Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет информационной безопасности

Кафедра инфокоммуникационных технологий

М. Н. Бобов, О. Г. Шевчук

ЗАЩИТА ИНФОРМАЦИИ В ИНФОКОММУНИКАЦИЯХ. МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ СРЕДЫ

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве учебно-методического пособия для специальности 1-45 80 01 «Системы и сети инфокоммуникаций»

Минск БГУИР 2022

УДК 004.056.5:002.6(076)
ББК 32.972.5я73
Б72

Р е ц е н з е н т ы:

кафедра связи учреждения образования
«Военная академия Республики Беларусь» (протокол №10 от 12.01.2021);

заместитель директора по науке научно-производственного республиканского унитарного предприятия «Научно-исследовательский институт технической защиты информации» кандидат технических наук, доцент С. Н. Касанин

Бобов, М. Н.

Б72 Защита информации в инфокоммуникациях. Механизмы обеспечения безопасности среды : учеб.-метод. пособие / М. Н. Бобов, О. Г. Шевчук. – Минск : БГУИР, 2022. – 96 с. : ил.

ISBN 978-985-543-663-9.

Рассмотрены основные механизмы обеспечения безопасности сетевой среды и её защиты от возможных атак: межсетевое экранирование, системы обнаружения вторжений, сканеры безопасности, средства защиты коммутаторов и маршрутизаторов. Приведены алгоритмы функционирования межсетевых экранов, их администрирование рассмотрено на примере Cisco ASA 5520. Описаны различные системы мониторинга безопасности в инфокоммуникациях.

Предназначено для студентов, изучающих дисциплину «Методы защиты сетей инфокоммуникаций».

УДК 004.056.5:002.6(076)
ББК 32.972.5я73

ISBN 978-985-543-663-9

Содержание

ВВЕДЕНИЕ

Реализация информационных технологий невозможна без создания компьютерных сетей, в которых сочетается как различного рода аппаратура по обработке, хранению и передаче информации, так и различные виды обслуживания. Сети являются связующим звеном между базами данных, терминалами пользователей и вычислительными машинами. В основном вычислительные сети создаются для того, чтобы дать возможность большому числу территориально разобщённых пользователей одновременно обращаться к вычислительным ресурсам.

Безопасность инфраструктуры предполагает решения по защите инфраструктурных компонентов системы. Цель решений – защитить инфраструктуру от неавторизованного доступа и обеспечить высокую отказоустойчивость и доступность сервисов. Для защиты информационных ресурсов инфраструктуры используются типовые решения, изложенные в документе ИСО 7498–2–99 «Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура зашиты информации», в котором определены базовые сетевые механизмы защиты, в том числе аутентификация, управление доступом, шифрование, электронная цифровая подпись, контроль целостности, аудит. Кроме базовых для защиты инфокоммуникаций используются специфические механизмы, обеспечивающие поддержку безопасной сетевой среды и её защиту от возможных атак. К таким механизмам относятся: межсетевое экранирование, системы обнаружения вторжений, сканеры безопасности, средства защиты коммутаторов и маршрутизаторов.

1 КЛАССИФИКАЦИЯ И ФУНКЦИОНАЛЬНАЯ СТРУКТУРА МЕЖСЕТЕВЫХ ЭКРАНОВ

Межсетевые экраны (МСЭ) обеспечивают барьер между сетями и предотвращают или блокируют нежелательный или несанкционированный трафик.

Межсетевой экран – система или группа систем, используемая для управления доступом между доверенными и недоверенными сетями на основе предварительно сконфигурированных правил [1–3].

Для построения межсетевого экрана используется определённый метод проверки пакета. В каждом из этих методов используется информация от различных уровней модели взаимосвязи открытых систем. Известны три типа межсетевых экранов [1]:

1) пакетные фильтры (Packet filtering);

2) МСЭ с контролем состояния (Stateful packet inspection);

3) шлюз прикладного уровня или прокси-серверы (Application-level gateways or proxies).

Для обеспечения повышенных возможностей по безопасности в МСЭ реализуются гибридные методы проверки пакетов на основе комбинации из известных типов.

1.1 Пакетные фильтры

Пакетные фильтры (рисунок 1.1) – самый простой метод проверки пакета. Процесс фильтрования пакета заключается в исследовании информации, содержащейся в заголовке, и сравнении её с предварительно сконфигурированной группой правил или фильтрами. Каждый пакет может исследоваться индивидуально без отношения к другим пакетам, несмотря на то, что они могут являться частью одного трафика.

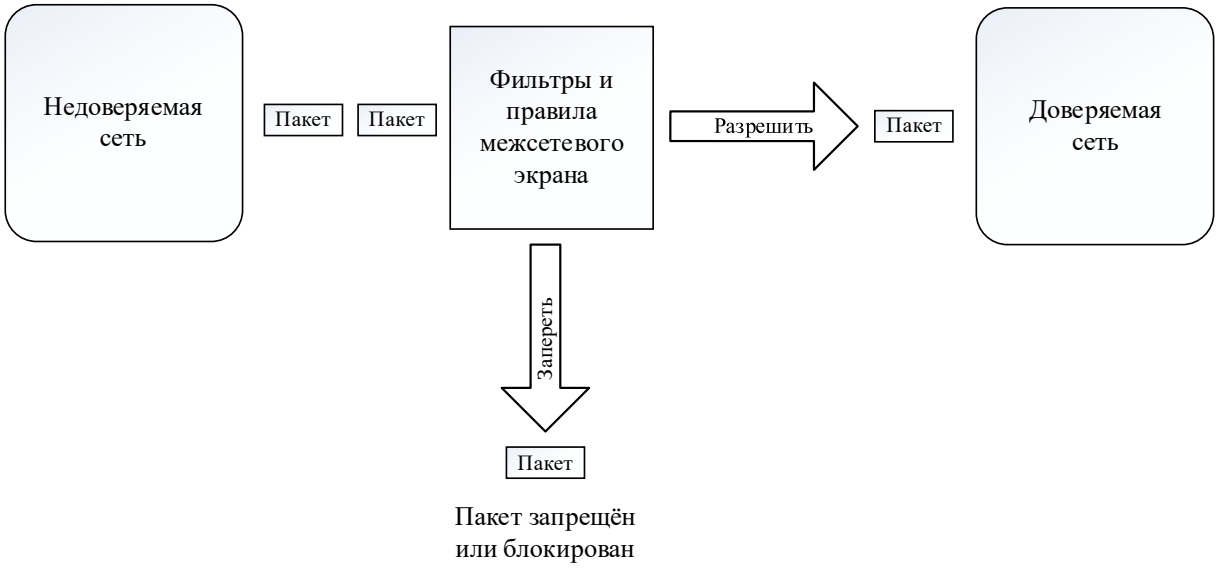


Рисунок 1.1 – Межсетевой экран – пакетный фильтр

Пакетные фильтры часто называют межсетевыми экранами уровня сети, потому что процесс фильтрования происходит на сетевом (третий уровень) или транспортном уровне (четвертый уровень) модели OSI. Рисунок 1.2 показывает отношение между пакетным фильтром и моделью OSI.

Рисунок 1.2 – Пакетный фильтр и уровни OSI

Правила пакетной фильтрации или фильтры могут быть сконфигурированы на основе разрешения или запрета. Конфигурация правил фильтрования пакета основывается на одном (или более) из следующих параметров:

– адрес источника;
– адрес назначения;
– тип протокола;
– порт источника;
– порт назначения.

Достоинства пакетных фильтров:

1) функционируют быстрее, чем другие типы МСЭ, фильтруют пакеты на более низких уровнях модели OSI, также при корректной настройке оказывают очень малое влияние на работу сети;

2) могут быть установлены «прозрачным» образом, они не требуют никакой дополнительной конфигурации для клиентов;

3) дешевле, чем другие методы проверки пакета;

4) являются независимыми от приложения, их решения основаны на информации, содержащейся в заголовке пакета, а не на информации, которая имеет отношение к определённому приложению.

Недостатки пакетных фильтров:

1) если порт был открыт МСЭ, то он открыт для всех проходящих трафиков через этот порт;

2) определение правил и фильтров в этом методе является сложной задачей, поэтому у администратора сети должно быть хорошее знание услуг и протоколов для выполнения требований безопасности;

3) проверка точности выполнения правил на пакетном фильтре является очень трудной задачей. Даже если правила кажутся простыми и явными, проверка их корректности путём тестирования отнимает много времени и не всегда даёт необходимый результат.

1.2 МСЭ с контролем состояния

МСЭ с контролем состояния исследует информацию заголовков пакетов от сетевого до прикладного уровня модели OSI и проверяет, является ли данный пакет частью законного потока и используются ли допустимые протоколы. Рисунок 1.3 показывает отношение между МСЭ с контролем состояния и моделью OSI.

Рисунок 1.3 – МСЭ с контролем состояния и уровни OSI

МСЭ с контролем состояния работает следующим образом (рисунок 1.4). Заголовки TCP-пакета проверяются для определения, является ли пакет частью уже существующего и действующего потока передаваемых данных.

Межсетевой экран имеет активную таблицу всех текущих сеансов и сравнивает входящие пакеты с её данными в процессе контроля доступа. Если в таблице отсутствует соответствующий вход соединения, МСЭ проверяет пакет с использованием установленного набора правил, аналогичного фильтру пакетов. Если проверка по правилам фильтрации прошла успешно и передача пакета разрешается, МСЭ создаёт или обновляет свою таблицу соединений. Внесённый вход соединения будет использоваться для проверки последующих пакетов вместо правил фильтрации. В качестве параметров проверки состояния используются:

– адрес источника;
– адрес назначения;
– тип протокола;
– порт источника;
– порт назначения;
– состояние связи.

Состояние связи определяется из информации, собранной на основе анализа предыдущих пакетов. Это – существенный фактор в принятии решения при новых попытках открыть соединение. МСЭ с контролем состояния сравнивает

пакеты с правилами или фильтрами и затем по динамической таблице состояния проверяет, что все пакеты – часть действительной и установленной связи.

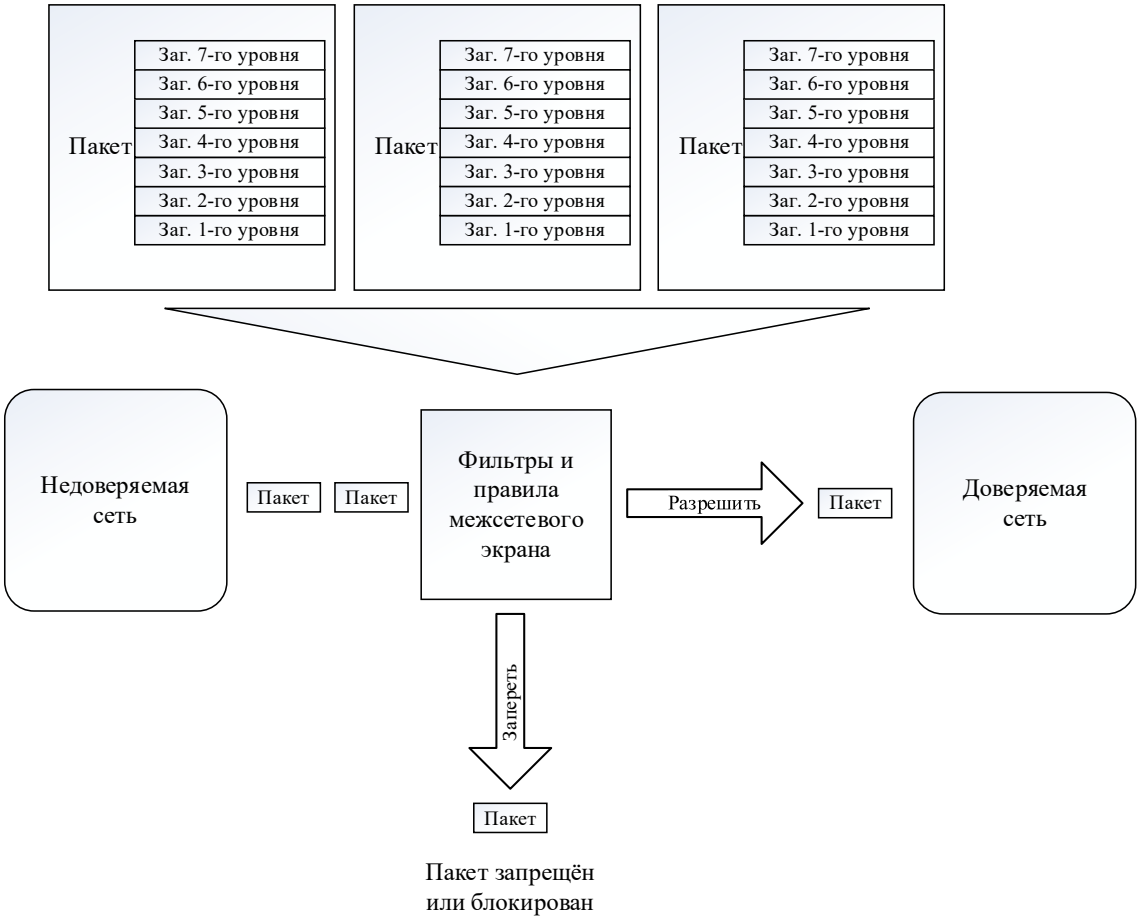


Рисунок 1.4 – МСЭ с контролем состояния

Этот метод защищает сети от атак лучше, чем методы экранирования пакетов, потому что он имеет возможность анализа состояния связи.

Достоинства МСЭ с контролем состояния:

1) оказывают очень небольшое влияние на работу сети, они реализуются «прозрачно» и являются независимыми от приложений;

2) более безопасны, чем пакетные фильтры, производят более глубокий анализ заголовка пакета для определения состояния связи между конечными точками;

3) анализируя информацию заголовка пакета, МСЭ с контролем состояния может проверить, что протоколы прикладного уровня работают правильно;

4) обычно имеют некоторые возможности по регистрации, которая может помочь идентифицировать и отследить различные типы трафиков, проходящие через межсетевой экран.

Недостатки МСЭ с контролем состояния:

1) не нарушает модель «клиент – сервер» и разрешает прямое соединение между этими двумя конечными точками;

2) правила и фильтры этого метода могут быть достаточно сложными и трудными для восприятия.

1.3 Прокси-серверы

Прокси-серверы обычно реализуются на безопасной системе хоста, формируемой с двумя интерфейсами сети. Прокси-серверы являются посредниками между этими двумя конечными точками. Этот метод проверки пакета нарушает модель «клиент – сервер» и осуществляет вместо этой модели две связи:

1) от источника к прокси-серверу;

2) от прокси-сервера к назначению.

Каждая конечная точка может общаться с другими точками, только проходя прокси-сервер.

Этот тип межсетевого экрана работает на прикладном уровне модели OSI. Для соединения конечных точек источников с точками назначений прокси-сервер должен быть реализован в каждом протоколе прикладного уровня. Рисунок 1.5 показывает отношение между прокси-серверами и моделью OSI.

Рисунок 1.5 – Прокси-серверы и уровни OSI

Прокси-сервер работает следующим образом (рисунок 1.6). Когда клиент делает запрос из недоверяемой сети, устанавливается связь с прокси-сервером. Прокси-сервер определяет, действителен ли запрос (сравнивая его с правилами или фильтрами), и затем посылает новый запрос от себя к назначению. При использовании этого метода прямая связь от доверяемой сети до недоверяемой сети никогда не осуществляется, и запрос представляется пришедшим от прокси-сервера.

Ответ отсылается назад к прокси-серверу и затем пересылается клиенту. Нарушая модель «клиент – сервер», этот тип межсетевого экрана может эффективно скрыть доверяемую сеть от недоверяемой сети.

Рисунок 1.6 – Прокси-сервер межсетевого экрана

В отличие от пакетного фильтра и МСЭ с контролем состояния, прокси-сервер может видеть все аспекты прикладного уровня, и, таким образом, может исследовать более определённую информацию. Например, он может найти различие между частью электронной почты, содержащей текст и графическое изображение, или между веб-страницами с использованием языка Java и веб-страницами без Java. С точки зрения безопасности прокси-серверы первичнее других типов экранирования пакета, но их использование не всегда является самым практичным.

Достоинства прокси-серверов:

1) не позволяют прямую связь между конечными точками, нарушают модель «клиент – сервер». В этом отношении данный метод действительно разделяет внутренние и внешние сети;

2) не реализуют прямой маршрут между сетями, никакая маршрутизация не осуществляется и данный метод обеспечивает трансляцию сетевых адресов (Network Address Translation (NAT));

3) позволяют администратору сети иметь больше контроля над трафиком, проходящим через межсетевой экран. Они могут разрешить или запретить определённые приложения или их особенности;

4) имеют лучшие технологии фильтрования, способность исследовать информационную часть пакета, следовательно, они могут принимать решения, основанные на содержании.

Недостатки прокси-серверов:

1) весь исходящий и входящий трафик проверяется на прикладном уровне, поэтому они медленнее, чем пакетные фильтры и МСЭ с контролем состояния, которые проверяют трафик на сетевом уровне. В этом методе все трафики должны пройти через все уровни модели OSI, в результате инспекционный процесс требует много времени обработки. Это может привести к тому, что МСЭ может стать узким местом в сети;

2) каждый протокол требует своей собственной привязки к прокси-серверу. Если такой привязки не существует, то соответствующий протокол не может проходить через межсетевой экран. Кроме того, для каждого протокола требуется свой собственный прокси-сервер, поэтому поддержка новых протоколов может стать трудным делом;

3) требуют дополнительных конфигураций клиента. Клиентам в сети может потребоваться специализированное программное обеспечение, чтобы быть в состоянии соединиться с прокси-сервером. Это может оказать сильное влияние на большие сети с многочисленными клиентами;

4) масштабируемость может быть проблемой с прокси-серверами, когда они установлены в больших сетях, потому что если число клиентов или число прокси-серверов, расположенных на одном хосте, увеличивается, то работа ухудшается;

5) прокси-серверы, установленные на операционных системах общего назначения, уязвимы для лазеек безопасности основной системы. Если основная система не безопасна, то и межсетевой экран не безопасен.

2 АЛГОРИТМ ФУНКЦИОНИРОВАНИЯ МСЭ

МСЭ анализирует трафик путём последовательного выполнения определённых функций, включающих в себя контроль целостности, трансляцию адреса, ведение таблицы соединения, управление доступом, инспектирование состояния соединения и проверку контента [3]. На рисунке 2.1 показан порядок выполнения этих функций от входного интерфейса X до выходного интерфейса Y.

Контроль целостности включает в себя проверку правильности адреса источника каждого пакета. Используя таблицу маршрутизации, МСЭ с помощью этой функции обнаруживает и блокирует пакеты с поддельными адресами источника (атака спуфинга).

Технология трансляции сетевых адресов осуществляется для скрытия внутренних адресов своей сети, чтобы не дать злоумышленнику возможности получить информацию о структуре и масштабах сети, а также о структуре и интенсивности исходящего и входящего трафиков.

Рисунок 2.1 – Последовательность функций анализа пакета в МСЭ

Функция ведения таблицы соединений служит для наблюдения состояния каждой связи, проходящей через МСЭ. Если связь разрешена, то в таблице соединения создаётся соответствующий вход и каждое изменение состояния этого входа должно контролироваться. Каждый вход в таблице соединения контролируется по следующим параметрам:

– локальный и глобальный адрес;
– локальный и глобальный номер порта;
– используемый протокол (TCP, UDP или ICMP);
– флаги состояния соединения;
– время простоя;
– счётчик байтов;
– последовательность номеров пакетов TCP.

Входы состояния соединения удаляются из таблицы в случаях, если сессия окончена, время простоя истекло или связи не установлены полностью.

Прежде чем соединение может быть установлено или позволена его актуализация, его трафик должен быть разрешён в соответствии со списком контроля доступа. В МСЭ может быть сформировано любое число списков, но только один список может быть установлен на каждом интерфейсе в определённом направлении.

Механизм инспектирования используется для проверки каждого соединения на предмет соответствия правил управления трафиком применяемому протоколу связи. Этот процесс называется *инспекцией протокола прикладного уровня*.

Некоторые протоколы просты и имеют очень слабые правила для управления трафиком между источником и получателем. К ним относятся некоммутируемые протоколы, такие как ICMP и UDP. Напротив, коммутируемые протоколы, такие как TCP, очень строги в подтверждении связи и обмене пакета между источником и назначением.

Функция экспертизы контента служит для проверки содержательной части пакетов на отсутствие нежелательных вставок, которые могут содержать в себе скрытые вирусы, сетевые черви или другой вредоносный код. Для реализации веб-серверов большинство веб-сайтов используют Java-апплеты и ActivX-скрипты. Эти реализации бывают в форме изображений, динамических содержаний, мультимедийных презентаций и других типов веб-элементов. Однако эти приложения могут быть использованы для разрушения, стирания или доступа к информации пользователя.

2.1 Алгоритм контроля целостности

Для предотвращения злонамеренного трафика на внутреннюю сеть МСЭ использует механизм поиска обратного маршрута (reverse path forwarding (RPF)), который заключается проверке правильности адреса источника в получаемых пакетах. Если адрес источника недействителен, пакет отбрасывается.

Механизм поиска обратного маршрута включает несколько способов проверки: свободный и строгий, а также их комбинацию. Кроме того, он поддерживает проверку маршрута по умолчанию. Выбор используемого способа на каждом интерфейсе межсетевого экрана зависит от реализации сегмента сети, связанного с этим интерфейсом.

2.1.1 Свободный способ RPF

В свободном способе RPF пакет должен быть получен от интерфейса, который будет использоваться межсетевым экраном для отправки возвращаемого пакета. В этом случае механизм RPF может блокировать законный трафик, если он поступает через интерфейс, который выбран межсетевым экраном для отправки возвращаемых пакетов. Эта проблема возникает тогда, когда в сети присутствуют асимметричные маршруты [2, 13].

Правильность адреса источника проверяется следующим образом (рисунок 2.2) [13]:

1 Блокируются пакеты с адресами источников (*SA*=255.255.255.255). Блокируются пакеты с нулевыми адресами источников (*SA*=0.0.0.0) и адресами

назначения (*DA*≠255.255.255.255). Пакет с адресом источника *SA*=0.0.0.0 и адресом назначения *DA*=255.255.255.255 может быть пакетом DHCP- или BOOT-протоколов и таким образом не отбрасывается.

2 Анализируется таблица адресации в соответствующей базе адресов (FIB): если адрес источника входящего пакета найден, пакет пропускается; если адрес источника не найден в таблице FIB, RPF принимает решение, основанное на маршруте по умолчанию и ключевом слове «маршрут по умолчанию разрешён».

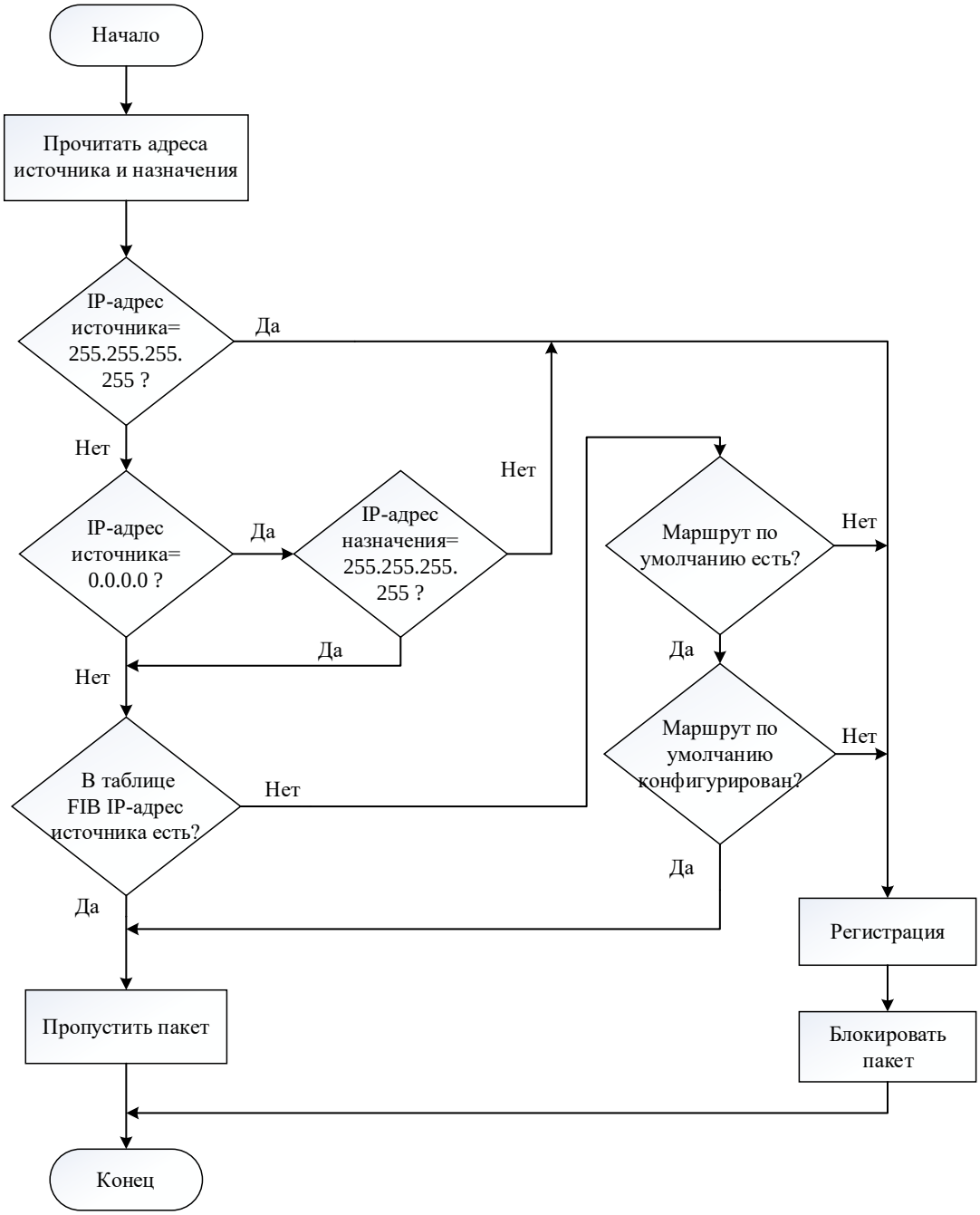


Рисунок 2.2 – Алгоритм свободного способа RPF

3 Если маршрут по умолчанию не установлен, пакет блокируется. Если маршрут по умолчанию доступен, но ключевое слово «маршрут по умолчанию

разрешён» не сформировано, то пакет блокируется. Если маршрут по умолчанию доступен и ключевое слово «маршрут по умолчанию разрешён» сформировано, то пакет пропускается.

2.1.2 Строгий способ RPF

В строгом способе RPF адрес источника должен находиться в таблице маршрутизации. Кроме того, если обратный маршрут до любого источника указывает на нулевой интерфейс, то все пакеты, содержащие адрес этого источника, блокируются. Строгий способ RPF на практике используется в сетях, которые содержат асимметричные маршруты. Алгоритм строгого способа RTF показан на рисунке 2.3.

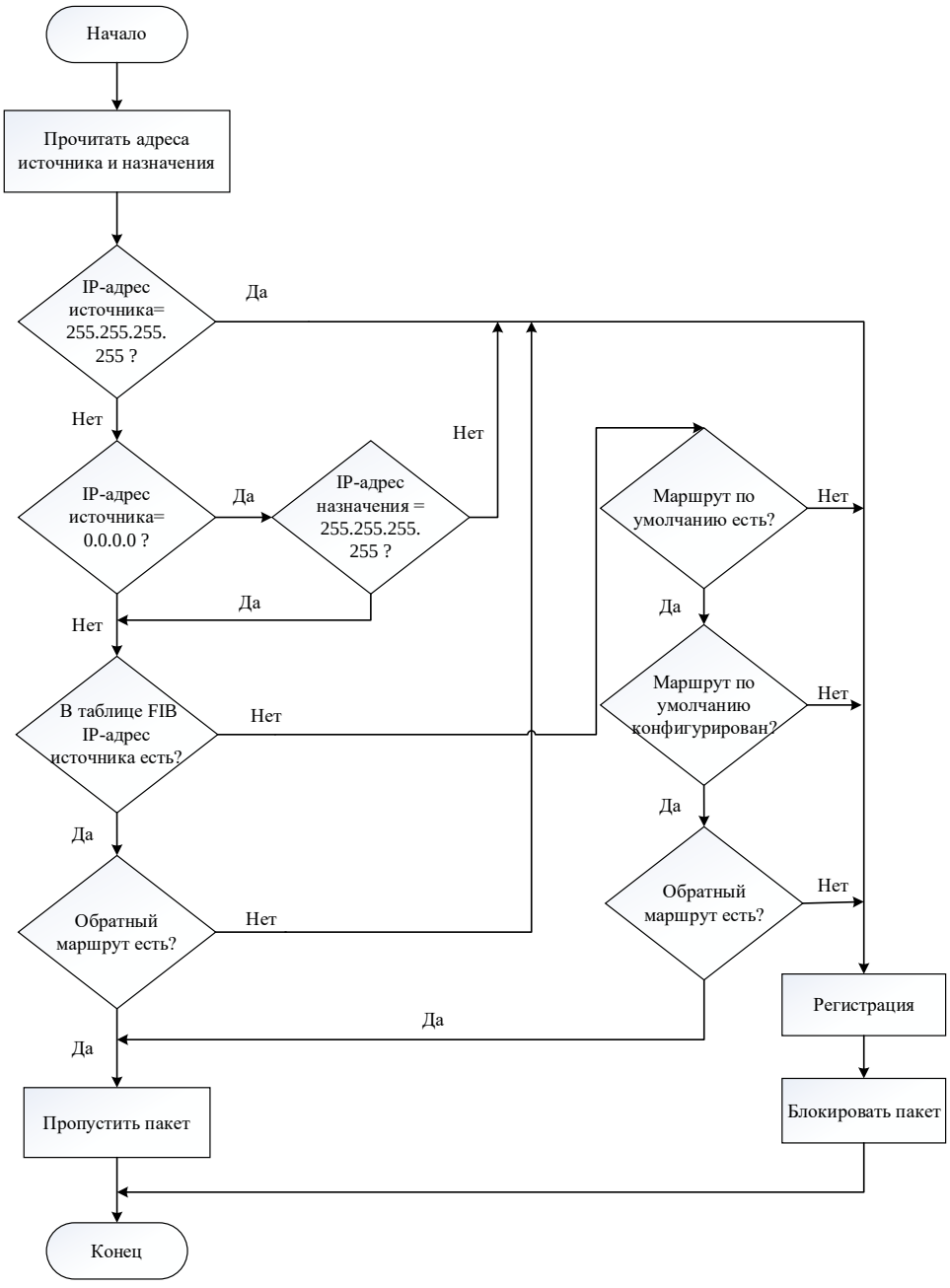


Рисунок 2.3 – Алгоритм строгого способа RPF

Как показано на рисунке 2.3, правильность адреса источника проверяется следующим образом [2, 13]:

1 Блокируются пакеты с адресами источников (*SA*=255.255.255.255). Блокируются пакеты с нулевыми адресами источников (*SA*=0.0.0.0) и адресами назначения (*DA*≠255.255.255.255). Пакет с адресом источника *SA*=0.0.0.0 и адресом назначения *DA*=255.255.255.255 может быть пакетом DHCP- или BOOT-протоколов и таким образом не блокируется.

2 Анализируется таблица FIB: если адрес источника входящего пакета найден, то осуществляется поиск обратных маршрутов (reverse route lookup) к адресу источника; если по крайней мере один исходящий интерфейс из таких маршрутов соответствует интерфейсу получения, пакет пропускается, иначе пакет блокируется.

3 Если адрес источника не найден в таблице FIB, принимается решение, основанное на маршруте по умолчанию и ключевом слове «allow-default-route». Если маршрут по умолчанию отсутствует, пакет блокируется. Если маршрут по умолчанию имеется, но ключевое слово «allow-default-route» не формируется, пакет блокируется. Если маршрут по умолчанию имеется, ключевое слово «allow-default-rout» сформировано и исходящий интерфейс маршрута по умолчанию является интерфейсом получения, то пакет пропускается. В противном случае пакет блокируется.

Механизм поиска обратного маршрута обеспечивает эффективную защиту локальных сетей от атаки спуфинга и блокирует пакеты с поддельными адресами от прохождения на внутреннюю сеть [2, 13].

2.2 Алгоритм трансляции адресов

Трансляция сетевых адресов (Network Address Translation, NAT) осуществляется на межсетевом экране, стоящем на границе между внутренней и внешней сетью. Перед посылкой пакетов во внешнюю сеть NAT транслирует внутренние локальные адреса в глобальные уникальные IP-адреса и наоборот. Эта технология осуществляется для скрытия внутренних адресов своей сети, чтобы не дать злоумышленнику возможности получить информацию о структуре и масштабах сети, а также структуре и интенсивности исходящего и входящего трафиков.

NAT включает в себя статическую и динамическую трансляцию, при этом статическая трансляция устанавливает взаимно однозначное соответствие между внутренними локальными адресами и внутренними глобальными адресами.

Динамическая трансляция устанавливает соответствие между внутренними локальными адресами и пулом глобальных адресов.

Статический метод трансляции [2]. На рисунке 2.4 показан МСЭ, транслирующий адрес источника при переходе пакета из внутренней сети во внешнюю.

Трансляция внутренних адресов источника статическим методом включает следующие шаги:

1) пользователь на хосте 1.1.1.1 открывает соединение с хостом B;

2) МСЭ получает пакет от хоста 1.1.1.1, читает информацию из заголовка и сверяется со своей NAT-таблицей;

3) если входа трансляции не существует в таблице, МСЭ блокирует пакет;

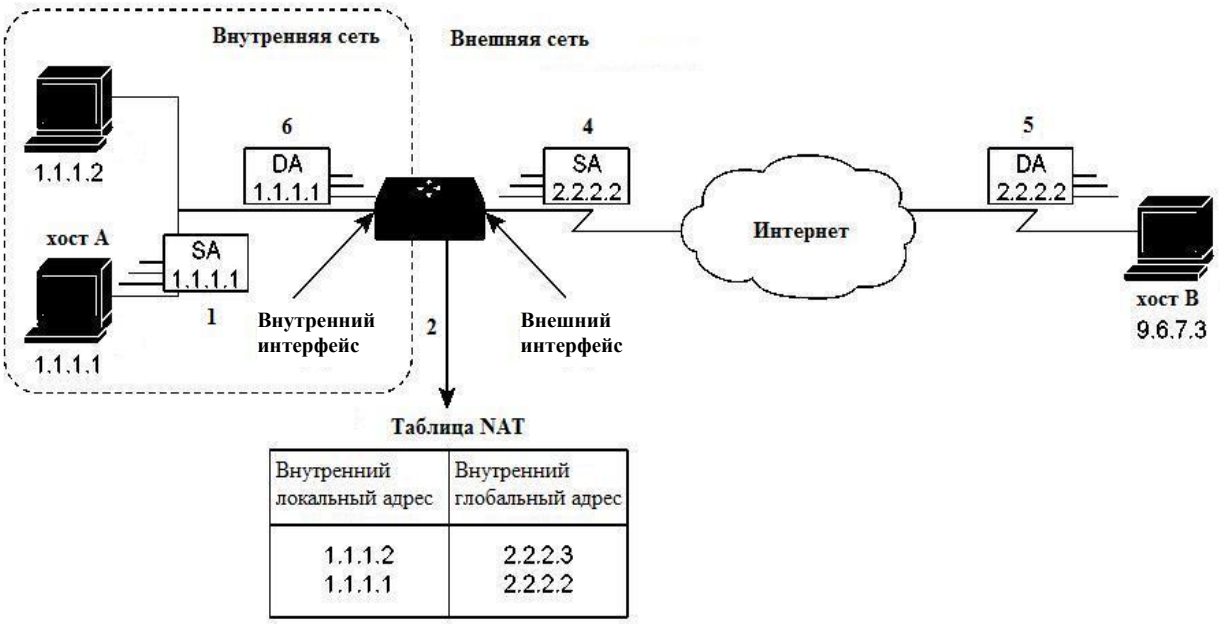
4) МСЭ заменяет внутренний локальный адрес источника хоста 1.1.1.1 (*SA*=1.1.1.1) на глобальный адрес в соответствии с входом в таблице и отправляет пакет;

5) хост B получает пакет и отвечает хосту 1.1.1.1, используя глобальный адрес назначения 2.2.2.2 (*DA*=2.2.2.2);

6) когда МСЭ получает пакет с внутренним глобальным адресом, он проверяет NAT-таблицу, транслирует адрес во внутренний локальный адрес хоста 1.1.1.1 и направляет пакет хосту 1.1.1.1;

7) хост 1.1.1.1 получает пакет и продолжает диалог с хостом B. Для каждого пакета МСЭ повторяет действия шагов со второго по пятый.

Алгоритм данного механизма представлен на рисунке 2.5.



SA – адрес источника (Source Address);
DA – адрес назначения (Destination Address)

Рисунок 2.4 – Трансляция внутренних адресов

Динамический метод трансляции [2]. Трансляция внутренних адресов источника динамическим методом включает следующие шаги:

1) пользователь на хосте 1.1.1.1 открывает соединение с хостом B;

2) МСЭ получает первый пакет от хоста 1.1.1.1, читает заголовок и сверяется со своей NAT-таблицей;

3) если вход трансляции не существует в таблице, МСЭ определяет, что адрес источника 1.1.1.1 должен транслироваться динамически, выбирает легальный глобальный адрес из пула динамических адресов и создаёт вход в таблице трансляции. Этот тип входа называется простым входом;

4) МСЭ заменяет внутренний локальный адрес источника хоста 1.1.1.1 (*SA*=1.1.1.1) на глобальный адрес в соответствии с входом в таблице и отправляет пакет;

5) хост B получает пакет и отвечает хосту 1.1.1.1, используя глобальный адрес назначения 2.2.2.2 (*DA*=2.2.2.2);

6) когда МСЭ получает пакет с внутренним глобальным адресом, он проверяет NAT-таблицу, транслирует адрес во внутренний локальный адрес хоста 1.1.1.1 и направляет пакет хосту 1.1.1.1;

7) хост 1.1.1.1 получает пакет и продолжает диалог с хостом B. Для каждого пакета МСЭ повторяет действия шагов со второго по пятый.

Алгоритм данного механизма представлен на рисунке 2.6.

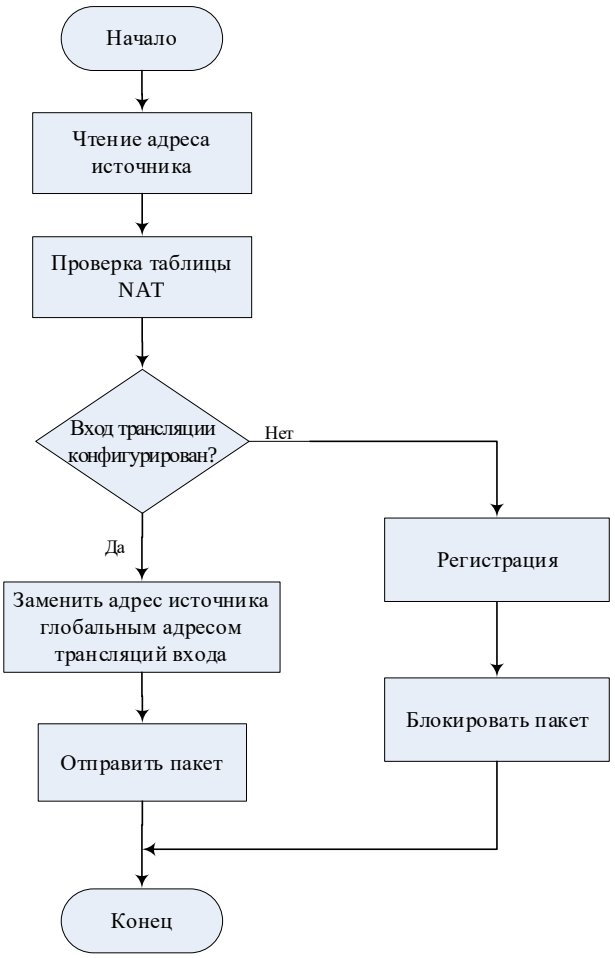


Рисунок 2.5 – Алгоритм трансляции внутреннего адреса источника статическим методом

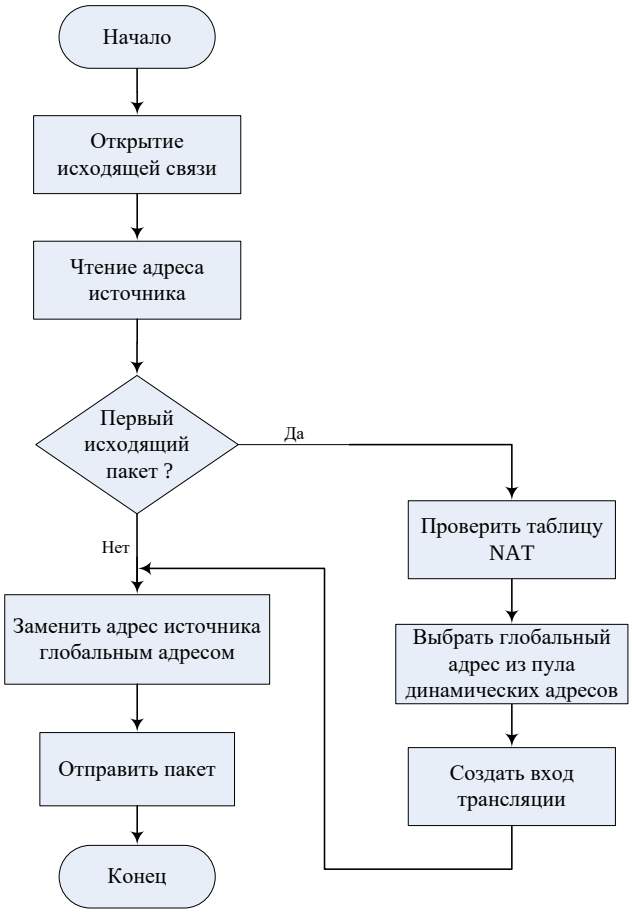


Рисунок 2.6 – Алгоритм трансляции внутреннего адреса источника динамическим методом

Смешанный метод трансляции [2]. Трансляция внутренних адресов источника смешанным методом включает следующие шаги:

1) пользователь на хосте 1.1.1.1 открывает соединение с хостом B;

2) МСЭ получает первый пакет от хоста 1.1.1.1, читает заголовок и сверяется со своей NAT-таблицей;

3) если статический вход трансляции был сконфигурирован, МСЭ следует на шаг 5;

4) если вход трансляции не существует в таблице, МСЭ определяет, что адрес источника 1.1.1.1 должен транслироваться динамическим методом, выбирает легальный глобальный адрес из пула динамических адресов и создаёт вход в таблице трансляции. Этот тип входа называется простым входом;

5) МСЭ заменяет внутренний локальный адрес источника хоста 1.1.1.1 (*SA*=1.1.1.1) на глобальный адрес в соответствии с входом в таблице и отправляет пакет;

6) хост B получает пакет и отвечает хосту 1.1.1.1, используя глобальный адрес назначения 2.2.2.2 (*DA*=2.2.2.2);

7) когда МСЭ получает пакет с внутренним глобальным адресом, он проверяет NAT-таблицу, транслирует адрес во внутренний локальный адрес хоста 1.1.1.1 и направляет пакет хосту 1.1.1.1;

8) хост 1.1.1.1 получает пакет и продолжает диалог с хостом B. Для каждого пакета МСЭ повторяет действия шагов со второго по пятый.

Алгоритм этого механизма представлен на рисунке 2.7.

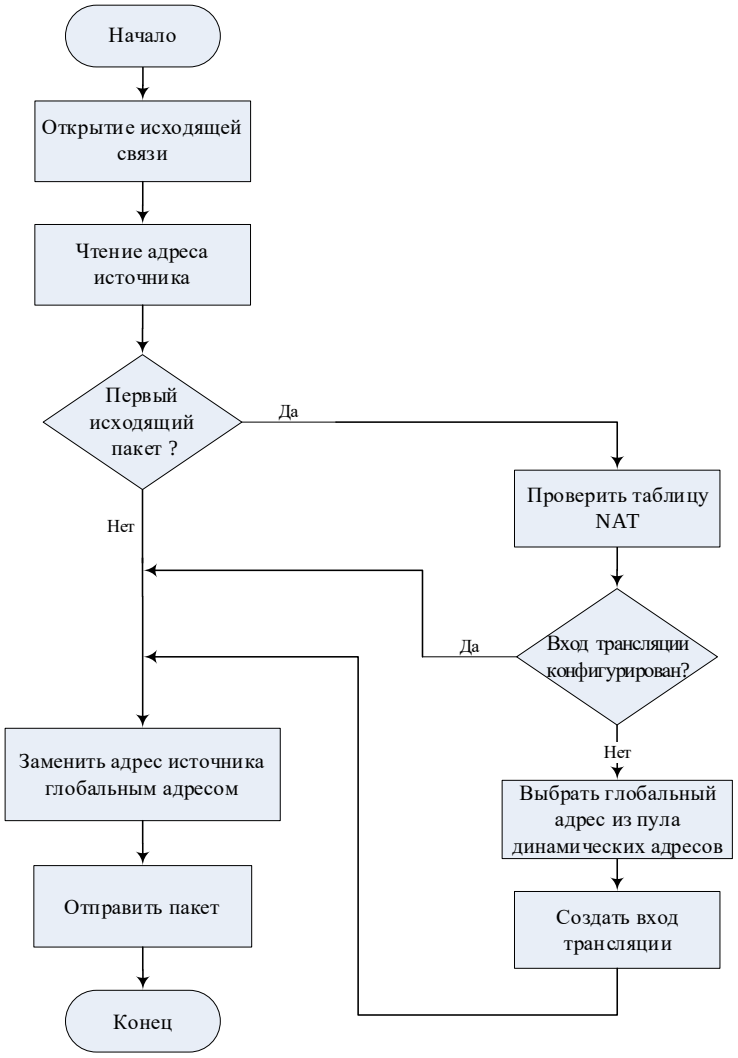


Рисунок 2.7 – Алгоритм трансляции внутреннего адреса источника смешанным методом

2.3 Функция ведения таблицы соединений

МСЭ проверяет и следит за изменением состояния каждого соединения, проходящего через него. Если соединение разрешено (поток трафика разрешён списком доступа), то каждое изменение состояния регистрируется в таблице соединений МСЭ. После начала соединения и выполнения входа в таблице соединений разрешается передача пакетов от источника к получателю. Обратная связь от получателя к источнику через МСЭ также разрешена.

Состояние соединения и движение пакетов от источника к получателю должны соответствовать правилам используемого протокола. При любых отклонениях от разрешённых действий соединение удаляется с регистрацией в журнале. Каждое подключение, заносимое в таблицу соединений, содержит следующие параметры:

– используемый протокол (TCP, UDP или ICMP);

– локальный и глобальный адрес;

– номер локального и глобального порта;

– флаги состояния соединения;

– счётчик времени простоя (увеличивается, если ни один из пакетов не использует соединение);

– счётчик байтов (общий объём трафика, используемый соединением);

– локальный и глобальный порядковый номер.

Поддерживаемое количество сессий МСЭ зависит от модели устройства и установленной лицензии.

2.3.1 Принцип работы функции ведения таблицы соединения

На рисунке 2.8 представлен пример проверки в МСЭ таблицы соединения:

1) пользователь А (ПК-А) внутри сети отправляет HTML-запрос на внешний сервер через МСЭ;

2) МСЭ собирает информацию о пользователе, отправившем запрос (адреса источника и получателя, протокол, номера портов источника и получателя), и сохраняет их в новом входе таблицы соединения;

3) МСЭ направляет HTTP-запрос на целевой веб-сервер.

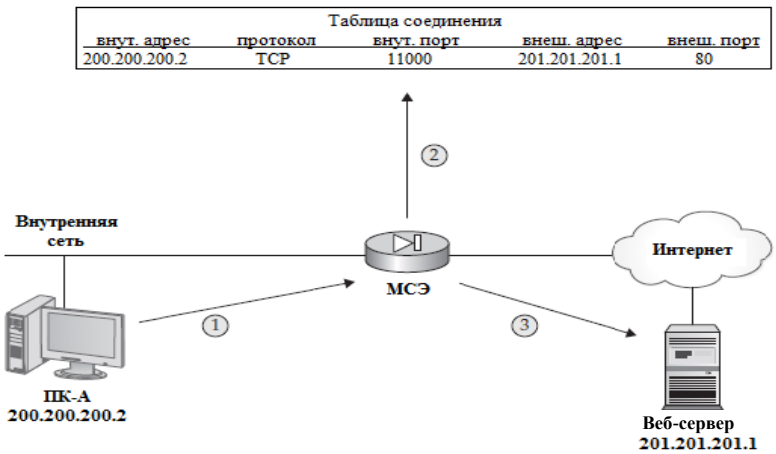


Рисунок 2.8 – Пример проверки таблицы соединений

На рисунке 2.9 показан возвращаемый трафик от HTTP-сервера к пользователю:

1) веб-сервер отправляет соответствующую веб-страницу пользователю;

2) МСЭ принимает ответ HTTP-сервера и сравнивает его с входами таблицы состояния;

3) если в таблице состояния найден соответствующий вход, то получение пакетов разрешено;

4) если в таблице состояния соответствующий вход не найден, то приходящие пакеты отбрасываются.

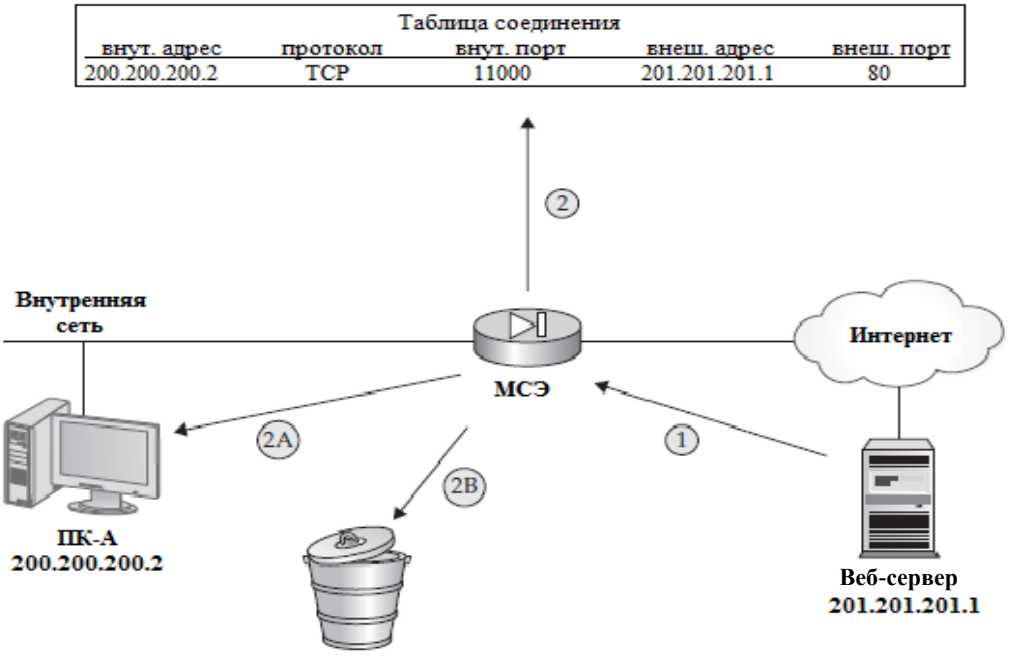


Рисунок 2.9 – Сверка МСЭ возвращаемого трафика с информацией из таблицы соединения

МСЭ сохраняет данную таблицу соединения до обнаружения запроса о прекращении соединения между источником и получателем. При получении данного запроса он удаляет соответствующие данные из таблицы соединения. Если соединение некоторое время не используется и установленное время простоя истекло, то данные о подключении также удаляются из таблицы соединения. На рисунке 2.10 показан алгоритм проверки таблицы соединения в МСЭ.

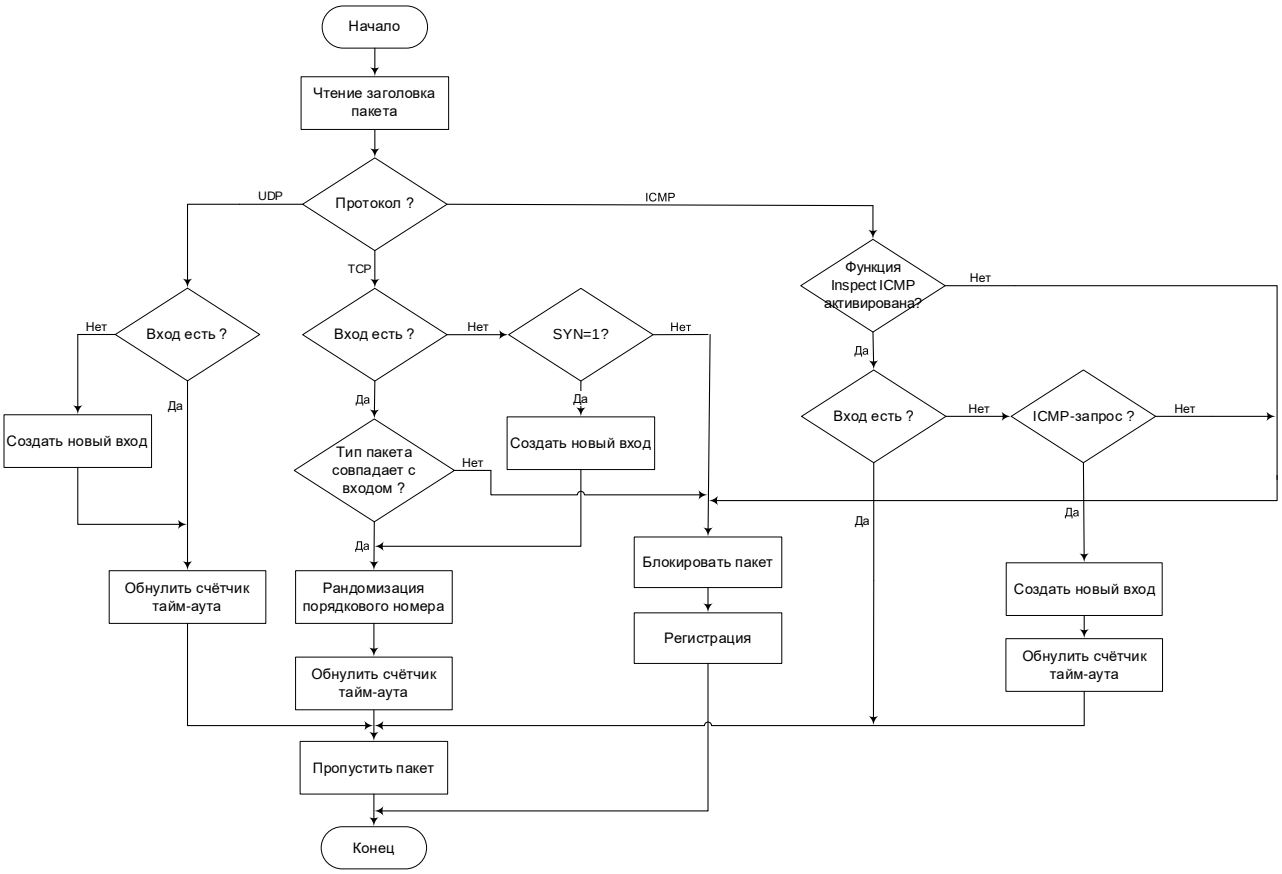


Рисунок 2.10 – Алгоритм функции ведения таблицы соединения в МСЭ

2.3.2 Удаление соединений

Определение момента завершения соединения и удаление его из таблицы состояния зависит от используемого протокола: TCP, UDP или ICMP. Для удаления входящего соединения TCP используются следующие критерии:

– флаги FIN и FIN/ACK в поле управления заголовка TCP;

– флаг RST в поле управления заголовка TCP;

– соединение TCP не используется более 3600 с (одного часа по умолчанию);

– соединение удаляется из таблиц соединения командой clear xlate.

Для протокола UDP используются следующие критерии удаления входа из таблицы состояния:

– соединение UDP не используется более 120 с (двух минут по умолчанию);

– для запроса DNS указывается соответствующий отклик DNS;

– соединение удаляется из таблиц сетевого экрана командой clear xlate.

Для протокола ICMP используются следующие критерии удаления входа из таблицы состояния:

– соединение ICMP не используется более 2 с (по умолчанию);

– соединение удаляется из таблиц сетевого экрана командой clear xlate.

Проверка соединения является современной и самой эффективной функцией, используемой МСЭ для защиты локальных сетей. Проверка соединения

обеспечивает прохождение только пакетов, являющихся частью одного соединения, пропуская их потоками. Таким образом, функция проверки соединения сокращает общее время обработки пакетов МСЭ и повышает пропускную способность МСЭ.

2.4 Алгоритм управления доступом

Управление доступом по спискам (Access Control List (ACL)) – одно из важнейших средств организации базовой безопасности сетей. Требование безопасности особенно актуально в локальных сетях, когда они связаны с Интернетом через МСЭ.

Списки доступа являются управляемыми фильтрами для проходящего трафика и при соответствующей настройке один трафик они могут беспрепятственно пропускать дальше, другой – блокировать (подавлять, отбрасывать). ACL устанавливаются на интерфейсах МСЭ, подавляя непредусмотренный сетевой трафик из Интернета в корпоративную сеть и наоборот и обеспечивая защиту периметра корпоративной сети. При этом критерием для фильтрации могут быть значения IP-адреса источника и назначения проходящих пакетов сетевого уровня, имена протоколов более высокого уровня, номера TCP- и UDP-портов и другие параметры.

Существует два вида списков доступа: стандартные (standard) и расширенные (extended). Стандартные ACL более простые и содержат в правилах только адрес источника пакета, а расширенные – как адрес отправителя (источника), так и адрес назначения, а также множество других контролируемых параметров.

В целом список доступа ACL представляет собой упорядоченный набор из одного или нескольких правил, используемых для сравнения с параметрами проходящих пакетов через данный интерфейс. Синтаксически отдельное правило – это одна строка, каждая строка называется входом управления доступом (Access Control Entry (ACE)). Рисунок 2.11 иллюстрирует конфигурации ACE.

Настройка списков доступа всегда состоит из двух этапов:

1) определения ACL, т. е. написания правил для сравнения;

2) активизации определённого ACL на заданном интерфейсе МСЭ.

Пока второй шаг не выполнен, списки доступа никакого влияния на фильтрацию пакетов не оказывают. Для каждого пакета список ACL просматривается заново, последовательно, начиная с первого правила:

– если параметры пакета не совпадают с параметрами данного правила, то правило игнорируется и рассматривается следующее по порядку правило;

– если параметры пакета правилу удовлетворяют, то выполняется одно из запрограммированных в правиле действий – разрешить (ключевое слово permit) прохождение пакета дальше или блокировать (deny) прохождение. При этом последующие правила в ACL для этого пакета уже не рассматриваются.

С целью большей безопасности в конце каждого списка ACL присутствует скрытое правило – «запретить все». Ниже приведен алгоритм проверки ACL (рисунок 2.12).

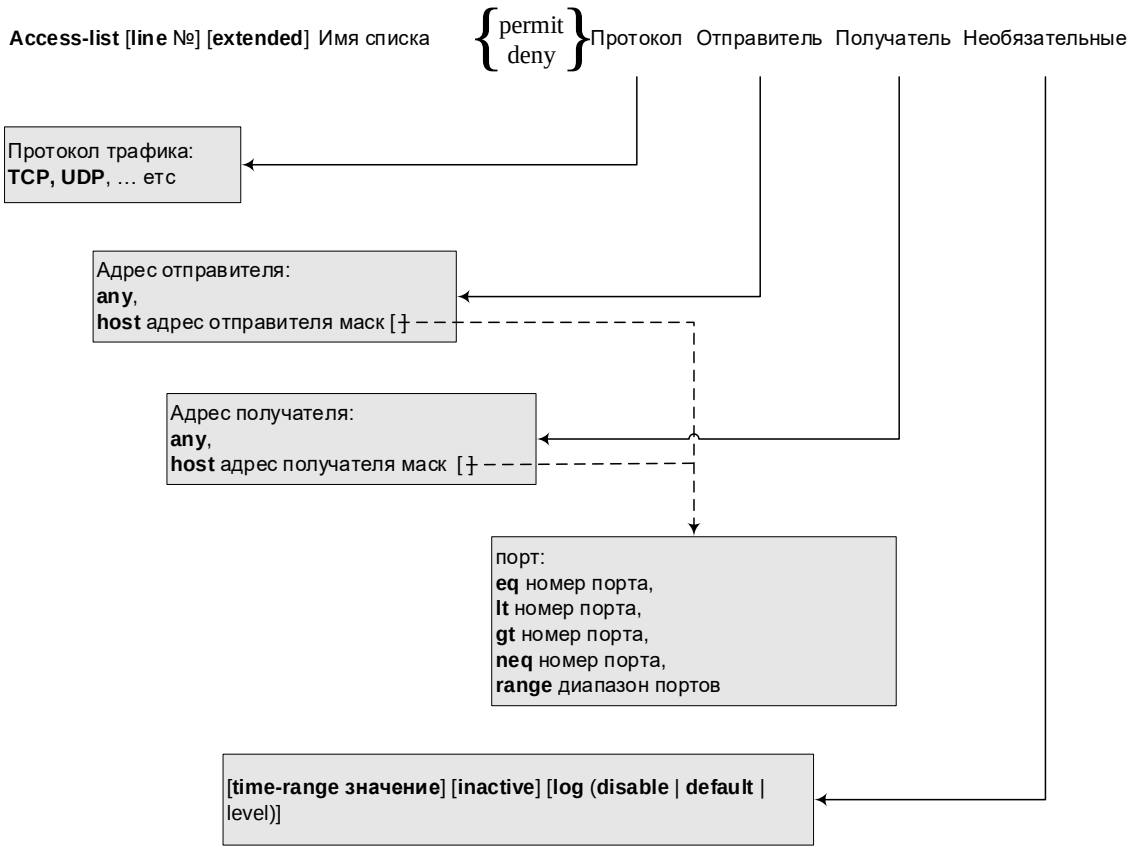


Рисунок 2.11 – Конфигурация ACE

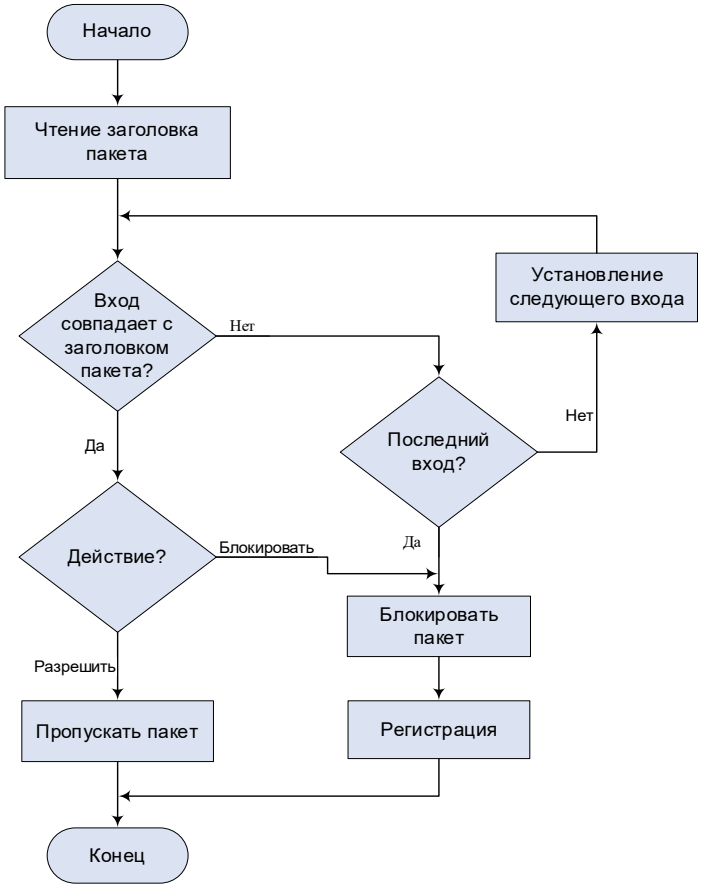


Рисунок 2.12 – Пример алгоритма проверки правил ACL

2.5 Инспектор состояния

2.5.1 Инспектирование протокола ICMP

Межсетевой протокол управляющих сообщений (Internet Control Message Protocol (ICMP)) – это некоммутируемый протокол, который позволяет одному хосту посылать сообщения другому хосту, не ожидая ответа. Вследствие этого МСЭ не может рассматривать или отслеживать состояние трафика между двумя хостами.

МСЭ проверяет трафики ICMP в зависимости от содержания таблицы трансляции и списков управления доступом (учитывая, что никакие соединения не используются в ICMP, для трафиков ICMP не создаются никакие входы соединений).

На рисунке 2.13 показано движение ICMP-пакетов между двумя хостами на разных интерфейсах МСЭ. Хост 1 посылает ICMP-пакет хосту 2. МСЭ необходим вход таблицы трансляции для одного или нескольких хостов. В зависимости от конфигурации вход осуществляется статическим или динамическим методом. ICMP-пакет также должен быть разрешён любым списком управления доступом, который применён к интерфейсу МСЭ.

Рисунок 2.13 – Прохождение ICMP-трафика через МСЭ

Как только вход таблицы трансляции создан и списки управления доступом разрешают прохождение трафика, два хоста могут свободно посылать ICMP-пакеты друг другу. Фактически другие хосты также могут посылать им ICMP-пакеты, если вход таблицы трансляции существует и список управления доступом разрешает трафик.

Если используется NAT, МСЭ позволяет соединениям ICMP оставаться открытыми в течение двух секунд после единственного ответа пакета ICMP. Для PAT условия немного отличаются: ICMP-соединение немедленно закрывается после первого ответа пакета. Алгоритм инспектирования протокола ICMP представлен на рисунке 2.14. В рамках проверки трафика ICMP МСЭ разрешает только один ответ на любой запрос ICMP: любой возвращаемый трафик после первого ответа блокируется; вход трансляции ICMP может оставаться активным до окончания времени простоя.

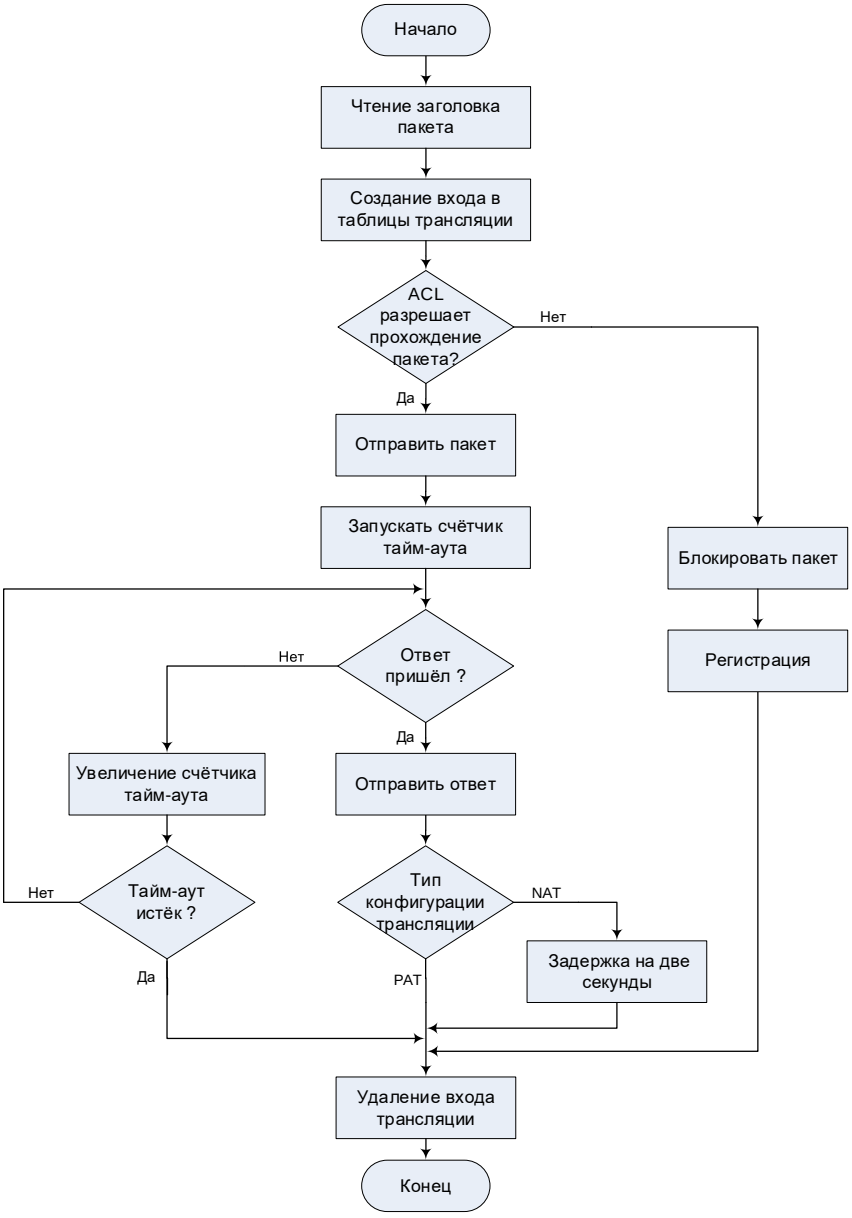


Рисунок 2.14 – Алгоритм инспектирования протокола ICMP

2.5.2 Инспектирование протокола UDP

Протокол пользовательских дейтаграмм (User Datagram Protocol (UDP)) также некоммутируемый протокол. Хост может посылать незапрашиваемые UDP-пакеты другому хосту, не ожидая ответа. Однако некоторые протоколы, например, DNS (Domain Name System), используют UDP для двухстороннего информационного обмена.

Для большинства трафика UDP МСЭ не может рассмотреть или отследить состояние информационного обмена. Пакеты UDP проверяются при помощи таблицы трансляции, списков управления доступом и входов таблицы соединения. Даже притом, что UDP является некоммутируемым протоколом, МСЭ создаёт входы соединения, поскольку пары хостов отправляют UDP-пакеты друг другу.

Хост 1 начинает сессию, посылая UDP-пакет хосту 2 через МСЭ (рисунок 2.15). Если списки управления доступом разрешают трафик, то МСЭ продолжает определять UDP-соединение. Для отправления пакетов МСЭ нужно создать существующий вход таблицы трансляции, если такового нет.

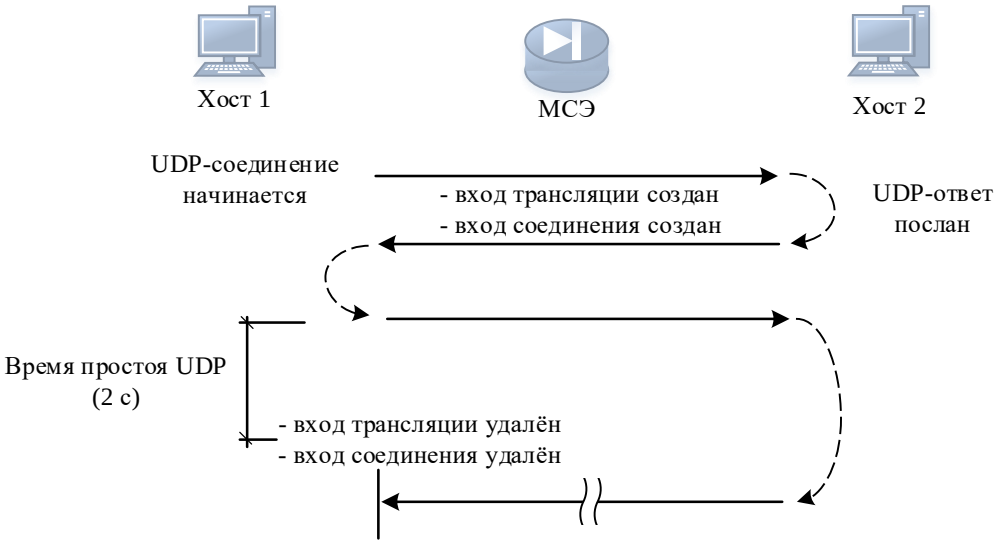


Рисунок 2.15 – Управление UDP-трафиком посредством МСЭ

Алгоритм инспектирования протокола UDP представлен на рисунке 2.16.

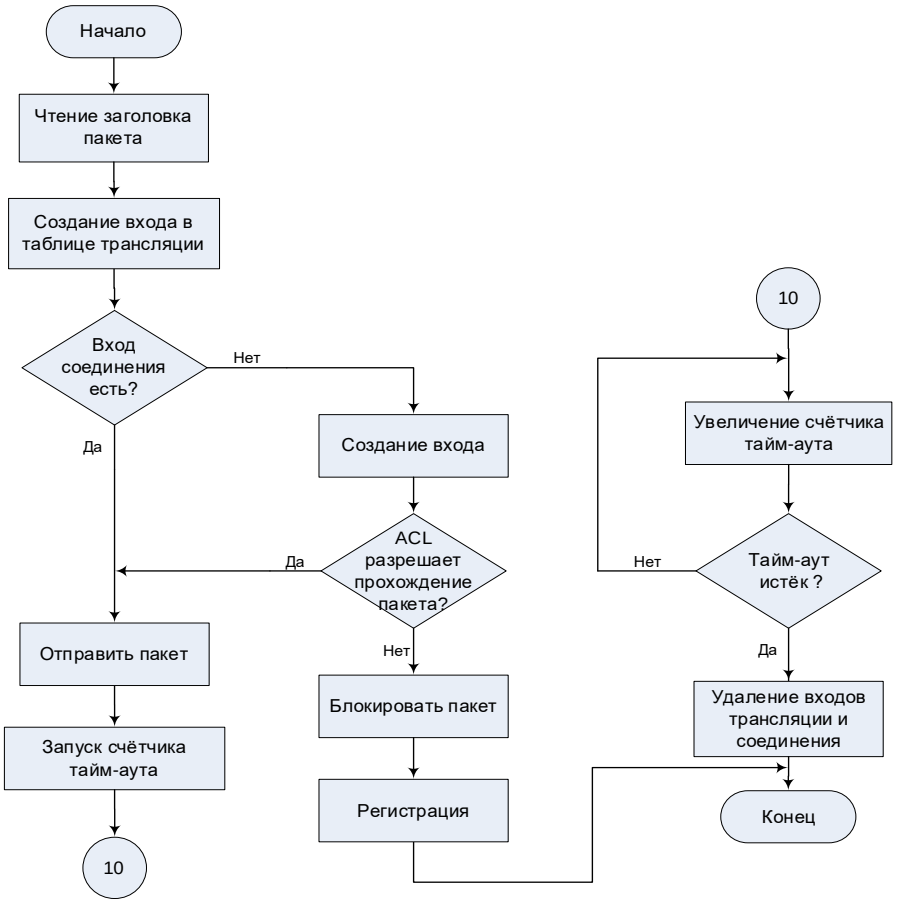


Рисунок 2.16 – Алгоритм инспектирования протокола UDP

С первым пакетом на сессии МСЭ создаёт новый вход в таблице соединения. Этот вход идентифицирует адреса источника и назначения и UDP-порты так, чтобы все пакеты, проходящие между парой хостов, могли идентифицироваться с этим определённым соединением. Теперь UDP-пакеты могут проходить между хостом 1 и хостом 2. МСЭ поддерживает связь столько, сколько нужно для прохождения пакетов через соединение. Если пакеты не прошли через соединение до истечения времени простоя UDP (2 мин по умолчанию), то вход удаляется из таблицы, соединение и связь закрываются. Это означает, что UDP-связи никогда сами не закрываются, потому что у них нет предусмотренного для этого механизма. Поэтому любые UDP-связи, созданные МСЭ, должны ожидать истечения времени простоя.

2.5.3 Инспектирование протокола TCP

Протокол управления передачей (Transmission Control Protocol (TCP)) является коммутируемым протоколом. Прежде чем два хоста смогут обменяться пакетами, они должны выполнить обмен сообщениями, чтобы установить TCP-соединение. После того как обмен пакетами произошёл, состояние связи всегда обновляется. Чтобы закрыть TCP-соединение, хосты должны выполнить модифицированный обмен сообщениями.

МСЭ может отследить состояние информационного обмена в любой момент времени, т. к. TCP является коммутируемым протоколом. Для каждого TCP-соединения МСЭ исследует адреса и порты источника и назначения, порядковый номер TCP (TCP sequence number), число подтверждений (acknowledgment value) и TCP-флаги. Пакеты, у которых есть неожидаемые числа, не могут быть частью существующего соединения, и МСЭ блокирует их.

TCP-соединения проверяются при помощи таблицы трансляции, списков управления доступом и таблицы соединения. У входов таблицы соединения также есть флаги, которые показывают текущее состояние соединения. На рисунке 2.17 показано, как МСЭ устанавливает TCP-трафик между двумя хостами на разных интерфейсах. Хост 1 начинает TCP-связь, отправляет хосту 2 пакет с SYN-флагом через МСЭ. МСЭ создаёт динамический вход трансляции, если такового нет. Также создаётся новый вход соединения для TCP-связи между этой парой хостов. МСЭ ожидает, что хост 2 ответит пакетом, у которого есть набор битов SYN и ACK.

На этом этапе соединение только полуоткрытое (не полностью сформированное). Если ожидаемый ответ не получен в течение 30 с, то время простоя истекает и соединение закрывается.

Наконец хост 1 заканчивает обмен сообщениями, посылая пакет с флагом ACK. МСЭ позволяет TCP-пакетам проходить через соединение. Каждый из пакетов проверяется, их параметры обновляются. Когда TCP-связь установлена, хост 1 посылает первый SYN-пакет с числом первого порядкового номера ISN (Initial Sequence Number) с тем, чтобы другой хост знал, как ответить. Число ISN

иногда предсказуемо, что в некоторых случаях может быть использовано злоумышленниками для захвата соединения. Алгоритм инспектирования протокола TCP представлен на рисунке 2.18.

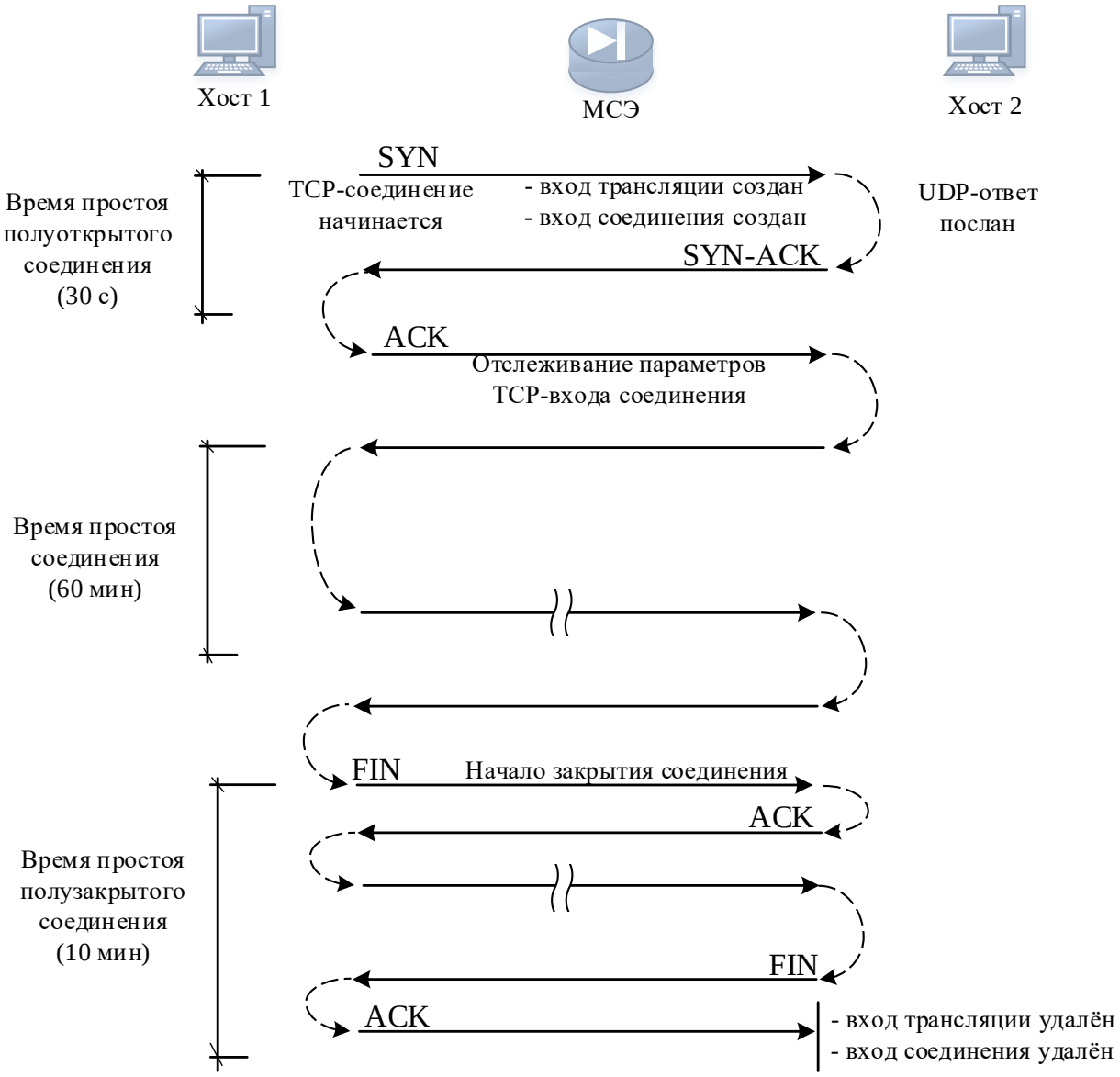


Рисунок 2.17 – Установление TCP-трафика межсетевым экраном

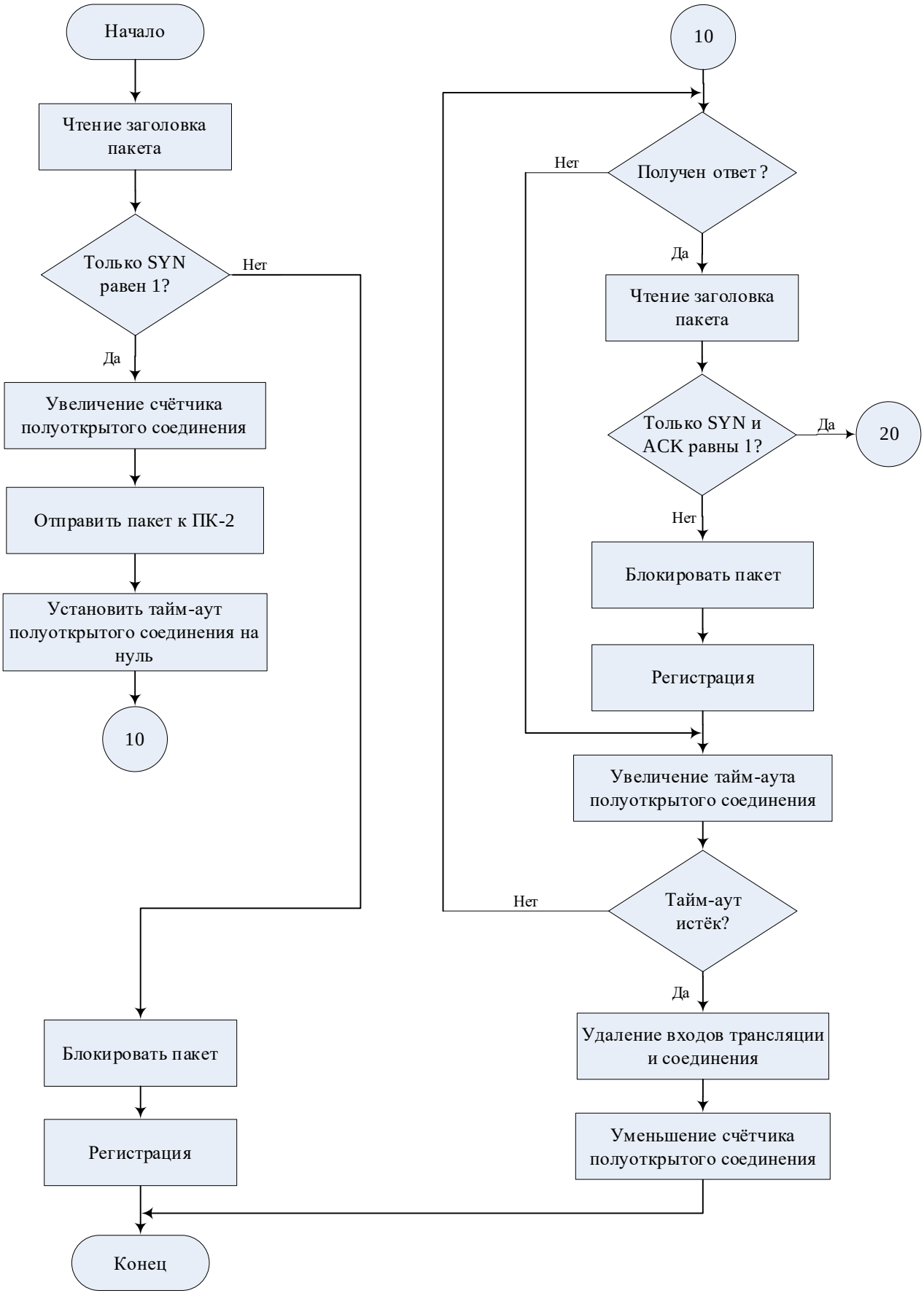


Рисунок 2.18 – Алгоритм инспектирования протокола TCP

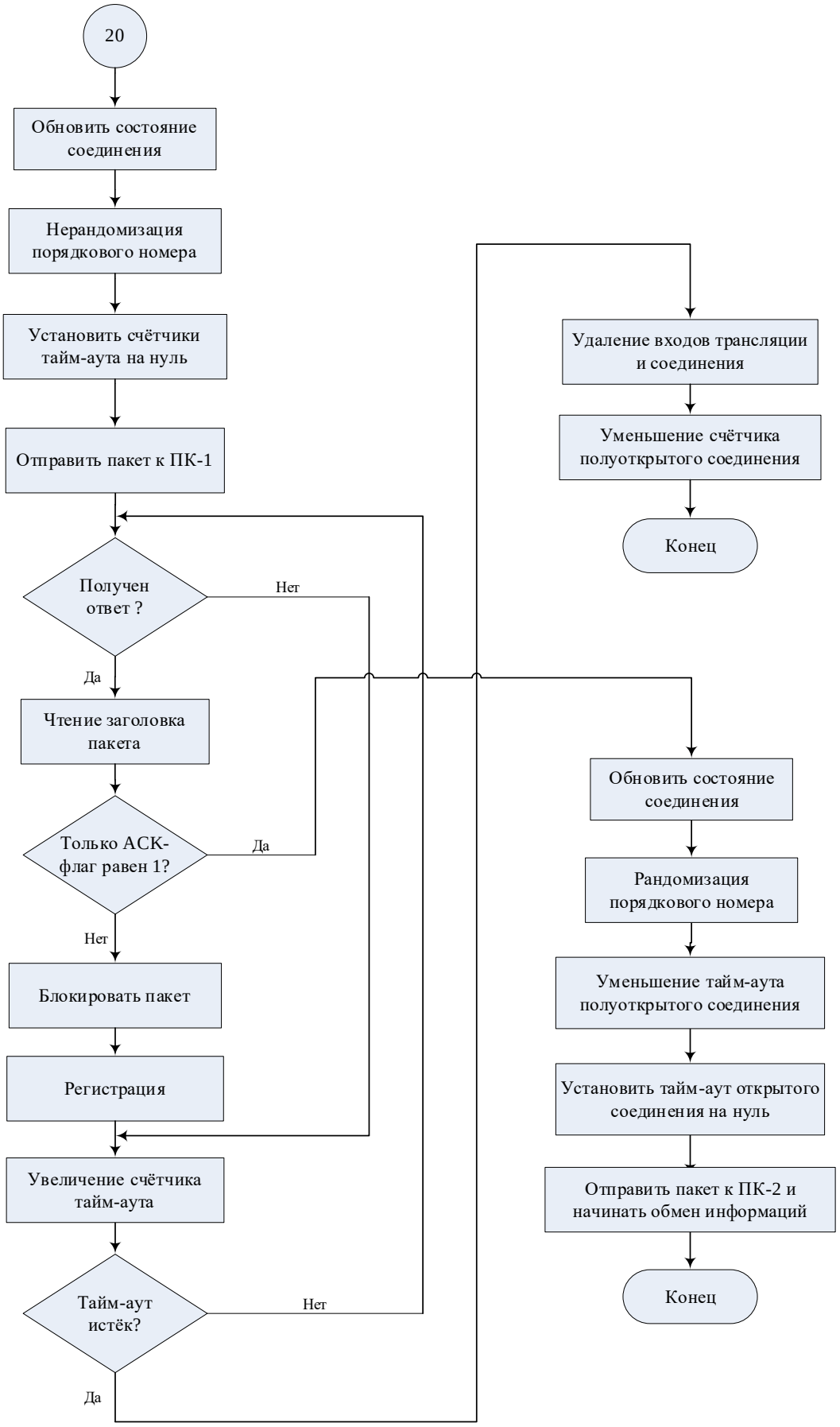


Рисунок 2.18, лист 2

2.6 Экспертная проверка контента

Используемые в МСЭ списки управления доступом обеспечивают фильтрование пакетов только на третьем и четвёртом уровнях OSI, но не могут фильтровать информацию в содержании пакетов. Например, хакер может создать Java-апплеты или ActiveX-скрипты, которые пользователи будут скачивать и выполнять. Эти приложения скачиваются протоколом HTTP (порт 80). С помощью списков управления доступом можно только пропустить или заблокировать весь трафик, проходящий через порт 80, который содержит апплеты, вставленные в сообщение, но нельзя фильтровать каждый апплет сам по себе.

Для решения данной задачи в МСЭ используется два механизма:

– фильтрование Java и ActiveX;

– фильтрование содержаний веб-страниц.

2.6.1 Реализация фильтра Java и ActiveX

МСЭ может фильтровать Java-апплеты и ActiveX-скрипты без использования дополнительных программных или аппаратных средств. Обычно, МСЭ проверяет и ищет HTML <object>-команды и заменяет их комментариями. Некоторые команды <object> содержат <applet>, <object> и <object classid>.

Эта операция запрещает загрузку злоумышленных апплетов и скриптов, когда пользователи скачивают веб-страницы. Процесс конфигурирования межсетевого экрана для фильтрования Java-апплетов и ActiveX-скриптов осуществляется следующим образом.

В межсетевом экране настройка фильтра на Java-апплеты осуществляется командой filter Java. Когда эта команда выполнена, процесс проверки автоматически активируется на всех интерфейсах, и все трафики, проходящие через МСЭ, проверяются на Java-апплеты. Например, в МСЭ Cisco ACA следующая команда устанавливает фильтр всех Java-апплетов HTTP соединений:

```
ciscoasa(config)# filter Java 80 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

или

```
ciscoasa(config)# filter Java http 0 0 0 0
```

Кроме фильтрования Java-апплетов, МСЭ может фильтровать ActiveX-скрипты путём установки команды filter activex. Синтаксис и метод работы этой команды похож на синтаксис и метод работы команды фильтрования Java. Следующая команда устанавливает фильтр всех ActiveX-скриптов:

```
ciscoasa(config)# filter activex 80 0 0 0 0
```

или

```
ciscoasa(config)# filter activex http 0 0 0 0
```

2.6.2 Процесс фильтрования веб-страниц

Для фильтрования содержания веб-страниц (веб-фильтрование) необходимо реализовать две функции:

– определить политику фильтрования;

– выполнить политику фильтрования.

Для выполнения этих функций используется два метода:

– прокси-приложение;

– модифицированный прокси.

В методе прокси-приложения две функции (определение и выполнение политики) выполняются в одном сервере. Веб-браузеры пользователей сконфигурированы таким образом, чтобы они обращались к прокси или их трафики переадресовывались на прокси.

В этом методе, когда пользователь скачивает веб-страницу, выполняются следующие шаги:

1) пользователь открывает веб-страницу;

2) все трафики переадресовываются в прокси-сервер приложения, который проверяет подлинность пользователя до того, как предоставить внешний доступ;

3) прокси-сервер проверяет соединение и сравнивает его с сконфигурированной политикой;

4) если соединение запрещено, то пользователю обычно показывается веб-страница о нарушении правил;

5) если соединение разрешено, то прокси открывает необходимые соединения для скачивания содержимого веб-страницы.

После этого содержимое отправляется пользователю через отдельное соединение и отображается в его веб-браузере. Приложения прокси хорошо работают на маленьких предприятиях, где одновременно скачивается немного веб-страниц. Когда число скачиваемых веб-страниц пользователями увеличивается, то пропускная способность через прокси уменьшается, т. к. для каждого соединения пользователя прокси выполняет два соединения: от пользователя до прокси и от прокси до внешнего веб-сервера. Этот процесс может быстро привести к перегрузке прокси, т. к. процессор и память постоянно заняты.

В модифицированном прокси политики фильтрации разделены: внешний сервер ведёт список правил доступа, а сетевое средство выполняет эти правила, когда через него проходит веб-трафик. МСЭ, который поддерживает метод модифицированного прокси для фильтрования содержания веб-страниц, должен взаимодействовать с внешним сервером веб-контента.

На рисунке 2.19 показано взаимодействие между пользователями, МСЭ, сервером правил и внешним веб-сервером. В этом случае выполняются следующие шаги:

1) пользователь отправляет HTML-запрос к внешнему веб-серверу через МСЭ;

2) МСЭ выполняет две задачи:

– отправку HTML-запроса (только информацию об URL) к серверу политики веб-контента;

– отправку HTML-запроса к внешнему веб-серверу;

3) сервер политики веб-контента сравнивает URL-запрос с внутренними политиками и отправляет результат к МСЭ;

4) внешний веб-сервер отправляет веб-страницу к МСЭ;

5) МСЭ реализует политику фильтрации на возвращаемом трафике в зависимости от полученного ответа от сервера политики веб-контента. Если сервер запретил трафик, то МСЭ блокирует возвращаемый трафик, иначе МСЭ разрешает прохождение разрешённого трафика к внутреннему пользователю.

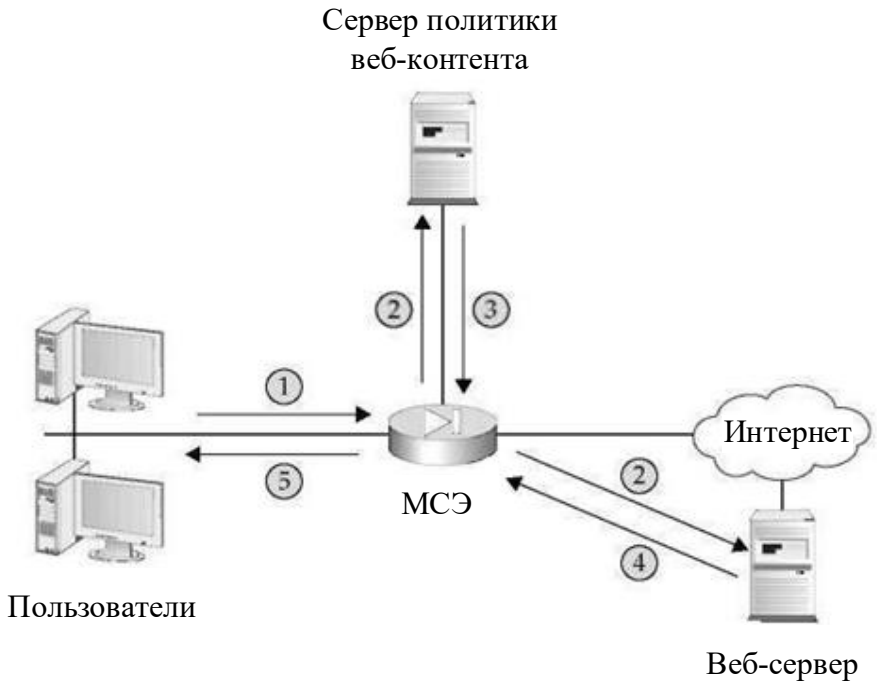


Рисунок 2.19 – Фильтрование содержания веб-страниц

Как видно из данного примера, МСЭ не осуществляет в реальном времени фильтрование открытого соединения. В этом процессе серверу контроля политики веб-контента вполне достаточно времени, чтобы проверить правила до прихода ответа от веб-сервера.

3 АДМИНИСТРИРОВАНИЕ МЕЖСЕТЕВОГО ЭКРАНА

Администрирование МСЭ проводится с целью настройки всех функций, которые предполагается задействовать при его использовании. Так как на практике используются МСЭ разного исполнения, рассмотрим администрирование наиболее типового МСЭ на примере межсетевого экрана Cisco ASA 5520 [1, 11].

3.1 Интерфейсы, используемые при конфигурировании МСЭ

Управление и конфигурирование МСЭ может осуществляться тремя способами:

1 Через консоль управления (рисунок 3.1).

В появившемся окне необходимо задать название соединения и выбрать любой значок. Далее необходимо указать, через какой порт осуществляется соединение. В данном случае это будет либо COM1, либо COM2, в зависимости от какого порта на ПЭМВ осуществляется подключение.

2 Через интерфейс GUI (рисунок 3.2).

Для того чтобы осуществить первичное подключение к устройству, необходимо включить питание МСЭ и подключить в сеть данное устройство через management port – порт управления (ПУ) на задней панели устройства. По умолчанию ПУ имеет IP-адрес 192.168.1.1. В поле адреса интернет-браузера необходимо написать https://192.168.1.1. После чего будет предложено принять сертификат от МСЭ. После принятия сертификата откроется SSL-соединение между МСЭ и ПЭВМ, с которой осуществляется конфигурирование.

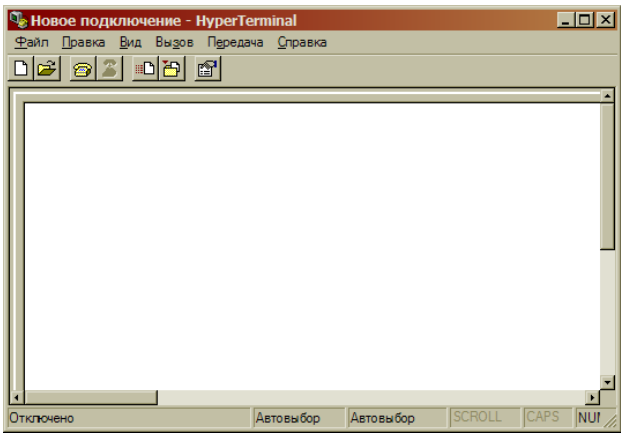


Рисунок 3.1– Вид окна программы Hyper Terminal

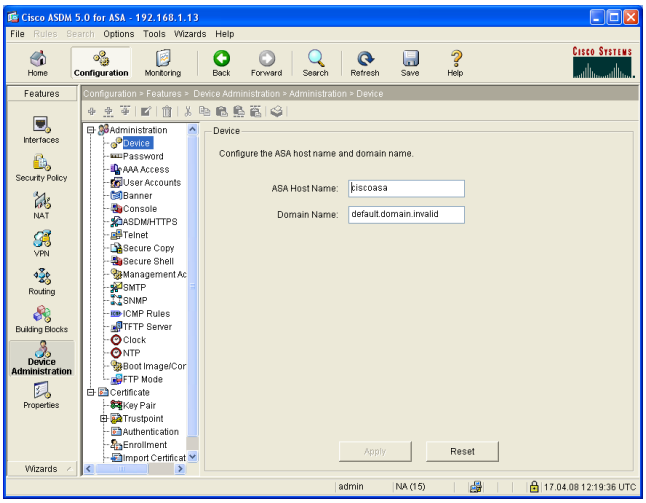


Рисунок 3.2 – Интерфейс GUI панели конфигурации МСЭ

3. Через интерфейс CLI (Comand Line Interface – интерфейс командной строки) (рисунок 3.3).

Доступ осуществляется через ПУ МСЭ, как и в случае с GUI. Для начала работы необходимо открыть командную строку Windows (**Пуск → Выполнить**.

В появившемся окне написать «cmd» и нажать клавишу **Enter**). Далее необходимо написать команду «telnet» и IP-адрес ПУ МСЭ – «telnet 192.168.1.1».

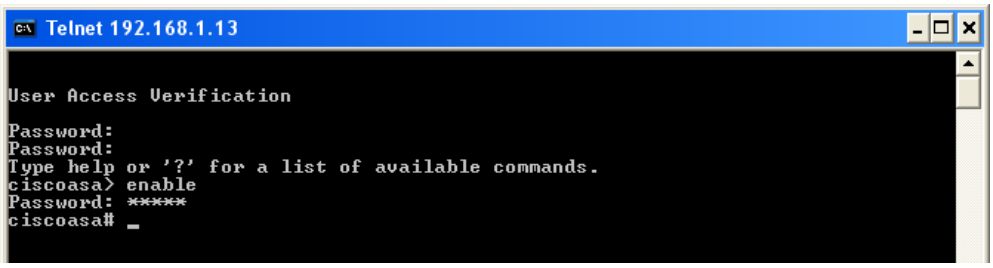


Рисунок 3.3 – Командная строка Windows

Дальнейшее рассмотрение настройки МСЭ будет осуществляться с применением последнего варианта подключения через CLI-интерфейс ПУ МСЭ. Подключившись к ПУ МСЭ, необходимо ввести имя пользователя и пароль, установленные по умолчанию.

При работе в командной строке различают несколько режимов доступа со своим набором команд. Определить, в каком режиме вы находитесь, можно по значку в начале строки консоли:

1) режим пользователя (>);
2) режим привилегированного пользователя (#);
3) режим конфигурирования (config);
4) режим детального конфигурирования (config-xxx).

Вход в привилегированный (#) режим из пользовательского (>) осуществляется командой enable, вход в конфигурационный (config) режим из привилегированного – командой configure termital, а сохранение конфигурации – командой write memory в привилегированном (#) режиме. Команда exit осуществляет переход на предыдущий режим.

3.2 Начальное конфигурирование Cisco ASA 5520

Конфигурирование Cisco ASA 5520 состоит из следующих шагов:

1 Задание имени устройства, подключённого в сеть.

В сети с большим многообразием сетевого оборудования необходимо давать уникальные имена любым устройствам. Для этого необходимо с помощью команды hostname осуществить следующее:

```
Cisco-ASA (config)# hostname ciscoasa
ciscoasa(config)#
```

2 Установка пароля для доступа к режиму администрирования Cisco.

Для того чтобы установить пароль доступа к настройкам МСЭ на команду «enable», необходимо ввести следующее:

```
ciscoasa(config)# enable password ******
```

3 Создание привилегированного пользователя.

Для того чтобы создать привилегированного пользователя, необходимо ввести следующее:

```
ciscoasa(config)# username admin password ***** encrypted privilege 15
```

4 Настройка интерфейсов.

На задней панели Cisco ASA 5520 находится четыре интерфейса с пропускной способностью 1 Гбит/с и один ПУ. Каждый интерфейс имеет уникальное имя (GigabitEthernet0/0, GigabitEthernet 0/1, GigabitEthernet 0/2, GigabitEthernet 0/3, Management0/0).

Для конфигурирования интерфейсов необходимо ввести следующие команды:

```
ciscoasa(config)# interface GigabitEhernet 0/X
ciscoasa(config-if)#
```

Здесь X – номер интерфейса.

По умолчанию все интерфейсы выключены. Их необходимо включить командой no shutdown:

```
interface{physical_interface[subinterface] | mapped_name}
```

5 Задание имени интерфейсу.

Команда nameif даёт имя интерфейсу на устройстве защиты:

```
ciscoasa(config-if)# nameif xxxxx
```

Здесь xxxxx – любое имя без пробелов, состоящее из латинских букв.

6 Присвоение IP-адреса.

Всем используемым интерфейсам МСЭ необходимо присвоить уникальный IP-адрес. Если при задании IP-адреса допущена ошибка, необходимо ввести команду заново. Командой clear configure ip сбрасываются IP-адреса на всех интерфейсах.

Для присвоения IP-адреса необходимо в режиме настройки интерфейса (см. пункт 4) выполнить следующую команду:

```
ciscoasa(config-if)# ip address xxx.xxx.xxx.xxx yyy.yyy.yyy.yyy
```

Здесь xxx.xxx.xxx.xxx – IP-адрес интерфейса, yyy.yyy.yyy.yyy – маска.

7 Установка уровня безопасности (Security level).

Значение уровня безопасности (УБ) «100» (максимальное) означает, что из этого сегмента сети по умолчанию доступны все сети, у которых УБ ниже 100.

Если у интерфейса значение УБ «0», то из сети, подсоединённой к этому интерфейсу, по умолчанию отсутствует доступ к интерфейсу с УБ больше нуля. Таким образом, с помощью задания на МСЭ УБ можно разделить коммутируемые сети по уровню доступа.

Для настройки маршрутизации и уровней безопасности каждого из интерфейсов МСЭ необходимо в соответствии с рисунком 3.4 произвести следующие операции:

```
ciscoasa# configure terminal
ciscoasa (config)# interface GigabitEthernet0/0
ciscoasa (config-if)# nameif outside
ciscoasa (config-if)# ip address 80.94.232.94
ciscoasa (config-if)# security-level 0
ciscoasa (config-if)#exit
ciscoasa (config)# interface GigabitEthernet0/1
ciscoasa (config-if)# nameif inside
ciscoasa (config-if)# ip address 10.7.7.253
ciscoasa (config-if)# security-level 100
ciscoasa (config-if)#exit
ciscoasa (config)# interface GigabitEthernet0/2
ciscoasa (config-if)# nameif dmz
ciscoasa (config-if)# ip address 10.8.8.1
ciscoasa (config-if)# security-level 50
ciscoasa (config-if)#exit
```

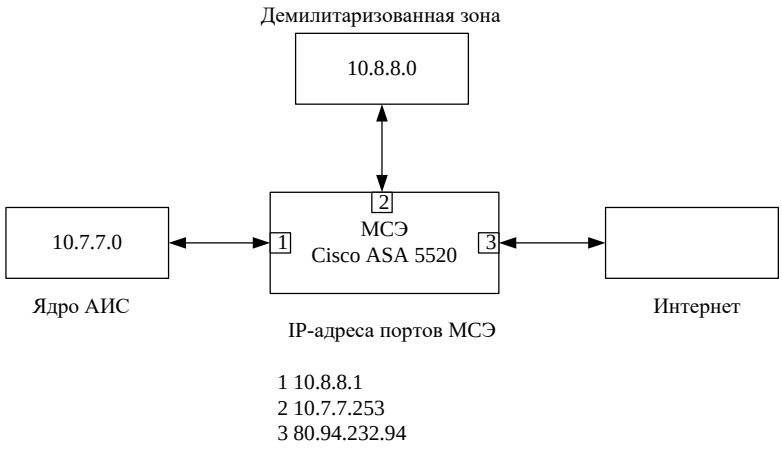


Рисунок 3.4 – Схема подключения устройств к МСЭ

8 Настройка NAT/PAT (Network Address Translation (трансляция сетевых адресов)/Port Address Translation (трансляция адреса портов)).

При прохождении пакетов через МСЭ внутренние адреса одной сети перед выходом из внешнего интерфейса транслируются в адреса другой сети. NAT/PAT конфигурируется с помощью команд nat, global и static.

Таким образом, необходимо, чтобы трафик из сети 10.7.7.0 транслировался в DMZ 10.8.8.0. В свою очередь трафик из сети 10.8.8.0 должен транслироваться в сеть Интернет. Для этого необходимо выполнить следующие команды:

```
ciscoasa (config)# nat (inside) 1 10.7.7.0 255.255.255.0
ciscoasa (config)# global (dmz) 1 interface
ciscoasa (config)# nat (dmz) 2 10.8.8.0 255.255.255.0
ciscoasa (config)# global (outside) 2 interface
ciscoasa (config)# nat (outside) 3 80.94.232.0 255.255.255.0
ciscoasa (config)# global (dmz) 3 interface
ciscoasa (config)# global (inside) 2 interface
```

Команда nat (inside) 1 10.7.7.0 255.255.255.0, где (inside) – имя интерфейса, 1 – идентификационный номер правила, 10.7.7.0 – подсеть ядра, 255.255.255.0 – маска, необходима для того, чтобы все сетевые адреса транслировались при выходе из интерфейса в другие адреса, описываемые командой global.

Команда global (dmz) 1 interface реализует правило, описанное в команде nat (dmz) 2 10.8.8.0 255.255.255.0, где nat (dmz) – имя интерфейса, на который должен поступать трафик с интерфейса (inside), 1 – идентификатор, связывающий с собой правило, описанное с помощью команды nat, interface означает, что весь трафик, исходящий от любого хоста в сети 10.7.7.0, будет нести в себе IP-адрес интерфейса и в качестве обратного идентификатора – порт.

Пример:

1) хост с адресом 10.7.7.5 генерирует HTTP-запрос серверу с адресом 10.8.8.2, находящийся в DMZ;

2) при переходе пакетов через интерфейс GigabitEthernet0/1, имеющим IP-адрес 10.7.7.253, и выходе из GigabitEthernet0/2, имеющим IP-адрес 10.8.8.1, IP-адрес посылаемых пакетов становится 10.8.8.1:port, где port – имя порта, зарезервированного в МСЭ. Порт необходим для того, чтобы отличить одного адресата из сети 10.7.7.0 от другого.

После настройки команд global для проверки их работы необходимо очистить таблицу трансляций, сгенерированную на основании предыдущих настроек. Для этого необходимо выполнить следующие действия:

```
ciscoasa (config)# clear xlate
```

9 Настройка DHCP-сервера.

МСЭ может выступать в роли DHCP-сервера. Это требуется, если IP-адреса контролируемой сети необходимо раздавать динамически. Для этого необходимо выполнить ряд шагов:

– активировать DHCP-сервер:

```
ciscoasa (config)# dhcp enable inside
```

– определить пул адресов, выдаваемых клиентам:

```
ciscoasa (config)# dhcp address 10.7.7.100-10.7.7.150 inside
```

– установить тайм-ауты DHCP-сервера:

```
ciscoasa (config)# dhcpd ping_timeout 50
ciscoasa (config)# dhcpd lease 3600
```

– определить дополнительные DHCP-опции, а именно указание шлюза (option 3) при выдаче IP-адреса хосту и DNS-сервера (option 6):

```
ciscoasa (config)# dhcpd option 3 ip 10.7.7.253
ciscoasa (config)# dhcpd option 6 ip 80.94.225.5 80.94.225.254
```

10 Настройка маршрутного пути во внешнюю сеть.

Для того чтобы пакеты из внутренней сети попадали во внешнюю, необходимо на МСЭ прописать маршрут, а именно шлюз, на который должен направляться весь исходящий трафик:

```
ciscoasa (config)# route outside 0.0.0.0 0.0.0.0 80.94.232.65 1
```

Здесь route – команда, означающая маршрут, outside – исходящий интерфейс «0.0.0.0 0.0.0.0», который означает, что это правило применимо для исходящего трафика, имеющего любой IP-адрес, 80.94.232.65 – IP-адрес шлюза провайдера, предоставляющего выход во внешнюю сеть, 1 – метрика.

11 Настройка аудита событий на базе Syslog-сервера.

МСЭ позволяет сохранять и пересылать данные аудита следующими способами:

– через встроенную память МСЭ;

– с помощью непосредственного отображения событий через службу терминала;

– по электронной почте;

– через консоль;

– через GUI-интерфейс ASDM;

– с помощью Syslog-сервера;

– с помощью SNMP NMS.

Для настройки аудита событий на одном из серверов должен быть установлен Syslog-сервер, который будет выполнять функции сбора данных аудита с межсетевого экрана. В зависимости от типа события можно активировать/деактивировать аудит этого события, а также назначить этому событию одну из приведённых ниже степеней важности:

– 0 – Emergencies: системные неиспользуемые сообщения;

– 1 – Alerts: принимает немедленное действие;

– 2 – Critical: критическое условие;

– 3 – Errors: сообщение об ошибке;

– 4 – Warnings: предупреждающее сообщение;

– 5 – Notifications: нормальное, но значительное условие;

– 6 – Informational: информирующее сообщение;

– 7 – Debugging: отладочные сообщения и аудит FTP-команд и World Wide Web.

Для настройки аудита с помощью Syslog-сервера необходимо ввести следующее:

```
ciscoasa (config)# logging host имя IP-адрес
ciscoasa (config)# logging trap debugging
ciscoasa (config)# logging timestamp
ciscoasa (config)# logging enable
```

12 Настройка доступа с помощью ACL (Access Control List).

При необходимости настройки ограничений либо предоставления доступа для конкретных IP-адресов, подсетей, сетей для доступа к ядру АИС, у которых УБ меньше, чем УБ интерфейса МСЭ ядра АИС, необходимо настроить списки доступа. Для этого необходимо:

– сконфигурировать PAT-трансляцию адресов для адреса веб-сервера, который скрыт от внешних пользователей;

– сконфигурировать ACL, который разграничивает доступ к хосту и протоколам, для обеспечения доступа только к необходимому ресурсу;

– применить ACL к интерфейсу.

Пример конфигурирования PAT приведён в пункте 8. Конфигурирование ACL осуществляется следующим образом. Команда access-list указывает разрешён или запрещён доступ к порту или протоколу. По умолчанию доступ к адресам в ACL запрещён. Таким образом, необходимо каждую ситуацию прописать в явном виде. Синтаксис команды access-list:

```
access-list id [line – number] [extended] {deny | permit} {protocol | object-group protocol_obj_grp_id} {host sip | sip mask | interface ifc_name | object-group network_obj_grp_id | any} [operator port [port] | object – group ser-vice_obj_grp_id ] {host dip | dip dmask | interface ifc_name | object – group network_obj_grp_id | any} [operator port [port] | object – group ser-vice_obj_grp_id | object – group icmp_type_obj_group_id] [log [[level] [inter-val secs] | disable | default]] [inactive | time – range time_range_name]
```

Пример применения команды access-list:

```
ciscoasa (config)# access – list ACLOUT permit tcp any host 10.8.8.1 eq www
```

Данный пример демонстрирует следующее:

– ACLOUT – имя списка доступа;

– параметр permit является разрешающим для последующих действий;

– tcp – протокол передачи данных;

– any означает, что доступ разрешён с любого внешнего хоста;

– host 10.8.8.1 означает, что доступ разрешён к внутреннему ресурсу с IP-адресом 10.8.8.1;

– eq – оператор, уточняющий предыдущие действия;

– www – доступ разрешён только к HTTP-ресурсу.

Таким образом, в результате выполненной команды будет создано правило, на основании которого трафик, приходящий из внешней сети, может получить доступ только к HTTP-серверу внутренней сети и только по HTTP-протоколу.

Для применения указанной выше команды списка доступа необходимо присвоить список доступа ACLOUT определённому интерфейсу. Для этого необходимо ввести следующую команду:

```
ciscoasa (config)# access-group ACLOUT in interface outside
```

Здесь access-group ACLOUT – имя ACL, in означает, что ACL применяется для входящих пакетов на интерфейс outside.

Настройка временного диапазона

Если требуется настроить временной диапазон, в течение которого будет применяться указанный выше ACL, необходимо сделать следующее:

```
ciscoasa (config)# time-range TEMP-WORKER
ciscoasa (config-time-range)# absolute start 00:00 1 August 2008 end 00:00 30 August 2006
```

или

```
ciscoasa (config-time-range)# periodic weekdays 08:00 to 17:00
```

Здесь absolute служит для задания точного периода времени, periodic служит для задания дневных интервалов, weekdays – означает период с понедельника по пятницу, daily – каждый день, weekend – выходные дни.

Для применения данных настроек необходимо выполнить следующие действия:

```
ciscoasa (config)# access – list ACLOUT permit tcp any host 80.94.232.9 eq www
time-range TEMP-WORKER
```

Настройка аудита для ACL событий

Для настройки аудита событий использования ACL необходимо:

```
ciscoasa (config)# access – list ACLOUT permit tcp any host 80.94.232.9 log 7
interval 600
```

Здесь log означает, что для аудита используется настроенная выше функция аудита событий с помощью Syslog-сервера, приведённая в пункте 11, цифра 7 означает степень важности, interval 600 означает 10 мин, в течение которых будет подсчитываться количество вызовов ACL.

13 Настройка пакетного фильтра (Java, Active X, FTP, HTTP).

Используется для блокировки потенциально опасных приложений, запущенных во внутренней сети с внешнего ресурса.

Пакетный фильтр настраивается командой filter:

```
filter {activex | Java} {port[-port] | except} local_ip local_mask for-eign_ip foreign_mask
```

Например, команда

```
ciscoasa (config)# filter activex 80 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

означает, что ActiveX блокируется для всего веб-трафика через порт 80 с любого хоста на любой хост.

Например, команда

```
ciscoasa (config)# filter Java http 192.168.3.3 255.255.255.255
```

означает, что происходит предотвращение загрузки Java-апплета из внешней сети на 192.168.3.3

14 Настройка AAA (Authentication, Authorization and Accounting) (аутентификация, авторизация и логирование действий пользователя).

Используется непосредственно на МСЭ. Данная функция необходима для учёта пользователей, которые пытаются установить сеанс связи с удалённым объектом, находящимся за пределами МСЭ. Например, если пользователи внутри сети хотят установить сеанс связи с сетью Интернет, то им придётся пройти процедуру аутентификации.

Ниже приведён пример конфигурационного файла Cisco ASA 5520.

```
ASA Version 7.0(7)
!
hostname ciscoasa
domain-name default.domain.invalid
enable password 2KFQnbNIdI.2KYOU encrypted
names
dns-guard
!
interface GigabitEthernet0/0
 speed 10
 duplex full
 nameif outside
 security-level 0
 ip address 80.94.232.94 255.255.255.224
!
interface GigabitEthernet0/1
 duplex full
 nameif inside
 security-level 90
 ip address 10.7.7.253 255.255.255.0
!
interface GigabitEthernet0/2
```

```
 nameif sku
 security-level 5
 ip address 192.168.2.222 255.255.255.0
!
interface GigabitEthernet0/3
 nameif dmz
 security-level 50
 ip address 10.8.8.1 255.255.255.0
!
interface Management0/0
 nameif management
 security-level 100
 ip address 192.168.1.13 255.255.255.0
 management-only
!
passwd 2KFQnbNIdI.2KYOU encrypted
ftp mode passive
access-list inside_access_in extended permit ip host 10.7.7.71 any
access-list inside_access_in extended deny tcp any any eq www
access-list outside_access_in extended permit ip any any
access-list inside_access_out extended permit ip any any
access-list outside_access_out extended permit ip any any
access-list outside_access_out extended permit tcp interface outside eq www any
access-list outside_access_out extended deny tcp any any eq www
pager lines 24
logging console informational
logging monitor warnings
logging asdm informational
mtu outside 1500
mtu inside 1500
mtu sku 1500
mtu dmz 1500
mtu management 1500
ip verify reverse-path interface outside
ip verify reverse-path interface inside
no failover
icmp permit any outside
icmp permit any inside
asdm image disk0:/asdm-507.bin
no asdm history enable
arp timeout 14400
global (outside) 1 interface
global (inside) 2 interface
nat (outside) 2 80.94.232.0 255.255.255.0
nat (inside) 1 10.7.7.0 255.255.255.0
access-group outside_access_in in interface outside
access-group outside_access_out out interface outside
access-group inside_access_in in interface inside
access-group inside_access_out out interface inside
route outside 0.0.0.0 0.0.0.0 80.94.232.65 1
timeout xlate 3:00:00
```

```
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00
timeout mgcp-pat 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
username admin password f3UhLvUj1QsXsuK7 encrypted privilege 15
http server enable
http 192.168.1.0 255.255.255.0 management
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart
telnet 192.168.1.0 255.255.255.0 management
telnet timeout 5
ssh timeout 5
console timeout 0
management-access management
dhcpd address 10.7.7.120-10.7.7.150 inside
dhcpd dns 80.94.225.5 80.94.225.254
dhcpd lease 3600
dhcpd ping_timeout 50
dhcpd option 3 ip 10.7.7.253
dhcpd option 6 ip 80.94.225.5 80.94.225.254
dhcpd enable inside
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map global_policy
 class inspection_default
 inspect dns maximum-length 512
 inspect ftp
 inspect h323 h225
 inspect h323 ras
 inspect rsh
 inspect rtsp
 inspect esmtp
 inspect sqlnet
 inspect skinny
 inspect sunrpc
 inspect xdmcp
 inspect sip
 inspect netbios
 inspect tftp
!
service-policy global_policy global
Cryptochecksum:f2dfa3df0015efe77beb88d5ce64ee18
: end
.
```

4 МЕХАНИЗМЫ ЗАЩИТЫ В КОММУТАТОРАХ И МАРШРУТИЗАТОРАХ

В современных локальных и глобальных сетях обмен информацией, как правило, предполагает передачу данных через сетевое оборудование: коммутаторы и маршрутизаторы.

Поэтому само сетевое оборудование и протоколы, которые используют коммутаторы и маршрутизаторы, могут быть целью атак. Более того, некоторые настройки коммутаторов и маршрутизаторов (как правило, это настройки по умолчанию) позволяют выполнить ряд атак и получить несанкционированный доступ к сети или вывести из строя сетевые устройства.

Для защиты сетевой инфраструктуры оборудование сети и, в частности, коммутаторы и маршрутизаторы оснащаются механизмами сетевой безопасности. Ниже рассмотрены механизмы сетевой безопасности в коммутаторах и маршрутизаторах компании Cisco System. Описанные механизмы сетевой безопасности также реализованы и другими производителями телекоммуникационного оборудования, но имеют свою специфику настройки и терминологию.

4.1 Механизмы сетевой безопасности в коммутаторах

Механизмы сетевой безопасности, реализованные в коммутаторах, как правило, включают в себя следующее [5]:

– Port Security;
– DHCP snooping;
– Dynamic ARP Inspection;
– IP Source Guard;
– BPDU Guard и Root Guard;
– аутентификация при доступе к сети;
– списки контроля доступа;
– VLAN.

4.1.1 Port Security

Первое, что необходимо реализовать в коммутируемой сети с точки зрения безопасности – это предотвращение подключения чужих устройств. Сделать это можно достаточно легко, и многие производители предлагают такую возможность (в Cisco это Port Security).

Реализация блокировки подключения чужих устройств (третья команда блокирует порт на 600 мин):

```
set port security 2/1 enable
set port security 2/1 enable 00-90-2b-03-34-08
set port security 2/1 shutdown 600
```

Описанный выше подход достаточно сложен в администрировании и абсолютно не масштабируем. Например, если требуется предоставить доступ к сети ранее не зарегистрированному компьютеру, жёсткая привязка портов коммутатора к MAC-адресам сделает эту задачу трудновыполнимой. Тем более что подделка MAC-адреса в настоящее время не является сложной задачей. Поэтому рекомендуется пойти немного другим путём: с помощью механизма 802.1x блокировать подключение несанкционированных устройств, а функцию Port Security использовать для динамической авторизации на коммутаторе.

Иными словами, вместо указания самих MAC-адресов указывается количество адресов, которые могут работать на данном порту. Порт коммутатора в динамическом режиме запоминает первые адреса, которые к нему «обратились», и в течение заданного администратором времени разрешает трафик только с них. При этом если на порт попал трафик с неразрешённых адресов, возможно применение двух режимов – shutdown и restricted. В первом случае блокируется работа самого порта, во втором – приём трафика с неразрешённых адресов.

Реализация механизма Port Security для IOS:

```
switchport port-security
switchport port-security maximum 3
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
```

Здесь авторизуется только три MAC-адреса на порту, и при превышении их числа порт не блокируется. Время «привязки» адресов к порту (время устаревания информации об авторизованных адресах) составляет 2 мин. Если используется IP-телефония, то на каждом порту необходимо разрешать три адреса (рабочая станция, IP-телефон и мини-коммутатор в IP-телефоне). При подключении только рабочей станции достаточно указать максимальное количество адресов на порту, равное единице.

Механизм Port Security помимо блокирования «чужих» адресов может быть использован и для предотвращения атак MAC Flood (переполнения таблицы коммутации) или DHCP Starvation (истощения DHCP).

4.1.2 DHCP snooping

Одной из распространённых атак, которая встречается в локальных сетях, является перехват трафика путём его перенаправления на себя. Злоумышленник, выдавая себя за DHCP-сервер, подменяет адреса отдельных узлов в сети (например, маршрутизатора), тем самым меняя маршруты информационных потоков. Другим применением этой атаки может служить атака «отказ в обслуживании», когда на определённые адреса трафик может вообще не доходить.

Ещё один пример атак с применением DHCP – истощение адресов (DHCP Starvation). Генерируя большой поток ложных служебных сообщений о выделении адресов, злоумышленник может «выбрать» весь пул адресов, и для авторизованных

пользователей их просто не останется, что приведёт к невозможности их работы. Защититься от этого позволяет встроенная функция коммутаторов Cisco Catalyst – DHCP Snooping.

DHCP snooping обеспечивает:

1) защиту клиентов в сети от получения адреса от неавторизованного DHCP-сервера;

2) предупреждение клиентов о том, какие сообщения протокола DHCP отбрасывать, какие перенаправлять и на какие порты.

Для правильной работы DHCP snooping необходимо определить доверенные (trusted) и недоверенные (untrusted) порты коммутатора:

1) недоверенные (Untrusted) – порты, к которым подключены клиенты. DHCP-ответы, приходящие с этих портов, отбрасываются коммутатором. Для ненадёжных портов выполняется ряд проверок сообщений DHCP и создаётся база данных привязки DHCP (DHCP snooping binding database);

2) доверенные (Trusted) – порты коммутатора, к которым подключён другой коммутатор или DHCP-сервер. DHCP-пакеты, полученные с доверенных портов не отбрасываются.

По умолчанию коммутатор отбрасывает DHCP-пакет, который пришёл на недоверенный порт, если:

1) приходит одно из сообщений от DHCP-сервера (DHCPOFFER, DHCPACK, DHCPNAK или DHCPLEASEQUERY);

2) приходит сообщение DHCPRELEASE или DHCPDECLINE, в котором содержится MAC-адрес из базы данных привязки DHCP, но информация об интерфейсе в таблице не совпадает с интерфейсом, на котором был получен пакет;

3) в пришедшем DHCP-пакете не совпадают MAC-адрес, указанный в DHCP-запросе, и MAC-адрес отправителя;

4) приходит DHCP-пакет, в котором есть опция 82.

4.1.3 Dynamic ARP Inspection

Протокол ARP при некорректной настройке позволяет злоумышленникам осуществлять перехват данных, «отказывать в обслуживании» и вносить хаос в работу сети. Например, атаки ARP-spoofing позволяют перехватывать трафик между узлами, которые расположены в пределах одного широковещательного домена, и осуществлять подмену адресов. Для защиты от ARP-атак в коммутаторах Cisco существует специальный механизм – Dynamic ARP Inspection (DAI).

Dynamic ARP Inspection позволяет:

– защитить клиентов в сети от атак с использованием протокола ARP;

– определить, какие сообщения протокола ARP отбрасывать, а какие перенаправлять.

Для правильной работы Dynamic ARP Inspection необходимо установить доверенные (trusted) и недоверенные (untrusted) порты коммутатора.

Доверенными считаются порты коммутатора, к которым подключен другой коммутатор. Сообщения протокола ARP, полученные с доверенных портов, не отбрасываются.

Недоверенными считаются порты, к которым подключены клиенты. Для недоверенных портов выполняется ряд проверок сообщений ARP.

Механизм Dynamic ARP Inspection работает следующим образом. Если порт недоверенный, коммутатор перехватывает все ARP-запросы и ARP-ответы на недоверенных портах и, прежде чем перенаправлять их, проверяет соответствие MAC-адреса IP-адресу.

Проверка соответствия MAC-адреса IP-адресу может выполняться на основании информации базы данных привязки DHCP и статических записей.

Пример настройки DAI для IOS:

```
ip arp inspection vlan 4,104
ip arp inspection trust
ip arp inspection limit rate 15
```

Первая команда связывает определённую VLAN с механизмом DAI, вторая – определяет доверенные порты, а третья – ограничивает полосу пропускания для защиты от DoS-атак.

Включение дополнительных проверок:

```
ip arp inspection validate < [src-mac] [dest-mac] [ip]>
```

Параметры команды:

– src-mac – коммутатор проверяет, соответствует ли MAC-адрес источника в заголовке пакета MAC-адресу отправителя в теле ARP-пакета (проверяются ARP-запросы и ARP-ответы). Если эти два адреса не совпадают, то коммутатор отбрасывает пакет;

– dest-mac – коммутатор проверяет, соответствует ли MAC-адрес назначения в заголовке пакета MAC-адресу получателя в теле ARP-пакета. Если эти два адреса не совпадают, то коммутатор отбрасывает пакет;

– ip – коммутатор проверяет, не передаётся ли «неправильный» IP-адрес в теле ARP-пакета вместо IP-адреса отправителя и получателя. Если будет обнаружен «неправильный» IP-адрес, то коммутатор отбросит пакет.

«Неправильные» адреса:

– 0.0.0.0;

– 255.255.255.255;

– IP-адреса класса D;

– IP-адреса класса E.

4.1.4 IP Source Guard

Злоумышленник может подменять не только MAC-адреса, реализуя различные ARP-атаки, но и организовывать IP-спуфинг. Например, до 95 % DoS-атак осуществляется именно с подменой IP-адреса, поэтому защита от этой угрозы является достаточно актуальной. Но если на периметре это сделать достаточно просто с помощью любого коммутатора или маршрутизатора, то в локальной

сети это достаточно серьёзная проблема. В коммутаторах Cisco Catalyst можно использовать для этой цели механизм IP Source Guard.

Функция защиты IP-адреса (IP Source Guard) предназначена для фильтрации трафика, принятого с интерфейса, на основании таблицы соответствий DHCP snooping и статических соответствий IP Source Guard. Таким образом, IP Source Guard позволяет бороться с подменой IP-адресов в пакетах.

Поскольку функция защиты IP-адреса использует таблицы соответствий DHCP snooping, то для её использования необходимо предварительно настроить и включить DHCP snooping.

Пример настройки функции защиты IP-адреса:

1 Включить функцию защиты IP-адреса для фильтрации трафика на основании таблицы соответствий DHCP snooping и статических соответствий IP Source Guard. Интерфейс в первой группе VLAN:

```
console(config)# ip dhcp snooping
console(config)# ip dhcp snooping vlan 1
console(config)# ip source-guard
```

2 Создать статическую запись в таблице соответствия для интерфейса, например, для gigabitethernet /0/1: IP-адрес клиента – 192.168.1.210, его MAC-адрес – 00:60:70:4A:AB:AF:

```
console(config)# ip source-guard binding 00:60:70:4A:AB:AF 1 192.168.1.210
gigabitethernet 1/0/1
```

3 Включить функцию защиты IP-адреса для интерфейса gigabitethernet /0/1:

```
console(config)# ip source-guard
```

4.1.5 BPDU Guard и Root Guard

Другой проблемой для локальных сетей является набор уязвимостей протокола Spanning Tree (STP), которые были обнаружены российскими экспертами в области безопасности – Олегом Артемьевым и Владиславом Мяснянкиным. До сих пор встречается оборудование известных сетевых вендоров, которые подвержены данным уязвимостям, что приводит к хаосу в сети и её отказу в обслуживании. В коммутаторах Cisco существует два механизма: BPDU Guard и Root Guard.

BPDU – Bridge Protocol Data Unit – фреймы, которыми обмениваются коммутаторы для выбора корневого (root) устройства при реализации протокола STP (Spanning Tree Protocol).

Чтобы гарантировать безопасность сети и неприкосновенность статуса root избранного устройства, на всех коммутаторах уровня доступа настраивается функция BPDU Guard, которая препятствует подключению к сети любого

устройства, активно отсылающего в неё BPDU-пакеты. Таким образом, BPDU Guard защищает неизменность топологии сети, а главное – препятствует злоумышленнику подключить устройство с низким приоритетом, чтобы заставить остальных коммутаторов думать, что в сети появился новый root, и перестроить всю топологию относительно атакующего устройства, тем самым позволив ему пропускать через себя все данные, предоставляя злоумышленнику такую информацию, как IP-адреса, пароли и т. п.

Подобно функции port-security, правильно настроенный BPDU Guard заблокирует порт в тот момент, когда обнаружит попытку передачи в сеть несанкционированного пакета BPDU через него. Настройка BPDU Guard начинается с команды spanning-tree portfast, которая прописывается на всех интерфейсах коммутаторов уровня доступа, предназначенных для подключения оконечных устройств. Далее есть два пути:

1) включить BPDU Guard в режиме глобальной конфигурации командой spanning-tree portfast bpduguard, которая включает защиту на всех интерфейсах с функцией spanning-tree portfast;

2) включить BPDU Guard на каждом интерфейсе по отдельности командой spanning-tree bpduguard enable.

Когда сработает защита и порт заблокируется, привести его в рабочее состояние можно будет только вручную, поочередно прописав на интерфейсе команды shut и no shut. При желании можно автоматизировать этот процесс и воспользоваться командой errdisable recovery cause bpduguard, которая включит порт через 300 с. Изменить длительность таймера можно командой errdisable recovery interval 400, где 400 – количество секунд из дипазона от 30 до 86 400:

```
(config-if)#spanning-tree portfast
(config-if)#spanning-tree bpduguard enable
(config)#errdisable recovery cause bpduguard
(config)#errdisable recovery interval 400
```

Функция защиты корня quard root обеспечивает возможность задать расположение корневого моста в сети. Это обеспечивает уверенность в том, что порт, на котором активизирована функция защиты корня, является назначенным. Обычно все порты корневого моста являются назначенными, если два или более портов корневого моста не соединены вместе. Если мост получает высокоприоритетные STP-элементы данных протокола управления мостами в корневом порту, для которого включена функция защиты корня, защита корня переводит порт в состояние STP, называемое несогласованностью корня. Состояние несогласованности корня аналогично состоянию прослушивания. Трафик через порт в таком состоянии не пересылается. Таким образом, защита корня задаёт расположение корневого моста. Функцию защиты корня необходимо включить на всех портах, которые не должны стать корневыми.

4.1.6 Аутентификация при доступе к сети

Протокол 802.1X работает на канальном уровне и определяет механизм контроля доступа к сети на основе принадлежности к порту (рисунок 4.1).

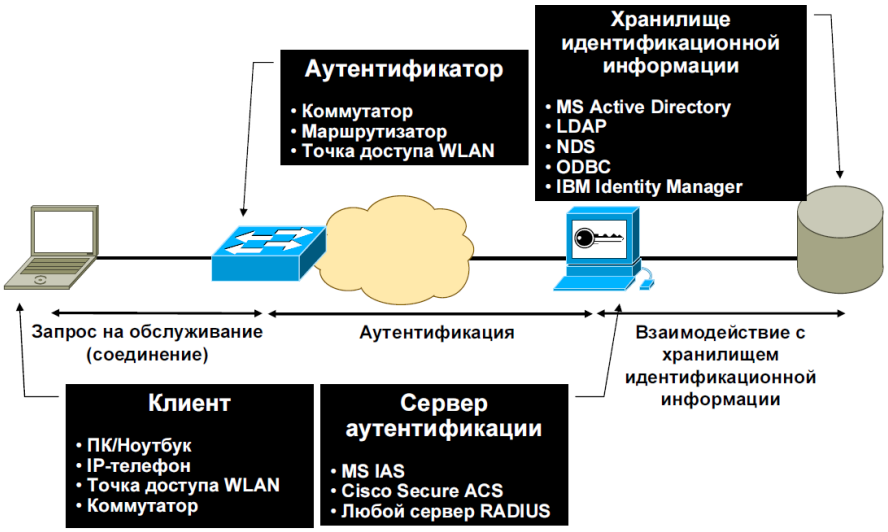


Рисунок 4.1 – Модель контроля доступа к портам стандарта 802.1X

Согласно протоколу 802.1X доступ к сети получают только клиенты, прошедшие аутентификацию, если аутентификация не была пройдена, доступ с соответствующего порта будет запрещён.

В протоколе 802.1X предполагается использование модели «точка – точка». То есть он не может быть применён в ситуациях, когда несколько хостов соединяются с коммутатором (на котором настроена аутентификация 802.1X) через хаб или через другой коммутатор.

Аутентификатор (authenticator) – устройство, контролирующее физический доступ к сети, которое выполняет роль посредника (proxy) между клиентом и сервером аутентификации, основываясь на статусе аутентификации клиента.

Ключевым моментом здесь является то, что сетевые устройства – аутентификаторы – могут быть достаточно простыми, поскольку для реализации функций 802.1X в них требуются минимальные аппаратные затраты, в то время как весь интеллект концентрируется в RADIUS-сервере. Такая схема имеет дополнительные выгоды и позволяет организовать тесную интеграцию управления сетевым оборудованием и сетевым ПО, что значительно облегчает управление информационной системой большого предприятия в целом. Протокол передачи EAP-сообщений в стандарте 802.1X называется EAPOL (EAP encapsulation over LAN) и в настоящее время определён для Ethernet локально-вычислительной сети (ЛВС), а также беспроводных сетей стандартов серии IEEE 802.11 и ЛВС, использующих технологии token ring и FDDI.

Для каждого порта коммутатора (с включенным 802.1X) создаётся два виртуальных порта:

– контролируемый порт (controlled port), который открывается только после авторизации по 802.1X; также называется авторизованный (authorized) порт;

– неконтролируемый порт (uncontrolled port), который разрешает передавать только EAPOL-трафик, также называется неавторизованный (unauthorized) порт.

До тех пор пока клиент не авторизован, на неконтролируемом порту разрешён только EAPOL-трафик.

4.1.7 Списки контроля доступа

ACL (Access Control Lists) – списки контроля доступа, которые могут использоваться в большом количестве различных сетевых операций, таких как управление маршрутизацией, контроль доступа к маршрутам и фильтрации системных выводов с CLI-интерфейсов, контроль над параметрами внешних шлюзов, таких как BGP AS-path. ACL также могут использоваться для контроля доступа к NAT (Network Address Translation – преобразование сетевых адресов) или фильтрации протоколов, не относящихся к IP. В зависимости от опций, установленных на Cisco IOS (Internetwork Operating System – межсетевая операционная система), ACL могут быть использованы для шифрования.

ACL представляют собой набор правил и действий, разделённых порядковыми номерами. Действия в ACL называются ACE (Access Control Entries – записи контроля доступа). Каждая ACE-запись определяет, какое действие с пакетом разрешено или запрещено выполнять. Чтобы выполнилось правило, все условия должны быть выполненными. В каждом правиле можно указать источник и направление трафика, который удовлетворяет данному правилу. Также можно указать в качестве источника и получателя определённый узел или группу узлов в сети, или сеть (подсеть) целиком.

Неявные запрещающие списки ACL имеют неявные правила. Эти правила не видны в рабочей конфигурации. Коммутатор исполняет эти правила, когда трафик не попадает ни под одно из описанных в ACL правил. Все списки ACL для IP (IPv4) включают в себя неявное правило «deny ip any any». Данное правило запрещает прохождение через коммутатор IP-трафика, который не описан ни в одном правиле. Чтобы отменить это правило для IP ACL, нужно выполнить команду «permit ip any any». Если данную команду не прописать в конец ACL, то всегда будет выполняться правило «deny ip any any», которое скрыто в списке каждого ACL.

4.1.8 VLAN

VLAN (Virtual Local Area Network) – группа устройств, имеющих возможность взаимодействовать между собой напрямую на канальном уровне, хотя физически при этом они могут быть подключены к разным сетевым коммутаторам. И наоборот, устройства, находящиеся в разных VLAN, невидимы друг для друга на канальном уровне, даже если они подключены к одному коммутатору, и связь между этими устройствами возможна только на сетевом и более высоких уровнях.

В современных сетях VLAN – главный механизм для создания логической топологии сети, не зависящей от её физической топологии. VLAN используются для сокращения широковещательного трафика в сети. Имеют большое значение с точки зрения безопасности, в частности, как средство борьбы с атаками ARP-spoofing.

Технология VLAN позволяет следующее:

1 Гибко разделять устройства на группы. Как правило, одному VLAN соответствует одна подсеть. Устройства, находящиеся в разных VLAN, будут находиться в разных подсетях. Но в то же время VLAN не привязан к местоположению устройств и поэтому устройства, находящиеся на расстоянии друг от друга, всё равно могут быть в одном VLAN независимо от местоположения.

2 Уменьшать количество широковещательного трафика в сети. Каждый VLAN – это отдельный широковещательный домен. Так как коммутатор – это устройство второго уровня модели OSI, то все порты на коммутаторе, где нет VLAN, находятся в одном широковещательном домене. Создание VLAN на коммутаторе означает разбиение коммутатора на несколько широковещательных доменов. Если один и тот же VLAN есть на разных коммутаторах, то порты разных коммутаторов будут образовывать один широковещательный домен.

3 Усиливать безопасность и управляемость сети. Когда сеть разбита на VLAN, упрощается задача применения политик и правил безопасности. С VLAN политики можно применять к целым подсетям, а не к отдельному устройству. Кроме того, переход из одного VLAN в другой предполагает прохождение через устройство третьего уровня, на котором, как правило, применяются политики, разрешающие или запрещающие доступ из VLAN в VLAN.

Компьютер при отправке трафика в сеть не содержит информации о том, в каком VLAN он размещён. Такая информация содержится в коммутаторе, который знает распределение компьютеров, подключенных к определённым портам, по соответствующим VLAN. Трафик, приходящий на порт определённого VLAN, ничем особенным не отличается от трафика другого VLAN. Другими словами, никакой информации о принадлежности трафика определённому VLAN в нём нет.

Однако, если через порт может прийти трафик разных VLAN, коммутатор должен его как-то различать. Для этого каждый кадр (frame) трафика должен быть помечен каким-то особым образом. Пометка должна говорить о том, какому VLAN принадлежит трафик.

Наиболее распространённый сейчас способ ставить такую пометку описан в открытом стандарте IEEE 802.1q. Коммутатор – устройство второго уровня и изначально все порты коммутатора находятся, как правило, в VLAN 1 и, следовательно, в одном широковещательном сегменте. Это значит, что если один из хостов, подключенных к коммутатору, отправит широковещательный фрейм, то все остальные хосты, подключенные к нему, также получат его.

Порты коммутатора, поддерживающие VLAN (с некоторыми допущениями), можно разделить на два множества:

– тегированные порты (или транковые порты, trunk-порты в терминологии Cisco);

– нетегированные порты (или порты доступа, access-порты в терминологии Cisco).

Тегированные порты нужны для того, чтобы через один порт была возможность передать несколько VLAN и, соответственно, получать трафик нескольких

VLAN на один порт. Информация о принадлежности трафика VLAN, как было сказано выше, указывается в специальном теге. Если порт нетегированный в каком-то VLAN, то трафик этого VLAN передаётся без тега. На Cisco нетегированным порт может быть только в одном VLAN, на некоторых других коммутаторах (например, ZyXEL, D-Link или Planet) данного ограничения нет.

Если порт тегирован для нескольких VLAN, то в этом случае весь нетегированный трафик будет приниматься специальным родным VLAN (native VLAN). Обычно по умолчанию все порты коммутатора считаются нетегированными членами VLAN 1. В процессе настройки или работы коммутатора они могут перемещаться в другие VLAN.

Существуют два подхода к назначению порта в определённый VLAN:

1) статическое назначение – принадлежность порта VLAN задаётся администратором в процессе настройки;

2) динамическое назначение – принадлежность порта VLAN определяется в ходе работы коммутатора с помощью процедур, описанных в специальных стандартах, таких, например, как 802.1X. При использовании 802.1X для того чтобы получить доступ к порту коммутатора, пользователь проходит аутентификацию на RADIUS-сервере. По результатам аутентификации порт коммутатора размещается в том или ином VLAN.

4.2 Механизмы сетевой безопасности в маршрутизаторах

Дополнительно к описанным в подразделе 4.1 механизмам защиты, которые реализованы в коммутаторах, только в маршрутизаторах (маршрутизаторы Cisco), встроены [5]:

– прозрачный межсетевой экран Cisco IOS Firewall [11];

– средство построения VPN (IPSec или SSL) Cisco IOS VPN;

– прозрачная система предотвращения атак (Intrusion Prevention System) Cisco IOS IPS.

Помимо этих хорошо известных механизмов существует и множество других не менее важных и полезных функций, делающих из обычного маршрутизатора полноценное защитное устройство, ориентированное на защиту небольших и средних офисов. Маршрутизатор может быть логически разделен на четыре функциональных компонента, отвечающих за решение своих задач:

1) Data Plane – уровень данных, через который проходит весь сетевой трафик;

2) Control Plane – уровень построения и обновления таблиц маршрутизации;

3) Management Plane – уровень управления маршрутизатором (SSH, SNMP, syslog и т. д.);

4) Service Plane – уровень обеспечения качества сервиса и уровня обслуживания.

Очевидно, что механизмы защиты маршрутизаторов должны быть применены ко всем этим уровням без исключения. Причём защита уровней управления и контроля зачастую является даже более важной, чем безопасность уровня данных. Списки контроля доступа (Access Control List, ACL), однонаправленная проверка передачи по обратному маршруту (Unicast Reverse Path Forwarding, uRPF),

ограничение полосы пропускания (Committed Access Rate, CAR) и так далее очень важны, но позволяют ограничить только определённые типы трафика. А вот недооценка вопросов самозащиты самого маршрутизатора может повлечь за собой катастрофические последствия – захват и компрометация всего устройства, локальное или дистанционное изменение таблиц маршрутизации, перехват трафика, реализация атак «отказ в обслуживании» (Denial of Service, DoS) и т. п.

4.2.1 AutoSecure

В маршрутизаторах Cisco, начиная с версии IOS 12.3, появился механизм AutoSecure, который [5]:

– запрещает потенциально опасные глобальные сервисы (Finger, Packet assembler and disassembler, TCP/UDP Small Services, Bootp Server, HTTP Server, CDP, NTP, Source Routing);

– запрещает потенциально опасные сервисы по интерфейсам (ICMP, Proxy-Arp, Broadcast, MOP, ICMP Unreachable, ICMP Reply);

– включает расширенные механизмы защиты (шифрование паролей, настройка баннеров, взаимодействие с серверами аутентификации, антиспуфинг, Cisco Express Forwarding, блокирование зарезервированных адресов IANA, установка маршрута по умолчанию NULL 0, CBAC, Netflow);

– обеспечивает защиту самого маршрутизатора (SSH и SCP, настройка паролей и учётных записей, блокирование SNMP);

– включает регистрацию событий безопасности.

Cisco AutoSecure может функционировать в двух режимах – интерактивном и автоматическом. В первом случае администратор отвечает на вопросы о своей собственной сети, а во втором – настройка осуществляется автоматически, в соответствие с параметрами по умолчанию. Причём включить Cisco AutoSecure можно всего одной командой: Router# auto secure.

По окончании работы сервиса на экран выводится список всех установленных настроек, и администратор должен разрешить все сделанные изменения. Проверка может быть осуществлена двумя путями – с помощью Security Device Manager (SDM) и команды IOS EXEC, которая показывает настройки, сделанные после AutoSecure. Наиболее интересен именно первый путь (функция Security Audit), т. к. он позволяет в удобном виде получить ответ на вопрос: «Какие из существующих механизмов защиты включены, а какие нет?».

4.2.2 Расширения IOS Login

Начиная с версии IOS 12.2(25)S, маршрутизаторы Cisco могут существенно усложнить жизнь злоумышленникам, желающим получить несанкционированный доступ к сетевому оборудованию. Одна из распространённых атак, позволяющих получить такой доступ, – подбор пароля. Для этого используются различные утилиты, к примеру, THC-Hydra или Brutus. Самый простой путь блокировать эту атаку – увеличить время задержки между попытками ввода логина и пароля. Сделать это можно тремя путями: через уже описанную функцию AutoSecure или с помощью специальных команд – login delay и login block-for.

Эти команды можно использовать и в паре.

```
Router(config)# login block-for 100 attempts 5 with-in 50
Router(config)# login quiet-mode access-class myacl
Router(config)# login delay 10
Router(config)# login on-failure log
Router(config)# login on-success log
```

Первая команда должна вводиться до использования любых других команд login. Она на 100 с блокирует любые попытки подключения к устройству, если в течение 50 с было осуществлено пять неудачных регистраций на маршрутизаторе. Если есть адреса, которые не должны быть блокированы (например, административные), то они описываются командой login quiet-mode access-class. Команда login delay определяет время задержки перед разрешением повторной регистрации. Если её не указать, то автоматическая задержка будет осуществлена по команде login block-for на 1 с. Последние две команды включают регистрацию успешных и неудачных попыток подключения к маршрутизатору.

Проверить настройки подсистемы регистрации можно путём использования команды show login. А команда show login failures показывает все неудачные попытки подключения к устройству [5].

4.2.3 Защита уровня контроля

Почти все конструкции маршрутизаторов уязвимы к атакам «отказ в обслуживании». При атаке на сетевое оборудование это несёт серьёзную опасность, т. к. выведение его из строя приводит к неработоспособности всей сети. Необходимо оградить процессор маршрутизатора от обработки вредоносного трафика и, начиная с версии IOS 12.2, такая возможность появилась и стала носить название Control Plane Policing. С её помощью можно [5]:

1) классифицировать и ограничить каждый класс трафика, поступающий на обработку в уровень контроля;

2) обеспечить механизм раннего отбрасывания пакетов, направленных на закрытые или иные TCP/UDP-порты;

3) обеспечить защиту от протокольного флудинга (технический флуд представляет собой хакерскую атаку с большим количеством запросов, приводящую к отказу в обслуживании);

4) обеспечить QoS для пакетов, направленных на уровень контроля;

5) обеспечить надёжность, защищённость и доступность.

Для реализации данного механизма необходимо пройти четыре обязательных и два опциональных шага:

1) задать критерии для классификации пакетов;

2) определить политики сервиса;

3) перейти в режим настройки;

4) применить политики;

5) настроить политики фильтрации портов (для раннего отбрасывания пакетов);

6) настроить политики пороговых значений (защита от протокольного флудинга).

Для реализации первой задачи необходимо использовать две команды. Первая задаёт имя класса трафика (class-map), вторая описывает критерии для данного трафика (match). Параметр match-any говорит о том, что хотя бы один критерий классификации должен встретиться в трафике (использование параметра match-all требует обнаружения всех критериев):

```
Router(config)# class-map match-any control-plane-class
Router(config-cmap)# match access-group name cpp-icmp-acl
```

Для определения политики необходимо выполнить три команды, задающие имя политики (policy-map), класс (class) и определяющие политику (police):

```
Router(config)# policy-map control-plane-policy
Router(config-pmap)# class control-place-class
Router(config-pmap-c)# police rate 50000 pps con-form-action transmit exceed-
    action drop
```

Применение политики осуществляется двумя задачами, которые связывают политику с субинтерфейсом (control-plane) и указывают имя используемой политики:

```
Router(config)# control-plane host
Router(config-cp)# service-policy input control-plane-policy
```

Для оставшихся двух опциональных задач необходимо использование команды class-map type, схожей по синтаксису с командами, описанными выше. Фильтрация портов и пороговых значений описывается следующим образом:

```
Router(config)# class-map type port-filter match-all pf-class
Router(config-cmap)# match closed-ports
Router(config-cmap)# exit
Router(config)# policy-map type port-filter cppr-pf-policy
Router(config-pmap)# class pf-class
Router(config-pmap-c)# drop
Router(config-pmap-c)# end
Router(config)# control-plane host
Router(config)# service-policy input cppr-pf-policy
Router(config)# class-map type queue-threshold qt-snmp-class
Router(config-cmap)# match protocol snmp
Router(config-cmap)# class-map type queue-threshold qt-telnet-class
Router(config-cmap)# match protocol telnet
Router(config-cmap)# class-map type queue-threshold qt-other-class
Router(config-cmap)# match host-protocols
Router(config-cmap)# exit
Router(config)# policy-map type queue-threshold qt-policy
```

```
Router(config-pmap)# class qt-snmp-class
Router(config-pmap-c)# queue-limit 50
Router(config-pmap-c)# class qt-telnet-class
Router(config-pmap-c)# queue-limit 50
Router(config-pmap-c)# class qt-other-class
Router(config-pmap-c)# queue-limit 150
Router(config-pmap-c)# end
```

Проверить настройки подсистемы регистрации можно путём использования команды show policy map control-plane.

4.2.4 Защита уровня управления

Механизм Control Plane Policing (CoPP) позволяет защитить маршрутизатор от обработки вредоносного трафика и не дать ему попасть в защищаемую сеть. Однако всё равно остается проблема защиты самого устройства от несанкционированного доступа. Эту задачу решает механизм Management Plane Policing (MPP), который позволяет описать один или несколько интерфейсов маршрутизатора как управляющие, что, в свою очередь, блокирует любые попытки управления с «не управляющих» интерфейсов. Иными словами, ограничивается доступ по протоколам FTP, HTTP, HTTPS, SSH, Telnet, SNMP и TFTP. Это, конечно, можно было бы реализовать и с помощью списков контроля доступа (ACL), но в этом случае снижается производительность и масштабируемость маршрутизатора, вынужденного тратить ресурсы на обработку ACL. Настройка данного механизма осуществляется достаточно просто [5]:

```
Router(config)# control-plane host
Router(config-cp-host)# management-interface FastEthernet 0/0 allow ssh snmp
```

Первая команда включает режим конфигурации, а вторая – задаёт его настройки. После параметра allow можно указать протоколы, которые разрешаются на данном интерфейсе (в приведённом примере – только SSH и SNMP).

Проверить наличие и настройки управляющих интерфейсов можно командой Router# show management-interface.

4.2.5 CPU и Memory Thresholding Notification

Очень часто признаком атаки «отказ в обслуживании» или другой вредоносной активности является перегрузка центрального процессора или нехватка памяти, вызванные наличием какого-нибудь процесса, «забирающего» все ресурсы «под себя». Механизм контроля аналогичных действий есть в маршрутизаторах Cisco: в CPU и Memory Threshholding Notification [5].

В первом случае система сигнализирует, когда загрузка процессора превышает максимально заданную или падает ниже минимально заданной границы.

Делается это следующим образом:

```
Router(config)# snmp-server enable traps cpu thresh-old
```

```
Router(config)# snmp-server host 192.168.0.0 traps public cpu
Router(config)# process cpu threshold type total rising 80 interval 5 falling
    20 interval 5
```

Первая команда разрешает посылать уведомления о нарушении, связанном с загрузкой процессора. Вторая описывает адрес, на который посылается SNMP Trap. Третья команда устанавливает пороговые значения: верхняя граница – 80 % и нижняя граница – 20 % (5 – это интервал запроса значения загрузки CPU).

Задание уведомления о критическом превышении доступной памяти выполняется аналогичным образом. При этом сигнал тревоги посылается оператору, когда в маршрутизаторе остается меньше 20 Кбайт свободной процессорной памяти или памяти ввода/вывода:

```
Router(config)# memory free low-watermark processor 20000
```

или

```
Router(config)# memory free low-watermark io 20000
```

С сигнализацией о нехватке памяти связан механизм выделения определённого объёма памяти под критичные задачи (например, под регистрацию событий). Это позволяет быть уверенным, что важная операция всё равно будет произведена даже при условии нехватки памяти. При этом резервируемая память не должна превышать 25 % от всего объёма доступной памяти:

```
Router(config)# memory reserve critical 1000
```

4.2.6 IOS Software Image Verification

Ещё одним видом атаки на сетевое оборудование является встраивание «чужого» кода в Cisco IOS. Но с самого начала целостность кода, загружаемого на маршрутизатор, можно проверять – достаточно сравнить вычисленный хэш-код образа IOS, установленного на устройство, с хэш-кодом, показанном на сайте cisco.com. Однако не многие пользователи осуществляли такую проверку, ссылаясь на сложность и длительность процедуры. Чтобы облегчить такую «непростую» задачу, у пользователей в версии 12.0(26)S появилась команда verify, которую можно и удобно использовать в трёх случаях [5]:

1 Глобальная и автоматическая проверка целостности образа (после любой попытки копирования или перезагрузки):

```
Router(config)# file verify auto
```

2 Проверка целостности образа после копирования из какого-либо источника:

```
Router(config)# copy /verify tftp://10.1.1.1/jdoe/-c7200-js-mz disk0:
```

3 Проверка целостности образа после перезагрузки устройства:

```
Router# reload /verify
```

4.2.7 Flexible Packet Matching

Многие слышали о том, что в маршрутизаторы Cisco встроена система предотвращения атак Cisco IOS IPS. Но очень мало кто знает о Flexible Packet Matching, которая позволяет описывать и обнаруживать любые интересующие события, например, атаки, для которых ещё никто не написал сигнатуры. Делается это с помощью XML, который позволяет описать любые поля заголовка пакета и тела данных любого протокола. Для наиболее распространённых из них существуют специальные файлы описания заголовка протокола – Protocol Header Definition File, PHDF [5].

4.2.8. Advanced Application Inspection and Control

Решения Cisco давно вышли из определения, данного в любом компьютерном словаре термину «маршрутизатор». Например, когда говорят о контроле доступа к защищаемым ресурсам (внешним или внутренним), то обычно первое, что приходит в голову, – списки контроля доступа (Access Control List, ACL), существующие в любом маршрутизаторе. Однако, когда речь заходит о контроле прикладного трафика (например, блокировании Instant Messaging или P2P), то многие обращаются к механизмам прикладного уровня. Однако в маршрутизаторах Cisco реализованы и такие функции защиты. В дополнение к описанному выше Flexible Packet Matching или известному механизму Network-Based Application Recognition (NBAR) для контроля того же прикладного трафика можно использовать команду ip inspect.

Ниже приведён фрагмент конфигурации для дополнительной проверки популярных протоколов на соответствие политике безопасности [5].

```
ip inspect name my-ios-fw http
ip inspect name my-ios-fw https
ip inspect name my-ios-fw esmtp
ip inspect name my-ios-fw pop3
ip inspect name my-ios-fw imap3
ip inspect name my-ios-fw dns
ip inspect name my-ios-fw ftp
ip inspect name my-ios-fw ntp
ip inspect name my-ios-fw icmp
```

Для не столь популярных протоколов ситуация сильно не меняется – надо добавить всего одну команду:

```
ip port-map user-vnc port tcp 5900
ip inspect name my-ios-fw user-vnc
```

После этого можно применить данные правила к нужному интерфейсу маршрутизатора:

```
interface fastethernet 0/1
ip inspect my-ios-fw in
```

А для инспекции разрешённого трафика, внутри которого может скрываться запрещённый трафик (именно так часто инкапсулируется Instant Messaging или P2P), достаточно использовать следующие команды:

```
appfw policy-name abuse-control
application http
port-misuse default action reset alarm
ip inspect name my-ios-fw appfw abuse-control
```

4.2.9 IP Source Tracker

Итак, имеется достаточное количество механизмов обнаружения и отражения атак и другой подозрительной активности. Когда пришёл сигнал о попытке несанкционированного доступа, необходимо быстро отследить источник атаки и собрать доказательства его вредоносной деятельности. Особенно важно сделать это при подмене адреса, когда не известно, с какого интерфейса маршрутизатора пришёл вредоносный трафик и куда двигаться в дальнейшем расследовании. Для решения этой задачи можно использовать механизмы маршрутизаторов Cisco IOS: ACL, NetFlow, uRPF и т. д. Но наиболее эффективный способ – задействование специальной функции IP Source Tracker [5]. Фрагмент конфигурации будет выглядеть следующим образом:

```
Router(config)# ip source-track 192.168.1.1
```

Эта команда позволяет отслеживать трафик, получаемый с адреса 192.168.1.1. Посмотреть статистику по данному адресу можно командой

```
show ip source-track
```

4.2.10 VPN

В настоящее время технологии построения виртуальных защищённых частных сетей (VPN) широко используются крупными компаниями для защиты сетей (банками, ведомствами, государственными структурами и т. д.). Причина такого интереса заключается в том, что VPN-технологии действительно дают возможность не только существенно сократить расходы на содержание выделенных каналов связи с удалёнными подразделениями (филиалами), но и повысить конфиденциальность обмена информацией.

VPN-технологии позволяют организовывать защищённые туннели как между офисами компании, так и к отдельным рабочим станциям и серверам. Потенциальным клиентам предлагается широкий спектр оборудования и ПО для

создания виртуальных защищённых сетей – от интегрированных многофункциональных и специализированных устройств до чисто программных продуктов.

Поскольку вся информация, исходящая из локальной сети, проходит через маршрутизатор, то вполне естественно возложить на него и задачи создания защищённых каналов. Ярким примером оборудования для построения VPN на маршрутизаторах (рисунок 4.2) является оборудование компании Cisco Systems. Начиная с версии программного обеспечения IOS 11.3(3)T, маршрутизаторы Cisco поддерживают протоколы L2TP и IPSec. Помимо простого шифрования проходящей информации Cisco поддерживает и другие функции VPN, такие как идентификация при установлении туннельного соединения и обмен ключами.

Для построения VPN Cisco использует туннелирование с шифрованием любого IP-потока. При этом туннель может быть установлен, основываясь на адресах источника и приёмника, номера порта TCP(UDP) и указанного качества сервиса (QoS).

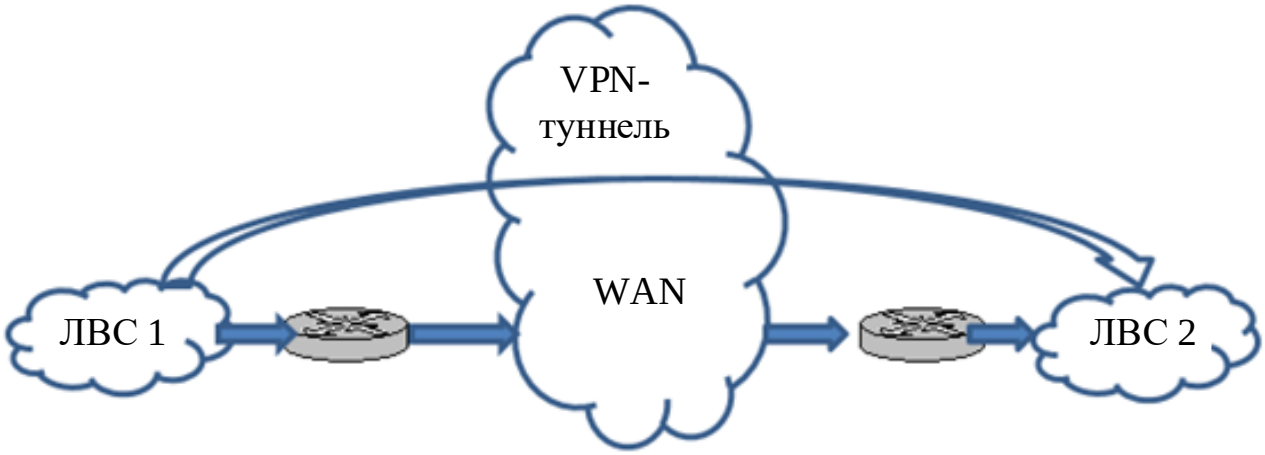


Рисунок 4.2 – VPN на базе маршрутизаторов

5 СИСТЕМЫ МОНИТОРИНГА БЕЗОПАСНОСТИ В ИНФОКОММУНИКАЦИЯХ

5.1 Архитектура систем мониторинга информационной безопасности

В последнее время системы мониторинга (СМ) информационной безопасности (ИБ) выделились в самостоятельное направление создания средств обеспечения защиты информации (ЗИ) в инфокоммуникациях. Предпосылками такого выделения являются:

– усложнение топологии инфокоммуникационных систем (ИКС);

– увеличение числа объектов ИКС;

– разнообразие факторов, воздействующих на объекты ИКС;

– усложнение атак на ИКС.

Влияние этих факторов потребовало развития механизмов сбора и обработки огромного объёма данных о событиях безопасности. Причём функциональный состав СМ ИБ определялся эволюционным характером их развития от SIM (Security Information Management) до SIEM (Security Information and Event Management) [4].

SIM предназначены для сбора событий преимущественно с сетевого оборудования. Основной задачей SIM-решения является анализ событий ИБ и отправка уведомлений о нарушениях в масштабе времени, близком к реальному.

SEM (Security Event Management) позволяют консолидировать и долгосрочно хранить события от систем информационной безопасности и приложений, осуществляют ретроспективный анализ и расследование инцидентов ИБ, анализ пользовательской активности, а также контроль над соблюдением политик и требований по ИБ.

SIEM (Security Information and Event Management) – решения, которые обрабатывают события безопасности, поступающие как от оборудования, так и от приложений различного уровня.

Важно понимать, что SIEM-системы в качестве самостоятельного (standalone) решения не предназначены и не способны предотвращать инциденты нарушения информационной безопасности. Их сущность заложена в их названии: анализ информации, поступающей из различных источников (DLP, IDS, антивирусы, межсетевые экраны и т. д.), и дальнейшее выявление отклонений от норм по заданным критериям.

В процессе функционирования система мониторинга и управления осуществляет сбор первичных данных мониторинга информационной безопасности с наблюдаемых подконтрольных объектов (ПКО) и производит их анализ. При этом мониторинг информационной безопасности может быть как активным, так и пассивным.

Активный мониторинг характеризуется тем, что на события, которые нарушают заданную политику безопасности, заранее определено воздействие, которое, как ожидается, должно привести к решению возникшей проблемы.

Пассивный мониторинг предполагает сбор информации с удалённых источников в режиме чтения. В случае выходов значений параметров за пределы, определённые как «нормальные», оператор предпринимает соответствующие

шаги для устранения возникшей ситуации и нормализации параметров. Такой вид аудита наиболее прост в реализации, но обеспечивает менее высокий уровень защиты по сравнению с активными методами.

По архитектуре построения СМ ИБ не имеют существенных различий и в общем виде могут быть представлены следующими структурными уровнями (рисунок 5.1).

Рисунок 5.1 – Уровни системы мониторинга

На уровне данных осуществляется сбор данных о событиях безопасности, их обобщение, нормализация и предварительная корреляция.

Уровень событий отвечает за распространение информационных потоков событий безопасности между потребителями в реальном времени. При этом следует отметить, что восходящий информационный поток, идущий от уровня данных к прикладному уровню, является более интенсивным, чем противоположный.

Прикладной уровень осуществляет обработку событий безопасности, моделирование, поддержку решений и реагирование, визуализацию, хранение событий в репозитории.

Данные о событиях безопасности формируются на уровне защищаемой инфраструктуры, подлежат предварительной обработке на уровне данных, распространяются с помощью уровня событий к требуемым элементам прикладного уровня и в конечном итоге окончательно обрабатываются последними элементами.

В соответствии со спецификой контроля и обслуживания объекты мониторинга также можно распределить на три уровня.

На нижнем, аппаратном уровне осуществляется сбор и отображение данных об отдельных узлах сети, их аппаратном обеспечении и операционных системах; проверяется доступность их по сети, загруженность процессоров, оперативной и дисковой памяти, состояние источников питания, температурный режим.

На сетевом уровне рассматриваются устройства и службы, обеспечивающие работу локальной сети: состояние памяти и загрузки процессоров коммутаторов, характеристики их портов; состояние внешнего канала и доступность необходимых для работы сетей.

На верхнем уровне – уровне служб – осуществляется контроль работы служб, предоставляемых конечным пользователям (персоналу, использующему подконтрольные системе мониторинга ресурсы для решения стоящих перед ним задач).

Причём между объектами указанных уровней могут существовать зависимости, например, корректная работа службы доступа к файлам (уровень служб) зависит от состояния компьютеров, предоставляющих для неё дисковое пространство (аппаратный уровень) и коммутатора, обслуживающего сегмент сети, в котором они располагаются (сетевой уровень) [6].

По функциональному назначению программные модули каждого структурного уровня разделяются на две категории [6]:

1) реализующие внутренние процессы самой системы мониторинга:

– информационные потоки между подсистемами;

– механизмы передачи данных, запуска процедур сбора, оповещения, сохранения и архивации данных;

– функции обработки данных;

2) осуществляющие взаимодействие с внешними объектами (их реализация зависит от вида этих объектов и предоставляемых ими протоколов взаимодействия или интерфейсов).

Функции и модули первой категории должны быть реализованы максимально просто и универсально. Их совокупность обычно называют ядром системы мониторинга. Ядро должно предоставлять пользователю, сопровождающему систему, гибкий инструментарий для построения оптимального решения стоящих перед ним задач, связанных с конкретным набором внешних объектов. С другой стороны, система должна позволять легко менять конкретную реализацию функций и модулей второй категории (и добавлять новые) в соответствии со стоящей задачей. Сопровождающий систему пользователь должен иметь возможность реализовать и использовать в системе собственные модули оповещения, вывода, сбора и анализа данных, работающие в рамках поддерживаемой и регламентируемой ядром схемы.

Подобные системы позволяют осуществлять мониторинг объектов любой природы – при условии разработки программного модуля, поставляющего интересующие данные, и выполнять в критической ситуации любые действия – в случае разработки модуля, реализующего при выполнении некоторого условия требуемую последовательность операций. На рисунке 5.2 приведена структурная схема такой модульной системы мониторинга [6].

Ядро системы составляют компоненты, реализующие её основные внутренние механизмы (получение и накопление данных об объектах мониторинга произвольной природы, диагностика нештатных ситуаций и оповещение о них, принятие решения о надлежащей реакции на обнаруженную нештатную ситуацию).

Компоненты ядра осуществляют вызов необходимых подключаемых модулей, реализующих решения частных задач, соответствующих области применения системы (мониторинг конкретных типов объектов и служб, взаимодействие с системами хранения данных, отображение состояния системы в различных форматах и т. п.). Функции подсистемы вывода целиком вынесены из ядра как не требующие взаимодействия с самими процессами мониторинга, а взаимодействующие только с поставляемыми системой хранения накопленными данными (как об объектах мониторинга, так и о работе самой системы).

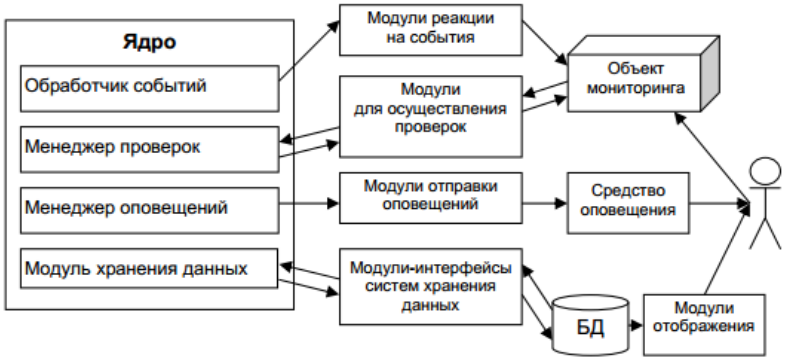


Рисунок 5.2 – Структура модульной системы мониторинга

По методам получения данных системы мониторинга разделяются на две группы: централизованные и децентрализованные. Централизованный подход подразумевает сбор и обработку данных об объектах мониторинга в одном определённом узле сети (сервере мониторинга). При децентрализованном подходе распределение этих действий осуществляется между несколькими узлами, например, с использованием автономных агентов (программ, работающих на объектах мониторинга независимо от сервера мониторинга, самостоятельно принимающих решения об отправке данных на сервер).

Другая концепция получения данных от объектов – это так называемые пассивные проверки (Passive Checks), при которых инициатива проверки принадлежит не ядру мониторинга, а самому программному модулю, который непрерывно работает и посылает данные на сервер мониторинга только при выполнении некоторых условий (например, при обнаружении нештатной ситуации) [6].

5.2 Типовой компонентный состав и перечень реализуемых функций систем мониторинга

Функциональная модель общей архитектуры системы мониторинга представлена на рисунке 5.3, на котором отображено распространение информационных потоков через уровни архитектуры и её элементы. Информация о событиях информационной безопасности собирается в граничных узлах и распространяется к прикладным сервисам. Под событиями информационной безопасности (СИБ) в

ИКС подразумеваются любые непредвиденные или нежелательные события, которые непосредственно или в совокупности с другими событиями нарушают (или могут нарушить) требования политики информационной безопасности, а именно:

– несанкционированное уничтожение и изменение информации;

– несанкционированное блокирование средств, обеспечивающих доступ к информации;

– создание нештатных режимов работы программных (программно-аппаратных) средств, в том числе и средств защиты;

– несанкционированное копирование данных и навязывание ложной информации.

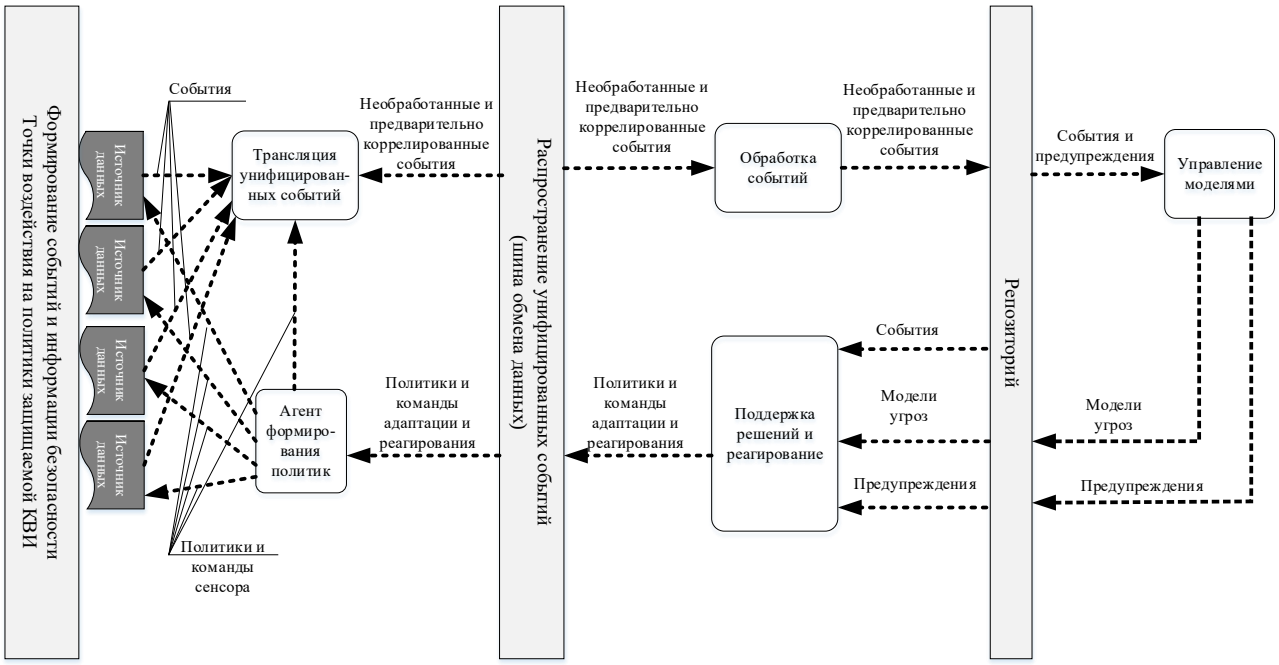


Рисунок 5.3 – Функциональная модель общей архитектуры СМ ИБ

Механизм обработки событий выполняет корреляцию событий, выделяемых из потока информации случайно или путём предварительной обработки, и помещает их на хранение в репозиторий.

Репозиторий обеспечивает непосредственное взаимодействие с другими прикладными модулями. Сервисы управления моделями выполняют моделирование поведения системы и вырабатывают дополнительные модели угроз и предупреждения безопасности, которые возвращаются обратно в репозиторий. Сервисы поддержки решений и реагирования анализируют входящие события. Модели угроз и предупреждения безопасности вырабатывают реакцию и контрмеры, приводящие к модификации политик безопасности, которые посылаются обратно к граничным узлам и воздействуют на удалённые источники данных, агенты и модули предварительной обработки событий безопасности.

Сбор событий [6]. Подсистема сбора данных осуществляет опрос объектов мониторинга с заданными временными интервалами для получения значений исследуемых параметров этих объектов. Может также включать в себя первичный

анализ полученных данных с целью, например, квалификации полученных значений как нормальных, требующих вмешательства оператора либо критических.

Обработка событий [6]. Подсистема анализа данных включает компоненты, производящие исследования данных, накопленных системой, их статистический анализ, нахождение корреляционного отношения величин и другие операции. Обработка событий осуществляется в масштабируемом модуле управления корреляцией, который ориентирован на систему параллельной обработки сложных событий, способную объединять вычислительные мощности для обработки большого количества событий в секунду и регулировать количество выделенных ресурсов для заданной системы. Поведение этого модуля может настраиваться через запросы, которые создаются из определяемых пользователем стандартных директив. Запросы определяют, каким образом следует абстрагировать, трансформировать, обобщать и коррелировать входные события. Запрос состоит из операторов.

Визуализация [6]. Визуализация данных о событиях безопасности, а также о решениях по её обеспечению является достаточно важной функцией систем мониторинга. Подсистема вывода отвечает за представление информации о работе системы и результатов проверок в виде, удобном для восприятия пользователем, причём независимо от его местонахождения и используемой операционной системы. Данное требование обуславливает реализацию подсистемы вывода в виде веб-интерфейса. Поскольку количество объектов мониторинга и объёмы собранных данных могут быть весьма большими, необходимо предусмотреть генерацию различных типов отчётов и сводок (таблицы, графики, секторные и иные диаграммы), а сам веб-интерфейс должен предоставлять средства удобной навигации и поиска необходимых данных.

Репозиторий [6]. Репозиторий является средством кросс-платформенной интеграции различных компонентов систем мониторинга. Подсистема хранения отвечает за накопление, хранение, архивацию данных о результатах проверок. Включает компоненты для работы с базами данных (БД) или иными репозиториями, программные средства сжатия данных для уменьшения объёма хранимой информации и т. п. В качестве основы для его реализации предлагается сервис-ориентированная архитектура (СОА), представляющая собой концепцию распределённой информационной среды, объединяющей модули программного обеспечения и приложений, основанные на интерфейсах и взаимодействиях между ними.

Уведомления [6]. Подсистема оповещения отвечает за уведомление лиц, ответственных за функционирование проверяемых объектов и самой системы мониторинга, о нештатных ситуациях и других значимых изменениях состояний объектов.

Подсистемы анализа данных, хранения, сбора данных решают задачи, относящиеся к фоновому (пассивному) мониторингу, т. е. систематическому долговременному накоплению, классификации и анализу данных о работе объектов мониторинга, не подразумевающему какую-либо реакцию на получаемые данные.

Все подсистемы в совокупности ориентированы на оперативный (активный) мониторинг, т. е. направленный на оценку текущей работоспособности и эффектив-

ности исследуемых объектов (как с помощью пользователя СМ, так и автоматически), а также на немедленную реакцию на обнаруженные нештатные ситуации.

Для обеспечения получения наиболее полной информации о событиях, происходящих в защищаемой системе и касающиеся её безопасности, реализуются следующие функции:

1 Фиксация в оперативном журнале результатов контроля целостности компонентов на ПКО, а также фиксация в собственном журнале фактов изменения настроек и попыток несанкционированного доступа к ПО системы мониторинга и управления. При этом контроль целостности собственного программного обеспечения и настроек, а также программного обеспечения, указанного как «объекты контроля» системы, обеспечивается с помощью возможностей систем защиты информации (СЗИ) от несанкционированного доступа (НСД), функционирующих в составе ПКО, в том числе серверного оборудования системы управления и мониторинга СЗИ от НСД.

2 Возможность просмотра настроек СЗИ от НСД на ПКО средствами ПО системы.

3 Удалённое управление процессами контроля целостности файловой системы ПКО средствами СЗИ от НСД.

4 Доставка и обновление ПО в соответствии с заданными политиками или по запросу пользователей. С использованием данного функционала администратор может:

– задавать требуемые политики для различных приложений, которые необходимо в дальнейшем будет рассылать на ПКО с возможностью получения от ПКО информации о версии установленных на них СЗИ от НСД;

– составлять расписание рассылки ПО (например, ежедневно/еженедельно/в фиксированное время/по изменению и т. д.);

– задавать каталог на ПКО, в который необходимо установить или откуда следует загрузить комплект ПО;

– задавать политику, применяемую в случае неудачной установки/доставки ПО на ПКО.

5 Повышение удобства использования системы мониторинга и управления СЗИ от НСД для её персонала в части:

– возможности назначения различных типов оповещений для разных видов событий ИБ. Система мониторинга и управления СЗИ от НСД должна предусматривать возможность задания приоритетов для обнаруживаемых атак или уязвимостей;

– возможность индивидуальной настройки эксплуатирующим персоналом системы параметров отображения оперативных журналов, получаемых от ПКО.

Источниками информации в системах мониторинга являются:

1) аппаратно-программные комплексы обнаружения компьютерных атак;

2) хостовые агенты мониторинга параметров объектов операционной системы, которые устанавливаются на хостах – автоматизированных рабочих местах (АРМ), в том числе изолированных от сети – и серверах, требующих мониторинга;

3) анализаторы текстовых журналов аудита и сообщений, обеспечивающие получение и анализ информации из указанных источников по настраиваемым алгоритмам анализа.

Рассмотрим принципы работы SIEM-систем.

На практике схема реализуется с помощью соответствующих компонентов:

– агенты (сбор данных из различных источников);

– серверы-коллекторы (аккумуляция информации, поступившей от агентов);

– сервер баз данных (хранение информации);

– сервер корреляции (анализ информации).

Входной информацией для SIEM-систем может служить практически любая информация, главное – правильно её подать. Как уже было сказано выше, сбор данных может осуществляться с помощью специальных агентов, которые представляют собой программу, которая локально собирает журналы событий и по возможности передаёт их на сервер. Для обработки событий от того или иного источника данных агент использует коллекторы – библиотеки для понимания конкретного журнала событий или системы. Коллекторы играют важную роль, т. к. разные источники могут именовать одно и то же событие по-своему. Например, Firewall одного производителя может записывать в отчёт deny, другого discard, третьего drop, хотя событие одно и то же. Коллекторы помогают привести все эти события к общему знаменателю. Если же для источника нет соответствующего коллектора, события можно попробовать отправлять как SYSLOG (при условии, что источник умеет это делать). Однако и здесь можно столкнуться с «проблемой синонимов» и необходимостью писать дополнительный обработчик для приведения данных в единый формат. Также информацию можно собирать удалённо при помощи соединения по протоколам NetBIOS, RPC, TFTP, FTP. Однако в этом случае может возникнуть проблема с нагрузкой на сеть, т. к. часть систем позволяет передавать только журнал целиком, а не актуальные записи. SIEM-системы могут использовать следующие источники информации:

– Access Control Authentication применяются для мониторинга контроля доступа к информационным системам и использования привилегий;

– DLP-системы содержат сведения о попытках инсайдерских утечек, нарушении прав доступа;

– IDS/IPS-системы несут данные о сетевых атаках, изменениях конфигурации и доступа к устройствам;

– антивирусные приложения генерируют события о работоспособности ПО, базах данных, изменении конфигураций и политик, вредоносном коде;

– журналы событий серверов и рабочих станций применяются для контроля доступа, обеспечения непрерывности, соблюдения политик информационной безопасности;

– межсетевые экраны содержат сведения об атаках, вредоносном ПО и прочем;

– сетевое активное оборудование используется для контроля доступа, учёта сетевого трафика;

– сканеры уязвимостей предоставляют данные об инвентаризации активов, сервисов, программного обеспечения, уязвимостей, поставках инвентаризационных данных и топологической структуры;

– системы инвентаризации и asset-management поставляют данные для контроля активов в инфраструктуре и выявления новых;

– системы веб-фильтрации предоставляют данные о посещении сотрудниками подозрительных или запрещённых веб-сайтов.

Получив информацию, система может её проанализировать. В основе анализа лежит практически «чистая» математика и статистика, но отправной точкой служат задаваемые вручную правила. К примеру, однократное событие «login failed» ничего не значит, в то время как три и более таких события от одной учётной записи уже могут свидетельствовать о попытках подбора пароля.

В простейшем случае в SIEM-системах правила представлены в формате RBR (Rule Based Reasoning) и содержат набор условий, триггеры, счётчики, сценарии действий. Например, учитывать параметры удалённости двух последних точек использования банковской карты за небольшой интервал времени: если в 17:00 её использовали для оплаты кофе в Москве, а через 10 мин пытаются снять дневной лимит в Гонконге, то на лицо – попытка мошенничества.

SIEM-системы способны выявлять:

– сетевые атаки во внутреннем и внешнем периметрах;

– вирусные эпидемии или отдельные вирусные заражения;

– попытки несанкционированного доступа к конфиденциальной информации;

– мошенничество;

– ошибки и сбои в работе информационных систем;

– уязвимости;

– ошибки конфигураций в средствах защиты и информационных системах;

– целевые атаки (APT).

Последовательность процессов функционирования SIEM показана на рисунке 5.4.

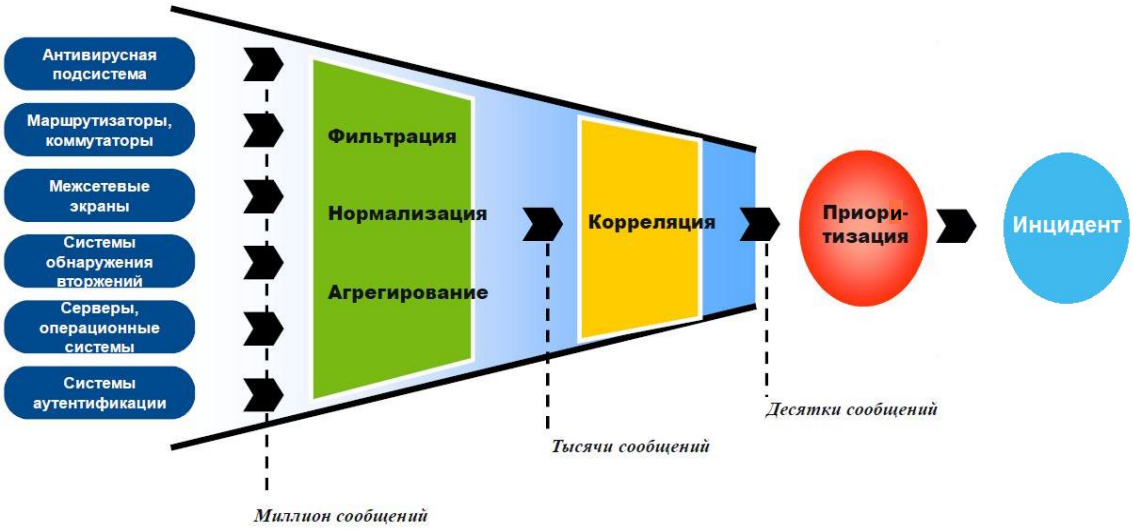


Рисунок 5.4 – Последовательность процесса выявления инцидентов ИБ

Таким образом, алгоритм функционирования SIEM включает в себя следующие операции:

1) сбор лог-файлов и обработка данных, поступающих от объектов мониторинга;

2) нормализация данных путём приведения сообщений о событиях к общему формату;

3) фильтрация и агрегация событий безопасности по заранее установленным признакам;

4) корреляция событий безопасности, заключающаяся в установлении взаимосвязей между разнородными событиями;

5) приоритизация событий безопасности – определение значимости и критичности событий на основании заданных правил;

6) оповещение об инцидентах информационной безопасности;

7) организация хранения лог-файлов и событий безопасности;

8) формирование отчётных документов.

Структура и взаимодействие компонентов типовой SIEM системы представлены на рисунке 5.5.

Типовая SIEM-система содержит следующие основные компоненты:

– хостовая система обнаружения вторжений (OSSEC [7]), которая осуществляет сбор и передачу лог-файлов из журналов событий операционных систем и прикладного ПО для анализа и выявления инцидентов информационной безопасности;

– сетевая система обнаружения вторжений (Suricata [8]), которая отслеживает сетевые вторжения, проверяет сетевой трафик и ведёт наблюдение за контролируемыми хостами;

– средство мониторинга доступности узлов сети (Nagios [6]) проверяет доступность хостов и сервисов, а также генерирует оповещения в зависимости от поведения контролируемых хостов и служб;

– сканер уязвимостей, осуществляющий активный мониторинг узлов вычислительной сети на предмет наличия проблем, связанных с информационной безопасностью;

– система обмена информацией о сетевых угрозах (OTX [10]), которая позволяет обмениваться информацией об угрозах между пользователями, зарегистрированными в международной базе данных Open Threat Exchange.

Типовая SIEM включает в себя серверную и клиентскую части.

В состав серверной части входят:

1) сенсор с программными обработчиками (далее – плагинами);

2) модуль корреляции;

3) база данных;

4) веб-интерфейс;

5) интерфейс командной строки;

6) программные компоненты с открытым исходным кодом:

– OSSEC – хостовая система обнаружения вторжений (HIDS);

– Suricata – сетевая система обнаружения вторжений (NIDS);

– Nagios – мониторинг доступности узлов сети;

– Nesus [9] – сканер уязвимостей;

– OTX – система для обмена информацией о сетевых угрозах;

– Snare – журнал для Windows-систем.

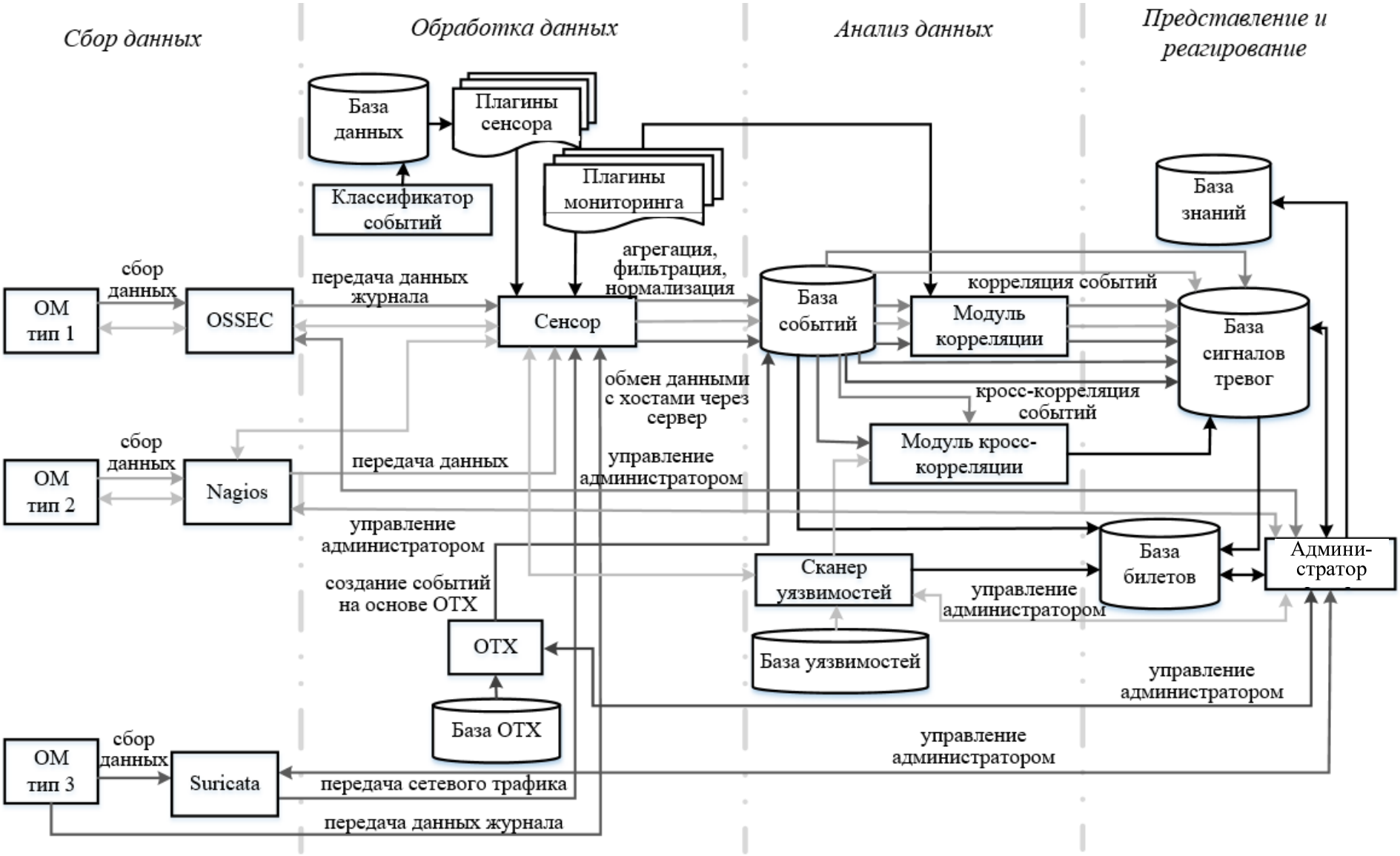


Рисунок 5.5 – Структура и взаимодействие компонентов типовой SIEM

В серверной части осуществляется сбор и обработка следующих данных:

– о событиях безопасности, получаемых с помощью программных агентов (OSSEC);

– об использовании ресурсов, получаемых с помощью программных агентов (Nagios);

– журналов событий, получаемых с помощью программных агентов (Snare);

– из лог-файлов коммутаторов сети;

– сетевого трафика сети.

В состав клиентской части типовой SIEM входят:

– программные агенты OSSEC, установленные на АРМ и серверах и осуществляющие в фоновом режиме сбор и передачу событий безопасности, получаемых из журналов событий контролируемых ими систем, на серверную часть;

– программные агенты Nagios, установленные на серверах и осуществляющие в фоновом режиме сбор и передачу событий об использовании вычислительных ресурсов контролируемых ими систем (загруженность процессора и памяти, доступность, производительность, свободное дисковое пространство) на серверную часть;

– программные агенты Snare, установленные на серверах и осуществляющие в фоновом режиме сбор и передачу данных журналов событий контролируемых ими систем на серверную часть;

– ПО коммутаторов сети, поддерживающее функцию логирования событий и передачу лог-файлов на серверную часть по протоколу Syslog.

Для осуществления функции администрирования и отображения результатов мониторинга предусмотрено подключение автоматизированного рабочего места администратора безопасности к серверной части.

5.3 Системы обнаружения и предотвращения вторжений

В настоящее время традиционные средства защиты, такие как межсетевой экран и антивирус, не способны обеспечить надлежащий уровень защиты внутренней сети организации, ведь вредоносное программное обеспечение может «замаскироваться» и отправлять пакеты, которые с точки зрения межсетевого экрана выглядят полностью легитимными. Вместе с тем существует класс решений, способных обеспечить надлежащий уровень защиты внутренней сети организации – это системы обнаружения и предотвращения вторжений. В англоязычной литературе – Intrusion Detection Systems (IDS) и Intrusion Prevention Systems (IPS) [1].

Различия между ними заключаются лишь в том, что одна может автоматически блокировать атаки, а другая просто предупреждает об этом. Данный класс средств защиты относится к методу отслеживания несанкционированных попыток получения доступа к защищаемым ресурсам организации, называемый мониторингом управления доступом. Он нацелен на выявление и регистрацию недостатков в безопасности внутренней инфраструктуры – сетевые атаки, попытки

несанкционированного доступа или повышения привилегий, работа вредоносного программного обеспечения и т. д. Таким образом, по сравнению с межсетевым экраном, контролирующим только параметры сессии, IDS и IPS анализируют передаваемые внутренние потоки данных, находя в них последовательности битов, которые могут представлять из себя вредоносные действия или события. Помимо этого, они могут осуществлять мониторинг системных журналов и других файлов регистрации деятельности пользователей.

Итак, **IDS** – система обнаружения вторжений, предназначенная для регистрации подозрительных действий в сети и уведомления о них ответственного за информационную безопасность сотрудника с помощью передачи сообщения на консоль управления, отправки электронного письма, SMS-сообщения на мобильный телефон и т. п.

Традиционная IDS состоит из сенсоров (просматривают сетевой трафик или журналы и передают информацию анализаторам), определяющих в полученных данных вредоносный характер, и в случае успешного обнаружения отправляет результаты в административный интерфейс. В зависимости от места расположения традиционные IDS делятся на сетевые (Network-based IDS (NIDS)) и хостовые (Host-based (HIDS)). Для более понятной классификации IDS необходимо выделить ещё два подмножества, которые делятся по типу анализируемого трафика: IDS, основанная на протоколе (Protocol-based IDS (PIDS)), которая анализирует коммуникационные протоколы со связанными системами или пользователями, а также IDS, основанная на прикладных протоколах (Application Protocol-based IDS (APIDS)), предназначенная для анализа данных, передаваемых с использованием специфичных для определённых приложений протоколов.

Вредоносную активность в анализируемом трафике можно обнаружить разными способами. Поэтому в IDS существуют следующие характеристики, отличающие друг от друга различные типы технологий IDS, которые можно описать следующим образом.

Сигнатурные IDS отслеживают определённые шаблоны в трафике и работают подобно антивирусному программному обеспечению. Недостатки данного подхода: сигнатуры должны быть в актуальном состоянии и IDS подобного типа не способны выявить незнакомые атаки.

Сигнатурные IDS также можно разделить на два вида:

– отслеживающие шаблоны – сравнивают сетевые пакеты с сигнатурами;

– отслеживающие состояние – сравнивают действия с шаблонами (здесь первоначальное состояние – перед началом атаки, а скомпрометированное состояние – после осуществления атаки, т. е. успешное заражение).

IDS, основанные на аномалиях. Данный тип IDS не использует сигнатур. Он основан на поведении системы и перед началом работы происходит этап обучения «нормальной» деятельности системы. Исходя из этого она впоследствии способна выявлять незнакомые атаки.

Аномалии, в свою очередь, в данной категории делятся на три класса:

1) статистические – IDS создаёт профиль штатной деятельности системы и сравнивает весь проходящий трафик и деятельность с этим профилем;

2) аномалии протоколов – IDS анализирует трафик с целью выявления фрагментов нелегитимного использования протоколов;

3) аномалии трафика – IDS выявляет нелегитимные действия в сетевом трафике.

IDS, основанные на правилах. Данные IDS используют программирование, основанное на правилах «ЕСЛИ ситуация, ТОГДА действие». IDS на основе правил похожи на экспертные системы, т. к. экспертная система представляет из себя совместную работу базы знаний, логических выводов и программирования на основе правил. В данном случае знания – это правила, а анализируемые данные можно назвать фактами, к которым применяются правила. Например: «ЕСЛИ пользователь administrator авторизовался в System1 И сделал изменение в File2, ЗАТЕМ запустил «Утилиту3», ТОГДА отправить уведомление», т. е. если пользователь зашел в систему (1) и сделал изменение в файле (2), а затем запустил утилиту (3), то отправить уведомление.

Стоит отметить, что IPS является подклассом IDS, поэтому основана на её методах обнаружения атак. IPS может работать как на уровне хоста (HIPS), так и на уровне сети (NIPS). Возможность предотвращения атак реализована за счёт того, что сетевая IPS, как правило, встраивается «в разрыв» сети и пропускает через себя весь трафик, а также имеет внешний интерфейс, на который приходит трафик и внутренний интерфейс, который пропускает трафик далее, если он признаётся безопасным. Существует также возможность работы с копией трафика в режиме мониторинга, но тогда теряется основной функционал данной системы.

IPS необходимо размещать таким образом, чтобы система могла наблюдать за подконтрольными ей сегментами сети. Чаще всего это выделенный компьютер, один интерфейс которого подключается после пограничных устройств и «смотрит» через них в незащищённые сети общего пользования (Интернет). Другой интерфейс IPS подключается на вход защищаемого сегмента, чтобы весь трафик проходил через систему и анализировался. В более сложных случаях защищаемых сегментов может быть несколько: в корпоративных сетях часто выделяют демилитаризованную зону (DMZ) с доступными из Интернета сервисами (рисунок 5.6).

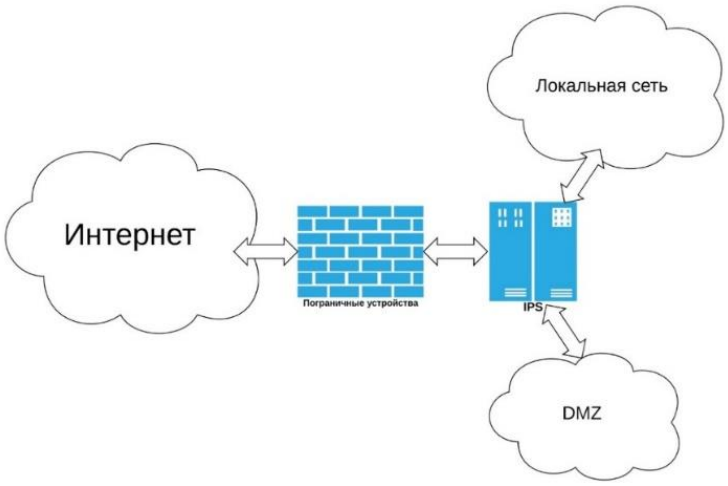


Рисунок 5.6 – Вариант расположения IPS

Такая IPS может предотвращать попытки сканирования портов или взлома с помощью перебора паролей, эксплуатацию уязвимостей в почтовом сервере, веб-сервере или в скриптах, а также другие разновидности внешних атак. В случае если компьютеры локальной сети будут заражены вредоносным ПО, IDS не позволит им связаться с расположенными снаружи серверами ботнета.

5.3.1 Хостовая система обнаружения вторжений OSSEC

Хостовая система обнаружения вторжений OSSEC является одним из программных компонентов SIEM, который предназначен для сбора и отображения данных о доступности объектов мониторинга, целостности системы и мониторинге политик. Как уже указывалось ранее, данный компонент имеет серверную часть (далее – OSSEC-сервер) и клиентскую часть на базе программных агентов OSSEC (далее – агенты OSSEC). OSSEC-сервер обеспечивает сбор и анализ данных, имеет веб-интерфейс, функционал которого встроен в веб-интерфейс SIEM. Агенты OSSEC устанавливаются на конечных узлах ПЭВМ (АРМ или серверах), которые собирают и пересылают лог-файлы из журналов событий операционных систем и прикладного ПО в шифрованном виде на OSSEC-сервер для анализа и выявления инцидентов безопасности.

Для анализа и выявления инцидентов безопасности OSSEC-сервер должен быть сконфигурирован и настроен на сбор и анализ следующих сообщений:

– сообщений журналов систем и сервисов, получаемых по протоколу syslog;

– сообщений о проверке целостности файлов и реестра Windows;

– сообщений о нарушении политики безопасности.

Настройка OSSEC-сервера заключается в выборе правил, по которым осуществляется сбор и анализ исходных данных, получаемых от объектов мониторинга. Правила обработки данных выбираются через веб-интерфейс OSSEC-сервера из списка «Disabled Rules» и добавляются в список «Enabled Rules». Например, список «Disabled Rules» может включать следующие правила обработки данных:

– alienvault-windows-workstation-logon-logoff-rules.xml;
– alienvault-windows-logon-logoff-rules.xml;
– alienvault-windows-password-change-rules.xml;
– alienvault-windows-shutdown-rules.xml;
– alienvault-windows-USB-rules.xml;
– alienvault-windows-FIM-rules.xml;
– alienvault-domain-rules.xml;
– alienvault-msqql-rules.xml;
– arpwatch-rules.xml;
– attack-rules.xml;
– local-rules.xml;
– ids-rules.xml;
– msauth-rules.xml;
– ossec-rules.xml.

Основные функции и особенности агента OSSEC:

– удалённое получение и обновление политик информационной безопасности, контролируемых агентом;

– контроль/принуждение к исполнению установленной политики информационной безопасности;

– синхронизация времени агента с временем управляющего сервера;

– надёжное скрытие всех элементов агента (файлов, используемых ключей системного реестра, сетевых соединений, процесса);

– возможность запуска сканирования по расписанию;

– удалённая визуализация и управление состоянием модулей агента (загрузка, выгрузка, запуск, остановка, смена текущей активной версии модуля);

– передача/получение сообщений через альтернативные маршруты (альтернативные транспортные серверы).

Настройка агентов OSSEC осуществляется автоматически при их установке на объекты мониторинга – АРМ и серверы.

OSSEC реализует функцию обнаружения вторжений для большинства операционных систем, включая Linux, OpenBSD, FreeBSD, Mac OS X, Solaris и Windows. Её кросс-платформенная архитектура позволяет легко управлять и наблюдать сразу несколько операционных систем. OSSEC ведёт очень подробный анализ логов, программа может сравнивать и анализировать логи одновременно нескольких приложений в нескольких форматах.

OSSEC-сервер работает и в распределённой сети, и автономно. Для включения серверного режима необходима установка хотя бы одной программы-агента. Веб-интерфейс, как и OSSEC-сервер, поддерживает ОС Unix, Solaris, BSD и Mac.

5.3.2 Сетевая система обнаружения вторжений Suricata

Suricata отслеживает сетевые вторжения, проверяет сетевой трафик и ведёт наблюдение за контролируемыми хостами. Пакет Suricata состоит из нескольких модулей – захвата, сбора, декодирования, обнаружения и вывода. Она получает доступ к сетевому трафику контролируемой сети от коммутационного оборудования, настроенного на «зеркалирование портов». Обработка сетевого трафика проходит в многопоточном режиме, в ходе которого данный компонент отслеживает сетевые вторжения, проверяет сетевой трафик и ведёт наблюдение за контролируемыми хостами.

По умолчанию до декодирования захваченный трафик идёт одним потоком, это оптимально с точки зрения детектирования, но больше нагружает систему. Но настройками можно переопределить такое поведение и указать, как будут распределяться потоки по процессорам после захвата. Это даёт широкие возможности для оптимизации обработки трафика на конкретном оборудовании в конкретной сети и позволяет осуществлять анализ потока со скоростью до 10 Гбит/с.

Одним из решений предотвращения вторжений в системы Suricata являются детекторы атак, которые предназначены для своевременного выявления множества вредоносных угроз. Работа детектора атак основана на анализе сигнатур и эвристике, что позволяет настраивать параметры работы системы для решения индивидуальных задач. Параметры работы детектора атак можно редактировать во вкладке настроек,

приведенной на рисунке 5.7. Здесь можно указать внутренние и внешние сети, диапазоны адресов различных серверов, а также используемые порты.

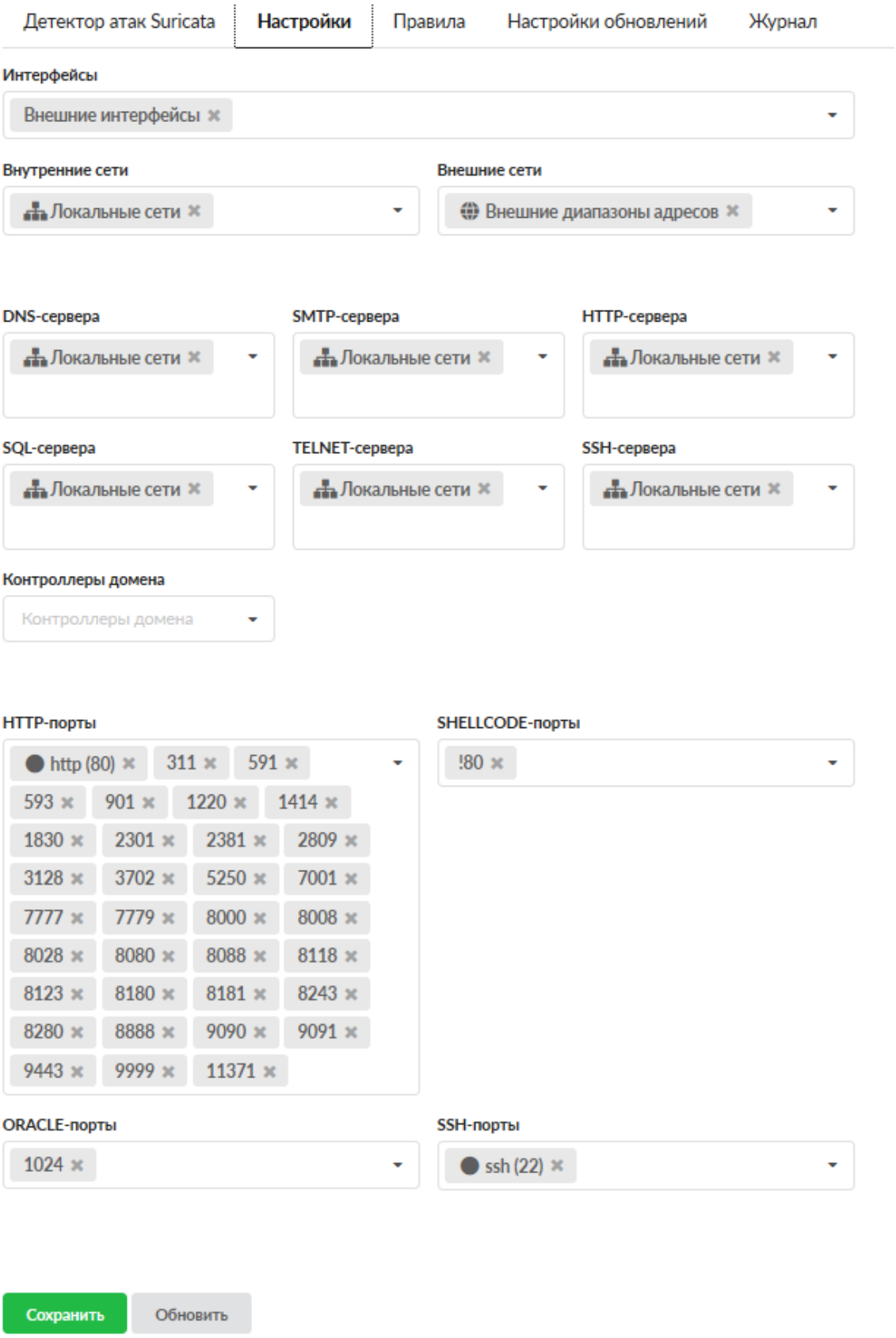
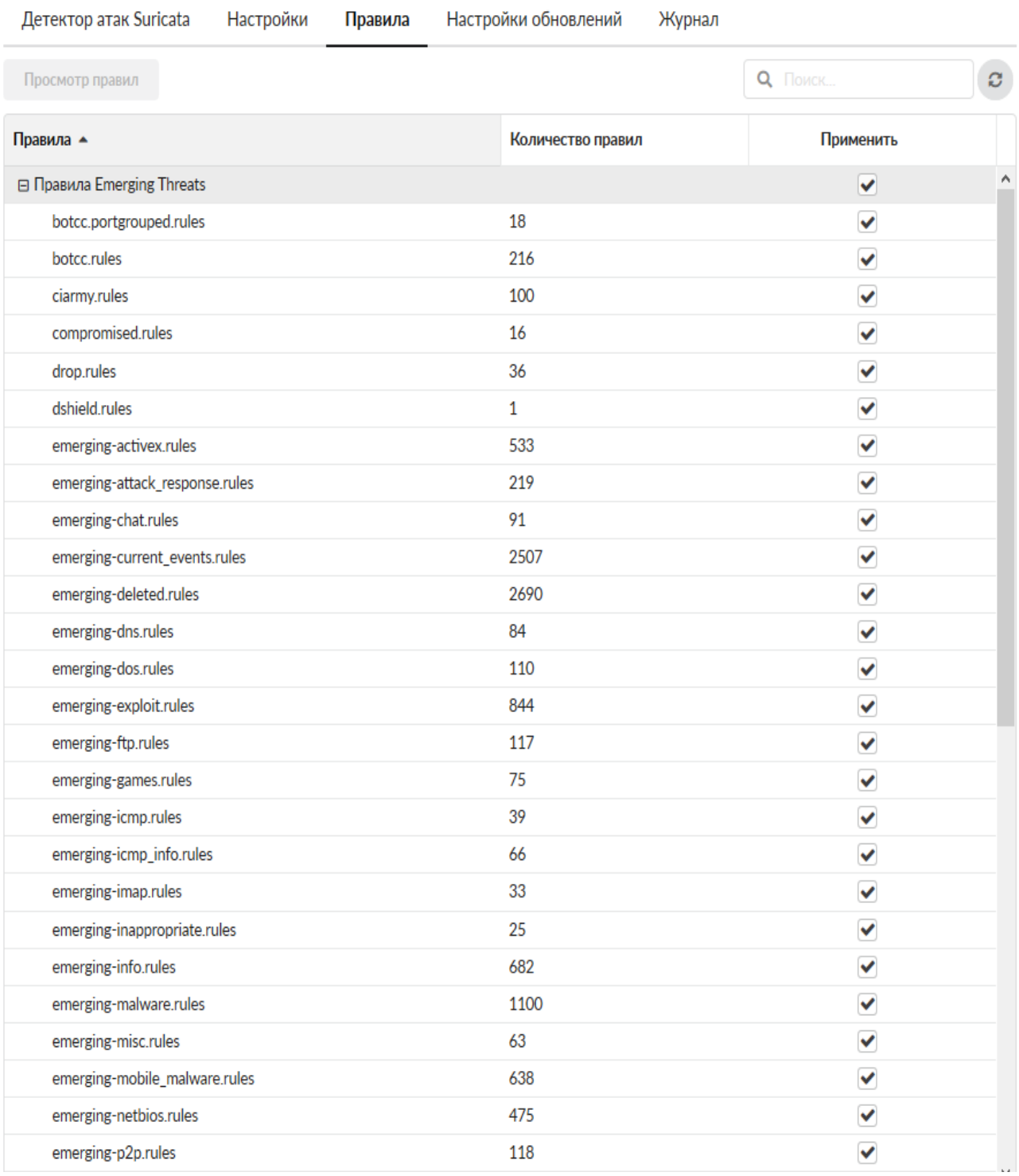


Рисунок 5.7 – Панель настроек пакета Suricata

К редактируемым параметрам Suricata также относятся правила, которым будет подчиняться анализ трафика, фильтры, ограничивающие вывод предупреждения администраторам, диапазоны адресов разных серверов, активные порты и сети. Пример подключения правил приведён на рисунке 5.8. На данной вкладке можно посмотреть наличие и содержимое того или иного файла с правилами, а также включить или выключить его действие (с помощью флажков справа). В правом верхнем углу располагается поиск по названию или по количеству правил в файле.



Правила	Количество правил	Применить
⊟ Правила Emerging Threats		☑
botcc.portgrouped.rules	18	☑
botcc.rules	216	☑
ciarmy.rules	100	☑
compromised.rules	16	☑
drop.rules	36	☑
dshield.rules	1	☑
emerging-activex.rules	533	☑
emerging-attack_response.rules	219	☑
emerging-chat.rules	91	☑
emerging-current_events.rules	2507	☑
emerging-deleted.rules	2690	☑
emerging-dns.rules	84	☑
emerging-dos.rules	110	☑
emerging-exploit.rules	844	☑
emerging-ftp.rules	117	☑
emerging-games.rules	75	☑
emerging-icmp.rules	39	☑
emerging-icmp_info.rules	66	☑
emerging-imap.rules	33	☑
emerging-inappropriate.rules	25	☑
emerging-info.rules	682	☑
emerging-malware.rules	1100	☑
emerging-misc.rules	63	☑
emerging-mobile_malware.rules	638	☑
emerging-netbios.rules	475	☑
emerging-p2p.rules	118	☑

Рисунок 5.8 – Панель подключения правил анализа трафика

К остальным техническим параметрам пакета Suricata необходимо отнести следующие возможности:

1 Поддерживаемые операционные системы: Linux, FreeBSD, OpenBSD, macOS/Mac OS X, Windows.

2 Анализ сетевых протоколов TCP/IP:

а) декодирование IPv6;

б) декодирование туннелей IPv4-in-IPv6, IPv6-in-IPv6, Teredo и др.

3 Обработка потока протокола TCP: контроль сессий, повторная сборка потока.

4 Разбор протоколов:

а) поддержка декодирования пакетов:

– IPv4, IPv6, TCP, UDP, SCTP, ICMPv4, ICMPv6, GRE;

– Ethernet, PPP, PPPoE, Raw, SLL, VLAN, QINQ, MPLS, ERSPAN, VXLAN, Geneve;

б) декодирование уровня приложения:

– HTTP, HTTP/2, SSL, TLS, SMB, DCERPC, SMTP, FTP, SSH, DNS, Modbus, ENIP/CIP, DNP3, NFS, NTP, DHCP, TFTP, KRB5, IKEv2, SIP, SNMP, RDP, RFB, MQTT;

– новые протоколы, разработанные на языке Rust, для безопасного и быстрого декодирования.

5 Инспектирование HTTP-трафика:

а) извлечение и проверка переданных по HTTP файлов;

б) разбор сжатого контента;

в) идентификация по URI, cookie, заголовкам, user-agent, телу запроса и ответа;

г) выделение контента при помощи регулярных выражений;

д) идентификация файлов по имени, типу или контрольной сумме.

6 Перехват трафика с помощью интерфейсов: NFQueue, IPFRing, LibPcap, IPFW, AF_PACKET, PF_RING.

Важной особенностью IPS Suricata является возможность функционирования в двух режимах:

– **NFQ** – через очередь NFQUEUE, которая может обрабатываться на уровне пользователя;

– **AF_PACKET** – через режим zero copy.

NFQ IPS-режим работает следующим образом:

1) пакет попадает в iptables;

2) правило iptables направляет его в очередь NFQUEUE;

3) из очереди NFQUEUE пакеты могут обрабатываться на уровне пользователя, что и делает Suricata;

4) Suricata прогоняет пакеты по настроенным правилам (rules) и в зависимости от них может вынести один из трёх вердиктов: NF_ACCEPT, NF_DROP и NF_REPEAT;

5) пакеты, попадающие в NF_REPEAT, могут быть промаркированы в системе и направлены обратно в начало текущей таблицы iptables, что даёт огромный потенциал для влияния на дальнейшую судьбу пакетов с помощью правил iptables.

AF_PACKET IPS-режим работает, используя режим системы zero copy, но с некоторыми ограничениями. Система должна работать в качестве шлюза с двумя сетевыми интерфейсами. Если пакет попадает под DROP-правило, то он просто не пересылается на второй интерфейс.

Следует обратить внимание на замечательное дополнение к процедуре iptables, такое как RAW DNAT/SNAT, которое позволяет при помощи Suricata направлять разные типы трафика на разные адреса назначения. Кроме того, Suricata умеет модифицировать пакеты «на лету».

Таким образом, Suricata – гибкий инструмент по обработке пакетов, который позволяет менять маршруты в зависимости от содержания пакета, детектировать атаки и предотвращать попадание «плохих» пакетов в систему (например, отбрасывать или подменять пакеты, пока они не дошли до веб-сервера).

5.4 Сканеры безопасности

5.4.1 Общая характеристика

Для высокого уровня безопасности необходимо применять не только межсетевые экраны, но и периодически проводить мероприятия по обнаружению уязвимостей, например, при помощи сканеров уязвимости. Своевременное выявление слабых мест в системе позволит предотвратить несанкционированный доступ и манипуляции с данными. Согласно статистике большинство атак происходит через известные и опубликованные «лазейки» безопасности, которые могут быть не ликвидированы по многим причинам, будь то нехватка времени, персонала или некомпетентность системного администратора [12]. Также следует понимать, что обычно нарушитель может проникнуть в систему несколькими способами, и если один из способов не сработает, то он всегда может опробовать другой. Для обеспечения максимального уровня безопасности системы требуется тщательный анализ рисков и дальнейшее составление чёткой модели угроз, чтобы предугадать все возможные действия гипотетического преступника.

В качестве наиболее распространённых уязвимостей можно назвать [12]:

– переполнение буфера;

– возможные ошибки в конфигурации маршрутизатора или межсетевого экрана;

– уязвимости веб-сервера;

– уязвимости почтовых серверов;

– уязвимости DNS-серверов;

– уязвимости серверов баз данных.

К уязвимостям системы также необходимо отнести:

– управление пользователями и файлами, поскольку обеспечение уровня доступа пользователя с минимальными привилегиями – специфическая задача, требующая компромисса между удобством работы пользователя и обеспечением защиты системы;

– проблему пустых или слабых паролей,

– проблему стандартных учётных записей;

– проблему общей утечки информации.

Сканер безопасности – это программное средство для удалённой или локальной диагностики различных элементов сети на предмет выявления в них большого спектра уязвимостей. Использование сканера безопасности позволяет значительно сократить время работы специалистов и облегчить поиск уязвимостей.

В настоящее время имеется большое количество сканеров безопасности, поэтому определим критерии их выбора, охватывающие все аспекты использования сканеров безопасности, начиная от методов сбора информации и заканчивая стоимостью.

Использование сканера безопасности начинается с планирования развёртывания и самого развёртывания. Поэтому **первая группа критериев** касается архитектуры сканеров безопасности, взаимодействия их компонентов, инсталляции, управления.

Вторая группа критериев – сканирование – должна охватить методы, используемые сравниваемыми сканерами для выполнения перечисленных действий, а также другие параметры, связанные с указанными этапами работы программного продукта. Также к важным критериям относятся результаты сканирования, в частности, как они хранятся и какие отчёты могут быть сформированы на их основе.

Третья группа критериев – это критерии обновления и поддержки, которые позволяют выяснить такие вопросы, как методы и способы обновления, уровень технической поддержки, наличие авторизованного обучения и т. д.

Четвёртая группа включает в себя единственный, но весьма важный критерий – стоимость.

Рассмотрим наиболее популярный используемый в системе сканер безопасности Nessus с позиций критериев, определённых выше.

5.4.2 Сканер безопасности Nessus

Сканер безопасности (уязвимостей) Nessus является одним из программных компонентов SIEM и предназначен для мониторинга узлов на предмет наличия проблем, связанных с информационной безопасностью. В основе его работы лежит коллекция тестов безопасности, позволяющих выявить уязвимость. С их помощью данный компонент выполняет определённые действия с узлом сети: сканирует открытые порты, посылает специальным образом сформированные пакеты для имитации атаки или даже авторизуется на узле, получает доступ к консоли управления и выполняет на нём команды. Затем выполняется анализ собранных данных и формируются выводы о наличии выявленных проблем с безопасностью.

Nessus – программа для автоматического поиска известных изъянов в защите инфокоммуникационных систем. Она способна обнаружить наиболее часто встречающиеся виды уязвимостей:

– наличие уязвимых версий служб или доменов;

– ошибки в конфигурации (отсутствие необходимости авторизации на SMTP-сервере);

– наличие паролей по умолчанию;

– наличие пустых или слабых паролей.

Сканер Nessus является мощным и надёжным средством, которое относится к семейству сетевых сканеров, позволяющих осуществлять поиск уязвимостей в сетевых сервисах, предлагаемых:

– операционными системами;

– межсетевыми экранами;

– фильтрующими маршрутизаторами;

– другими сетевыми компонентами.

Для поиска уязвимостей используются как стандартные средства тестирования и сбора информации о конфигурации и функционировании сети, так и специальные средства, эмулирующие действия злоумышленника по проникновению в системы, подключённые к сети. В программе существует возможность подключения собственных проверочных процедур или шаблонов. Для этого в сканере предусмотрен специальный язык сценариев, названный NASL (Nessus Attack Scripting Language). Имеется база уязвимостей, которая постоянно пополняется и обновляется. Сканер Nessus имеет следующие параметры для настройки:

1 Установку порогового значения метрики уязвимостей (Vulnerability Ticket Threshold: «2») для создания билетов в случае обнаружения уязвимостей.

2 Установку максимального количества одновременных процессов сканирования (Max Simultaneous Scans: «5»).

3 Настройку процедуры сканирования со следующими параметрами:

а) Job Name: Server-Internet / Server-Orcl;

б) Select Sensor: alientvault 192.168.2.250;

в) Profile: Default – Non destructive Full and Fast Scan;

г) Schedule Method: Immediately;

д) Type: Servers;

е) Only scan hosts that are alive (greatly speeds up the scanning process);

ж) Pre-Scan locally (do not pre-scan from scanning sensor).

4 Настройку процедуры сканирования уязвимостей без аутентификации со следующими параметрами:

а) Job Name: Test;

б) Select Sensor: alientvault 192.168.2.250;

в) Profile: Default – Non destructive Full and Fast Scan;

г) Schedule Method: Immediately;

д) Type: Servers;

е) Only scan hosts that are alive (greatly speeds up the scanning process) (должна быть установлена галочка);

ж) Pre-Scan locally (do not pre-scan from scanning sensor) (должна быть установлена галочка).

5 Настройку процедуры сканирования уязвимостей по расписанию со следующими параметрами:

а) Job Name: Test-2;

б) Select Sensor: alientvault 192.168.2.250;
в) Profile: Default – Non destructive Full and Fast Scan;
г) Schedule Method: Day of the Week:
– Begin In: 2016 / 8 / 5;
– Weekly: Friday;
– Frequency: Every 1 week;
– Time: 16:00;
д) Type: Servers;
е) Only scan hosts that are alive (greatly speeds up the scanning process;
ж) Pre-Scan locally (do not pre-scan from scanning sensor).

5.5 Программный компонент «Мониторинг доступности узлов сети» Nagios

Nagios является одним из программных компонентов SIEM и предназначен для сбора и отображения данных об использовании (загруженности) вычислительных ресурсов объектов мониторинга. Данный компонент имеет серверную и клиентскую части.

Серверная часть Nagios (далее – Nagios-сервер) разработана под операционную систему Linux, которая поставляется и устанавливается вместе с SIEM и обеспечивает обработку данных, полученных от клиентской части, а также осуществляет периодический контроль за состоянием контролируемых узлов и служб. Nagios-сервер имеет веб-интерфейс, функционал которого встроен в меню SIEM.

Клиентская часть Nagios представляет собой программные агенты (далее – агенты Nagios), разработанные под различные операционные системы. Агенты Nagios устанавливаются на конечные узлы ПЭВМ (АРМ пользователей или серверы приложений), где они собирают и пересылают на Nagios-сервер информацию о программной и аппаратной части контролируемых ими систем (доступность, производительность, свободное дисковое пространство, загруженность процессора и памяти).

Для того чтобы система «агент – сервер» Nagios функционировала, её серверную и клиентскую части необходимо настроить.

Настройка системы Nagios включает в себя:

1 Определение имён и адресов хостов, подлежащих контролю (АРМ и серверов).

2 Указание конкретных контролируемых устройств в определённых ранее хостах (CPU, MEMORY, DISK SPACE).

3 Настройку агентов Nagios на объектах мониторинга со следующими параметрами:

а) Allowed hosts: 192.168.2.250 (где 192.168.4.250 – IP-адрес Nagios-сервера);

б) Password: 12345 (пароль «12345», который использовался при установке программных агентов на контролируемые хосты).

в) Modules to load:

– Enabled common check plugins;
– Enabled Nsclient server;
– Enabled NRPE server/Safe Mode;
– Enabled NSCA client;
– Enabled web-server.

4 Настройку службы (NSClient) на объектах мониторинга с установленными агентами Nagios путём выбора параметра в её свойствах: «Разрешить взаимодействие с рабочим столом» на вкладке «Вход в систему».

5.6 База глобального сообщества исследователей угроз информационной безопасности OTX

OTX (Open Threat Exchange) – это открытое сообщество для обмена данными о новых угрозах, которое обеспечивает совместную защиту и исследование угроз. OTX обеспечивает открытый доступ для любой организации, что позволяет всем сотрудничать с сообществом профессионалов в области информационной безопасности. Доступ к базе OTX позволяет всем, кто занимается вопросами безопасности, активно делиться последними данными, тенденциями и методами распространения угроз. Такой доступ также ускоряет распространение свежей информации об угрозах и автоматизирует процесс обновления инфраструктуры безопасности.

Доступ SIEM к базе OTX необходим для получения данных по угрозам, включающим наборы индикаторов компрометации (IOCS) и репутацию IP-адресов сети Интернет. SIEM начинает получать данные по угрозам сразу после регистрации и подключения к базе OTX. Затем SIEM коррелирует эти данные и создаёт соответствующие события безопасности.

Настройка доступа SIEM к базе OTX включает в себя:

1) регистрацию на сайте глобального сообщества исследователей угроз по ссылке: https://otx.alienvault.com;

2) создание на сайте глобального сообщества исследователей угроз персонального программного ключа «OTX key» для подключения SIEM к базе OTX;

3) настройку доступа SIEM к базе OTX с использованием программного ключа путём копирования сгенерированного персонального ключа «OTX key» на сайте глобального сообщества и помещения его в поле «OTX key» на вкладке «OTX Account» SIEM;

4) подписку на сайте глобального сообщества исследователей на получение данных по угрозам из базы OTX.

Очевидно, что доступ к базе OTX возможен только после подключения SIEM к сети Интернет. Доступ SIEM в сеть Интернет необходим также для получения обновлений стандартизованной базы уязвимостей CVE (Common Vulnerabilities and Exposures) из специализированных интернет-ресурсов.

5.7 Программный агент Snare

Snare предназначен для сбора и фильтрации данных аудита и журналов событий Windows, обеспечивая высокую надёжность их шифрования и доставки на ядро SIEM. После инсталляции и настройки агент Snare взаимодействует с подсистемой регистрации событий Windows, чтобы преобразовать входные данные службы Eventlog в текстовый формат Syslog, а затем отправить данные журналов (событий, безопасности, системы) в ядро SIEM, используя дополнительное шифрование TLS/SSL.

Для обеспечения данной функциональности настройка агентов Snare должна включать в себя:

1 Настройку агентов Snare при их инсталляции на объекты мониторинга (сервера) со следующими параметрами:

– Take over control of Eventlog Configuration;

– Use system account;

– Use Web-access.

2 Настройку агентов Snare для отправки лог-файлов в ядро SIEM путём удалённого подключения администратора к объектам мониторинга (серверам) с помощью прикладного ПО «Remote Desktop Connection» и использования веб-интерфейса агента Snare для установки следующих параметров:

– Destination Snare Server address;

– Destination Port: 514;

– Enable Syslog Header.

Данные параметры необходимы для того, чтобы включить в заголовок системного журнала следующие значения:

– SYSLOG Facility: User;

– SYSLOG Priority: Notice.

3 Редактирование конфигурационного файла snare.conf на SIEM для обеспечения возможности ядру SIEM получать лог-файлы от агентов Snare.

5.8 Функционирование типовой SIEM-системы

Система мониторинга информационной безопасности функционирует в штатном режиме круглосуточно и позволяет проводить текущий анализ событий, зарегистрированных на объектах мониторинга. При возникновении событий информационной безопасности на объектах мониторинга данные об этом записываются в журналы состояний этих объектов и в режиме времени, близком к реальному, отправляются на обработку в ядро системы программными агентами, установленными на АРМ и серверах, или собственными средствами коммуникационного оборудования сети.

В свою очередь, ядро принимает эти данные, агрегирует и записывает их в журналы в соответствии с типом источника событий с помощью утилиты rsyslog. Сенсор обрабатывает журналы в соответствии с логикой плагинов, обеспечива-

ющих поддержку работы с конечными устройствами и системами различных типов, после чего нормализованные данные поступают в базу данных событий, где каждому событию ИБ присваивается персональный идентификатор. После этого выполняется обработка данных модулем корреляции в соответствии с заранее определёнными правилами корреляции и политиками, на основании которых выявляются инциденты ИБ, уведомления о выявлении которых передаются средствами веб-интерфейса администратору безопасности. Нормализованные события ИБ, данные об инцидентах ИБ и информация о настройках системы хранятся в реляционной базе данных.

Взаимодействие администратора безопасности с ядром SIEM-системы осуществляется с использованием веб-интерфейса и интерфейса командной строки, с помощью которых администратор получает возможности:

– осуществлять анализ поступающих событий для выявления значимых или незначимых событий безопасности и связанных с ними инцидентов информационной безопасности;

– выполнять настройку директив корреляции на основании выявленных значимых событий безопасности;

– производить настройку политик и действий, сопровождающих выявление инцидентов информационной безопасности (оповещение по электронной почте, создание тикетов);

– формировать различные отчёты на основании аналитических данных;

– осуществлять резервное копирование и восстановление данных.

На рисунках 5.9–5.14 приведены изображения окон взаимодействия администратора безопасности с одной из популярных SIEM-систем HP ArcSight.

Преимущества применения SIEM:

– оперативность сбора и обработки данных о событиях безопасности;

– высокая достоверность данных об инцидентах информационной безопасности за счёт выявления взаимосвязей между событиями безопасности;

– централизованное и децентрализованное конфигурирование и управление программными агентами, установленными на объектах мониторинга;

– использование в сетях с разветвлённой и иерархической структурой;

– возможность масштабирования и модернизации системы собственными силами;

– составление наглядных отчётов с таблицами и диаграммами.

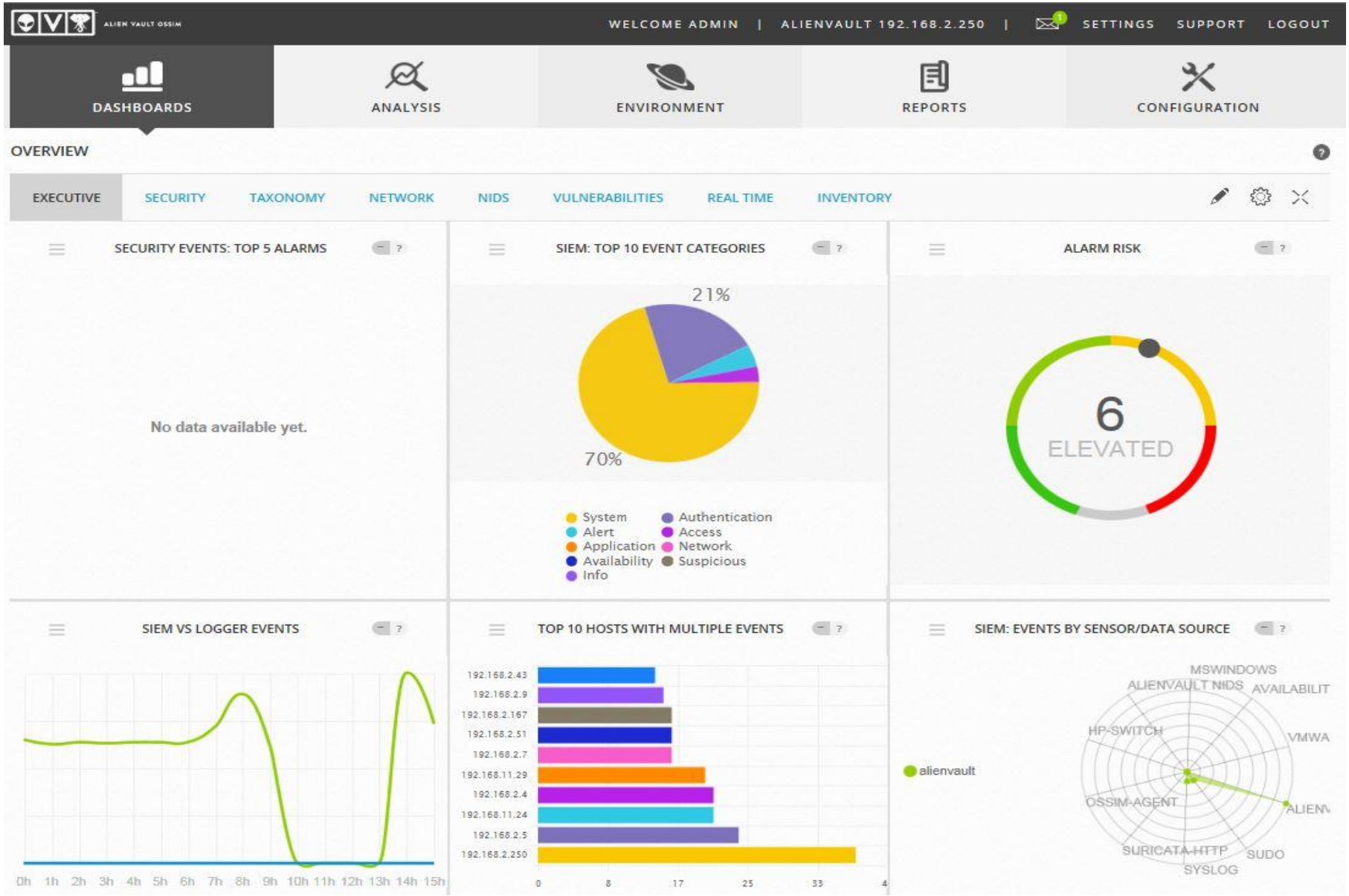


Рисунок 5.9 – Главное окно веб-интерфейса

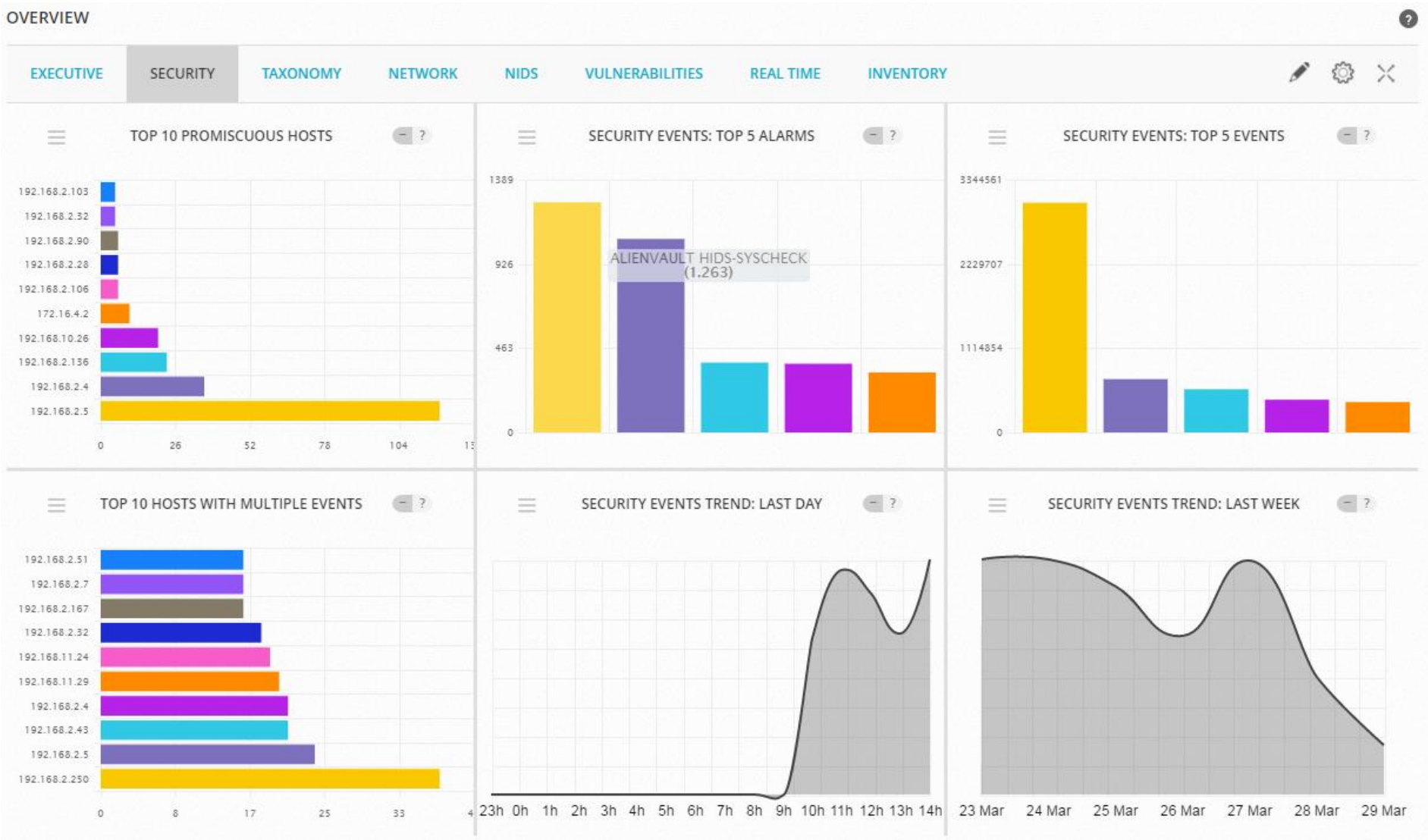


Рисунок 5.10 – Окно виджетов «Security» веб-интерфейса

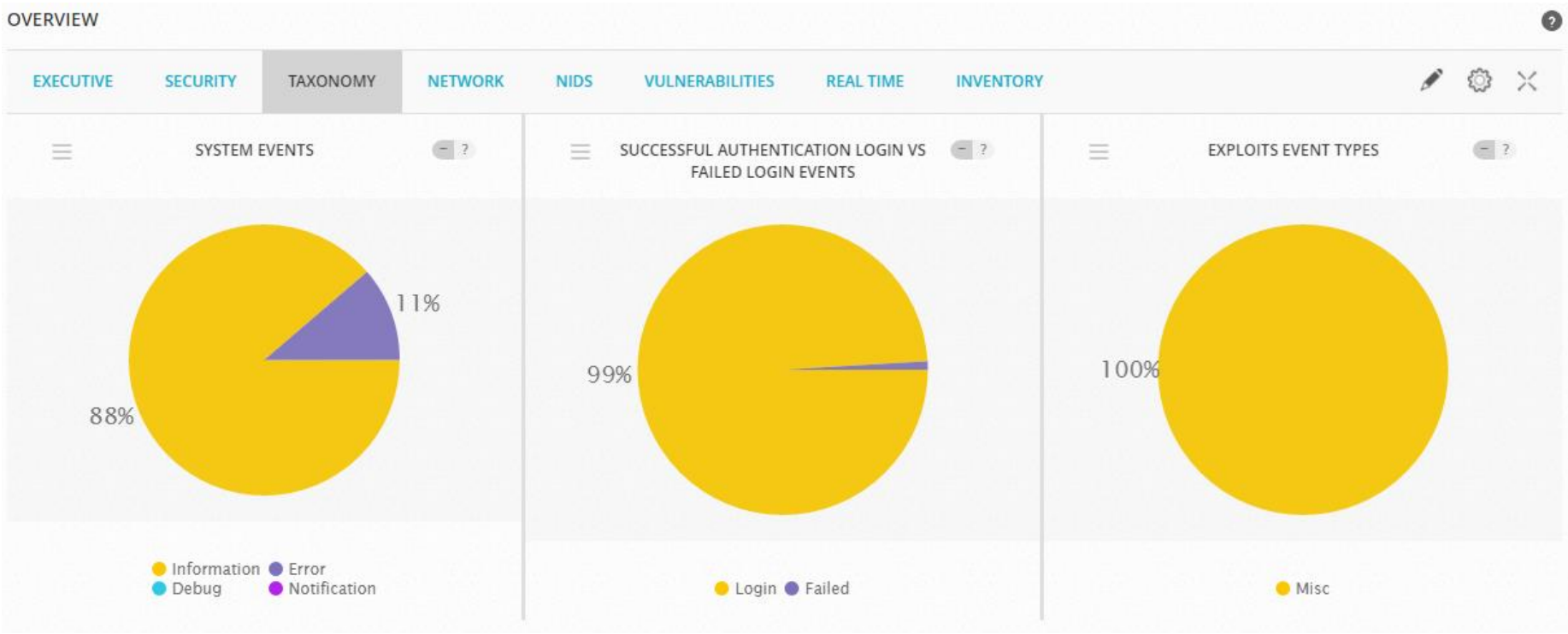


Рисунок 5.11 – Окно виджетов «Taxonomy» веб-интерфейса

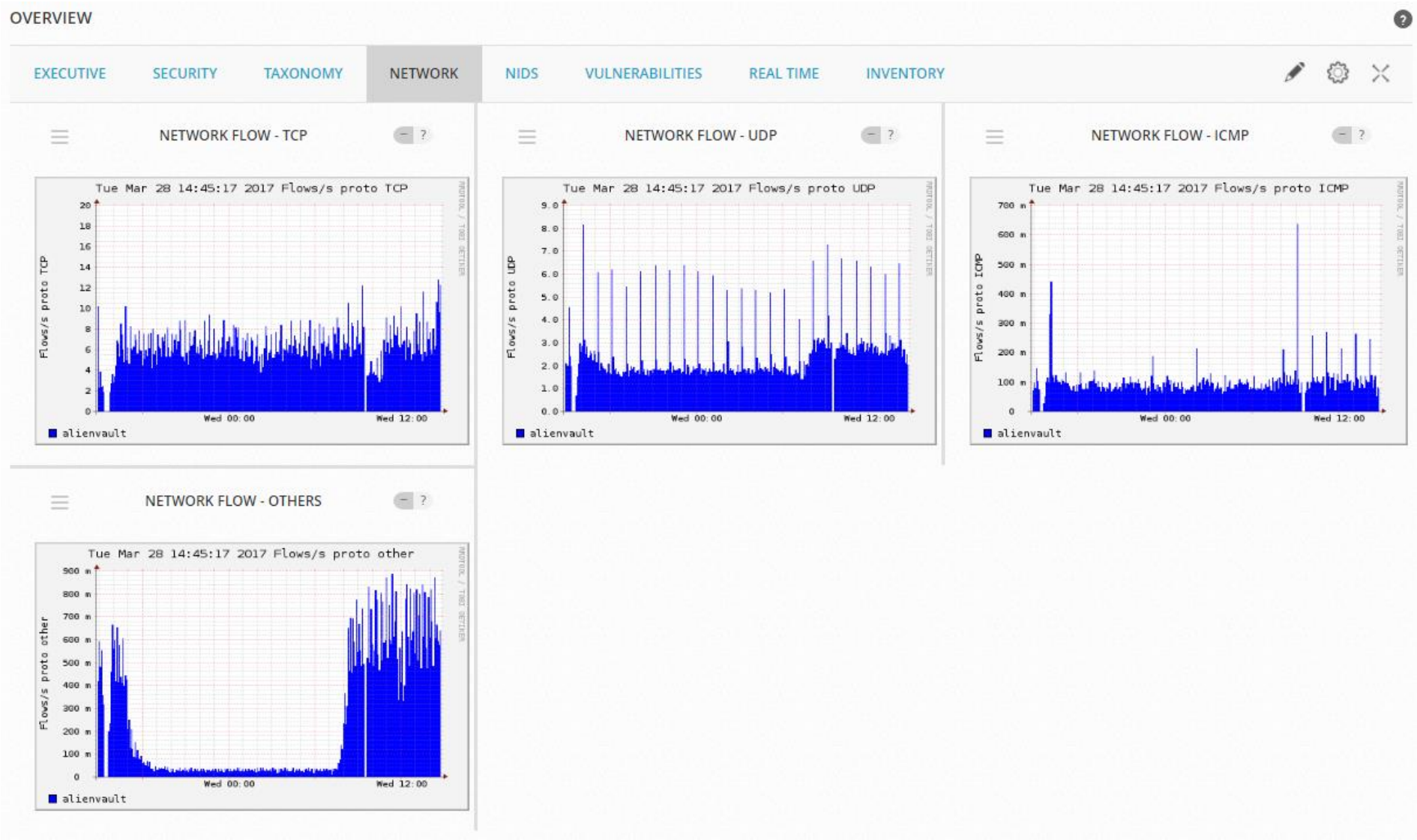


Рисунок 5.12 – Окно виджетов «Network» веб-интерфейса

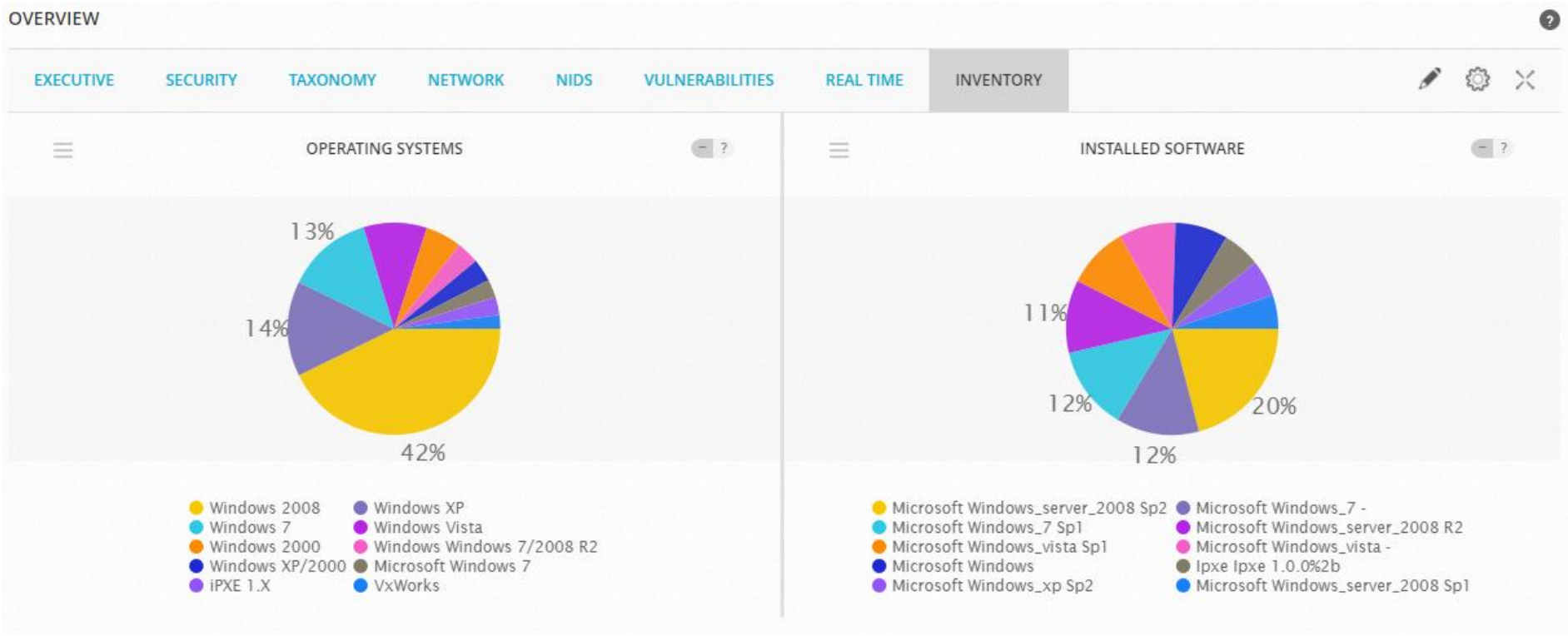


Рисунок 5.13 – Окно виджетов «Inventory» веб-интерфейса

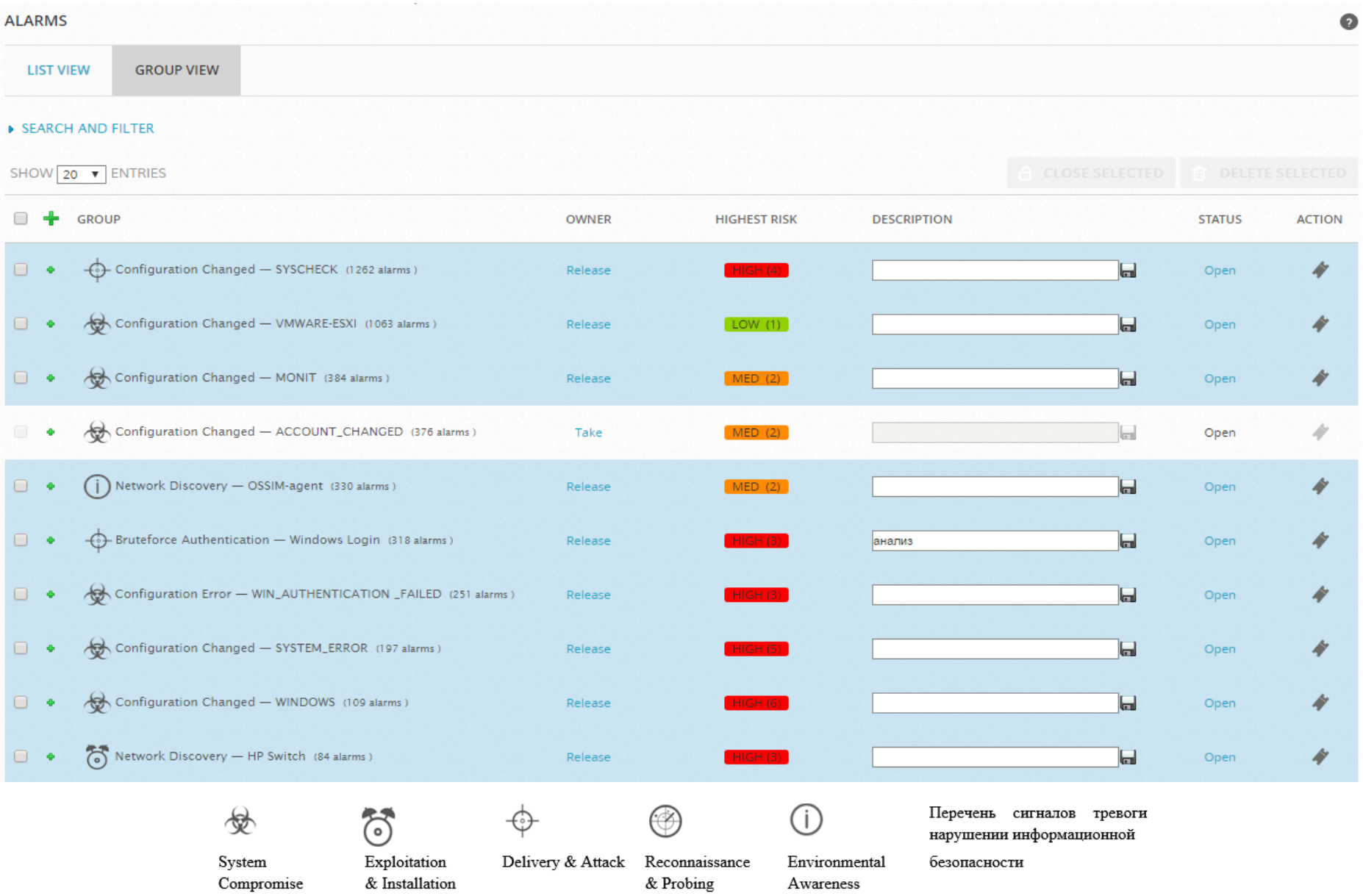


Рисунок 5.14 – Окно виджетов сигналов тревоги

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Лапонина, О. Р. Основы сетевой безопасности. В 2 ч. Ч. 1. : Межсетевые экраны: учеб. пособие / О. Р. Лапонина ; под ред. В. А. Сухомлина. – М. : Национальный Открытый Университет «ИНТУИТ», 2014. – 378 с.

2 Мохаммед, Ф. О. Адаптивное управление межсетевыми экранами в инфотелекоммуникациях : дис. … канд. техн. наук : 05.13.19 / Ф. О. Мохаммед. – Минск, 2012.

3 Бобов, М. Н. Принципы функционирования межсетевых экранов / М. Н. Бобов, Ф. О. Мохаммед // Теоретические и прикладные проблемы информационной безопасности в Республике Беларусь : материалы междунар. науч.-практ. конф., Минск, 31 марта 2010 г. / Минск : Академия МВД, 2010. – С. 64–69.

4 Бобов, М. Н. Оценка показателей мониторинга безопасности информационных систем / М. Н. Бобов // Телекоммуникации: сети и технологии, алгебраическое кодирование и безопасность данных : материалы междунар. науч.-техн. семинара, Минск, апрель – декабрь 2015 г. / Минск : БГУИР, 2015. – С. 33–38.

5 Электронный журнал «Хакер», №68 [Электронный ресурс]. – 2019. – Режим доступа : https://xakep.ru/issues/xa/068.

6 Кореньков, В. В. Архитектура системы мониторинга центрального информационно-вычислительного комплекса ОИЯИ / В. В. Кореньков, В. В. Мицын, П. В. Дмитриенко // Информационные технологии и вычислительные системы. – 2012. – №3. – С. 31–42.

7 Хостовая система обнаружения вторжений OSSEC [Электронный ресурс]. – 2019. – Режим доступа : https://ossec.net.

8 Сетевая система обнаружения вторжений Suricata [Электронный ресурс]. – 2019. – Режим доступа : https://redmine.openinfosecfoundation.org/projects/suricata/wiki.

9 Сканер уязвимостей Nesus [Электронный ресурс]. – 2019. – Режим доступа : https://abnet.am/Nesus.html.

10 Система обмена информацией о сетевых угрозах OTX [Электронный ресурс]. – 2019 – Режим доступа : https://cybersecurity.att.com/open-threat-exchange.

11 Официальный сайт компании CISCO [Электронный ресурс]. – 2019. – Режим доступа : https://cisco.com.

12 Рожкова, Е. О. Обзор и сравнение сканеров уязвимостей / Е. О. Рожкова, И. В. Ильин // Научное сообщество студентов XXI столетия. Технические науки: сб. ст. по материалам XXX междунар. студ. науч.-практ. конф., Новосибирск, апрель 2015 г. / Новосибирск : Изд. «СибАК», 2015. – №3(29). – С. 77–87.

13 Бобов, М. Н. Методы использования механизма поиска обратного маршрута для защиты локальных сетей от атаки спуфинга / М. Н. Бобов, Ф. О. Мохаммед // Доклады БГУИР. – 2010. – №5(51). – С. 72–75.

Св. план 2021, поз. 33

Учебное издание

Бобов Михаил Никитич
Шевчук Оксана Геннадьевна

ЗАЩИТА ИНФОРМАЦИИ В ИНФОКОММУНИКАЦИЯХ. МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ СРЕДЫ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *Е. С. Юрец*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *Е. Г. Бабичева*

Подписано в печать 16.11.2022. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 5,7. Уч.-изд. л. 6,0. Тираж 30 экз. Заказ 193.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск