоящему соглашению и с предварительного согласия Правообладателя или его уполномоченного Партнера. При этом, правопреемник Пользователя в отношении использования ПО становится законным правопреемником Пользователя в отношении использования ПО только при условии полного замещения Пользователя по настоящему соглашению.

2.6. Вы самостоятельно несете ответственность и обеспечиваете соблюдение применимого экспортного и импортного законодательства, а также применимых торговых санкций и эмбарго в отношении передачи прав и использования ПО.

3. Ограничения

3.1. Вы не вправе декомпилировать, дизассемблировать, модифицировать или выполнять производные работы, основанные на ПО, целиком или частично, за исключением случаев, предусмотренных законодательством.

3.2. Запрещается сдавать ПО в аренду, прокат, временное пользование, уступать закладывать и продавать третьим лицам, а так же разглашать результаты стендовых испытаний Системы.

3.3. Система лицензируется только для Вашей внутренней деятельности. Вы можете установить и использовать Систему только в пределах помещений, которыми Вы владеете или контролируете.

3.4. Запрещается передавать право на использование ПО третьим лицам за исключением случая, указанного в п. 2.5.

3.5. Запрещается передавать и предоставлять доступ к лицензионному ключу третьим лицам в нарушение положений настоящего Соглашения, за исключением случая, указанного в п.2.5 настоящего Соглашения. Лицензионный ключ являются конфиденциальной информацией. Правообладатель оставляет за собой право использовать средства для проверки подлинности установленного у Вас лицензионного ключа.

3.6. Правообладатель имеет право заблокировать лицензионный ключ в случае нарушения Пользователем условий настоящего Соглашения.

3.7. При использовании Вами ПО, предназначенного для ознакомительных целей, Вы не имеете права передавать имеющийся у Вас экземпляр ПО третьим лицам.

3.8. Вы не вправе использовать ПО для любых целей или способом ограниченным или запрещенным применимым законодательством. Вы самостоятельно несете ответственность за неправомерное использование ПО.

3.9. В случае нарушения Вами какого-либо из условий данного Соглашения Правообладатель вправе прервать действие Соглашения на использование ПО в любое время без Вашего уведомления и без возмещения стоимости ПО или его части.

4. Ограниченная гарантия и отказ от предоставления гарантий

4.1. Правообладатель гарантирует работу ПО в соответствии с описанием, изложенным в Документации к Системе.

4.2. Вы соглашаетесь с тем, что никакое ПО не свободно от ошибок и Вам рекомендуется регулярно создавать резервные копии своих файлов.

4.3. Правообладатель не гарантирует работоспособность ПО при нарушении условий, описанных в Документации к Системе, а также в случае нарушения Пользователем условий настоящего Соглашения.

4.4. ЗА ИСКЛЮЧЕНИЕМ УСТАНАВЛИВАЕМОЙ В НАСТОЯЩЕМ ПУНКТЕ ОГРАНИЧЕННОЙ ГАРАНТИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПОСТАВЛЯЕТСЯ "КАК ЕСТЬ". ПРАВООБЛАДАТЕЛЬ И ЕГО ПАРТНЕРЫ НЕ ДАЮТ НИКАКИХ ГАРАНТИЙ НА ЕГО ИСПОЛЬЗОВАНИЕ ИЛИ ПРОИЗВОДИТЕЛЬНОСТЬ. ЗА ИСКЛЮЧЕНИЕМ ГАРАНТИЙ, УСЛОВИЙ, ПРЕДСТАВЛЕНИЙ ИЛИ ПОЛОЖЕНИЙ, СТЕПЕНЬ КОТОРЫХ НЕ МОЖЕТ БЫТЬ ИСКЛЮЧЕНА ИЛИ ОГРАНИЧЕНА ПРИМЕНИМЫМ ЗАКОНОДАТЕЛЬСТВОМ, ПРАВООБЛАДАТЕЛЬ И ЕГО ПАРТНЕРЫ НЕ ДАЮТ НИКАКИХ ГАРАНТИЙ, УСЛОВИЙ, ПРЕДСТАВЛЕНИЙ ИЛИ ПОЛОЖЕНИЙ (ВЫРАЖАЕМЫХ В ЯВНОЙ ИЛИ В ПОДРАЗУМЕВАЕМОЙ ФОРМЕ) НА ВСЕ, ВКЛЮЧАЯ БЕЗ ОГРАНИЧЕНИЙ НЕНАРУШЕНИЕ ПРАВ ТРЕТЬИХ ЛИЦ, КОММЕРЧЕСКОЕ КАЧЕСТВО, ИНТЕГРАЦИЮ ИЛИ ПРИГОДНОСТЬ ДЛЯ ОПРЕДЕЛЕННЫХ ЦЕЛЕЙ. ВЫ СОГЛАШАЕТЕСЬ С ТЕМ, ЧТО ВЫ НЕСЕТЕ ОТВЕТСТВЕННОСТЬ ЗА ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ДОСТИЖЕНИЯ НУЖНЫХ РЕЗУЛЬТАТОВ, ЗА УСТАНОВКУ И ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, А ТАКЖЕ ЗА РЕЗУЛЬТАТЫ, ПОЛУЧЕННЫЕ С ЕГО ПОМОЩЬЮ.

5. Ограничение ответственности

В МАКСИМАЛЬНОЙ СТЕПЕНИ, ДОПУСКАЕМОЙ ПРИМЕНИМЫМ ЗАКОНОДАТЕЛЬСТВОМ, ПРАВООБЛАДАТЕЛЬ И/ИЛИ ЕГО ПАРТНЕРЫ НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ЗА КАКИЕ-ЛИБО УБЫТКИ И/ИЛИ УЩЕРБ (В ТОМ ЧИСЛЕ УБЫТКИ В СВЯЗИ С НЕДОПОЛУЧЕННОЙ КОММЕРЧЕСКОЙ ПРИБЫЛЬЮ, ПРЕРЫВАНИЕМ ДЕЯТЕЛЬНОСТИ, УТРАТОЙ ИНФОРМАЦИИ ИЛИ ИНОЙ ИМУЩЕСТВЕННЫЙ УЩЕРБ), ВОЗНИКАЮЩИЕ В СВЯЗИ С ИСПОЛЬЗОВАНИЕМ ИЛИ НЕВОЗМОЖНОСТЬЮ ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ДАЖЕ ЕСЛИ ПРАВООБЛАДАТЕЛЬ И ЕГО ПАРТНЕРЫ БЫЛИ УВЕДОМЛЕНЫ О ВОЗМОЖНОМ ВОЗНИКНОВЕНИИ ТАКИХ УБЫТКОВ И/ИЛИ УЩЕРБА. В ЛЮБОМ СЛУЧАЕ ОТВЕТСТВЕННОСТЬ ПРАВООБЛАДАТЕЛЯ И ЕГО ПАРТНЕРОВ ПО ЛЮБОМУ ИЗ ПОЛОЖЕНИЙ НАСТОЯЩЕГО СОГЛАШЕНИЯ ОГРАНИЧИВАЕТСЯ СУММОЙ, ФАКТИЧЕСКИ УПЛАЧЕННОЙ ВАМИ ЗА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ. НАСТОЯЩИЕ ОГРАНИЧЕНИЯ НЕ МОГУТ БЫТЬ ИСКЛЮЧЕНЫ ИЛИ ОГРАНИЧЕНЫ В СООТВЕТСТВИИ С ПРИМЕНИМЫМ ПРАВОМ.

6. Программные продукты

6.1. Технология третьих лиц, которая может быть целесообразна или необходима для использования в составе ПО определена в Документации к Системе. Такая технология лицензируется конечному пользователю на условиях лицензионного соглашения третьего лица, определенного в Документации к Системе. Такое лицензионное соглашение третьего лица дополняет положения настоящего Соглашения.

6.2. Входящие в состав Системы программные продукты лицензированы для ограниченного использования только в составе Системы. По истечении срока лицензионного использования ПО, пользователь обязан прекратить использование ПО и уничтожить все имеющиеся копии.

7. Открытое (свободное) программное обеспечение

Данный продукт может содержать программы, которые лицензируются (или сублицензируются) Пользователю в соответствии с общедоступной лицензией GNU или иными аналогичными лицензиями Open Source, которые помимо прочих прав разрешают пользователю копировать, модифицировать, перераспределять определенные программы или их части и получать доступ к исходному коду ("ПО с открытым исходным кодом"). Если такая лицензия предусматривает предоставление исходного кода пользователям, которым предоставляется ПО в формате исполняемого двоичного кода, исходный код делается доступным при осуществлении запроса Правообладателю или сопровождается с продуктом. Если какая-либо лицензия на ПО с открытым исходным кодом требует, чтобы Правообладатель предоставлял права на использование, копирование или модификацию ПО с открытым исходным кодом, выходящие за рамки прав, предоставляемых настоящим Соглашением, такие права имеют преимущественную силу над правами и ограничениями, оговоренными в настоящем Соглашении.

8. Права на интеллектуальную собственность

8.1. Вы соглашаетесь с тем, что ПО, документация, как и все другие объекты авторского права, а также системы, идеи и методы работы, другая информация, которая содержится в ПО, товарные знаки являются объектами интеллектуальной собственности Правообладателя или его Партнеров. Данное Соглашение не дает Вам никаких прав на использование объектов интеллектуальной собственности, включая товарные знаки и знаки обслуживания Правообладателя или его Партнеров, за исключением прав, предоставленных настоящим Соглашением.

8.2. Вы соглашаетесь с тем, что исходный код, лицензионный ключ для ПО являются собственностью Правообладателя или его соответствующих контрагентов.

8.3. Вы не можете удалять или изменять уведомления об авторских правах или другие проприетарные уведомления на любой копии ПО.

9. Контактная информация Правообладателя

Общество с ограниченной ответственностью Микроолап Текнолоджис
Телефон: +7 495 748 8105
Продажи: sales@microolap.ru
Техподдержка: support@microolap.ru
Веб-сайт: https://www.microolap.ru