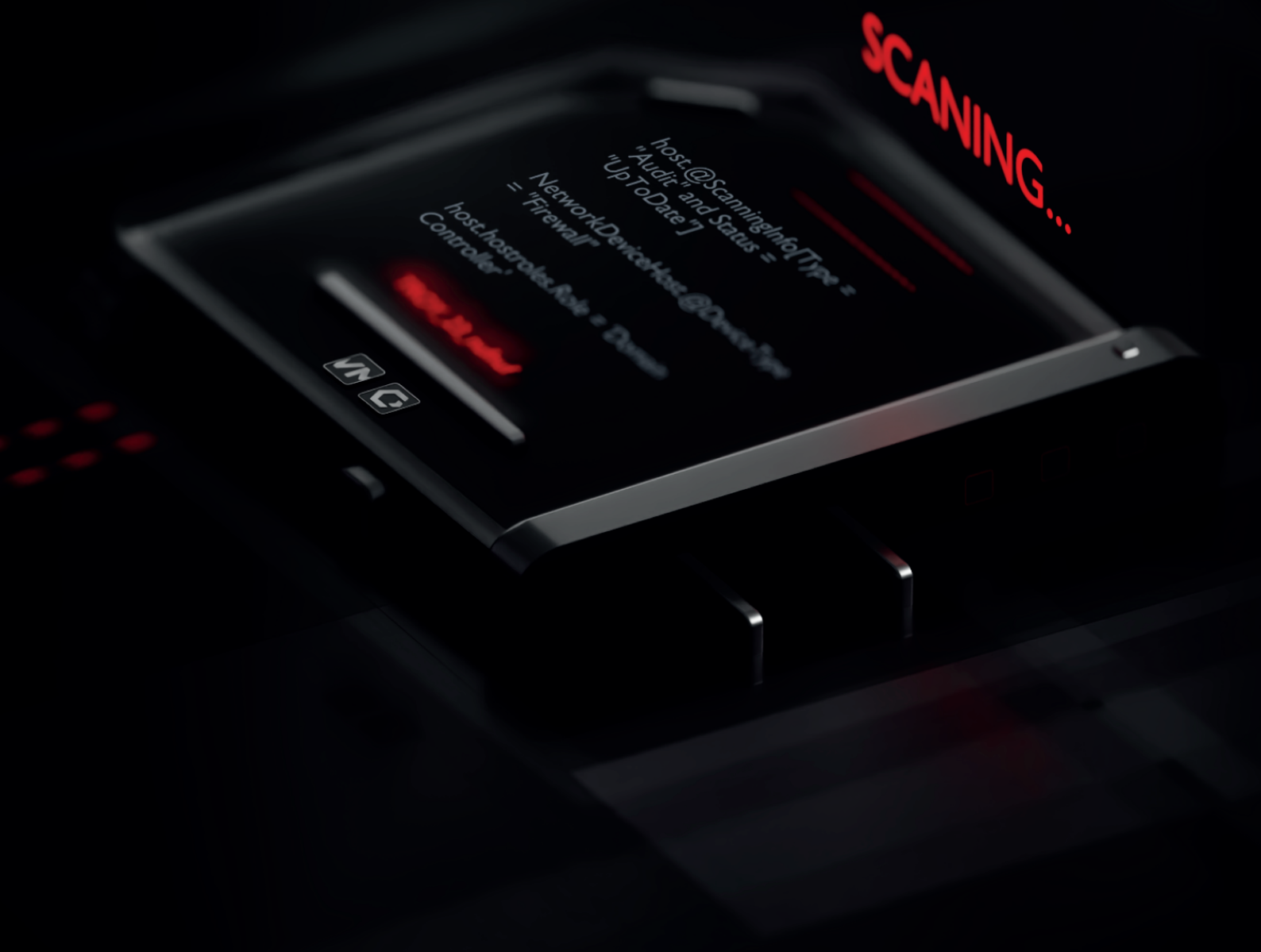
positive technologies

РУКОВОДСТВО ПО АУДИТУ АКТИВОВ



MaxPatrol Carbon

MaxPatrol VM

Дата редакции документа: 16.04.2025

Содержание

1. Об этом документе

Управление активами — одна из ключевых задач служб ИБ и ИТ, обеспечивающая устойчивость и безопасность современной ИТ-инфраструктуры. Для построения эффективного процесса необходимы глубокое понимание инфраструктуры, высокая квалификация специалистов, слаженное взаимодействие команд, а также современные инструменты, позволяющие автоматизировать инвентаризацию и аудит.

Полные и актуальные знания об ИТ-активах и сетевой топологии организации помогают своевременно выявлять уязвимости, моделировать потенциальные маршруты кибератак и оперативно устранять обнаруженные слабости инфраструктуры. Недостаток информации может привести к реализации значительных рисков, включая компрометацию данных, нарушение работоспособности систем и финансовые потери.

Руководство разработано для оказания помощи в построении процесса аудита ИТ-активов с использованием MaxPatrol VM. Документ содержит подробные инструкции и справочную информацию для оценки качества сканирования инфраструктуры. Он поможет службам ИБ и ИТ обеспечить контроль над ИТ-активами, а также подготовить инфраструктуру к внедрению MaxPatrol Carbon.

2. Сканирование активов

Сканирование активов рекомендуется выполнять в следующем порядке:

1. Сканирование портов для обнаружения активов (см. раздел 2.1).
2. Сканирование активов методом черного ящика (см. раздел 2.2).
3. Группировка активов (см. раздел 2.3).
4. Сканирование активов методом белого ящика (аудит) (см. раздел 2.4).
5. Оценка качества аудита (см. раздел 2.4.10).
6. Аудит активов, информации о которых по результатам оценки качества оказалось недостаточно.

Во время сканирования некоторые активы могут быть недоступны. Это может быть связано с тем, что актив выключен или не подключен к сети или существуют общие проблемы с доступом в конкретный сегмент сети. Поэтому для обнаружения новых активов и их классификации рекомендуется выполнять сканирование инфраструктуры по расписанию, не реже одного раза в неделю. Кроме того, необходимо периодически актуализировать данные об активах, полученные в результате аудита.

В этом разделе приведены инструкции по созданию задач на сбор данных и аудит и PDQL-запросы, которые необходимо использовать при выполнении аудита, для оценки качества аудита и поиска неотсканированных активов.

Информация о работе с активами, управлении задачами на сбор данных и аудит приведена в Руководстве по настройке источников MaxPatrol VM, а также в разделах «Работа с активами» и «Работа с задачами» Руководства оператора MaxPatrol VM.

В этом разделе

2.1. Сканирование портов для обнаружения активов

Сканирование портов, часто используемых для размещения активов, выполняется с помощью MaxPatrol VM в рамках задачи на сканирование с профилем **HostDiscovery** и позволяет обнаружить активы.

► Чтобы создать задачу на сканирование портов:

1. В главном меню выберите **Сбор данных** → **Задачи**.
2. Нажмите **Создать задачу** → **Сбор данных**.
3. Введите название задачи.
4. Выберите профиль **HostDiscovery**.
5. На вкладке **Способы проверки узлов** включите проверку с помощью эхо-запроса ICMP.
6. В поле **Порты TCP для проверки с помощью SYN-пакетов** введите `21, 22, 25, 53, 80, 110, 123, 139, 143, 443, 445, 465, 631, 993, 995, 3389`.

   Примечание. Выше приведен список часто используемых портов. Вы можете выполнить расширенное сканирование, используя порты 1—1719, 1721—3050, 3074, 3128, 3260-3300, 3389, 5000—5500, 7000—8100, 8000, 9000, 9001. Вы можете вводить несколько портов через запятую, диапазон портов — через дефис.
7. Включите отображение дополнительных параметров.
8. На вкладке **Параметры проверки узлов** включите определение операционной системы и проверку с помощью ARP-запросов.
9. В поле **Максимальное количество неудачных попыток** укажите `2`.
10. В поле **Интервал между проверками** укажите `900`.
11. В поле **Тайм-аут ответа** укажите `1000`.
12. В поле **Максимальная скорость отправки ARP-запросов** укажите `100`.
13. В поле **Максимальная скорость отправки пакетов** укажите `100`.
14. Нажмите **Сохранить**.

    Примечание. Задачи сканирования могут выполняться достаточно долго. Для ускорения сканирования вы можете уменьшить количество сканируемых портов или выполнять сканирование по отдельным сегментам сети.

    Примечание. Когда сканирование будет завершено, необходимо убедиться, что как минимум от части активов в каждом из сегментов сети, указанных в целях сканирования, был получен ответ. Если доступ к какому-либо сегменту сети отсутствует, рекомендуется настроить права доступа от сервера MP 10 Collector до сканируемых узлов, например путем модернизации правил маршрутизации, изменения списков управления доступом (ACL).

Сканирование рекомендуется выполнять в несколько итераций, изменяя значения параметров **Максимальная скорость отправки ARP-запросов** и **Максимальная скорость отправки пакетов**. Значения этих параметров могут повлиять на работоспособность IT-инфраструктуры во время сканирования. Это обусловлено следующими факторами:

1. Оборудование рассчитано на определенную скорость поступления пакетов. При обычной работе приложений и пользователей количество пакетов не превышает определенного значения. Во время сканирования оно сильно увеличивается, поэтому оборудование может быть перегружено потоком трафика.
2. Используется SYN-сканирование: на определенный порт отправляется специальный пакет, который инициирует соединение без отправки последующих пакетов. В результате в таблице со списком соединений накапливаются новые записи, что может привести к исчерпанию ресурсов и неработоспособности оборудования.

Поэтому для подбора значений параметров **Максимальная скорость отправки ARP-запросов** и **Максимальная скорость отправки пакетов** необходимо:

1. Сначала установить для каждого из параметров значение 100 пакетов в секунду. Это значение не приведет к нарушению работоспособности IT-инфраструктуры во время сканирования.
2. Уведомить сетевых инженеров о начале сканирования, согласовать с ними процедуру оперативной связи для остановки сканирования. На время сканирования сетевым инженерам необходимо будет включить мониторинг сетевого оборудования.
3. Запустить сканирование.
4. Когда сканирование закончено, сообщить об этом сетевым инженерам.
5. Запросить у сетевых инженеров информацию о том, можно ли увеличить скорость сканирования по результатам мониторинга.
6. Провести сканирование несколько раз, увеличивая значение параметров **Максимальная скорость отправки ARP-запросов** и **Максимальная скорость отправки пакетов** до тех пор, пока не будет подобрано оптимальное значение скорости пакетов в вашей инфраструктуре.

2.2. Сканирование активов методом черного ящика

Сканирование активов методом черного ящика (без использования данных учетных записей) выполняется с помощью MaxPatrol VM в рамках задачи на сканирование с профилем **Service Discovery** и позволяет идентифицировать открытые порты, сетевые сервисы и приложения, определять версию ОС. Эти данные необходимы для анализа инфраструктуры и используются при выполнении последующих задач.

- Чтобы создать задачу на сканирование активов методом черного ящика:

  1. В главном меню выберите **Сбор данных** → **Задачи**.
  2. Нажмите **Создать задачу** → **Сбор данных**.

3. Введите название задачи.
4. Выберите профиль **Service Discovery**.
5. На вкладке **Сканирование портов** в поле **Порты** введите список портов.

   В поле **Порты** приведен список портов по умолчанию. Вы можете выполнить сканирование в соответствии с этим списком, но такая задача может выполняться достаточно долго. Для сокращения продолжительности сканирования вы можете уменьшить количество сканируемых портов или выполнить сканирование отдельных групп активов, как описано ниже.

   Внимание! Необходимо обязательно выполнять сканирование портов 21, 22, 80, 443, 445, 3389 (TCP) (вне зависимости от аудита).
6. На вкладке **Сканирование UDP-служб** выключите сканирование UDP-служб.
7. На вкладке **Поиск уязвимостей** выключите поиск уязвимостей.
8. На вкладке **Подбор учетных данных** выключите подбор учетных данных.
9. На вкладке **Поиск файлов** выключите поиск файлов.
10. Включите отображение дополнительных параметров.
11. Нажмите **Сохранить**.

Сканирование групп активов

Вы можете создать динамические группы активов (см. раздел 2.3), для каждой группы создать задачу на сканирование с профилем **Service Discovery** и настроить периодичность сканирования. Благодаря этому сканирование будет выполняться более точечно, по отдельным группам активов.

Например, если в результате сканирования портов (см. раздел 2.1) были обнаружены активы с открытым TCP-портом 80 и (или) 443, вы можете объединить их в группу и настроить периодичность сканирования, выполнив следующие шаги:

1. Создать динамическую группу активов (см. раздел 2.3), указав условие фильтрации: `Host[Endpoints<TransportEndpoint>[Port in [80, 443] and Status = 'Open']]`
2. Создать задачу на сканирование этой группы активов с профилем **Service Discovery**, как описано в инструкции выше, при этом в списке портов указать `80/tcp;443/tcp;`.
3. В задаче на сканирование включить расписание.
4. Указать периодичность сканирования, например каждые две недели.
5. Сохранить группу.

Аналогично вы можете объединить в динамические группы и настроить периодичность сканирования активов с другими открытыми портами из обязательного (21, 22, 80, 443, 445, 3389 (TCP)) или расширенного списка портов.

Примечание. Вы можете создать динамическую группу активов с открытым TCP-портом 22 и (или) 3389 двумя способами: используя условие фильтрации с номерами портов (как показано выше) или условия фильтрации `UnixHost` (порт 22 — стандартный порт SSH в ОС семейства Unix), `WindowsHost` (порт 3389 — стандартный порт RDP в Windows).

2.3. Группировка активов

После выполнения первых задач сканирования для удобства работы с системой рекомендуется объединить активы в группы. Группировка активов осуществляется в MaxPatrol VM и используется для классификации активов, мониторинга источников, планирования задач на сканирование активов, сбора и фильтрации данных, а также при назначении правил политик на странице **Система** → **Политики**.

Информация о статических и динамических группах и инструкции по работе с ними содержатся в разделе «Группы активов» Руководства оператора MaxPatrol VM.

С помощью статических групп определяются категории IT-инфраструктуры, по которым планируется группировка активов (например, ОС, ПО, сетевые сегменты). Статическая группа может содержать вложенные группы — как статические, так и динамические. В статической группе отображаются активы из всех вложенных групп. С помощью динамических групп посредством PDQL-запросов определяются фильтры, по которым система автоматически собирает активы по тем или иным признакам (например, Cisco, Unix, TeamViewer, DMZ). Динамическая группа не может содержать вложенные группы.

Примечание. Вы можете просмотреть, в какие группы входит актив, нажав в карточке актива на значок  , и если требуется, с помощью флажков добавить актив в статические группы. Вы можете использовать «систему подтверждения кандидатов», когда активы добавляются в динамические группы, а пользователь просматривает их и определяет, добавлять ли активы из динамических в родительские статические группы, тем самым подтверждая принадлежность активов группам.

- Чтобы создать группу активов из карточки актива:

1. В главном меню выберите **Активы**.
2. Выберите актив.
3. В карточке актива на вкладке **Конфигурация** выберите атрибут актива и нажмите  .
4. Выберите условие фильтрации, например `Host`.

   Условие фильтрации отобразится в нижней части карточки актива.

   Примечание. Вы можете выбрать несколько условий фильтрации.
5. Нажмите **Перейти к созданию группы**.

   Откроется страница **Создание группы**. PDQL-запрос с выбранными условиями фильтрации будет автоматически добавлен в поле **Фильтр**.

   Примечание. Вы можете сформировать PDQL-запрос вручную.

Примечание. Сложные запросы, в том числе приведенные в следующих разделах (например, запросы с оператором `JOIN`, запросы для поиска активов на основании косвенных признаков) необходимо сначала выполнить в группе **Все активы**, а затем применить их результаты для создания других групп.

6. Введите название группы активов.
7. Укажите расположение группы активов, выбрав группу, к которой она относится.
8. Выберите тип группы активов.

   Примечание. PDQL-запросы применяются только для динамических групп.
9. Нажмите **Сохранить**.

В таблице ниже представлен пример группировки активов (С — статическая группа, Д — динамическая группа).

Таблица 1. Пример группировки активов

Группа уровня 1	Группа уровня 2	Группа уровня 3	PDQL-запрос для динамической группы
Обязательные группы для настройки аудита служб каталогов, сбора данных о веб-приложениях	Службы каталогов (Д)	—	`directoryservice`
	Веб-сайты (Д)	—	`website`
	Виртуальные образы (Д)	—	`ImageSet`
Выполнение сканирования (С)	Минимальный набор информации (С)	Обнаруженные через `ping`, события, сторонний аудит (Д)	`host[@audittime = null and @pentesttime = null and endpoints = null]`
		Обнаруженные через HostDiscovery (см. раздел 2.1) (Д)	`host[@audittime = null and @pentesttime = null and endpoints != null]`
	Выполнено сканирование только методом черного ящика (см. раздел 2.2) (С)	Сканирование выполняется своевременно (Д)	`host[@audittime = null and @ScanningInfo[Type = "Pentest" and Status = "UpToDate"]]`
		Проблемы сканирования (Д)	`host[@audittime = null and @ScanningInfo[Type = "Pentest" and Status != "UpToDate"]]`

Группа уровня 1	Группа уровня 2	Группа уровня 3	PDQL-запрос для динамической группы
	Выполнено сканирование методом белого ящика (см. раздел 2.4) (С)	Сканирование выполняется своевременно (Д)	`host.@ScanningInfo[Type = "Audit" and Status = "UpToDate"]`
		Проблемы сканирования (Д)	`host.@ScanningInfo[Type = "Audit" and Status != "UpToDate"]`
	Необходимо сканирование обоими методами (черного (см. раздел 2.2) и белого (см. раздел 2.4) ящика) (С)	Сканирование выполняется своевременно (Д)	`host[@ScanningInfo[Type = "Audit" and Status = "UpToDate"] and @pentesttime and endpoints<transportendpoint>[port in [80, 443] and Status = "Open"]]`
		Проблемы сканирования (Д)	`(host[@ScanningInfo[Type = "Audit" and Status != "UpToDate"] or @pentesttime = null]) and host.endpoints<transportendpoint>[port in [80, 443] and Status = "Open"]`
Инфраструктурная роль (С)	Сетевое оборудование (С)	Все сетевое оборудование (Д)	`NetworkDeviceHost`
		Межсетевые экраны (МСЭ) (Д)	`NetworkDeviceHost.@DeviceType = "Firewall"`
		Маршрутизаторы (Д)	`NetworkDeviceHost.@DeviceType = "Router"`
		Коммутаторы (Д)	`NetworkDeviceHost.@DeviceType = "Switch"`
		Точка доступа (Д)	`NetworkDeviceHost.@DeviceType = "Access Point"`
		Сетевое оборудование не определено	`NetworkDeviceHost.@DeviceType = "UnknownNetworkDevice"`
	Контроллеры домена (Д)	—	`host.hostroles.Role = 'Domain Controller'`

Группа уровня 1	Группа уровня 2	Группа уровня 3	PDQL-запрос для динамической группы
	Виртуализация (С)	HostRole Virtualization (Д)	`host.hostroles.Role = "Virtualization"`
		Hyper-V (Д)	`host.Hypervisors.VirtType = "Hyper-V"`
		VMware Vcenters (Д)	`host.softs<VCenterInstance>`
		VMware ESXi (Д)	`esxihost`
	Цепочки поставок ПО (С)	HostRole CI/CD (Д)	`host.hostroles.Role in ["CI/CD", "Version Control System", "File Server"]`
		Artifactory (Д)	`host.softs<Artifactory>`
		GitLab (Д)	`host.softs<gitlab>`
	Proxy (С)	HostRole Proxy (Д)	`host.hostroles.Role = "Proxy Server"`
		HAProxy (Д)	`host.softs<HAProxy>`
	VPN (С)	HostRole VPN (Д)	`host.hostroles.Role = "VPN"`
		OpenVPN (Д)	`host.softs<OpenVPN>.configs`
	Security Software (С)	HostRole AntiVirus Server (Д)	`host.hostroles.Role = "Antivirus Server"`
		KSC (Д)	`host.softs<KasperskySecurityCenter>`
ОС (С)	Unix (Д)	—	`UnixHost`
	Windows (С)	Рабочие станции (Д)	`WindowsHost.HostType = 'Desktop'`
		Серверы (Д)	`WindowsHost.HostType = 'Server'`
Сетевые сегменты (С)	DMZ (Д)	—	`Host.@ipaddresses.item in [192.168.2.0/24, 10.0.0.0/24]`
	IT_testlab (Д)	—	`Host.@ipaddresses.item in 192.168.1.0/24`
	SOC (Д)	—	`Host.@ipaddresses.item in 192.168.0.0/24`

Группа уровня 1	Группа уровня 2	Группа уровня 3	PDQL-запрос для динамической группы
Уязвимости (С)	Узлы, подверженные некоторым уязвимостям Microsoft Exchange (Д)	—	`Host.@Vulners[CVEs.Item in ['CVE-2021-26855', 'CVE-2021-26857', 'CVE-2021-26858', 'CVE-2021-27065'] and status != 'fixed']`
	Узлы, подверженные уязвимостям компонентов Microsoft Office (Д)	—	`Host.softs<OfficeComponent>.@Vulners.status != 'fixed'`

Примечание. Примеры проблем сканирования: не настроено расписание задачи на сканирование, не установлены сроки актуальности данных в политиках на странице **Система** → **Политики** (для выявления таких активов вы можете выполнить PDQL-запрос: `host.@ScanningInfo.Status = "NotDefined"`).

2.4. Сканирование активов методом белого ящика (аудит)

Сканирование активов методом белого ящика (аудит) выполняется с помощью MaxPatrol VM и позволяет получить более полную информацию об активах, используя данные об учетных записях.

> **Внимание!** Необходимо выполнить аудит всех систем, имеющихся в IT-инфраструктуре. Рекомендуется проводить аудит в соответствии с порядком следования разделов в этом документе. Наиболее важно выполнить аудит сетевых устройств (см. раздел 2.4.1).

Для выполнения аудита необходимо осуществить настройку актива, добавить учетную запись для доступа к активу, создать и запустить задачу на аудит с определенным профилем.
В параметрах профиля для задачи аудита необходимо указать учетную запись, которой предоставлены права на выполнение команд для сбора данных об активе. В параметрах задачи рекомендуется настроить ее расписание.

Инструкции по настройке источников, добавлению учетных записей для доступа к активам, созданию и запуску задач на аудит активов приведены в Руководстве по настройке источников MaxPatrol VM. Для сбора данных об активе на нем выполняются команды и используются порты, которые приведены в приложении к Руководству по настройке источников MaxPatrol VM.

Стандартные профили аудита описаны в разделе «Модуль Audit» Руководства по настройке источников MaxPatrol VM.

Типы учетных записей описаны в разделе «Работа с учетными записями» Руководства оператора MaxPatrol VM.

Примеры создания задач на аудит с разными профилями (для разных систем) представлены в разделах «Создание и запуск задачи на аудит актива» Руководства по настройке источников MaxPatrol VM.

После завершения аудита необходимо проверить собранную об активах информацию с помощью PDQL-запросов, которые приведены в следующих разделах. Синтаксис PDQL описан в документе «Синтаксис языка запросов PDQL» MaxPatrol VM.

В этом разделе

2.4.1. Аудит сетевых устройств

Аудит сетевых устройств позволяет получить:

- информацию из ARP-таблиц, таблиц маршрутизации, списков управления доступом (ACL) — для построения сетевой топологии и проверки достижимости;
- список терминированных подсетей — для определения сетевого периметра;
- существующие правила таблицы преобразования адресов (NAT) — для определения внутренних узлов, напрямую доступных из внешней сети;
- информацию о других устройствах, для которых требуется выполнить аудит.

Поиск сетевых устройств включает:

1. Поиск сетевых устройств без разделения по типам.
2. Поиск сетевых устройств с функцией фильтрации, в том числе внешних.
3. Поиск сетевых устройств через соседние устройства.
4. Поиск сетевых устройств через конечные узлы.

Примечание. Использовать поиск по портам при сканировании сетевых устройств нецелесообразно, так как существует много вариантов сетевых устройств и закрытых доступов.

Внимание! Для обеспечения качества аудита необходимо циклически выполнять запросы, представленные в этом разделе. Часто при оценке качества аудита могут быть найдены не выявленные ранее устройства. При выявлении нового узла необходимо снова выполнить остальные запросы, так как в их результатах могут быть серьезные изменения, требующие выполнения новых задач аудита. Например, выявив новый межсетевой экран, можно обнаружить новые точки входа, новые соседние сетевые устройства или даже подсеть.

Поиск сетевых устройств без разделения по типам

Поиск сетевых устройств без разделения по типам необходим прежде всего для оценки топологии сети и проверки того, что все устройства связаны друг с другом, а также для проверки актуальности аудита.

Запрос для поиска сетевых устройств без разделения по типам:

```
select(
    @networkdevicehost, networkdevicehost.@id,
    networkdevicehost.@DeviceType, networkdevicehost.@iplist,
    networkdevicehost.vendor, networkdevicehost.modelNumber,
    networkdevicehost.isvirtual, networkdevicehost.@audittime) |
unique() |
calc(TotalDays(now() - networkdevicehost.@AuditTime) as dur_audit) |
sort(networkdevicehost.@DeviceType)
```

Результат запроса будет содержать значения параметров `dur_audit` и `networkdevicehost.@DeviceType`. Для устройств с функцией маршрутизации, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), а также устройств, тип которых не определен (`networkdevicehost.@DeviceType = "unknownnetworkdevice"`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Если требуется, вы можете узнать номера моделей сетевых устройств с помощью запроса:

```
select(networkdevicehost.modelNumber) |
filter(networkdevicehost.modelNumber != null) | unique()
```

Результат запроса будет содержать значение параметра `networkdevicehost.modelNumber`. Описание устройства по номеру модели вы можете найти в открытых источниках.

Поиск сетевых устройств с функцией фильтрации

Поиск сетевых устройств, способных осуществлять фильтрацию, таких как маршрутизаторы и межсетевые экраны, рекомендуется осуществлять в первую очередь. Определить такие устройства можно с помощью запросов, представленных ниже.

Запрос для поиска маршрутизаторов и межсетевых экранов:

```
select(
    @networkdevicehost, networkdevicehost.@DeviceType,
    networkdevicehost.@iplist, networkdevicehost.vendor,
    networkdevicehost.modelNumber, networkdevicehost.isvirtual,
    networkdevicehost.@audittime) |
filter(networkdevicehost.@DeviceType not in
      ["switch", "unknownnetworkdevice"]) |
calc(TotalDays(now() - networkdevicehost.@AuditTime) as dur_audit) |
sort(networkdevicehost.@DeviceType, networkdevicehost.isvirtual,
    networkdevicehost.modelNumber)
```

Запрос для поиска внешних роутеров и межсетевых экранов:

```
filter(networkdevicehost.@DeviceType not in
       ["switch", "unknownnetworkdevice"]) |
select(
    @networkdevicehost, networkdevicehost.@id,
    networkdevicehost.@DeviceType, networkdevicehost.vendor,
    networkdevicehost.modelNumber, networkdevicehost.isvirtual,
    networkdevicehost.@audittime, networkdevicehost.@ipaddresses) |
filter(networkdevicehost.
    @ipaddresses.item not in
      [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16, 0.0.0.0/32, 169.254.0.0/16,::1/128,
127.0.0.0/8, 100.64.0.0/10]) |
select(
    @networkdevicehost, networkdevicehost.@id,
    networkdevicehost.@DeviceType, networkdevicehost.vendor,
    networkdevicehost.modelNumber, networkdevicehost.isvirtual,
    networkdevicehost.@audittime,
    compactunique(networkdevicehost.@ipaddresses)) |
sort(@networkdevicehost)
```

Необходимо проверить, что все задачи аудита выполнены и качество аудита обеспечено.

Вы можете использовать в запросе адреса, указанные на сетевых интерфейсах, вместо общего поля `@IpAddresses`:

```
filter(networkdevicehost.@DeviceType not in
     ["switch", "unknownnetworkdevice"])|
select(@host, host.@id, Host.Interfaces.id,
    Host.Interfaces.L3Settings.Address.Address.Address as ip,
    Host.Interfaces.L3Settings.Address.Address.NetworkID as net_id,
    Host.Interfaces.L3Settings.Address.Address.Prefix as pref) |
filter(ip and ip not in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16,
0.0.0.0/32,127.0.0.0/8, 169.254.0.0/16,fe80::/10, ::1/128, 100.64.0.0/10]) |
select(@host, host.@id, Host.Interfaces.id, ip, net_id, pref) |
unique() | sort(net_id, @host)
```

Поиск сетевых устройств через соседние устройства

Запрос для поиска сетевых устройств, не попавших в результаты сканирования, с помощью соседних устройств (запрос применим, если возможность поиска через соседние устройства включена):

```
select(@networkdevicehost, networkdevicehost.@DeviceType, networkdevicehost.@Id,
networkdevicehost.@AuditTime, networkdevicehost.Neighbors.OwnInterface,
networkdevicehost.Neighbors.DeviceID, networkdevicehost.Neighbors.SysDescr,
networkdevicehost.Neighbors.OsName, networkdevicehost.Neighbors.IPAddresses,
networkdevicehost.Neighbors.neighborInterface) |
filter(networkdevicehost.Neighbors.DeviceID) | unique() |
join(filter(networkdevicehost.Neighbors != null) |
select(networkdevicehost.@id, networkdevicehost.interfaces.Id,
networkdevicehost.interfaces.isenabled,
networkdevicehost.interfaces.L3Settings.address.address.address) as own,
own.networkdevicehost.@id = networkdevicehost.@id and
own.networkdevicehost.interfaces.Id = networkdevicehost.Neighbors.OwnInterface) |
join(select(@networkdevicehost, networkdevicehost.@DeviceType, networkdevicehost.@id,
networkdevicehost.@audittime, networkdevicehost.fqdn, networkdevicehost.hostname,
networkdevicehost.@ipaddresses) as nei, nei.networkdevicehost.fqdn =
networkdevicehost.Neighbors.DeviceID OR nei.networkdevicehost.hostname =
networkdevicehost.Neighbors.DeviceID OR nei.networkdevicehost.@ipaddresses =
networkdevicehost.Neighbors.IPAddresses) |
select (@networkdevicehost, networkdevicehost.@DeviceType,
networkdevicehost.@AuditTime, networkdevicehost.Neighbors.OwnInterface,
own.networkdevicehost.interfaces.isenabled,
own.networkdevicehost.interfaces.L3Settings.address.address.address,
networkdevicehost.Neighbors.DeviceID, nei.@networkdevicehost,
nei.networkdevicehost.@DeviceType, nei.networkdevicehost.@audittime) |
calc(TotalDays(now() - nei.networkdevicehost.@AuditTime) as dur_audit) | unique() |
sort(@networkdevicehost, networkdevicehost.Neighbors.OwnInterface,
networkdevicehost.Neighbors.DeviceID)
```

Результат запроса будет содержать список устройств, которые являются соседними для известных сетевых устройств. Если через устройство `@networkdevicehost` на интерфейсе `networkdevicehost.Neighbors.OwnInterface` с адресом `own.networkdevicehost.interfaces.L3Settings.address.address.address` (может быть пустым) получена информация об устройстве `networkdevicehost.Neighbors.DeviceID`, а в результатах сканирования такого устройства нет, то значение параметра `nei.@networkdevicehost` будет пустым. Такие устройства необходимо отсканировать дополнительно, чтобы получить о них полную информацию. Если аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо найти задачу на аудит (см. раздел 2.4) и изменить расписание ее запуска или выяснить, почему аудит не был выполнен, и устранить причины.

> **Примечание.** Результат запроса будет содержать дублирующиеся записи. Это необходимо, чтобы отобразить, для какого устройства является соседним данное устройство. Если требуется, вы можете получить список уникальных записей. Для этого в запросе, приведенном выше, в последнем условии `select` нужно удалить все параметры, кроме `networkdevicehost.Neighbors.DeviceID, nei.@networkdevicehost, nei.networkdevicehost.@DeviceType, nei.networkdevicehost.@audittime`.

Поиск сетевых устройств через конечные узлы

Если на этапах сканирования портов (см. раздел 2.1) и сканирования активов методом черного ящика (см. раздел 2.2) получена информация о разных подсетях и проведен аудит хотя бы одного узла из каждой подсети, вы можете определить шлюзы подсети и проверить, был ли выполнен их аудит, с помощью запроса:

```
select(host.RoutingTables.Routes.Gateway as gate) |
filter(gate) | unique() |
join(select(@host, host.@ipaddresses, host.@audittime) as gw, gw.host.@ipaddresses =
gate) |
select(gate, gw.host.@audittime) |
filter(gw.host.@audittime = null) | unique() | sort(gate)
```

Если в результате запроса найдены данные об одном или нескольких шлюзах, то необходимо создать (запустить) задачу на аудит (см. раздел 2.4) каждого шлюза с адресом из параметра `gate`.

Оценка качества аудита сетевых устройств

Для оценки качества аудита сетевых устройств вы можете выполнить запрос:

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@DeviceType,
    compactunique(networkdevicehost.Interfaces),
    compactunique(networkdevicehost.ARPTable),
    compactunique(networkdevicehost.routingtables),
    networkdevicehost.RoutingSettings.IPv4Unicast,
    compactunique(networkdevicehost.appliedacl.name),
    compactunique(networkdevicehost.accesslists.name),
    compactunique(networkdevicehost.nattable),
    compactunique(networkdevicehost.VRF)) |
unique() | sort(networkdevicehost.@audittime, @networkdevicehost) |
group(networkdevicehost.@DeviceType) | sort(networkdevicehost.@DeviceType)
```

Результат запроса будет содержать значения параметров:

— `@audittime` — время последнего аудита;

— `DeviceType` — тип устройства;

— `Interfaces` — интерфейсы устройства.

Необходимо проверить выполнение следующих условий:

— Каждый из параметров `@audittime`, `DeviceType`, `Interfaces` содержит хотя бы одно значение.

— Со времени последнего аудита прошло не больше двух недель (`@audittime <= 14`).

— Нет устройств с неизвестным типом (`DeviceType = "unknownnetworkdevice"`) или устройств, у которых отсутствует интерфейс.

Примечание. Исключением могут являться менеджеры кластера, например, для менеджера кластера устройств Check Point после аудита будет указано значение `DeviceType = "unknownnetworkdevice"`.

- Для всех устройств, кроме коммутаторов, заполнено значение параметра `networkdevicehost.RoutingSettings.IPv4Unicast`.
- Для маршрутизаторов заполнены значения параметров `ARPTable`, `routingtables` и `VRF`.
- Для межсетевых экранов заполнены значения параметров `appliedacl`, `accesslists`, `nattable` или межсетевой экран специфически настроен. Запрос для поиска межсетевых экранов, для которых не заполнено значение хотя бы одного из этих параметров:

```
select(
    @networkdevicehost, networkdevicehost.@devicetype,
    compactunique(networkdevicehost.appliedacl) as apl_acl,
    compactunique(networkdevicehost.accesslists) as acl,
    compactunique(networkdevicehost.nattable) as nat) |
filter(
    networkdevicehost.@devicetype = "firewall"
    and (apl_acl = null OR acl = null OR nat = null))
```

Если хотя бы одно из перечисленных выше условий не выполнено, то считается, что качество аудита не обеспечено. Необходимо проверить, созданы ли и запущены ли задачи на аудит (см. раздел 2.4). Если задач нет и сетевые устройства не найдены с помощью запросов, представленных в этом разделе, то необходимо создать (запустить) задачи на аудит (см. раздел 2.4).

Если аудит активов проводился больше двух недель назад (`@audittime > 14`), то необходимо найти задачи на аудит (см. раздел 2.4) и изменить расписание их запуска или выяснить, почему аудит не был выполнен, и устранить причины.

В этом разделе

2.4.1.1. Аудит устройств с помощью таблиц преобразования адресов (NAT)

Таблицы NAT позволяют при активном соединении определить внутренний адрес устройства по внешнему адресу. Вы можете найти узлы, доступные из внешней сети, а также проверить, выполнялся ли аудит внутренних узлов, указанных в таблицах NAT.

Запрос для поиска узлов, доступных из внешней сети

```
select(@NetworkDeviceHost,
    NetworkDeviceHost.@id as id,
    NetworkDeviceHost.NATTable.Name as name,
    NetworkDeviceHost.NATTable.Rules.Comment as comment,
    NetworkDeviceHost.NATTable.Rules.LineNumber as ln,
    NetworkDeviceHost.NATTable.Rules.Type as type,
    compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedSource.Values.address)
as source,
    NetworkDeviceHost.NATTable.Rules.NormalizedProtocol.Protocol.Values as proto,
    NetworkDeviceHost.NATTable.Rules.NormalizedProtocol.TCPUDPOptions.DestinationPort
s.Values as dst_port,
    compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedDestination.Values.addre
ss) as destination) |
filter(source != null and source.item != ::) |
join(select (NetworkDeviceHost.@id as id,
    NetworkDeviceHost.NATTable.Name as name,
    NetworkDeviceHost.NATTable.Rules.LineNumber as ln,
    compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedSourceAddress.
Values.Address) as ntsrc,
    compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAdd
ress.Values.Address) as ntdst,
    compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAdd
ress.Values.Prefix) as ntdst_pref,
    NetworkDeviceHost.NATTable.rules.NormalizedTranslatedProtocol.TCPUDPOptions.Desti
nationPorts.Values as ntpdst_port) |
filter(name != null) as Q, id = Q.id and Q.name= name and Q.ln = ln) | filter((source
and source in 0.0.0.0/32) or (Q.ntsrc and Q.ntsrc not in [172.16.0.0/12, 10.0.0.0/8,
192.168.0.0/16, ::1/128, 100.64.0.0/10]))
| select(name, comment, ln, type, source, Q.ntsrc, proto, destination, dst_port,
Q.ntdst, Q.ntpdst_port) | unique()
```

Преобразование адресов осуществляется как для входящего, так и для исходящего трафика. Входящий или исходящий сетевой пакет достигает устройства, на котором настроено правило NAT. Сетевой пакет содержит значения параметров `source`, `source port`, `destination`, `destination port`. Если они соответствуют правилу NAT, то в пакете значение параметра `source` заменяется на `NormalizedTranslatedSourceAddress`, значение параметра `destination` — на `NormalizedTranslatedDestinationAddress`. Если в правиле NAT указано значение `source`, равное `0.0.0.0`, то правило не накладывает ограничений на адрес источника соединения. В этом случае в правиле может быть определен порт или соответствие правилу проверяется по значению параметра `destination`. Если в правиле NAT значение параметра `NormalizedTranslatedSourceAddress` пусто, то значение `source` не меняется.

Условия применения правил NAT проверяются на каждом устройстве, на котором настроены правила NAT. Внутри устройства правила применяются последовательно по возрастанию номера строки в конфигурации (`ln`). Таким образом, если сначала указано более общее правило, а затем — более узкое, то применяется более узкое.

Результат запроса будет содержать правила NAT, позволяя определить узлы, которые являются точками входа в инфраструктуру или выхода во внешнюю сеть. Необходимо периодически выполнять этот запрос, чтобы отслеживать состав таких узлов и его изменения.

Чтобы конкретизировать результаты запроса, приведенного выше, вы можете использовать вместо условия `| filter((source and source in 0.0.0.0/32) or (Q.ntsrc and Q.ntsrc not in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16, ::1/128, 100.64.0.0/10]))` другие условия фильтрации.

Если указать условие фильтрации `| filter(source and source in 0.0.0.0/32)`, то результат запроса будет содержать правила NAT с адресами узлов, доступных из внешней сети с любых адресов.

Если указать условие фильтрации `| filter(Q.ntdst_pref = 32 and Q.ntdst and Q.ntdst in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16, ::1/128, 100.64.0.0/10])`, то результат запроса будет содержать правила NAT, ведущие на внутренний единичный адрес, позволяя определить серверы со статическим адресом, доступные из внешней сети с любых или только с определенных адресов. Также в результат запроса попадут и внутренние правила NAT, которые осуществляют преобразование внутренних адресов во внутренние адреса.

Если указать условие фильтрации `| filter(Q.ntdst_pref = 32 and Q.ntdst and Q.ntdst in [10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, ::1/128, 100.64.0.0/10] and destination and destination not in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16, ::1/128, 100.64.0.0/10, 0.0.0.0, ::])`, то результат запроса будет содержать правила NAT, которые преобразуют значения параметра `destination` из внешних во внутренние, позволяя определить точки входа, доступные из внешней сети с любых или только с определенных адресов.

Примечание. Вы можете использовать одновременно несколько условий фильтрации.

Запросы для проверки выполнения аудита внутренних узлов из таблиц NAT

Запрос 1:

```
select(NetworkDeviceHost.NATTable.Rules.NormalizedSource.Values.address as address, NetworkDeviceHost.NATTable.Rules.NormalizedSource.Values.prefix as pref) | filter(pref = 32 and address) | unique() |

join(select(@host, host.@audittime, host.@ipaddresses) as NATed, NATed.host.@ipaddresses = address) | calc(TotalDays(now() - NATed.host.@AuditTime) as dur_audit) | SORT(address)
```

Запрос 2:

```
select(NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedSourceAddress.Values.address as address, NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedSourceAddress.Values.prefix as pref) | filter(pref = 32 and address) | unique() |
```

```
join(select(@host, host.@audittime, host.@ipaddresses) as NATed,
NATed.host.@ipaddresses = address) | calc(TotalDays(now() -
NATed.host.@AuditTime) as dur_audit) | SORT(address)
```

Запрос 3:

```
select(NetworkDeviceHost.NATTable.Rules.NormalizedDestination.Values.address as
address, NetworkDeviceHost.NATTable.Rules.NormalizedDestination.Values.prefix
as pref) | filter(pref = 32 and address) | unique() |

join(select(@host, host.@audittime, host.@ipaddresses) as NATed,
NATed.host.@ipaddresses = address) | calc(TotalDays(now() -
NATed.host.@AuditTime) as dur_audit) | SORT(address)
```

Запрос 4:

```
select(NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAddress.
Values.address as address,
NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAddress.Values.
prefix as pref) | filter(pref = 32 and address) | unique() |

join(select(@host, host.@audittime, host.@ipaddresses) as NATed,
NATed.host.@ipaddresses = address) | calc(TotalDays(now() -
NATed.host.@AuditTime) as dur_audit) | SORT(address)
```

Если в результате любого из этих запросов значение параметра `NATed.@host` пусто для внутренних узлов, то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) внутренних узлов, которые указаны в правилах NAT, и настроить их расписание.

Если узел, данные о котором указаны в результате запроса, уже не существует, то необходимо удалить правило NAT для этого узла. В противном случае злоумышленник может использовать сетевой адрес и получить точку входа в инфраструктуру, не изменяя настроек межсетевого экрана.

Если в результате любого из этих запросов значение параметра `NATed.@host` не пусто, но значение параметра `NATed.host.@AuditTime` пусто или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) таких узлов и настроить их расписание.

2.4.1.2. Аудит устройств с помощью списков управления доступом (ACL)

ACL определяют набор правил доступа к ресурсам системы или сети.

Запрос для поиска узлов с ACL

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@id, networkdevicehost.appliedacl.name) |
filter(networkdevicehost.appliedacl.name) | unique() |
```

```
join(
    select(
      networkdevicehost.@id,
      networkdevicehost.AccessLists.name as name) as acl,
    acl.networkdevicehost.@id = networkdevicehost.@id
    and acl.name = networkdevicehost.appliedacl.name) |
calc(TotalDays(now() - networkdevicehost.@AuditTime) as dur_audit)
```

Для узлов, у которых по результатам запроса значение параметра `networkdevicehost.appliedacl.name` заполнено, но значение параметра `acl.name` отсутствует, а также аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос таблицы ACL

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@id, networkdevicehost.appliedacl.name) |
filter(networkdevicehost.appliedacl.name) | unique() |
join(
    select(
        networkdevicehost.@id, networkdevicehost.AccessLists.name as name,
        networkdevicehost.AccessLists.rules <ActionRule>.linenumber as ln,
        networkdevicehost.AccessLists.rules <ActionRule>.normalizedAction as n_act,
        networkdevicehost.AccessLists.rules
<ActionRule>.Source.Address.Values.address as source,
        networkdevicehost.AccessLists.rules
<ActionRule>.NormalizedSource.Values.address as ntsrc,
        networkdevicehost.AccessLists.rules <ActionRule>.Direction as Direction,
        networkdevicehost.AccessLists.rules
<ActionRule>.Destination.Address.Values.address as dest,
        networkdevicehost.AccessLists.rules
<ActionRule>.NormalizedDestination.Values.address as ntdst) as acl,
    acl.networkdevicehost.@id = networkdevicehost.@id
    and acl.name = networkdevicehost.appliedacl.name) |
select(
    acl.n_act, acl.source, acl.ntsrc,
    acl.Direction, acl.dest, acl.ntdst) | unique()
```

Запросы для проверки выполнения аудита узлов из ACL

Запрос 1:

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@id, networkdevicehost.appliedacl.name) |
filter(networkdevicehost.appliedacl.name) | unique() |
join(
```

```
    select(
      networkdevicehost.@id,
      networkdevicehost.AccessLists.name as name,
      networkdevicehost.AccessLists.rules <ActionRule>.Source.Address.Values.address
as source,
      networkdevicehost.AccessLists.rules <ActionRule>.Source.Address.Values.prefix
as pref) as acl,
acl.networkdevicehost.@id = networkdevicehost.@id
and acl.name = networkdevicehost.appliedacl.name) |
select(acl.source, acl.pref) | filter(acl.source) | unique() |
join(
    select(@host, host.@audittime, host.@ipaddresses) as H,
      H.host.@ipaddresses = acl.source) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime, acl.source)
```

Запрос 2:

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@id, networkdevicehost.appliedacl.name) |
filter(networkdevicehost.appliedacl.name) | unique() |
join(
    select(
      networkdevicehost.@id, networkdevicehost.AccessLists.name as name,
      networkdevicehost.AccessLists.rules
<ActionRule>.NormalizedSource.Values.address as ntsrc,
      networkdevicehost.AccessLists.rules <ActionRule>.NormalizedSource.Values.prefix
as pref) as acl,
acl.networkdevicehost.@id = networkdevicehost.@id
and acl.name = networkdevicehost.appliedacl.name) |
select(acl.ntsrc, acl.pref) | filter(acl.ntsrc) | unique() |
join(
    select(@host, host.@audittime, host.@ipaddresses) as H,
    H.host.@ipaddresses = acl.ntsrc) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime, acl.ntsrc)
```

Запрос 3:

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@id, networkdevicehost.appliedacl.name) |
filter(networkdevicehost.appliedacl.name) | unique() |
join(
    select(
      networkdevicehost.@id,
      networkdevicehost.AccessLists.name as name,
      networkdevicehost.AccessLists.rules
<ActionRule>.Destination.Address.Values.address as dest,
```

```
      networkdevicehost.AccessLists.rules
<ActionRule>.Destination.Address.Values.prefix as pref) as acl,
      acl.networkdevicehost.@id = networkdevicehost.@id
and acl.name = networkdevicehost.appliedacl.name) |
select(acl.dest, acl.pref) | filter(acl.dest) | unique() |
join(
    select(@host, host.@audittime, host.@ipaddresses) as H,
    H.host.@ipaddresses = acl.dest) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime, acl.dest)
```

Запрос 4:

```
select(
    @networkdevicehost, networkdevicehost.@audittime,
    networkdevicehost.@id, networkdevicehost.appliedacl.name) |
filter(networkdevicehost.appliedacl.name) | unique() |
join(
    select(
      networkdevicehost.@id,
      networkdevicehost.AccessLists.name as name,
      networkdevicehost.AccessLists.rules
<ActionRule>.NormalizedDestination.Values.address as ntdst,
      networkdevicehost.AccessLists.rules
<ActionRule>.NormalizedDestination.Values.prefix as pref) as acl,
    acl.networkdevicehost.@id = networkdevicehost.@id
    and acl.name = networkdevicehost.appliedacl.name) |
select(acl.ntdst, acl.pref) | filter(acl.ntdst) | unique() |
join(
    select(@host, host.@audittime, host.@ipaddresses) as H,
    H.host.@ipaddresses = acl.ntdst) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime, acl.ntdst)
```

Для узлов, у которых по результатам запросов значение параметра `pref` равно `32`, IP-адрес существует, но значение параметра `@host` пусто или аудит не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

2.4.1.3. Аудит устройств с внешними адресами

Вы можете предоставить доступ к устройству из внешней сети следующими способами:

1. Создать правило преобразования адресов (NAT) (см. раздел 2.4.1.1) на сетевом устройстве, имеющем прямой доступ к сети провайдера.
2. Назначить устройству внешний адрес из диапазона адресов маршрутизации и прописать правила ACL (см. раздел 2.4.1.2) на межсетевом экране или добавить узел с внешним адресом напрямую в сеть провайдера. Ниже приведены запросы для поиска таких устройств.

   Примечание. Добавление узла с внешним адресом напрямую в сеть провайдера увеличивает риски безопасности. Рекомендуется контролировать и ограничивать внешний доступ с использованием межсетевых экранов, ACL, NAT, VPN и других механизмов защиты.

Запрос для поиска узлов с внешними адресами по адресам на узлах

```
filter(not networkdevicehost)|
select(@host, host.@id, Host.@ipaddresses, host.@audittime) |
filter(host.@ipaddresses.item not in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16,
0.0.0.0/32, 169.254.0.0/16,::1/0, 127.0.0.0/8, 100.64.0.0/10]) |
select(@host, host.@id, compactunique(Host.@ipaddresses), host.@audittime) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(host.@audittime ASC,
@host ASC)
```

Результат запроса будет содержать список узлов с внешними адресами. Значения параметра `@IPAdresses` могут быть неактуальными, если внешние адреса указаны у узлов, на которых они уже не используются.

Внешние адреса на узлах могут появиться по разным причинам. Если в результатах слишком много узлов, то это может быть связано с ПО, которое может отдавать внешнюю адресацию узла в MaxPatrol VM (MaxPatrol SIEM). Рассмотрим пример, когда в инфраструктуре установлены MaxPatrol SIEM и MaxPatrol EDR. Необходимо выполнить запрос для поиска узлов MaxPatrol EDR, представленный ниже, и проанализировать причины появления внешних адресов.

```
filter(not networkdevicehost)|
select(@host, host.@type, host.@updatetime, host.@id, Host.@ipaddresses,
host.@audittime) |
filter(host.@ipaddresses.item not in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16,
0.0.0.0/32, 169.254.0.0/16, ::1/128, 127.0.0.0/8, FE80::/10, 100.64.0.0/10, ::1/0]) |
select(@host, host.@type, host.@updatetime, host.@id,
compactunique(Host.@ipaddresses) as ips, host.@audittime) |
join(select(host.@id, host.softs <XDRAgent>) as edr, edr.host.@id = host.@id) |
filter(edr.host.softs <XDRAgent>) |
select(@host, host.@audittime, ips) | calc(TotalDays(now() - host.@AuditTime) as
dur_audit) | sort(host.@audittime, @host)
```

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

> **Примечание.** В MaxPatrol SIEM на вкладке **События** вы можете применить фильтр `event_src.title = "xdr" and msgid = "register_pt_edr_agent"`. Из этих событий от MaxPatrol EDR в MaxPatrol SIEM создаются активы.

Запрос для поиска узлов с внешними адресами с пользовательскими ОС

```
filter(not networkdevicehost) |
select(@host, host.osname, Host.interfaces.name, Host.interfaces.isenabled,
Host.interfaces.l3settings.address.address.address as address, host.@audittime) |
filter(host.osname match "^Windows ([\w.]{2,3}|Vista)$" and address not in
[172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16, 0.0.0.0/32, 169.254.0.0/16, ::1/128,
127.0.0.0/8, FE80::/10, ::1/0, 100.64.0.0/10]) |
select(@host, host.osname, Host.interfaces.name, Host.interfaces.isenabled,
compactunique(address), host.@audittime) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(host.osname, @host)
```

> **Примечание.** Запрос, приведенный выше, применим для узлов с Windows. Для ОС семейства Unix необходимо удалить из запроса фильтрацию по `host.osname match "^Windows ([\w.]{2,3}|Vista)$"`, сгруппировать данные и определить, узлы с какой ОС могут иметь внешние адреса. Запрос для группировки данных об ОС:

```
filter(not networkdevicehost) |
select(@host, host.osname) |
filter(host.osname and not host.osname match "^Windows") |
group(host.osname, COUNT(*)) | sort("COUNT(*)" DESC)
```

Если в результатах запроса есть данные об узлах с пользовательскими ОС, то необходимо удалить внешние адреса для них. Как правило, не рекомендуется организация доступа к пользовательским машинам напрямую из внешней сети, иначе они будут являться точками входа в инфраструктуру. Если, тем не менее, использование внешних адресов на узлах с пользовательскими ОС необходимо, то рекомендуется обучить пользователей правилам безопасности и своевременно выполнять аудит этих узлов.

Запрос для поиска узлов с внешними адресами с серверными ОС

```
filter(not networkdevicehost) |
select(@host, host.osname, Host.interfaces.name, Host.interfaces.isenabled,
Host.interfaces.l3settings.address.address.address as address, host.@audittime) |
filter(host.osname not match "^Windows ([\w.]{2,3}|Vista)$" and address not in
[172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16, 0.0.0.0/32, 169.254.0.0/16, :: 1/128,
127.0.0.0/8, FE80::/10, ::1/0, 100.64.0.0/10]) |
select(@host, host.osname, Host.interfaces.name, Host.interfaces.isenabled,
compactunique(address), host.@audittime) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(host.osname,
host.@audittime, @host)
```

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить, обосновано ли использование внешних адресов, и если нет, то отказаться от использования внешних адресов для этих узлов.

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для поиска узлов с внешними адресами по данным с сетевых устройств

Запрос правил маршрутизации:

```
select(@networkdevicehost,
networkdevicehost.RoutingTables.Routes.Destination.NetworkID as net_id,
networkdevicehost.RoutingTables.Routes.Destination.prefix,
networkdevicehost.RoutingTables.Routes.Destination.address) |
filter(net_id and net_id not in [172.16.0.0/12, 10.0.0.0/8, 192.168.0.0/16,
0.0.0.0/32, 169.254.0.0/16, :: 1/128, 127.0.0.0/8, FE80::/10, 100.64.0.0/10, :: 1/0])
| unique() | sort(net_id ASC)
```

Запрос для поиска подсетей без дублирования с разных устройств:

```
select(networkdevicehost.RoutingTables.Routes.Destination.NetworkID as net_id,
networkdevicehost.RoutingTables.Routes.Destination.prefix) |
filter(net_id and net_id not in [172.16.0.0/12, 10.0.0.0/8,     192.168.0.0/16,
0.0.0.0/32, 169.254.0.0/16, ::1/128, 127.0.0.0/8, FE80::/10, 100.64.0.0/10, ::1/0]) |
unique() | sort(net_id ASC)
```

В результате этого запроса будет получен список подсетей. Для поиска узлов с внешними адресами вы можете использовать этот список, выполнив запрос:

```
filter(not networkdevicehost)|
select(@host, host.@id, Host.@ipaddresses, host.@audittime) |
filter(host.@ipaddresses.item in [%Полученный список подсетей%]) |
select(@host, host.@id, compactunique(Host.@ipaddresses) as ips, host.@audittime) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
sort(host.@audittime ASC, @host ASC)
```

Если известно, какие из полученных подсетей используются только для выхода в интернет, необходимо удалить их из запроса, настроив правила маршрутизации.

Чтобы сгруппировать результаты и найти подсети с большим количеством активов, вы можете добавить в конец фильтра `| group(ips, COUNT(*)) | sort("COUNT(*)" DESC)`.

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Поиск серверов, доступных из внешней сети, с помощью PT NAD

Если в инфраструктуре установлен PT NAD, вы можете использовать его для мониторинга трафика и поиска серверов, доступных из внешней сети, выполнив следующие шаги:

1. Применить фильтр `src.groups == EXTERNAL_NET && dst.groups == HOME_NET && !dhcp && !app_proto == "dhcp" && !flags == "SPLIT"`.
2. На странице **Дашборды** экспортировать данные виджета **Серверы по сессиям и трафику**, нажав . В результате будут найдены данные о серверах и узлах, которые являются точками входа в инфраструктуру или выхода во внешнюю сеть, а также данные об использовании адресов link-local, предназначенных для коммуникаций в пределах одного сегмента сети.
3. Создать группы узлов LINK_LOCAL (169.254.1.0/16) и USER_NET (включает подсети, которые используются для выхода в интернет).

   Примечание. Создание групп описано в разделе «Управление группами узлов и портов» Руководства оператора PT NAD.
4. Применить фильтр `src.groups == EXTERNAL_NET && !src.groups == LINK_LOCAL && dst.groups == HOME_NET && dst.groups != USER_NET && !dhcp && !app_proto == "dhcp" && !flags == "SPLIT"`. В результате будут найдены данные о серверах, доступных из внешней сети, которые являются точками входа в инфраструктуру. Вы можете сравнить полученные результаты с результатами запроса для поиска узлов, доступных из внешней сети, с помощью правил NAT (см. раздел 2.4.1.1).

2.4.2. Аудит доменов и контроллеров доменов

Аудит доменов и контроллеров доменов рекомендуется выполнять в следующем порядке:

1. Получить информацию хотя бы об одном контроллере домена в каждом защищаемом лесе доменов, а также учетные записи для выполнения задач аудита (см. раздел 2.4).
2. Выполнить аудит узлов с определенным профилем (см. раздел 2.4).
3. Выполнить аудит узлов с профилем Microsoft Active Directory Audit (далее также — аудит по LDAP).
4. Выполнить аудит всех контроллеров доменов.

Поиск доменов и контроллеров доменов, для которых не был выполнен аудит

Если между доменами А и Б установлены отношения доверия (trusts), то в домене А есть информация о существовании домена Б. Если был выполнен аудит для домена А, то вы можете проверить, был ли выполнен аудит для домена Б, с помощью запроса:

```
select(Activedirectory.forestTrusts, Activedirectory.domains.domainTrusts) |
calc(Activedirectory.forestTrusts + Activedirectory.domains.domainTrusts as domain) |
```

```
select (domain) | unique() |
join(
    select(@activedirectory, activedirectory.domains.name as name,
    ActiveDirectory.@updatetime) as AD, AD.name = domain)
```

Для доменов, данные о которых найдены в результате запроса (заполнены значения параметров `AD.@activedirectory` и `AD.name`), необходимо найти контроллеры доменов и создать (запустить) задачи (см. раздел 2.4) на аудит по LDAP. Если аудит проводился больше месяца назад (`ActiveDirectory.@updatetime > 30`), то необходимо найти задачу на аудит (см. раздел 2.4) и изменить расписание ее запуска или выяснить, почему аудит не был выполнен, и устранить причины.

Вы можете найти контроллеры доменов, для которых был выполнен аудит, но не был выполнен аудит по LDAP, выполнив запрос:

```
select(Activedirectory.forestTrusts, Activedirectory.domains.domainTrusts) |
calc(Activedirectory.forestTrusts + Activedirectory.domains.domainTrusts as domain) |
select (domain) | unique() |
join(select (@activedirectory, activedirectory.domains.name) as AD,
    AD.activedirectory.domains.name = domain) |
join(filter(host.hostroles.role = "domain controller") |
    select (@windowshost, windowshost.domain) as DC,
    DC.windowshost.domain = domain)
```

Результат запроса содержит список контроллеров доменов. Если значение параметра `AD.activedirectory.domains.name` пусто, то для контроллера домена был выполнен аудит, но не был выполнен аудит по LDAP. Для таких контроллеров необходимо создать (запустить) задачи (см. раздел 2.4) на аудит по LDAP. Чтобы в результате запроса отображались только такие контроллеры доменов, вы можете добавить в запрос фильтр `| filter(AD.activedirectory.domains.name = null)`.

Поиск доменов, информация о которых есть на конечных узлах

Запрос для поиска доменов, информация о которых есть на конечных узлах:

```
select(windowshost.domain) | filter(windowshost.domain) | unique()
```

Запрос для поиска доменов, информация о которых есть на конечных узлах, но которые не были найдены в результате аудита:

```
select (windowshost.domain) | filter(windowshost.domain) | unique() |
join(
    select(
      @activedirectory, activedirectory.domains.shortname as
      shortname, activedirectory.domains.name as name) as AD,
AD.name = windowshost.domain OR AD.shortname = windowshost.domain) |
filter(AD.@activedirectory = null)
```

Запрос для поиска узлов, на которых есть информация о доменах, не найденных в результате аудита:

```
select(windowshost.domain) | filter(windowshost.domain) | unique() |
join(
    select(@activedirectory, activedirectory.domains.shortname as
      shortname, activedirectory.domains.name as name) as AD,
AD.name = windowshost.domain OR AD.shortname = windowshost.domain) |
filter(AD.@activedirectory = null)
```

Результат запроса будет содержать данные об узлах с Windows вне доменов (то есть в домене `WORKGROUP`) и в доменах (кроме `WORKGROUP`).

Если найдены данные об узлах вне доменов (в домене `WORKGROUP`), то в зависимости от политики безопасности организации вы можете либо включить эти узлы в один из доменов и централизованно управлять политиками безопасности и обновлений, используя Domain Management Server, либо исключить данные об этих узлах из результатов запроса, заменив в запросе фильтр `filter(windowshost.domain)` на `filter(windowshost.domain and windowshost.domain != "WORKGROUP")`.

Если найдены данные об узлах в доменах (кроме `WORKGROUP`), то необходимо проверить, был ли выполнен аудит домена. Если аудит домена не был выполнен и контроллер домена не найден при выполнении приведенных ниже запросов, то чтобы его найти, необходимо расширить фильтры для сканирования с профилем HostDiscovery или выявить особенности IT-инфраструктуры, из-за которых запросы могут быть заблокированы.

Запрос для поиска контроллеров доменов из неизвестных доменов:

```
select(@windowshost, windowshost.domain,
    windowshost.hostroles.role, windowshost.@audittime) |
filter(windowshost.hostroles.role = "domain controller"
    and windowshost.domain) | unique() |
join(
    select(
      @activedirectory, activedirectory.domains.shortname as shortname,
activedirectory.domains.name as name) as AD,
    AD.name = windowshost.domain OR AD.shortname = windowshost.domain) |
filter(AD.@activedirectory = null)
```

Примечание. Это частный случай предыдущего запроса по поиску узлов, на которых есть информация о доменах, не найденных в результате аудита.

Результат запроса будет содержать в каждой строке данные о контроллере домена, для которого был выполнен аудит (значение параметра `windowshost.@audittime` не пусто), но для домена при этом не выполнялся аудит по LDAP. Для каждого такого домена необходимо создать (запустить) задачу (см. раздел 2.4) на аудит по LDAP, указав учетную запись, с которой была выполнена задача на аудит контроллера домена.

Оценка качества аудита доменов и контроллеров доменов

Для оценки качества аудита доменов и контроллеров доменов вы можете выполнить запрос:

```
select(
    @ActiveDirectory, ActiveDirectory.@updatetime,
    ActiveDirectory.domains.name,
    ActiveDirectory.domains.objectSid as objectSid,
    compactunique(ActiveDirectory.domains.Users.SamAccountName)
    as users) |
join(
    select(ActiveDirectory.domains.objectSid as objectSid,
      compactunique(ActiveDirectory.domains.Groups.SamAccountName) as groups) as G,
G.objectSid = objectSid) |
join(
    select(ActiveDirectory.domains.objectSid as objectSid,
      compactunique(
        ActiveDirectory.domains.Computers.SamAccountName) as computers) as C,
C.objectSid = objectSid) |
select(
    @ActiveDirectory, ActiveDirectory.@updatetime,
    ActiveDirectory.domains.name, objectSid,
    users, G.groups, C.computers)
```

Результат запроса будет содержать значения параметров:

- `updatetime` — время последнего аудита;
- `name` — FQDN;
- `objectSid` — SID домена;
- `users` — все пользователи в домене;
- `groups` — все группы в домене;
- `computers` — все компьютеры в домене.

Качество аудита обеспечено, если каждый из этих параметров содержит хотя бы одно значение и полученный список доменов совпадает с предоставленным для аудита списком доменов. Если в результате аудита какой-либо из доменов не обнаружен, то необходимо проверить, существует ли и запущена ли задача на аудит (см. раздел 2.4). Если задачи нет и домен не найден с помощью запросов, представленных в этом разделе, то необходимо создать (запустить) задачу на аудит (см. раздел 2.4).

В этом разделе

2.4.2.1. Поиск контроллеров доменов, не включенных в домены

Запрос для поиска контроллеров доменов по портам (HostDiscovery)

```
select(@host, host.@id, host.@audittime, Host.Endpoints<TransportEndpoint>.Port as
Port, Host.Endpoints<TransportEndpoint>.Protocol as Protocol,
Host.Endpoints<TransportEndpoint>.Status as Status) | limit (0) |
filter(Port in [53, 88, 135, 389, 445, 636, 3268, 3269] and Protocol = 'tcp' and
Status = 'Open') |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
calc(if Host.@audittime = null then "no audit" else if dur_audit > 30 then "old
audit" else "timely audit" as for_sorting) |
join(select(host.@id, host.hostroles.role as role) |
filter(role = "domain controller") as DC, DC.host.@id = host.@id) |
select(@host, dur_audit, DC.role, countunique(Port), compactunique(Port),
for_sorting) |
filter(countunique(Port) > 4) | sort(for_sorting ASC, dur_audit DESC, @host ASC)
```

Примечание. Сведения о портах, используемых контроллерами доменов, и примеры названий сервисов, представлены на сайте learn.microsoft.com.

Результат запроса будет содержать значение параметра `countunique(Port)` для каждого узла. Если значение параметра равно `8`, то этот узел — контроллер домена. Если значение параметра меньше `8`, то узел тоже может быть контроллером домена, у которого закрыты порты для сканирования. Поэтому в обоих случаях необходимо создать (запустить) задачи на аудит (см. раздел 2.4) узлов, данные о которых найдены в результате запроса.

Чтобы в результате запроса получить данные только об узлах, для которых не был выполнен аудит, вы можете добавить в конец последнего фильтра условие `and (not dur_audit or dur_audit > 30)`. Для таких узлов необходимо создать (запустить) задачи на аудит (см. раздел 2.4). Также вы можете просмотреть значение параметра `for_sorting`, полученное для каждого узла. Для узлов, у которых это значение равно `no audit` или `old audit`, необходимо создать (запустить) задачи на аудит (см. раздел 2.4), и в результате их корректного выполнения значение параметра `for_sorting` будет равно `timely audit`.

После аудита контроллеров доменов необходимо проверить, что выполняется аудит для доменов.

Запрос для поиска контроллеров доменов по сервисам за портами (Service Discovery)

```
select(@host, host.@audittime, host.@id,
Host.Endpoints<TransportEndpoint>.service.@type as type,
Host.Endpoints<TransportEndpoint>.port as port) |
filter(type in ["LdapService", "LdapSslService", "KerberosService"]) |
join(select(host.@id, host.hostroles.role as role) |
```

```
filter(role = "domain controller") as DC, DC.host.@id = host.@id) |
select(@host, host.@audittime, DC.role, compactunique(type), countunique(type),
compactunique(port)) |
filter(countunique(type) > 1)
```

Чтобы в результате запроса получить данные только об узлах, для которых не был выполнен аудит, вы можете добавить в конец последнего фильтра условие `and (not dur_audit or dur_audit > 30)`. Для таких узлов необходимо создать (запустить) задачи на аудит (см. раздел 2.4).

После аудита контроллеров доменов необходимо проверить, что выполняется аудит для доменов.

Запросы для поиска контроллеров доменов, для которых настроен аудит

Запрос 1:

```
select(@host, host.@audittime, host.hostroles.role) |
filter(host.hostroles.role = "Domain Controller")
```

Запрос 2:

```
select(@windowshost, windowshost.@audittime, windowshost.domainrole) |
filter(windowshost.domainrole like "%Domain Controller%")
```

Запрос 3:

```
select(@WindowsHost, WindowsHost.Roles.Name) |
filter(WindowsHost.Roles.Name = "Active Directory Domain Services") | unique()
```

Запрос 4:

```
select (@windowshost, windowshost.user.name) |
filter(windowshost.user.name = "krbtgt")
```

> **Примечание.** Локальная учетная запись krbtgt используется службой Windows Server Kerberos.

Вы можете использовать полученные результаты для группировки активов (см. раздел 2.3) и анализа инфраструктуры.

2.4.2.2. Поиск узлов, для которых не был выполнен аудит, по информации из домена

Запросы, приведенные ниже, позволяют проанализировать информацию, полученную в результате аудита по LDAP. Перед их выполнением необходимо выполнить аудит доменов и контроллеров доменов (см. раздел 2.4.2) и поиск контроллеров доменов, не включенных в домены (см. раздел 2.4.2.1).

Поиск контроллеров домена в закрытом сегменте сети

Поиск контроллеров домена по информации из домена позволяет проверить наличие контроллеров домена в закрытом сегменте сети.

Запрос для поиска контроллеров домена по информации из домена:

```
select(
    @ActiveDirectory, ActiveDirectory.Domains.Name as name,
    ActiveDirectory.Domains.Computers.DnsHostName,
    ActiveDirectory.Domains.Computers.DomainRole,
    ActiveDirectory.Domains.Computers.objectSid) |
filter(
    ActiveDirectory.Domains.Computers.DomainRole like '%Domain Controller%' OR dn
like "%OU=Domain Controllers,%") | sort(name)
```

Запрос для поиска контроллеров домена по информации из домена (через группы):

```
select(
    @ActiveDirectory, ActiveDirectory.Domains.objectSid,
    ActiveDirectory.Domains.Name as name,
    ActiveDirectory.Domains.groups.SamAccountName as SamAccountName,
    ActiveDirectory.Domains.groups.allmembers.objectSid as mem) |
filter(
    SamAccountName like "%Domain Controller%" and mem) |
join(
    select(
      ActiveDirectory.Domains.objectSid,
      ActiveDirectory.Domains.computers.objectSid as objectSid) as
      Comp, Comp.ActiveDirectory.Domains.objectSid =
ActiveDirectory.Domains.objectSid and Comp.objectSid = mem)
```

После выполнения запросов, приведенных выше, вы можете проверить выполнение аудита для существующих активов с помощью запроса:

```
select(
    ActiveDirectory.Domains.Name as name,
    ActiveDirectory.Domains.Computers.DnsHostName as DnsHostName,
    ActiveDirectory.Domains.Computers.DomainRole,
    ActiveDirectory.Domains.Computers.objectSid) |
filter(
    ActiveDirectory.Domains.Computers.DomainRole like '%Domain Controller%') |
join(
    select (@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = DnsHostName) |
sort(name) |
select(name, DnsHostName, H.@host, H.host.@audittime) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime)
```

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

> **Примечание.** Значение параметра `H.@host` может быть пустым из-за того, что по FQDN не удалось определить узел, указанный в домене.

Поиск серверов с Microsoft System Center Configuration Manager (SCCM)

Серверы с SCCM, как и контроллеры доменов, позволяют конфигурировать любые узлы домена и поэтому могут быть целью злоумышленника.

Запрос для поиска серверов с SCCM:

```
select(
    @ActiveDirectory, ActiveDirectory.domains.groups.objectsid as gr_sid,
    ActiveDirectory.domains.groups.SamAccountName as gr_name,
    ActiveDirectory.domains.groups.allmembers.SamAccountName as mem_name,
    ActiveDirectory.domains.groups.allmembers.ObjectSid as mem_id) |
filter(
    gr_name match "^(SMS( |_)|SCCM.*Servers)"
    and mem_name like "%$") | UNIQUE() |
join(
    select(
      ActiveDirectory.domains.Name as name,
      ActiveDirectory.domains.shortname,
      ActiveDirectory.domains.computers.objectSid as hostSID,
      ActiveDirectory.domains.computers.dnsHostName as dnsHostName
    ) as gr,
    gr.hostSID = mem_id) |
select (@ActiveDirectory, mem_name, gr_name, gr.dnsHostName) |
join(
    select (@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = gr.dnsHostName) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime) | group(@ActiveDirectory)
```

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значение параметра `gr.dnsHostName`. Если оно пусто, то учетная запись узла есть в группе `gr_name`, но отсутствует в домене. В этом случае необходимо удалить учетную запись узла из группы `gr_name`, иначе злоумышленник может узнать имя узла и создать виртуальную машину с таким именем.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

Поиск почтовых серверов

Сбор событий с почтовых серверов рекомендуется выполнять с помощью модулей Eventlog, Exchange Eventlog, Custom Event Collector, FileMonitor. Описание модулей представлено в Справочном руководстве MaxPatrol SIEM.

Запрос для поиска почтовых серверов:

```
select(
    @ActiveDirectory, ActiveDirectory.domains.groups.objectsid as gr_sid,
    ActiveDirectory.domains.groups.SamAccountName as gr_name,
    ActiveDirectory.domains.groups.allmembers.objectSid as mem_id,
    ActiveDirectory.domains.groups.allmembers.SamAccountName as mem_name) |
filter(gr_name like "Exchange Servers" and mem_name like "%$") |
UNIQUE() |
join(
    select(
      ActiveDirectory.domains.Name as name,
      ActiveDirectory.domains.shortname,
      ActiveDirectory.domains.computers.objectSid as hostSID,
      ActiveDirectory.domains.computers.dnsHostName as dnsHostName)
as gr, gr.hostSID = mem_id) |
select (@ActiveDirectory, mem_name, gr.name, gr.dnsHostName) |
join(
    select(@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = gr.dnsHostName) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime) | group(@ActiveDirectory)
```

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значение параметра `gr.dnsHostName`. Если оно пусто, то учетная запись узла есть в группе `gr_name`, но отсутствует в домене. В этом случае необходимо удалить учетную запись узла из группы `gr_name`, иначе злоумышленник может узнать имя узла и создать виртуальную машину с таким именем.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4)

и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

Поиск серверов, поддерживающих подключение по протоколу RDP или через RDG

Серверы, поддерживающие подключение по протоколу удаленного рабочего стола (Remote Desktop Protocol, RDP) или через шлюз удаленных рабочих столов (Remote Desktop Gateway, RDG), могут быть использованы злоумышленником для входа в инфраструктуру организации или в закрытые сегменты сети.

Запрос для поиска RDP- и RDG-серверов:

```
select(
    @ActiveDirectory, ActiveDirectory.domains.groups.objectsid as
    gr_sid, ActiveDirectory.domains.groups.SamAccountName as
    gr_name, ActiveDirectory.domains.groups.allmembers.objectSid as
    mem_id, ActiveDirectory.domains.groups.allmembers.SamAccountName
     as mem_name) |
filter(gr_sid = "S-1-5-32-561" and mem_name like "%$") | UNIQUE() |
join(
    select(
      ActiveDirectory.domains.Name as name,
      ActiveDirectory.domains.shortname,
      ActiveDirectory.domains.computers.objectSid as hostSID,
      ActiveDirectory.domains.computers.dnsHostName as dnsHostName
    ) as gr, gr.hostSID = mem_id) |
select(@ActiveDirectory, mem_name, gr.name, gr.dnsHostName) |
join(
    select (@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = gr.dnsHostName) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime) | group(@ActiveDirectory)
```

> **Примечание.** Описание значений параметра `gr_sid` (идентификаторов безопасности) представлено на сайте learn.microsoft.com.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значение параметра `gr.dnsHostName`. Если оно пусто, то учетная запись узла есть в группе `gr_name`, но отсутствует в домене. В этом случае необходимо удалить учетную запись узла из группы `gr_name`, иначе злоумышленник может узнать имя узла и создать виртуальную машину с таким именем.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4)

и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

Поиск серверов со службой сертификатов Active Directory (AD CS)

Серверы со службой сертификатов Active Directory (Active Directory Certificate Services, AD CS) обеспечивают управление цифровыми сертификатами, которые используются для аутентификации, шифрования и защиты данных.

Запрос для поиска серверов с AD CS:

```
select(
    @ActiveDirectory, ActiveDirectory.domains.objectSid as domain_sid,
    ActiveDirectory.domains.groups.objectsid as gr_sid,
    ActiveDirectory.domains.groups.SamAccountName as gr_name,
    ActiveDirectory.domains.groups.allmembers.objectSid as mem_id,
    ActiveDirectory.domains.groups.allmembers.SamAccountName as mem_name) |
calc(domain_sid+"-517" as sid_cert_pub) |
filter(gr_sid = sid_cert_pub and mem_name like "%$") | UNIQUE() |
join(
    select(
      ActiveDirectory.domains.Name as name,
      ActiveDirectory.domains.shortname,
      ActiveDirectory.domains.computers.objectSid as hostSID,
      ActiveDirectory.domains.computers.dnsHostName as dnsHostName) as gr, gr.hostSID
= mem_id) |
select(@ActiveDirectory, mem_name, gr.name, gr.dnsHostName) |
join(select(@host, host.fqdn, host.@audittime) as H,
H.host.fqdn = gr.dnsHostName) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit)|
sort(H.host.@audittime) | group(@ActiveDirectory)
```

> **Примечание.** Описание значений параметра `gr_sid` (идентификаторов безопасности) представлено на сайте learn.microsoft.com.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значение параметра `gr.dnsHostName`. Если оно пусто, то учетная запись узла есть в группе `gr_name`, но отсутствует в домене. В этом случае необходимо удалить учетную запись узла из группы `gr_name`, иначе злоумышленник может узнать имя узла и создать виртуальную машину с таким именем.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4)

и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

Поиск серверов доменов

Рекомендуется сначала выполнить поиск серверов, описанных выше, а затем поиск других серверов доменов с помощью запроса:

```
select(
    ActiveDirectory.Domains.Name as name,
    ActiveDirectory.Domains.Computers.DnsHostName as DnsHostName,
    ActiveDirectory.Domains.Computers.DomainRole) |
filter(
    ActiveDirectory.Domains.Computers.DomainRole like '%Domain Server%') |
join(
    select (@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = DnsHostName) | sort(name) |
select(name, DnsHostName, H.@host, H.host.@audittime) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime, H.@host)
```

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

Поиск узлов домена

После поиска серверов рекомендуется выполнить поиск остальных узлов домена с помощью запроса:

```
select(
    ActiveDirectory.Domains.Name as name,
    ActiveDirectory.Domains.Computers.DnsHostName as DnsHostName,
    ActiveDirectory.Domains.Computers.DomainRole) |
filter(DnsHostName) |
join(
    select(@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = DnsHostName) | sort(name) |
select(name, DnsHostName, H.@host, H.host.@audittime) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime, H.@host)
```

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

Поиск учетных записей администраторов домена и узлов с учетной записью администратора домена

Если на предыдущих этапах было обеспечено качество аудита, то в результате запроса, приведенного ниже, будут получены учетные записи администраторов домена, у которых нет активных сессий. В противном случае в результате запроса будут получены учетные записи администраторов без привязки к узлу: это значит, что для узлов не был выполнен аудит.

Запрос для поиска администраторов домена и их компьютеров:

```
select(
    @ActiveDirectory, ActiveDirectory.domains.Name as domain,
    ActiveDirectory.Domains.shortname as short,
    ActiveDirectory.Domains.Groups.SamAccountName as gr_name,
    ActiveDirectory.Domains.Groups.ObjectSid as group_sid,
    ActiveDirectory.Domains.Groups.AllMembers.objectType as mem_type,
    ActiveDirectory.Domains.Groups.AllMembers.SamAccountName as member_sam) |
filter(
    mem_type = "DomainUser"
    and (
      group_sid in [ "S-1-5-32-544", "S-1-5-32-548",
      "S-1-5-32-549", "S-1-5-32-551" ]
      or group_sid match "-51[289]$") and member_sam) |
calc(
    short + "\" + member_sam as user) |
    select(@ActiveDirectory, domain, member_sam, short, user) |
    unique() | join(select(@windowshost, windowshost.@audittime,
     windowshost.Processes.Owner as owner) | filter(owner) |
    unique() as PR, pr.owner = user) |
    select(@ActiveDirectory, domain, short, member_sam,
    PR.@windowshost, PR.windowshost.@audittime) |
    unique() | sort(PR.windowshost.@audittime, member_sam) |
    group(domain)
```

Примечание. Описание значений параметра `group_sid` (идентификаторов безопасности) представлено на сайте learn.microsoft.com.

Если в результате запроса найдены данные об узлах, для которых значение параметра `R.@windowshost` пусто, то необходимо выяснить, почему существуют учетные записи администраторов без активных сессий (например, учетная запись может быть резервной), а при необходимости — удалить их.

Если аудит узлов был выполнен, то вы можете найти узлы с учетной записью администратора домена с помощью запроса:

```
select(
    @windowshost, windowshost.netBiosName, windowshost.@id,
    windowshost.@audittime, windowshost.Processes.Owner as owner) |
filter(
    owner     and owner != "N/A"
    and owner not like "NT AUTHORITY\\%"
    and owner not like "Window Manager\\%"
    and owner not like "Font Driver Host\\%"
    and owner not like "NT SERVICE\\%") |
calc(windowshost.netBiosName + "\\%" as netb) |
filter(owner not like netb) | unique() |
calc(lower(owner) as l_owner) |
join(
    select (
      @ActiveDirectory, ActiveDirectory.domains.Name as domain,
      ActiveDirectory.Domains.shortname as short,
      ActiveDirectory.domains.users.ObjectSid,
      ActiveDirectory.domains.users.SamAccountName as acc_name,
      ActiveDirectory.domains.users.allparents.objectSid as group_sid) |
filter(
    group_sid in [ "S-1-5-32-544", "S-1-5-32-548", "S-1-5-32-549",
     "S-1-5-32-551" ] or group_sid match "-51[289]$") |
calc(
    lower(short + "\" + acc_name) as long_domain_uname) as D,
    D.long_domain_uname = owner) | filter(D.@ActiveDirectory) |
    select(@windowshost, owner, D.@ActiveDirectory) | unique()
```

Примечание. Описание значений параметра `group_sid` (идентификаторов безопасности) представлено на сайте learn.microsoft.com.

Поиск узлов, для которых выполнен аудит, из доверенных доменов

Если между доменами установлены отношения доверия (trusts), то вы можете найти узлы из доверенных доменов с помощью запроса:

```
select(
    @ActiveDirectory, ActiveDirectory.domains.Name as domain,
    ActiveDirectory.domains.foreignSecurityPrincipals.objectSid
    as sid, ActiveDirectory.domains.foreignSecurityPrincipals.samAccountName
    as name, ActiveDirectory.domains.foreignSecurityPrincipals.foreignDomainName as
foreignDomain) |
filter(name like "%$") | UNIQUE() |
join(
    select(
      ActiveDirectory.domains.Name as name,
      ActiveDirectory.domains.shortname,
      ActiveDirectory.domains.computers.objectSid as hostSID,
      ActiveDirectory.domains.computers.dnsHostName as dnsHostName)
    as fD, fD.ActiveDirectory.domains.shortname = foreignDomain
    and fD.hostSID = sid) |
select(@ActiveDirectory, domain, name, foreignDomain,
    fD.name, fD.dnsHostName) |
join(select (@host, host.fqdn, host.@audittime) as H,
    H.host.fqdn = fD.dnsHostName) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(H.host.@audittime) | group(@ActiveDirectory)
```

Если полученное в результате запроса значение параметра `fD.name` пусто, то узел имеет доступ к ресурсам домена, но не обнаружен в доверенном домене или для доверенного домена не был выполнен аудит. Необходимо проверить, существует ли доверенный домен и был ли выполнен его аудит. Если домен не существует, необходимо удалить его из зоны доверия. Если домен существует, необходимо выполнить его аудит (см. раздел 2.4.2). Если аудит домена выполнен, но узел не обнаружен, то необходимо удалить его из зоны доверия.

Для всех узлов, данные о которых найдены в результате запроса, необходимо проверить значения параметров `H.@host`, `H.host.@audittime`, `dur_audit`. Если значения параметров `H.@host` или `H.host.@audittime` пусты или аудит проводился больше месяца назад (`dur_audit > 30`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание. При создании (запуске) задачи на аудит узла с пустым значением параметра `H.@host` необходимо установить флажок **Выполнить обнаружение узлов до начала сбора данных**.

2.4.3. Аудит систем виртуализации

Перед аудитом систем виртуализации рекомендуется выполнить сканирование (аудит), как описано в предыдущих разделах. Необходимо сначала выполнить аудит гипервизоров, а затем выполнить поиск активов отдельно для гипервизоров Hyper-V (см. раздел 2.4.3.1) и VMware vSenter Server (см. раздел 2.4.3.2).

Запрос данных о гипервизорах:

```
select(@host, host.Hypervisors.VirtType) |
filter(host.Hypervisors.VirtType) |
group(host.Hypervisors.VirtType, COUNT(*))
```

Запрос данных о виртуальных машинах:

```
filter(host.IsVirtual = true) | select(@host, host.vmid)
```

Значение параметра `host.vmid` может быть пустым, если не отсканирован управляющий сервер виртуальных машин или гипервизор. И наоборот, если значение параметра `host.vmid` заполнено, то это значит, что был отсканирован управляющий сервер или гипервизор и можно определить, какому гипервизору принадлежит виртуальная машина.

Запрос данных о виртуальных машинах, для которых не определена их принадлежность гипервизору:

```
filter(host.IsVirtual = true) | select(@host, host.vmid) | filter(host.vmid =
null)
```

Запрос данных об отсканированных виртуальных машинах, принадлежащих гипервизорам:

```
select(
    host.@name, host.hypervisors.VMs.id as vm_id,
    host.hypervisors.VMs.name as vm_name,
    host.hypervisors.VMs.fqdn as vm_fqdn,
    host.hypervisors.VMs.hostname as vm_hostname,
    compactunique(host.hypervisors.VMs.NetworkInterfaces.ipaddress.address) as
vm_ips,
    host.hypervisors.VMs.state as state) |
filter(state in ["poweredOn", "Running"]) |
join(select(@host, host.@audittime, host.vmid) |
filter(host.vmid) as H, H.host.vmid = vm_id) |
select(host.@name, vm_name, vm_fqdn, vm_hostname,
    vm_ips, H.@host, H.host.@audittime) |
```

```
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit)
```

После аудита гипервизоров все включенные виртуальные машины будут созданы.

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для проверки актуальности аудита гипервизоров:

```
select(@host, host.hypervisors, host.@audittime) |
filter(host.hypervisors) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
sort(host.@audittime)
```

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

В этом разделе

2.4.3.1. Аудит узлов с Hyper-V

Запрос данных о гипервизорах, для которых проводилось сканирование портов (с профилем HostDiscovery):

```
filter(Host.Endpoints<TransportEndpoint>[Port = 2179 and Protocol = 'tcp' and
Status = 'Open'])
| select(@host, compactunique(host.hypervisors),
compactunique(host.softs<HyperV>), host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)
```

Для узлов, у которых значение параметра `host.@audittime` пусто или аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос данных о гипервизорах, для которых проводилось сканирование методом черного ящика (с профилем Service Discovery):

```
filter(Host.Softs<HyperV>)
| select(@host, compactunique(host.hypervisors),
compactunique(host.softs<HyperV>), host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)
```

Для узлов, у которых значение параметра `host.@audittime` пусто или аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для поиска узлов с Hyper-V, для которых настроен аудит:

```
filter(Host.Hypervisors.VirtType = 'Hyper-V')

| select(@host, host.@audittime) | calc(TotalDays(now() - host.@AuditTime) as
dur_audit)
```

Для группировки активов (см. раздел 2.3) и анализа инфраструктуры вы можете использовать результаты этого запроса, а также запроса данных об отсканированных виртуальных машинах, принадлежащих гипервизорам (см. раздел 2.4.3), и запроса для проверки актуальности аудита гипервизоров (см. раздел 2.4.3). Чтобы в результатах этих запросов отображались данные об узлах только с Hyper-V, необходимо добавить в начало запросов условие `filter(Host.Hypervisors.VirtType = 'Hyper-V') |`.

2.4.3.2. Аудит узлов с VMware vCenter Server

Получив данные о VMware vCenter Server (см. раздел 2.4.3), вы можете найти гипервизоры и виртуальные машины.

Запросы данных об узлах с VMware vCenter Server, для которых не проводился аудит

Запросы, приведенные ниже, основаны на данных, полученных о VMware vCenter Server по результатам сканирования портов (см. раздел 2.1) и сканирования методом черного ящика (см. раздел 2.2).

Запрос для поиска серверов VMware vCenter по портам:

```
select(@host, host.endpoints<transportendpoint>.port as port,
host.endpoints<transportendpoint>.status as status, host.osName,
compactunique(Host.Softs<VCenterInstance>.datacenters) as datacenters,
host.@audittime) |
filter(status = "Open" and port in [22, 80, 389, 443, 902, 2014, 5480, 7090, 8010,
8084, 9084, 9087, 9097, 15007, 15008, 16666, 16667]) |
select(@host, compactunique(port), countunique(port) as p_cnt, host.osName,
datacenters, host.@audittime) |
filter(p_cnt > 5) | unique() | calc(TotalDays(now() - host.@AuditTime) as dur_audit)
| sort(p_cnt DESC)
```

Список портов для сканирования приведен на сайте ports.esp.vmware.com. Чтобы отфильтровать список по серверам VMware vCenter, для параметра `Destination` необходимо указать значение `vcenter`, для параметра `Purpose` — значение `base`.

Запрос для поиска серверов VMware vCenter по установленному ПО:

```
filter(Host.Softs<VCenterInstance>) |
select(@host, host.@audittime, host.osName,
compactunique(Host.Softs<VCenterInstance>.datacenters) as datacenters) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit)
```

После выполнения запросов, приведенных выше, если аудит узлов не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) с профилями Unix Audit и vSphere Audit и настроить их расписание.

> **Примечание.** Рекомендуется выполнить обе эти задачи аудита независимо от значения параметра `host.@audittime`.

Для узлов, у которых по результатам запроса значение параметра `host.osName` пусто, необходимо создать (запустить) задачи на аудит (см. раздел 2.4) с профилем Unix Audit и настроить их расписание.

Для узлов, у которых по результатам запроса значение параметра `datacenters` пусто, необходимо создать (запустить) задачи на аудит (см. раздел 2.4) с профилем vSphere Audit и настроить их расписание.

Запрос для поиска гипервизоров, для которых нет актива с VMware vCenter Server:

```
filter(esxihost) |
select(@EsxiHost, EsxiHost.hostname, EsxiHost.@audittime, EsxiHost.HostMobId as
MobId, esxihost.vCenterUUID as vC_uuid) |
calc(TotalDays(now() - EsxiHost.@AuditTime) as dur_audit) |
filter(vC_uuid != null) | unique() |
join(filter(host.softs<VCenterInstance>.vCenterUUID) |
select(@host, host.hostname, host.FQDN, host.softs<VCenterInstance>.vCenterUUID as
v_uuid) as vC, vC.v_uuid = vC_uuid) |
select(EsxiHost.hostname, dur_audit, MobId, vC_uuid, vC.@host, vC.host.hostname) |
filter(vC.host.hostname = null) | sort(vC_uuid, MobId)
```

Для всех VMware vCenter Server, к которым относятся полученные в ответе на запрос гипервизоры, необходимо создать (запустить) задачи на аудит (см. раздел 2.4) с профилями Unix Audit и vSphere Audit и настроить их расписание.

Запросы данных об узлах с VMware vCenter Server, для которых проводился аудит

Вы можете проверить актуальность аудита доменов (см. раздел 2.4.2), указанных для аутентификации на сервере VMware vCenter, выполнив запрос:

```
select(host.softs<VCenterInstance>.domains) |
filter(host.softs<VCenterInstance>.domains and host.softs<VCenterInstance>.domains !=
"VSPHERE.LOCAL") | unique() |
calc(lower(host.softs<VCenterInstance>.domains) as domain) |
join(select(@activedirectory, activedirectory.domains.name,
activedirectory.domains.ShortName, ActiveDirectory.@updatetime) |
```

```
calc(lower(activedirectory.domains.name) as name) |
calc(lower(activedirectory.domains.ShortName) as ShortName) as AD, AD.name = domain
or AD.ShortName = domain) |
select(domain, AD.@activedirectory, AD.ActiveDirectory.@updatetime) |
calc(TotalDays(now() - AD.ActiveDirectory.@updatetime) as dur_audit)
```

Вы можете сравнить полученные данные с данными о доменах, полученными в результате аудита.

Если значение параметра `AD.@activedirectory` пусто, то необходимо найти домены, указанные в `domain`, и выполнить аудит (см. раздел 2.4.2).

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос данных о пользователях VMware vCenter Server:

```
select(@host, Host.@id, Host.Softs<VCenterInstance>,
Host.Softs<VCenterInstance>.users.domain as u_domain,
Host.Softs<VCenterInstance>.users.name as u_name,
Host.Softs<VCenterInstance>.users.IsGroup as IsGroup,
Host.Softs<VCenterInstance>.users.permissions.DCID as DCID) |
filter(Host.Softs<VCenterInstance>) |
join(select(Host.@id, Host.Softs<VCenterInstance>.datacenters.mobid as mobid,
Host.Softs<VCenterInstance>.datacenters.name as name) |
filter(mobid) as D, D.mobid = DCID) |
select(@host, u_domain, u_name, IsGroup, compactunique(D.name), compactunique(DCID))
| sort(@host, u_domain, IsGroup, u_name)
```

Запрос данных об узлах с VMware vSphere Hypervisor (ESXi), для которых не проводился аудит

Запрос, приведенный ниже, основан на данных, полученных о VMware ESXi по результатам сканирования портов (см. раздел 2.1) и сканирования методом черного ящика (см. раздел 2.2).

Запрос для поиска узлов с VMware ESXi по портам:

```
select(@host, host.endpoints<transportendpoint>.port as port,
host.endpoints<transportendpoint>.status as status, host.osName, host.@type,
compactunique(host.hypervisors), host.@audittime) |

filter(status = "Open" and port in
[22, 80, 443, 902, 1564, 1565, 2233, 5001, 5201,
8000, 8080, 9000, 9080, 9095, 9096, 12321, 12345, 12443, 23451]) |

select(@host, compactunique(port), countunique(port) as p_cnt, host.osName,
host.@type, compactunique(host.hypervisors), host.@audittime) |

filter(p_cnt > 5) | unique() | calc(TotalDays(now() - host.@AuditTime) as
dur_audit) | sort(p_cnt DESC)
```

Список портов для сканирования приведен на сайте ports.esp.vmware.com. Чтобы отфильтровать список по узлам с VMware ESXi, для параметра `Destination` необходимо указать значение `esxi`.

> **Примечание.** Поиск портам не всегда позволяет обнаружить все узлы с VMware ESXi, так как в большинстве инфраструктур эти порты закрыты для доступа извне.

Если аудит узлов не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи (см. раздел 2.4) на аудит с профилями Unix Audit и vSphere Audit и настроить их расписание.

Запросы данных об узлах с VMware vSphere Hypervisor (ESXi), для которых проводился аудит

Запрос для поиска узлов с VMware ESXi, для которых проводился аудит:

```
select(@esxihost, esxihost.@audittime) |
calc(TotalDays(now() - esxihost.@AuditTime) as dur_audit) |
sort(esxihost.@AuditTime)
```

Если аудит узлов не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи (см. раздел 2.4) на аудит с профилями Unix Audit и vSphere Audit и настроить их расписание.

Запрос для поиска отсканированных серверов VMware vCenter с известными VMware ESXi, которые относятся к кластеру:

```
filter(host.softs<VCenterInstance>.vCenterUUID) |

select(@host, host.@name, host.softs<VCenterInstance>.vCenterUUID as v_uuid,
host.softs<VCenterInstance>.Datacenters.Name as dc_name,
host.softs<VCenterInstance>.Datacenters.Clusters.name as CL_name,
host.softs<VCenterInstance>.Datacenters.Clusters.ESXiHosts as esxi) |

select(host.@name, v_uuid, dc_name, CL_name, esxi) |

filter(esxi) |

join(filter(EsxiHost.HostMobId != null) |

select(@EsxiHost, EsxiHost.@audittime, EsxiHost.HostMobId as MobId,
EsxiHost.vCenterUUID as vC_uuid) as ESXi, esxi = ESXi.MobId and ESXi.vC_uuid =
v_uuid) |

calc(TotalDays(now() - ESXi.EsxiHost.@AuditTime) as dur_audit) |

select(host.@name, dc_name, CL_name, v_uuid, esxi, ESXi.@EsxiHost, dur_audit) |
sort(host.@name, dc_name, CL_name, esxi)
```

Если аудит узлов не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи (см. раздел 2.4) на аудит с профилями Unix Audit и vSphere Audit и настроить их расписание.

Запрос для поиска отсканированных серверов VMware vCenter с известными VMware ESXi, которые относятся к дата-центру:

```
filter(host.softs<VCenterInstance>.vCenterUUID) |

select(@host, host.@name, host.softs<VCenterInstance>.vCenterUUID as v_uuid,
host.softs<VCenterInstance>.Datacenters.Name as dc_name,
host.softs<VCenterInstance>.Datacenters.ESXiHosts as esxi) |

select(host.@name, v_uuid, dc_name, esxi) |

filter(esxi) |

join(filter(EsxiHost.HostMobId != null) |

select(@EsxiHost, EsxiHost.@audittime, EsxiHost.HostMobId as MobId,
EsxiHost.vCenterUUID as vC_uuid) as ESXi, esxi = ESXi.MobId and ESXi.vC_uuid =
v_uuid) |

calc(TotalDays(now() - ESXi.EsxiHost.@AuditTime) as dur_audit) |

select(host.@name, dc_name, v_uuid, esxi, ESXi.@EsxiHost, dur_audit) |
sort(host.@name, dc_name, esxi)
```

Если аудит узлов не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи (см. раздел 2.4) на аудит с профилями Unix Audit и vSphere Audit и настроить их расписание.

Запрос для поиска узлов с VMware ESXi, о которых нет информации на активах с VMware vCenter Server:

```
filter(esxihost.ipaddress)

| select(@esxihost, esxihost.hostMobId as hostMobId, esxihost.vCenterUUID as
e_vUUID, esxihost.@audittime)

| join(filter(host.softs<VCenterInstance>.vCenterUUID) |

select(@host, host.@name, host.softs<VCenterInstance>.vCenterUUID as v_uuid,
host.softs<VCenterInstance>.Datacenters.Name as dc_name,
host.softs<VCenterInstance>.Datacenters.ESXiHosts as esxi) |

select(host.@name, v_uuid, dc_name, esxi) as VCD, VCD.esxi = hostMobId and
e_vUUID = VCD.v_uuid)

| filter(VCD.dc_name = null)

| join(filter(host.softs<VCenterInstance>.vCenterUUID) |

select(@host, host.@name, host.softs<VCenterInstance>.vCenterUUID as v_uuid,
host.softs<VCenterInstance>.Datacenters.Name as dc_name,
host.softs<VCenterInstance>.Datacenters.Clusters.name as CL_name,
host.softs<VCenterInstance>.Datacenters.Clusters.ESXiHosts as esxi) as VCC,
VCC.esxi = hostMobId and e_vUUID = VCC.v_uuid)

| filter(VCC.dc_name = null)
```

```
| select(@esxihost, hostMobId, e_vUUID, esxihost.@audittime) |
calc(TotalDays(now() - EsxiHost.@AuditTime) as dur_audit)
```

Если для узлов с VMware ESXi значение параметра `hostMobId` заполнено, то на них есть информация об их принадлежности VMware vCenter Server, но на активах с VMware vCenter Server информации об этих узлах нет. В этом случае рекомендуется проверить результаты последнего аудита.

Если аудит узлов не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи (см. раздел 2.4) на аудит с профилями Unix Audit и vSphere Audit и настроить их расписание.

Запросы данных о виртуальных машинах

Запрос данных о виртуальных машинах на VMware vCenter Server:

```
filter(host.softs<VCenterInstance>.vCenterUUID) |

select(@host, host.@Name, host.softs<VCenterInstance>.vCenterUUID as v_uuid,
host.softs<VCenterInstance>.datacenters.name as dc_name,
host.softs<VCenterInstance>.datacenters.VMs.id as vm_id,
host.softs<VCenterInstance>.datacenters.VMs.fqdn as vm_fqdn,
host.softs<VCenterInstance>.datacenters.VMs.hostname as vm_hostname,
host.softs<VCenterInstance>.datacenters.VMs.NetworkInterfaces.MacAddress as
vm_macs,
host.softs<VCenterInstance>.datacenters.VMs.NetworkInterfaces.ipaddress.address
as vm_ips, host.softs<VCenterInstance>.datacenters.VMs.state as state) |

filter(state = "poweredOn") |

join(select(@host, host.@audittime, host.fqdn, host.hostname, host.vmid,
host.@macaddresses, host.@ipaddresses) as H, H.host.vmid = vm_id) |

select(host.@name, dc_name, vm_fqdn, vm_hostname, compactunique(vm_ips),
compactunique(vm_macs), vm_id, H.@host, H.host.@audittime) |

calc( TotalDays(now() - H.host.@AuditTime) as dur_audit)
```

Примечание. Запрос может выполняться достаточно долго.

Если значение параметра `@host` пусто (то есть виртуальная машина и гипервизор, управляющий ею, не сканируются, но виртуальная машина включена), то необходимо выполнить поиск гипервизоров и виртуальных машин.

Если значение параметра `host.vmid` пусто, то необходимо выполнить аудит всех гипервизоров — тогда оно будет заполнено для всех виртуальных машин.

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос данных о виртуальных машинах на гипервизорах:

```
select(esxihost.@name, esxihost.hypervisors.VMs.id as vm_id,
esxihost.hypervisors as hyper, esxihost.hypervisors.VMs.name as vm_name,
esxihost.hypervisors.VMs.fqdn as vm_fqdn, esxihost.hypervisors.VMs.hostname as
vm_hostname,
compactunique(esxihost.hypervisors.VMs.NetworkInterfaces.ipaddress.address) as
vm_ips, esxihost.hypervisors.VMs.state as state) |

filter(state = "poweredOn") |

join(select(@host, host.@audittime, host.vmid) |

filter(host.vmid) as H, H.host.vmid = vm_id) |

select(esxihost.@name, hyper, vm_name, vm_fqdn, vm_hostname, vm_ips, H.@host,
H.host.@audittime) |

calc(TotalDays(now() - H.host.@AuditTime) as dur_audit)
```

После аудита гипервизоров все включенные виртуальные машины будут созданы.

Для узлов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

2.4.4. Аудит систем CI/CD

Серверы CI/CD используются для автоматизации процессов сборки, тестирования и развертывания приложений. Как правило, серверы CI/CD предоставляют веб-интерфейсы для работы с их функциями. В большинстве случаев серверы CI/CD доступны из внутренней сети по порту 443 (TCP), реже — по порту 80 (TCP). Эти признаки используются для поиска серверов CI/CD в запросах, приведенных ниже.

> **Примечание.** По этим признакам могут быть выявлены как серверы CI/CD, так и другие узлы, например балансировщики HTTP-трафика.
>
> **Внимание!** Перед аудитом серверов CI/CD необходимо выполнить сканирование портов для обнаружения активов (см. раздел 2.1) с открытыми портами 443 и 80.

Активы с открытыми портами 443 и 80

Запрос для поиска активов, которые предоставляют веб-интерфейсы и доступны по портам 443 и 80 (по результатам сканирования портов):

```
filter(host.endpoints<transportendpoint>[port in [443, 80] and status
= 'Open']) |

select(@host, host.@PentestTime, host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit) | sort(host.@audittime)
```

Для найденных активов необходимо проверить, выполнялся ли аудит или как минимум сканирование методом черного ящика, с помощью запроса или условия фильтрации, приведенных ниже.

> **Примечание.** Выполнение аудита (см. раздел 2.4) — это выполнение задач с определенным профилем из модуля Audit. Выполнение сканирования методом черного ящика (см. раздел 2.2) — это выполнение задач с определенным профилем из модуля Pentest (в частности, с профилем Service Discovery). Профили описаны в разделах «Модуль Audit» и «Модуль Pentest» Руководства по настройке источников MaxPatrol VM.

Запрос для поиска активов, для которых аудит и сканирование методом черного ящика не выполнялись или выполнялись больше двух недель назад:

```
filter(host.endpoints<transportendpoint>[port in [443, 80] and status
= 'Open']) |

select(@host, host.@PentestTime, host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)  | calc(TotalDays(now() - host.@PentestTime) as
dur_pentest) |

filter((dur_audit = null OR dur_audit > 14) and (dur_pentest = null OR
dur_pentest > 14)) | sort(host.@audittime)
```

Условие фильтрации активов, для которых аудит и сканирование методом черного ящика не выполнялись или данные о которых устарели:

```
host.endpoints<transportendpoint>[port in [443, 80] and status = 'Open'] and
host.@ScanningInfo[Type = 'Audit' and Status != 'UpToDate'] and
host.@ScanningInfo[Type = 'Pentest' and Status != 'UpToDate']
```

> **Примечание.** Для активов вы можете указывать время, по истечении которого данные о них устаревают. Подробнее в разделе «Устаревание активов» Руководства оператора MaxPatrol VM.

Если в результате запроса или по условию фильтрации найдены активы, то необходимо объединить их в динамическую группу (см. раздел 2.3) и запустить для группы задачу на сканирование с профилем Service Discovery (см. раздел 2.2) (в параметрах задачи необходимо указать порты 443 и 80). После выполнения задачи существующие активы распределятся по другим группам. Если после выполнения задачи какие-либо активы остались в группе, то необходимо убедиться, что они не существуют, и удалить их.

Объем трафика через порты 443 и 80

Если в инфраструктуре установлен PT NAD, вы можете использовать его для мониторинга трафика, выполнив следующие шаги:

1. Применить условие фильтрации `dst.port in [443, 80] && dst.groups == HOME_NET`.

   > **Примечание.** Также для просмотра списка узлов вы можете сгруппировать их на вкладке **Узлы** по типу **Сервер**, группе **HOME_NET** и протоколу **HTTP.**

2. На странице **Дашборды**, в виджете **Серверы по сессиям и трафику** отсортировать данные по значениям столбца **Отправлено**.

3. Экспортировать данные виджета, нажав  .

Получив информацию об объеме трафика, необходимо:

1. Создать (запустить) задачи на сканирование с профилем Service Discovery (см. раздел 2.2) и на аудит (см. раздел 2.4) для всех узлов, объем входящего трафика которых через порты 443 и 80 составляет более 100 МБ в сутки.
2. Создать (запустить) задачи на сканирование с профилем Service Discovery (см. раздел 2.2) и на аудит (см. раздел 2.4) для всех узлов, использующих HTTP-трафик. Необходимый минимум — задачи на сканирование с профилем Service Discovery с указанием портов 443 и 80. После выполнения задач вы можете проанализировать флаги пакетов данных (флаги на портах) и определить установленные сервисы или приложения, которые связаны с этими портами.
3. Разместить ключевые серверы за межсетевыми экранами для защиты приложений (application firewall).

Названия активов, связанные с CI/CD

После сканирования активов с профилем Service Discovery (см. раздел 2.2) (с указанием портов 443 и 80) вы можете просмотреть названия отсканированных серверов, выполнив запросы, представленные ниже.

Запрос 1:

```
select(@host, host.endpoints<transportendpoint>.port as port,
host.endpoints<transportendpoint>.status as status,
host.endpoints<transportendpoint>.service.checks<GetHTTPHeaders>.Headers.Header
as Header) |

filter(port in [443, 80] and status = 'Open' and Header != null) |
group(Header, COUNT(*)) | SORT("COUNT(*)" DESC)
```

Запрос 2:

```
select(@host, host.endpoints<transportendpoint>.port as port,
host.endpoints<transportendpoint>.status as status,
host.endpoints<transportendpoint>.Service.Checks.Arguments<GetHTTPHeadersArgume
nt>.Headers.Header as Header) |

filter(port in [443, 80] and status = 'Open' and Header != null) |
group(Header, COUNT(*)) | SORT("COUNT(*)" DESC)
```

Если среди найденных названий есть связанные с CI/CD, то для таких активов необходимо создать (запустить) задачи на аудит (см. раздел 2.4). Например, могут быть найдены названия `X-GITLAB-META`, `X-ARTIFACTORY-ID`, `TEAMCITY-NODE-ID` для активов с GitLab, JFrog Artifactory, JetBrains TeamCity соответственно.

Данные из пакетов экспертизы MaxPatrol SIEM

Если в инфраструктуре установлен MaxPatrol SIEM и используются пакеты экспертизы «Аномалии в цепочках поставок», «Атаки методом перебора» и «Профилирование доступа к критически важным системам», то вы можете выборочно просмотреть их табличные списки (tabular_lists). Некоторые табличные списки заполняются автоматически, например после аудита, после сбора событий. Названия табличных списков и правил, связанных с цепочками поставок, начинаются с `SupplyChain_`. Иногда в названиях табличных списков или правил встречаются названия ПО, на события которого ориентированы правила.

Чтобы просмотреть табличные списки в MaxPatrol SIEM, вы можете перейти на страницу **Сбор данных** → **Табличные списки**. Для табличных списков отображаются вкладки **Правила обогащения** и (или) **Правила корреляции** — в зависимости от наличия правил, посредством которых заполняются табличные списки или которые получают данные из них.

> **Примечание.** Описание табличных списков представлено в разделах «Страница **Табличные списки**» и «Работа с табличными списками» Руководства оператора MaxPatrol SIEM.

Чтобы просмотреть табличные списки в веб-интерфейсе **Knowledge Base** MaxPatrol SIEM, вы можете на странице **Табличные списки** перейти по ссылке **Открыть в Knowledge Base** и в иерархическом списке **Папки** открыть следующие папки:

— Аномалии в цепочках поставок → tabular_lists;

— Атаки методом перебора → tabular_lists;

— Профилирование доступа к критически важным системам → tabular_lists.

> **Примечание.** Описание пакетов экспертизы представлено в разделах «Страница **Пакеты экспертизы**» и «Интерфейс **Knowledge Base**» Руководства оператора MaxPatrol SIEM.

В папках в веб-интерфейсе **Knowledge Base** есть табличные списки и относящиеся к ним правила, названия которых включают `SupplyChain_` и, например, `gitlab`, `artifactory`, `teamcity`, `merge`, `branches`, `release_build`. Вы можете просмотреть их и проверить заполнение табличных списков, применимых в вашей инфраструктуре.

В этом разделе

2.4.4.1. Аудит GitLab

После сканирования активов с профилем Service Discovery (см. раздел 2.2) (с указанием портов 443 и 80) вы можете выполнить запрос для поиска активов с названием `gitlab`:

```
filter(host.endpoints<transportendpoint>[port in [443, 80] and status
= 'Open' and Service.Checks.Arguments<GetHTTPHeadersArgument>.Headers.Header
= "X-GITLAB-META"]) |
```

```
select(@host, host.@PentestTime, host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)  | calc(TotalDays(now() - host.@PentestTime) as
dur_pentest)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Для аудита GitLab необходимо выполнить задачи с профилем Unix Audit. После этого вы можете использовать запросы, представленные ниже.

Запрос для поиска активов с GitLab:

```
filter(host.softs<gitlab>) |

select(@host, host.@PentestTime, host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)  | calc(TotalDays(now() - host.@PentestTime) as
dur_pentest)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

С помощью следующих запросов вы можете просмотреть информацию, полученную в результате аудита ПО.

IP-адреса и порты

Запрос для поиска IP-адресов и портов, по которым доступен GitLab:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>.openports.Protocol,
host.softs<gitlab>.openports.address, host.softs<gitlab>.openports.Port,
compactunique(host.interfaces.l3settings.address.address.address) as addresses)
| filter(host.softs<gitlab>.openports.Protocol)
```

Если в качестве адреса указано значение `0.0.0.0`, то GitLab доступен со всех IP-адресов актива (они перечислены в поле `addresses`).

Репозитории проектов

Запрос для поиска репозиториев проектов:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>, host.softs<gitlab>.projects.ID as ID,
host.softs<gitlab>.projects.Name as Name, host.softs<gitlab>.projects.FullName
as FullName, host.softs<gitlab>.projects.FullPath as FullPath,
host.softs<gitlab>.projects.Visibility as Visibility,
compactunique(host.softs<gitlab>.projects.DirectMembers) as DirectMembers,
compactunique(host.softs<gitlab>.projects.AllMembers) as AllMembers,
```

```
host.softs<gitlab>.projects.GitLabCI as GitLabCI,
compactunique(host.softs<gitlab>.projects.AvailableRunners) as
AvailableRunners) | sort(ID)
```

По результатам запроса вы можете просмотреть проекты GitLab, а также права доступа пользователей к репозиториям проектов, чтобы проверить отсутствие избыточно выданных прав.

Пользователи и группы

Запрос для поиска данных о пользователях:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>, host.softs<gitlab>.users.LogonName as
LogonName, host.softs<gitlab>.users.IsAdmin as IsAdmin,
host.softs<gitlab>.users.State as State, host.softs<gitlab>.users.Type as Type,
host.softs<gitlab>.users.RequireOTP as RequireOTP,
compactunique(host.softs<gitlab>.users.Identities.Provider) as Providers,
compactunique(host.softs<gitlab>.users.Identities.Identity) as Identities) |
sort(LogonName)
```

По результатам запроса вы можете просмотреть список пользователей, которые могут подключиться к серверу GitLab (если значение параметра `State` равно `Active`) или которые не могут подключиться к серверу GitLab (если значение параметра `State` не равно `Active`). Это позволит, например, найти отключенных пользователей, которые не должны быть отключены, или сравнить полученный список пользователей, которые могут подключиться к серверу GitLab, со списком сотрудников, которым должен быть предоставлен доступ для подключения.

Запрос для поиска данных о группах и пользователях, входящих в группы:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>, host.softs<gitlab>.groups.ID as ID,
host.softs<gitlab>.groups.Name as Name, host.softs<gitlab>.groups.FullName as
FullName, host.softs<gitlab>.groups.FullPath as FullPath,
host.softs<gitlab>.groups.ParentID as ParentID,
host.softs<gitlab>.groups.Visibility as Visibility,
compactunique(host.softs<gitlab>.groups.DirectMembers) as DirectMembers,
compactunique(host.softs<gitlab>.groups.AllMembers) as AllMembers) | sort(ID)
```

По результатам запроса вы можете просмотреть список групп и состав пользователей в группах. Это позволит, например, найти пользователей с правами, избыточно выданными через группы. Также вы можете узнать, в какое количество групп входит каждый пользователь, и найти пользователей, которые входят в большое количество групп, выполнив запрос:

```
filter(host.softs<gitlab>) |
```

```
select(host.softs<gitlab>.groups.AllMembers.Name as member,
host.softs<gitlab>.groups.Name as Name) | filter(member) | unique() |
group(member, count(*)) | sort(count(*) DESC)
```

Агенты (runners)

Запрос для поиска данных об агентах (runners):

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>, host.softs<gitlab>.runners.ID as ID,
host.softs<gitlab>.runners.Token as Token, host.softs<gitlab>.runners.Type as
Type, host.softs<gitlab>.runners.Platform as Platform,
host.softs<gitlab>.runners.ExecutorType as ExecutorType,
host.softs<gitlab>.runners.State as State, host.softs<gitlab>.runners.Active as
Active, host.softs<gitlab>.runners.AccessLevel as AccessLevel,
host.softs<gitlab>.runners.IPAddress as IPAddress,
host.softs<gitlab>.runners.RunUntaggedJobs as RunUntaggedJobs) | sort(ID)
```

По результатам запроса вы можете просмотреть, какие агенты защищены (для них в веб-интерфейсе GitLab отображается значок защищенного ключа, а в параметрах устанавливается флаг `Protected`), и к каким подсетям относятся агенты.

Если в инфраструктуре есть статичные агенты (static runners), которые привязаны к конкретным серверам или узлам, вы можете проверить, был ли выполнен их аудит, с помощью запроса:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>, host.softs<gitlab>.runners.ID as ID,
host.softs<gitlab>.runners.Token as Token, host.softs<gitlab>.runners.Type as
Type, host.softs<gitlab>.runners.Platform as Platform,
host.softs<gitlab>.runners.ExecutorType as ExecutorType,
host.softs<gitlab>.runners.State as State, host.softs<gitlab>.runners.Active as
Active, host.softs<gitlab>.runners.AccessLevel as AccessLevel,
host.softs<gitlab>.runners.IPAddress as IPAddress,
host.softs<gitlab>.runners.RunUntaggedJobs as RunUntaggedJobs) |

filter(host.softs<gitlab> and State = "online") |

join(select(@host, host.@audittime, host.ipaddress) as H, H.host.ipaddress =
IPAddress) | calc(totaldays(now() - H.host.@audittime) as dur_audit)|
sort(H.host.@audittime, ID)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Параметры безопасности (security settings)

Запрос для поиска данных об интеграции сервера GitLab с сервером LDAP (`LDAP configuration`) и с сервером Kerberos (`Kerberos configuration`):

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>,
host.softs<gitlab>.SecuritySettings.SignUpAllowed as SignUpAllowed,
host.softs<gitlab>.SecuritySettings.WebPasswordAuthEnabled as
WebPasswordAuthEnabled,
host.softs<gitlab>.SecuritySettings.GitPasswordAuthEnabled as
GitPasswordAuthEnabled, host.softs<gitlab>.SecuritySettings.LDAPConfigurations
as LDAPConfigurations,
host.softs<gitlab>.SecuritySettings.KerberosConfiguration as
KerberosConfiguration) | sort(@host)
```

Если результат запроса не пуст, то вы можете просмотреть параметры интеграции сервера GitLab с сервером LDAP и с сервером Kerberos. Эти серверы используются для аутентификации пользователей.

Запрос о параметрах интеграции с сервером LDAP:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.ID as ID,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.Label as Label,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.Host as Host,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.BindDN as BindDN,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.PlaintextPassword as
PlaintextPassword,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.Encryption as
Encryption, host.softs<gitlab>.SecuritySettings.LDAPConfigurations.BaseDN as
BaseDN, host.softs<gitlab>.SecuritySettings.LDAPConfigurations.LDAPFilter as
LDAPFilter) | sort(@host)
```

Результат запроса будет содержать список доменов, к которым подключен GitLab. Вы можете проверить, выполнен ли аудит найденных доменов, с помощью запроса:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.ID as ID,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.Label as Label,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.Host as Host,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.BindDN as BindDN,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.PlaintextPassword as
PlaintextPassword,
host.softs<gitlab>.SecuritySettings.LDAPConfigurations.Encryption as
```

```
Encryption, host.softs<gitlab>.SecuritySettings.LDAPConfigurations.BaseDN as
BaseDN, host.softs<gitlab>.SecuritySettings.LDAPConfigurations.LDAPFilter as
LDAPFilter) |

join(select(@directoryservice, directoryservice.@updatetime,
DirectoryService.Domains.DefaultNamingContext as DN) as DS, DS.DN = BaseDN) |
sort(@host)
```

Для доменов, у которых значение параметра `DS.directoryservice.@updatetime` пусто, необходимо выполнить аудит по LDAP (см. раздел 2.4.2).

Запрос о параметрах интеграции с сервером Kerberos:

```
filter(host.softs<gitlab>) |

select(@host, host.softs<gitlab>,
host.softs<gitlab>.SecuritySettings.KerberosConfiguration.KeytabPath as
KeytabPath,
host.softs<gitlab>.SecuritySettings.KerberosConfiguration.KerberosPrincipal as
KerberosPrincipal,
host.softs<gitlab>.SecuritySettings.KerberosConfiguration.KerberosPort as
KerberosPort, host.softs<gitlab>.SecuritySettings.KerberosConfiguration.HTTPS
as HTTPS,
host.softs<gitlab>.SecuritySettings.KerberosConfiguration.AllowUserCreation as
AllowUserCreation,
host.softs<gitlab>.SecuritySettings.KerberosConfiguration.BlockAutoCreatedUsers
as BlockAutoCreatedUsers) | sort(@host)
```

2.4.4.2. Аудит JFrog Artifactory

После сканирования активов с профилем Service Discovery (см. раздел 2.2) (с указанием портов 443 и 80) вы можете выполнить запрос для поиска активов с названием `artifactory`:

```
filter(host.endpoints<transportendpoint>[port in [443, 80] and status
= 'Open' and service.checks<GetHTTPHeaders>.Headers.Header = "X-ARTIFACTORY-
ID"]) |

select(@host, host.@PentestTime, host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)  | calc(TotalDays(now() - host.@PentestTime) as
dur_pentest)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Для аудита JFrog Artifactory необходимо выполнить задачи с профилем Unix Audit (для получения информации об узле) и с профилем Web API Audit (для получения информации о ПО и заполнения значений параметра `host.softs<Artifactory>`). После этого вы можете использовать запросы, представленные ниже.

> **Примечание.** Если аудит был проведен, но нет возможности определить, с каким именно профилем, то вы можете просмотреть параметры задач и убедиться, что задачи с обоими профилями выполняются корректно.

Запрос для поиска активов с JFrog Artifactory:

```
filter(host.softs<Artifactory>) |

select(@host, host.@PentestTime, host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)  | calc(TotalDays(now() - host.@PentestTime) as
dur_pentest)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

С помощью следующих запросов вы можете просмотреть информацию, полученную в результате аудита ПО.

IP-адреса и порты

Запрос для поиска IP-адресов и портов, по которым доступен JFrog Artifactory:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>.openports.Protocol,
host.softs<Artifactory>.openports.address,
host.softs<Artifactory>.openports.Port,
compactunique(host.interfaces.l3settings.address.address.address) as addresses)
| filter(host.softs<Artifactory>.openports.Protocol)
```

Если в качестве адреса указано значение `0.0.0.0`, то JFrog Artifactory доступен со всех IP-адресов актива (они перечислены в поле `addresses`).

Репозитории

В запросах, представленных ниже, используются следующие классы репозиториев JFrog Artifactory:

— Repository. Общий класс, который содержит URL и тип репозитория, включая удаленные репозитории. URL удаленного репозитория указывает на внешний источник, откуда будут загружаться артефакты. Удаленный репозиторий синхронизируется с внешним источником и является точкой входа в инфраструктуру организации при атаке на цепочку поставок,

направленную на поставщика удаленного репозитория. Поэтому необходимо определить состав удаленных репозиториев и обеспечить их мониторинг с помощью систем проверки пакетов (анализа зависимостей).

— LocalRepository. Локальные репозитории дают возможность сохранять артефакты, которые сгенерированы или загружены внутри организации. URL локального репозитория указывает на расположение репозитория в хранилище артефактов.

— VirtualRepository. Виртуальные репозитории объединяют несколько репозиториев (как локальных, так и удаленных), предоставляют интерфейс для работы с ними через единый URL и обеспечивают централизованный доступ к артефактам. URL виртуального репозитория указывает на логическую точку доступа, которая объединяет несколько реальных репозиториев.

Для поиска репозиториев вы можете использовать запросы, представленные ниже. Рекомендуется использовать запросы для конкретных классов репозиториев.

Запрос для поиска репозиториев общего класса:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>, host.softs<Artifactory>.Repositories.Key
as Key, host.softs<Artifactory>.Repositories.URL as URL,
host.softs<Artifactory>.Repositories.Type as Type,
host.softs<Artifactory>.Repositories.PackageType as PackageType)
```

Запрос для поиска удаленных репозиториев:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>, host.softs<Artifactory>.Repositories.Key
as Key, host.softs<Artifactory>.Repositories.URL as URL,
host.softs<Artifactory>.Repositories.Type as Type,
host.softs<Artifactory>.Repositories.PackageType as PackageType) |

filter(host.softs<Artifactory> and Type = "REMOTE") | sort(@host, URL)
```

Запрос для поиска локальных репозиториев:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>,
host.softs<Artifactory>.Repositories<LocalRepository>.Key as Key,
host.softs<Artifactory>.Repositories<LocalRepository>.URL as URL,
host.softs<Artifactory>.Repositories<LocalRepository>.Type as Type,
host.softs<Artifactory>.Repositories<LocalRepository>.PackageType as
PackageType,
compactunique(host.softs<Artifactory>.Repositories<LocalRepository>.Entities.Pa
th) as Entities_path,
compactunique(host.softs<Artifactory>.Repositories<LocalRepository>.Permissions
) as Permissions) |

filter(host.softs<Artifactory> and Key) | sort(@host, URL)
```

Результат запроса будет содержать значения параметров `Entities_path` и `Permissions`. Если они не заполнены, то необходимо проверить выполнение задач на аудит (см. раздел 2.4) и настройку прав доступа (permissions).

Запрос для поиска виртуальных репозиториев:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>,
host.softs<Artifactory>.Repositories<VirtualRepository>.Key as Key,
host.softs<Artifactory>.Repositories<VirtualRepository>.URL as URL,
host.softs<Artifactory>.Repositories<VirtualRepository>.Type as Type,
host.softs<Artifactory>.Repositories<VirtualRepository>.PackageType as
PackageType,
compactunique(host.softs<Artifactory>.Repositories<VirtualRepository>.IncludedR
epoKeys) as IncludedRepoKeys) |

filter(host.softs<Artifactory> and Type) | sort(@host, URL)
```

Группы и пользователи в группах

Группы используются для назначения прав доступа пользователям, входящим в группы.

Запрос для поиска данных о группах:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>, host.softs<Artifactory>.Groups.Name as
Name, host.softs<Artifactory>.Groups.Realm as Realm,
host.softs<Artifactory>.Groups.AdminPrivileges as AdminPrivileges,
host.softs<Artifactory>.Groups.LDAPGroupDN as LDAPGroupDN) | sort(@host)
```

Параметр `Realm` обозначает источник аутентификации, который предоставляет учетные данные пользователей. По значению этого параметра вы можете выявить внутренние группы (например, для них могут использоваться значения `internal` или `artifactory`) и LDAP-группы (например, для них может использоваться значение `LDAP` или другие значения, указывающие на источник аутентификации).

Запрос для поиска данных о пользователях:

```
filter(host.softs<Artifactory>) |

select(@host, host.softs<Artifactory>, host.softs<Artifactory>.users.ID as ID,
host.softs<Artifactory>.users.Name as Name,
host.softs<Artifactory>.users.IsAdmin as IsAdmin,
host.softs<Artifactory>.users.Realm as Realm,
compactunique(host.softs<Artifactory>.users.Groups) as Groups,
host.softs<Artifactory>.users.LastLoggedIn as LastLoggedIn,
host.softs<Artifactory>.users.Disabled as Disabled) | sort(@host, ID)
```

Запрос для поиска данных о группах с пользователями:

```
filter(host.softs<Artifactory>) |
```

```
select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Groups.Name as Name,
host.softs<Artifactory>.Groups.Realm as Realm,
host.softs<Artifactory>.Groups.AdminPrivileges as AdminPrivileges,
host.softs<Artifactory>.Groups.LDAPGroupDN as LDAPGroupDN) |

join(select(host.@id, host.softs<Artifactory>, host.softs<Artifactory>.users.ID
as ID, host.softs<Artifactory>.users.Name as Name,
host.softs<Artifactory>.users.IsAdmin as IsAdmin,
host.softs<Artifactory>.users.Realm as Realm,
host.softs<Artifactory>.users.Groups as "Group",
host.softs<Artifactory>.users.LastLoggedIn as LastLoggedIn,
host.softs<Artifactory>.users.Disabled as Disabled) as Usr, Usr.host.@id =
host.@id and Usr.host.softs<Artifactory> = host.softs<Artifactory> and Name =
Usr.Group) | sort(Usr.Name) | group(Realm, Name, COUNT(*)) |
sort("COUNT(*)" DESC)
```

Запрос для поиска данных о пользователях, входящих в группы:

```
filter(host.softs<Artifactory>) |

host.softs<Artifactory>.users.ID as ID, host.softs<Artifactory>.users.Name as
Name, host.softs<Artifactory>.users.IsAdmin as IsAdmin,
host.softs<Artifactory>.users.Realm as Realm,
host.softs<Artifactory>.users.Groups as "Group",
host.softs<Artifactory>.users.LastLoggedIn as LastLoggedIn,
host.softs<Artifactory>.users.Disabled as Disabled) |

join(select(host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Groups.Name as Name,
host.softs<Artifactory>.Groups.Realm as Realm,
host.softs<Artifactory>.Groups.AdminPrivileges as AdminPrivileges,
host.softs<Artifactory>.Groups.LDAPGroupDN as LDAPGroupDN) as Grp,
Grp.host.@id = host.@id and Grp.host.softs<Artifactory> =
host.softs<Artifactory> and Grp.Name = "Group")
```

Права доступа (permissions)

Система управления доступом в JFrog Artifactory определяет, какие пользователи и группы могут осуществлять те или иные действия с репозиториями, артефактами, пользователями. Из запросов, представленных в этом разделе, для анализа результатов аудита рекомендуется в первую очередь использовать запросы, связанные с правами доступа (permissions).

Запрос для поиска репозиториев, для доступа к которым не назначены явные права доступа пользователей или групп:

```
filter(host.softs<Artifactory>) |
```

```
select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Repositories.Key as Key,
host.softs<Artifactory>.Repositories.URL as URL,
host.softs<Artifactory>.Repositories.Type as Type,
host.softs<Artifactory>.Repositories.PackageType as PackageType) |

join(select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Permissions.Name as Name,
compactunique(host.softs<Artifactory>.Permissions.IncludesPatterns) as
IncludesPatterns,
compactunique(host.softs<Artifactory>.Permissions.ExcludesPatterns) as
ExcludesPatterns, host.softs<Artifactory>.Permissions.Repositories as
Repositories, compactunique(host.softs<Artifactory>.Permissions.Targets) as
Targets) |

filter(host.softs<Artifactory>) as Perm, Perm.host.@id = host.@id and
Perm.host.softs<Artifactory> = host.softs<Artifactory> and Perm.Repositories =
Key) |

filter(Perm.Repositories = null) | sort(Key)
```

Для каждого из полученных репозиториев необходимо проверить права доступа и возможность ограничения анонимного доступа. Иначе может быть потерян контроль доступа к таким репозиториям: они могут быть доступны для всех пользователей (если права доступа не ограничены на уровне системы) или полностью недоступны для пользователей.

Запрос для поиска прав доступа к репозиториям, которые были удалены:

```
filter(host.softs<Artifactory>) |

select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Permissions.Name as Name,
host.softs<Artifactory>.Permissions.IncludesPatterns as IncludesPatterns,
host.softs<Artifactory>.Permissions.ExcludesPatterns as ExcludesPatterns,
host.softs<Artifactory>.Permissions.Repositories as Repositories,
host.softs<Artifactory>.Permissions.Targets as Targets) |

filter(host.softs<Artifactory> and not Repositories in ["ANY", "ANY
LOCAL", "ANY REMOTE"]) |

join(select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Repositories.Key as Key,
host.softs<Artifactory>.Repositories.URL as URL,
host.softs<Artifactory>.Repositories.Type as Type,
host.softs<Artifactory>.Repositories.PackageType as PackageType) |

filter(host.softs<Artifactory>) as Rep, Rep.host.@id = host.@id and
Rep.host.softs<Artifactory> = host.softs<Artifactory> and Repositories =
Rep.Key) |

filter(Rep.Key = null)
```

Необходимо удалить права доступа к репозиториям, которые были удалены.

Запрос для поиска прав доступа, которые были назначены не группе, а отдельному пользователю:

```
filter(host.softs<Artifactory>) |

select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.Permissions.Name as Name,
compactunique(host.softs<Artifactory>.Permissions.IncludesPatterns) as
IncludesPatterns,
compactunique(host.softs<Artifactory>.Permissions.ExcludesPatterns) as
ExcludesPatterns,
compactunique(host.softs<Artifactory>.Permissions.Repositories) as
Repositories, compactunique(host.softs<Artifactory>.Permissions.Targets) as
Targets, host.softs<Artifactory>.Permissions.Targets as Target) |

join(select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.users.ID as ID, host.softs<Artifactory>.users.Name as
Name, host.softs<Artifactory>.users.IsAdmin as IsAdmin,
host.softs<Artifactory>.users.Realm as Realm,
compactunique(host.softs<Artifactory>.users.Groups) as Groups,
host.softs<Artifactory>.users.LastLoggedIn as LastLoggedIn,
host.softs<Artifactory>.users.Disabled as Disabled) |

filter(host.softs<Artifactory>) as Usr, Usr.host.@id = host.@id and
Usr.host.softs<Artifactory> = host.softs<Artifactory> and Usr.Name = Target) |

filter(Usr.Name) | group(Usr.Name, compactunique(Name), COUNT(*)) |
sort("COUNT(*)" DESC)
```

Необходимо уточнить, с какой целью права доступа назначались отдельным пользователям. Вы можете определить целесообразность такого назначения прав доступа, или включить пользователей в подходящие группы, или удалить избыточные права доступа.

Параметры безопасности (security settings)

Для оценки параметров безопасности вы можете выполнить запросы, представленные ниже.

Запрос 1:

```
filter(host.softs<Artifactory>) |

select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.SecuritySettings.AnonymousAccess as AnonymousAccess,
host.softs<Artifactory>.SecuritySettings.BasicAuthentication as
BasicAuthentication,
host.softs<Artifactory>.SecuritySettings.PasswordEncryption as
PasswordEncryption, host.softs<Artifactory>.SecuritySettings.PasswordMaxAge as
PasswordMaxAge, host.softs<Artifactory>.SecuritySettings.FailedLoginAttempts as
FailedLoginAttempts)
```

Запрос 2:

```
filter(host.softs<Artifactory>) |

select(@host, host.@id, host.softs<Artifactory>,
host.softs<Artifactory>.SecuritySettings.LDAPSettings.Key as Key,
host.softs<Artifactory>.SecuritySettings.LDAPSettings.Host as Host,
host.softs<Artifactory>.SecuritySettings.LDAPSettings.BindDN as BindDN,
host.softs<Artifactory>.SecuritySettings.LDAPSettings.BaseDN as BaseDN,
host.softs<Artifactory>.SecuritySettings.LDAPSettings.LDAPFilter as LDAPFilter,
host.softs<Artifactory>.SecuritySettings.LDAPSettings.CreatingNewUsersAllowed
as CreatingNewUsersAllowed)
```

По результатам обоих запросов необходимо проверить значения всех параметров (все они булевы). Если получены пустые значения (`null`), то необходимо проверить выполнение задач на аудит (см. раздел 2.4) и настройку прав доступа (permissions).

2.4.5. Аудит HAProxy

HAProxy — это серверное ПО, которое используется для маршрутизации трафика между несколькими серверами в целях повышения доступности и отказоустойчивости TCP- и HTTP-приложений.

Перед аудитом серверов с HAProxy необходимо выполнить аудит сетевых устройств (см. раздел 2.4.1). Это позволит найти прокси-серверы, которые находятся во внутренней сети и защищены правилами NAT (см. раздел 2.4.1.1). Чтобы найти прокси-серверы, которые не управляются правилами NAT, необходимо регулярно выполнять следующие действия:

1. Сканирование портов (см. раздел 2.1) для обнаружения активов с открытыми портами 443 и 80.
2. Сканирование активов методом черного ящика (см. раздел 2.2) с профилем Service Discovery (в параметрах задачи необходимо указать порты 443 и 80).
3. Аудит серверов из общих подсетей и с открытыми портами 443 и 80 (см. раздел 2.4.4).
4. Мониторинг трафика (см. раздел 2.4.4) и аудит серверов по результатам мониторинга.

   Примечание. Если в инфраструктуре установлен PT NAD, вы можете использовать его для мониторинга трафика.

Для проверки выполнения аудита вы можете выполнить запрос:

```
filter(host.softs<HAProxy>)

| select(@host, host.@audittime, host.softs<HAProxy>.Vendor as Vendor,
host.softs<HAProxy>.Name as Name, host.softs<HAProxy>.Version as Version,
compactunique(host.softs<HAProxy>.ConfFiles) as ConfFiles,
compactunique(host.softs<HAProxy>.Frontends.Name) as F_Names,
compactunique(host.softs<HAProxy>.Backends.Name) as B_Names) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(host.@audittime)
```

Для активов, у которых по результатам запроса значение хотя бы одного из параметров не заполнено или аудит которых проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

После выполнения аудита вы можете выполнить запросы, представленные ниже.

Параметры HAProxy

Запрос для просмотра параметров HAProxy:

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Vendor as Vendor, host.softs<HAProxy>.Name
as Name, host.softs<HAProxy>.InstallPath as InstallPath,
host.softs<HAProxy>.Version as Version, host.softs<HAProxy>.Architecture as
Architecture, host.softs<HAProxy>.OsFamily as OsFamily,
compactunique(host.softs<HAProxy>.Ports) as Ports,
host.softs<HAProxy>.UninstallKey as UninstallKey,
compactunique(host.softs<HAProxy>.ConfFiles) as ConfFiles,
compactunique(host.softs<HAProxy>.ConfDir) as ConfDir,
host.softs<HAProxy>.Chroot as Chroot, host.softs<HAProxy>.User as User,
host.softs<HAProxy>.Group as "Group", host.softs<HAProxy>.ExternalCheck as
ExternalCheck, host.softs<HAProxy>.InsecureForkWanted as InsecureForkWanted,
host.softs<HAProxy>.InsecureSetuidWanted as InsecureSetuidWanted,
compactunique(host.softs<HAProxy>.LuaLoad) as LuaLoad,
compactunique(host.softs<HAProxy>.LuaLoadPerThread) as LuaLoadPerThread,
host.softs<HAProxy>.UnixBind as UnixBind)

| filter(Vendor) | sort(@host)
```

Запрос для поиска IP-адресов и портов, по которым доступно HAProxy:

```
filter(host.softs<HAProxy>)

| select(@host, host.@id, host.softs<HAProxy>.openports.Protocol as proto,
host.softs<HAProxy>.openports.address as op_address,
compactunique(host.softs<HAProxy>.openports.Port) as op_ports,
compactunique(host.endpoints<ipendpoint>.address) as end_addresses)

| filter(proto)

| join(select(host.@id, host.endpoints<ipendpoint>.address as address) as End,
End.address = op_address)

| filter(op_address = 0.0.0.0 or End.address)

| select(@host, op_address, End.address, op_ports, end_addresses)

| sort(@host, op_address, op_ports)
```

> **Примечание.** В HAProxy могут использоваться мнимые (виртуальные) IP-адреса, которые назначаются на несколько серверов (они не привязаны к какому-либо физическому серверу). Мнимые адреса применяются, например, в интерфейсе обратной связи

(loopback) для обеспечения связи между приложениями или сервисами, работающими на одном устройстве, или при использовании адресов link-local, предназначенных для коммуникаций в пределах одного сегмента сети. Запрос, представленный выше, построен без использования мнимых адресов.

Запрос для поиска IP-адресов HAProxy, указанных в правилах NAT на межсетевых экранах:

```
filter(host.softs<HAProxy>.Name != null)
| select(@host, host.endpoints<ipendpoint>.address as ip)
| filter(ip not in [127.0.0.1, ::1])
|
join(select(compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedSource.Values.address) as source,
NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAddress.Values.address as address,
NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAddress.Values.prefix as pref,
compactunique(NetworkDeviceHost.NATTable.rules.NormalizedTranslatedProtocol.TCPUDPOptions.DestinationPorts.Values) as ntpdst_ports)
| filter(address and pref = 32) | unique() as NAT, NAT.address = ip)
| unique() | sort(@host, ip, NAT.source)
```

Для серверов с HAProxy, у которых по результатам запроса заполнено значение параметра `NAT.source`, критически важно своевременно и корректно выполнять аудит, чтобы обеспечить эффективную и безопасную маршрутизацию трафика и доступность приложений. Рекомендуется присвоить серверам с HAProxy, а также серверам, которые обрабатывают поступающий через HAProxy трафик, высокую значимость, как описано в разделе «Присвоение значимости активам» Руководства оператора MaxPatrol VM.

Элементы конфигурации HAProxy

Вы можете настроить проксирование в конфигурационном файле HAProxy.

К элементам конфигурации HAProxy относятся:

- Global. Определяет общие параметры системы, параметры производительности и безопасности.
- Frontend. Определяет параметры приема пакетов входящего трафика (например, порты, IP-адреса, правила маршрутизации).
- Backend. Определяет параметры передачи пакетов для обработки. Регулирует балансировку нагрузки.

— Default. Определяет параметры по умолчанию, если они не определены в других элементах.

— Listen. Определяет параметры приема и передачи пакетов в одном элементе (объединяет элементы frontend и backend). Используется для упрощения конфигурации, когда нужно указать минимальный набор параметров и нет сложных правил для маршрутизации трафика и балансировки нагрузки.

Запрос для поиска сервисов приема пакетов (frontends):

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Frontends.Name as Name,
host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.NetBinds.Protocol as NetBinds_Protocol,
host.softs<HAProxy>.Frontends.NetBinds.Address as NetBinds_Address,
compactunique(host.softs<HAProxy>.Frontends.NetBinds.Port) as NetBinds_Ports,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Path) as
SocketBinds_paths,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Type) as
SocketBinds_types, compactunique(host.softs<HAProxy>.Frontends.Backends) as
Backends, host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)
```

Запрос для поиска сервисов приема пакетов (frontends), которые не относятся к интерфейсам обратной связи (loopback):

```
filter(host.softs<HAProxy>)

| select(@host, host.@id, host.softs<HAProxy>.Frontends.Name as Name,
host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.Status as Status,
host.softs<HAProxy>.Frontends.NetBinds.Protocol as NetBinds_Protocol,
host.softs<HAProxy>.Frontends.NetBinds.Address as NetBinds_Address,
compactunique(host.softs<HAProxy>.Frontends.NetBinds.Port) as NetBinds_Ports,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Path) as
SocketBinds_paths,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Type) as
SocketBinds_types, compactunique(host.softs<HAProxy>.Frontends.Backends) as
Backends, host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)

| join(select(host.@id, host.endpoints<ipendpoint>.address as address) as End,
End.address = NetBinds_Address)

| filter(NetBinds_Address = 0.0.0.0 or NetBinds_Address = :: or End.address)

| select(@host, Name, Mode, Status, NetBinds_Address, NetBinds_Ports,
SocketBinds_paths, SocketBinds_types, Backends, DefaultBackend) | unique() |
sort(@host, Name, NetBinds_Address)
```

Запрос для поиска сервисов приема пакетов (frontends), для которых не указаны сервисы передачи пакетов (backends):

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Frontends.Name as Name,
host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.Status as Status,
compactunique(host.softs<HAProxy>.Frontends.NetBinds) as NetBinds,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Path) as
SocketBinds_paths,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Type) as
SocketBinds_types, compactunique(host.softs<HAProxy>.Frontends.Backends) as
Backends, host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)

| filter(Backends = null and DefaultBackend = null)
```

Как правило, найденные сервисы приема пакетов (frontends) предназначены для изменения URL-адресов, перенаправления на новый домен или протокол (redirect) или для блокировки или отказа в обслуживании некоторых запросов (deny). Если они есть в таблицах NAT, то необходимо убедиться, что для всего трафика прослушивающих серверов настроены только правила перенаправления (redirect) и блокировки или отказа в обслуживании (deny).

Запрос для поиска сервисов приема пакетов (frontends), для которых указаны несуществующие сервисы передачи пакетов (backends, отсутствующие в конфигурационном файле):

```
filter(host.softs<HAProxy>)

| select(@host, host.@id, host.softs<HAProxy>.Frontends.Name as Name,
host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.NetBinds.Protocol as NetBinds_Protocol,
host.softs<HAProxy>.Frontends.NetBinds.Address as NetBinds_Address,
compactunique(host.softs<HAProxy>.Frontends.NetBinds.Port) as NetBinds_Ports,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Path) as
SocketBinds_paths,
compactunique(host.softs<HAProxy>.Frontends.SocketBinds.Type) as
SocketBinds_types, host.softs<HAProxy>.Frontends.Backends as Backend,
host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)

| filter(Backend or DefaultBackend)

| join(filter(host.softs<HAProxy>)

| select(@host, host.@id, host.softs<HAProxy>.Backends.Name as Name,
host.softs<HAProxy>.Backends.Mode as Mode, host.softs<HAProxy>.Backends.Status
as Status, host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress) as B,
B.host.@id = host.@id and (B.name = Backend or DefaultBackend = B.name))

| filter(B.name = null) | sort(@host)
```

Как правило, при применении такой конфигурации возникает ошибка. Необходимо исправить параметры в конфигурационном файле.

Запрос для поиска сервисов передачи пакетов (backends):

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Backends.Name as Name,
host.softs<HAProxy>.Backends.Mode as Mode, host.softs<HAProxy>.Backends.Status
as Status, host.softs<HAProxy>.Backends.Servers.Name as SName,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress,
host.softs<HAProxy>.Backends.Servers.Hostname as Hostname,
host.softs<HAProxy>.Backends.Servers.Protocol as Protocol,
host.softs<HAProxy>.Backends.Servers.Port as Port)
```

Запрос для поиска сервисов передачи пакетов (backends), которые не определяют параметры передачи:

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Backends.Name as Name,
host.softs<HAProxy>.Backends.Mode as Mode, host.softs<HAProxy>.Backends.Status
as Status, host.softs<HAProxy>.Backends.Servers.Name as SName,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress,
host.softs<HAProxy>.Backends.Servers.Hostname as Hostname,
host.softs<HAProxy>.Backends.Servers.Protocol as Protocol,
host.softs<HAProxy>.Backends.Servers.Port as Port)

| filter(Name and IpAddress = null)
```

Найденные сервисы backends не участвуют в передаче пакетов, при этом усложняют конфигурацию. Поэтому необходимо выяснить, почему они существуют, но не используются, и внести соответствующие изменения в параметры конфигурации. Чтобы найти сервисы приема пакетов (frontends), которые распределяют трафик на такие сервисы backends, вы можете выполнить запрос:

```
filter(host.softs<HAProxy>)

| select(@host, host.@Name, host.@id, host.softs<HAProxy>.Frontends.Name as
Name, host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.Status as Status,
host.softs<HAProxy>.Frontends.NetBinds.Protocol as NetBinds_Protocol,
host.softs<HAProxy>.Frontends.NetBinds.Address as NetBinds_Address,
host.softs<HAProxy>.Frontends.NetBinds.Port as NetBinds_Port,
host.softs<HAProxy>.Frontends.Backends as Backend,
host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)

| filter(Name and (Backend or DefaultBackend))

| join(filter(host.softs<HAProxy>) | select(@host, host.@id,
host.softs<HAProxy>.Backends.Name as Name, host.softs<HAProxy>.Backends.Mode as
Mode, host.softs<HAProxy>.Backends.Status as Status,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress) as B, B.host.@id =
host.@id and (B.name = Backend or DefaultBackend = B.name))

| filter(B.Name and B.IpAddress = null)
```

```
| select(host.@Name, Name, Status, compactunique(NetBinds_Address),
compactunique(NetBinds_Port), B.Name, B.Status, B.IpAddress) |
filter(B.IpAddress)| unique()

| sort(host.@Name, Name)
```

Запрос для поиска сервисов передачи пакетов (backends), определенных параметром `DefaultBackend` (то есть не подходящих под другие правила маршрутизации):

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Defaults.Name as Name,
host.softs<HAProxy>.Defaults.Mode as Mode, host.softs<HAProxy>.Defaults.Status
as Status, host.softs<HAProxy>.Defaults.DefaultBackend as DefaultBackend)

| sort(@host)
```

Если результат запроса не пуст, то наличие сервисов передачи пакетов (backends), определенных параметром `DefaultBackend`, может усложнить понимание конфигурации HAProxy. Рекомендуется определить правила маршрутизации в параметрах конфигурации для сервисов frontends.

Запрос для поиска сервисов listen:

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Listen.Name as Name,
host.softs<HAProxy>.Listen.Mode as Mode, host.softs<HAProxy>.Listen.Status as
Status, compactunique(host.softs<HAProxy>.Listen.NetBinds) as NetBinds,
compactunique(host.softs<HAProxy>.Listen.SocketBinds.Path) as
SocketBinds_paths, compactunique(host.softs<HAProxy>.Listen.SocketBinds.Type)
as SocketBinds_types, compactunique(host.softs<HAProxy>.Listen.Backends) as
Listen, host.softs<HAProxy>.Listen.DefaultBackend,
host.softs<HAProxy>.Listen.Servers.Name as SName,
host.softs<HAProxy>.Listen.Servers.IpAddress as IpAddress,
host.softs<HAProxy>.Listen.Servers.Hostname as Hostname,
host.softs<HAProxy>.Listen.Servers.Protocol as Protocol,
host.softs<HAProxy>.Listen.Servers.Port as Port) | filter(Name) | sort(@host,
Name)
```

Иногда сервисы listen включают только сервисы backends — например, если они используются для изменения URL, перенаправления на новый домен или протокол (redirect). В этом случае для поиска активов в дополнение к запросам, представленным ниже в разделе «Активы, связанные с HAProxy», вы можете выполнить запрос:

```
filter(host.softs<HAProxy>)

| select(@host, host.softs<HAProxy>.Listen.Name as Name,
host.softs<HAProxy>.Listen.Mode as Mode, host.softs<HAProxy>.Listen.Status as
Status, compactunique(host.softs<HAProxy>.Listen.NetBinds) as NetBinds,
compactunique(host.softs<HAProxy>.Listen.SocketBinds.Path) as
SocketBinds_paths, compactunique(host.softs<HAProxy>.Listen.SocketBinds.Type)
```

```
as SocketBinds_types, compactunique(host.softs<HAProxy>.Listen.Backends) as
Listen, host.softs<HAProxy>.Listen.DefaultBackend,
host.softs<HAProxy>.Listen.Servers.Name as SName,
host.softs<HAProxy>.Listen.Servers.IpAddress as IpAddress,
host.softs<HAProxy>.Listen.Servers.Hostname as Hostname,
host.softs<HAProxy>.Listen.Servers.Protocol as Protocol,
host.softs<HAProxy>.Listen.Servers.Port as Port) | filter(Name)

| join(select(@host, host.endpoints<ipendpoint>.address as address,
host.@audittime) as H, H.address = IpAddress)

| select(@host, Name, Status, IpAddress, H.@host, H.host.@audittime) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit)

| sort(@host, Name)
```

Активы, связанные с HAProxy

К активам, связанным с HAProxy, относятся физические или виртуальные серверы или узлы, которые управляются сервисами backends и обрабатывают запросы.

Запрос для поиска активов, связанных с HAProxy:

```
filter(host.softs<HAProxy>)

| select(@host, host.@Name, host.@id, host.softs<HAProxy>.Frontends.Name as
Name, host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.Status as Status,
host.softs<HAProxy>.Frontends.NetBinds.Protocol as NetBinds_Protocol,
host.softs<HAProxy>.Frontends.NetBinds.Address as NetBinds_Address,
host.softs<HAProxy>.Frontends.NetBinds.Port as NetBinds_Port,
host.softs<HAProxy>.Frontends.Backends as Backend,
host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)

| filter(Name and (Backend or DefaultBackend))

| join(filter(host.softs<HAProxy>) | select(@host, host.@id,
host.softs<HAProxy>.Backends.Name as Name, host.softs<HAProxy>.Backends.Mode as
Mode, host.softs<HAProxy>.Backends.Status as Status,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress) as B, B.host.@id =
host.@id and (B.name = Backend or DefaultBackend = B.name))

| select(host.@Name, Name, Status, NetBinds_Address, NetBinds_Port, B.Name,
B.Status, B.IpAddress)

| filter(B.IpAddress)| unique()

| join(select(@host, host.@AuditTime, host.endpoints<ipendpoint>.address as
address) as H, H.address = B.IpAddress)
```

```
| select(host.@Name, Name, Status, compactunique(NetBinds_Address),
compactunique(NetBinds_Port), B.Name, B.Status, B.IpAddress, H.@host,
H.host.@AuditTime) | calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(host.@Name, Name)
```

Для поиска активов, связанных с HAProxy, для которых аудит не проводился вообще или проводился больше месяца назад, вы можете выполнить любой из запросов, приведенных ниже.

Запрос 1 (с использованием параметров `frontends` и `backends`):

```
filter(host.softs<HAProxy>)

| select(@host, host.@Name, host.@id, host.softs<HAProxy>.Frontends.Name as
Name, host.softs<HAProxy>.Frontends.Mode as Mode,
host.softs<HAProxy>.Frontends.Status as Status,
host.softs<HAProxy>.Frontends.NetBinds.Protocol as NetBinds_Protocol,
host.softs<HAProxy>.Frontends.NetBinds.Address as NetBinds_Address,
host.softs<HAProxy>.Frontends.NetBinds.Port as NetBinds_Port,
host.softs<HAProxy>.Frontends.Backends as Backend,
host.softs<HAProxy>.Frontends.DefaultBackend as DefaultBackend)

| filter(Name and (Backend or DefaultBackend))

| join(filter(host.softs<HAProxy>) | select(@host, host.@id,
host.softs<HAProxy>.Backends.Name as Name, host.softs<HAProxy>.Backends.Mode as
Mode, host.softs<HAProxy>.Backends.Status as Status,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress) as B, B.host.@id =
host.@id and (B.name = Backend or DefaultBackend = B.name))

| select(host.@Name, Name, Status, NetBinds_Address, NetBinds_Port, B.Name,
B.Status, B.IpAddress)

| filter(B.IpAddress)| unique()

| join(select(@host, host.@AuditTime, host.endpoints<ipendpoint>.address as
address) as H, H.address = B.IpAddress)

| select(host.@Name, Name, Status, compactunique(NetBinds_Address),
compactunique(NetBinds_Port), B.Name, B.Status, B.IpAddress, H.@host,
H.host.@AuditTime) | calc(TotalDays(now() - H.host.@AuditTime) as dur_audit)

| filter(H.@host = null or H.host.@AuditTime = null or dur_audit > 30) |
sort(host.@Name, Name)
```

Запрос 2 (с использованием параметров `backends`):

```
filter(host.softs<HAProxy>) | select(@host, host.@Name, host.@id,
host.softs<HAProxy>.Backends.Name as Name, host.softs<HAProxy>.Backends.Mode as
Mode, host.softs<HAProxy>.Backends.Status as Status,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress)

| select(host.@Name, Name, Status, IpAddress)
```

```
| filter(IpAddress)| unique()

| join(select(@host, host.@AuditTime, host.endpoints<ipendpoint>.address as
address) as H, H.address = IpAddress)

| select(host.@Name, Name, Status, IpAddress, H.@host, H.host.@AuditTime) |
calc(TotalDays(now() - H.host.@AuditTime) as dur_audit)

| filter(H.@host = null or H.host.@AuditTime = null or dur_audit > 30) |
sort(host.@Name, Name)
```

Запрос 3 (с использованием параметров `backends` и исключением повторяющихся результатов, получаемых от кластерных групп серверов или от самих узлов с HAProxy):

```
filter(host.softs<HAProxy>)
| select(@host, host.@Name, host.@id, host.softs<HAProxy>.Backends.Name as
Name, host.softs<HAProxy>.Backends.Mode as Mode,
host.softs<HAProxy>.Backends.Status as Status,
host.softs<HAProxy>.Backends.Servers.IpAddress as IpAddress)

| select(Name, IpAddress)

| filter(IpAddress)| unique()

| join(select(@host, host.@AuditTime, host.endpoints<ipendpoint>.address as
address) as H, H.address = IpAddress)

| select(Name, IpAddress, H.@host, H.host.@AuditTime) | calc(TotalDays(now() -
H.host.@AuditTime) as dur_audit)

| filter(H.@host = null or H.host.@AuditTime = null or dur_audit > 30) |
sort(IpAddress)
```

Для активов, данные о которых найдены в результате этих запросов, необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

2.4.6. Аудит OpenVPN

В OpenVPN для установки соединений по умолчанию используется порт 1194 (UDP). Сервер OpenVPN может быть настроен на работу с другими портами, а также вместо протокола UDP может использоваться TCP.

Для поиска серверов OpenVPN, использующих порт 1194 (UDP), вы можете выполнить запрос:

```
filter(Host.Endpoints<TransportEndpoint>[Port = 1194 and Protocol = 'udp' and
Status = 'Open'])

| select(@host, host.@audittime, compactunique(host.softs<OpenVPN>.Configs))

| calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(@host)
```

Если в инфраструктуре установлен PT NAD, то для поиска серверов OpenVPN, использующих другие порты (и протокол UDP или TCP) вы можете использовать его и выполнить следующие шаги:

1. Применить условие фильтрации `app_proto == "openvpn"`.
2. На странице **Дашборды** экспортировать данные виджета **Серверы по сессиям и трафику**, нажав  .
3. Если среди перечисленных серверов есть серверы OpenVPN с внешними IP-адресами, то имеется исходящий трафик. В этом случае вы можете убедиться, что все сессии легитимны.
4. Если среди перечисленных серверов есть серверы OpenVPN с внутренними IP-адресами, то имеется входящий трафик, который преобразуется правилами NAT (см. раздел 2.4.1.1). В этом случае вы можете просмотреть правила NAT и определить, к каким внутренним устройствам или сервисам направляется трафик.

Для проверки выполнения аудита вы можете выполнить запрос:

```
filter(host.softs<OpenVPN>.configs)

| select(@host, host.@audittime, host.softs<OpenVPN>.Vendor as Vendor,
host.softs<OpenVPN>.Name as Name, host.softs<OpenVPN>.Version as Version,
compactunique(host.softs<OpenVPN>.configs.ConfigPath) as ConfFiles,
host.softs<OpenVPN>.configs.Mode as mode,
compactunique(host.softs<OpenVPN>.configs.routes) as routes,
compactunique(host.softs<OpenVPN>.configs.clients.clientName) as clients)

| calc(TotalDays(now() - host.@AuditTime) as dur_audit)

| sort(@host)
```

Для активов, у которых по результатам запроса значение хотя бы одного из параметров не заполнено, или аудит которых проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

После выполнения аудита вы можете выполнить запросы, представленные ниже.

Запрос для просмотра параметров серверов OpenVPN:

```
filter(host.softs<OpenVPN>.Configs)

| select(@host, host.@audittime, compactunique(host.softs<OpenVPN>.Configs))

| calc(TotalDays(now() - host.@AuditTime) as dur_audit)  | sort(@host)
```

Запрос для поиска IP-адресов и портов, по которым доступны серверы OpenVPN:

```
filter(host.softs<OpenVPN>.configs)
```

```
| select(@host, host.@id, host.softs<OpenVPN>.openports.Protocol as proto,
host.softs<OpenVPN>.openports.address as op_address,
compactunique(host.softs<OpenVPN>.openports.Port) as op_ports,
compactunique(host.endpoints<ipendpoint>.address) as end_addresses) |
filter(proto)

| join(select(host.@id, host.endpoints<ipendpoint>.address as address) as End,
End.address = op_address)

| filter(op_address = 0.0.0.0 or End.address)

| select(@host, op_address, End.address, op_ports, end_addresses)

| sort(@host, op_address, op_ports)
```

Запрос для поиска IP-адресов серверов OpenVPN, указанных в правилах NAT на межсетевых экранах:

```
filter(host.softs<OpenVPN>.Configs)

| select(@host, host.endpoints<ipendpoint>.address as ip)

| filter(ip not in [127.0.0.1, ::1])

|
join(select(compactunique(NetworkDeviceHost.NATTable.Rules.NormalizedSource.Val
ues.address) as source,
NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAddress.Values.
address as address,
NetworkDeviceHost.NATTable.Rules.NormalizedTranslatedDestinationAddress.Values.
prefix as pref,
compactunique(NetworkDeviceHost.NATTable.rules.NormalizedTranslatedProtocol.TCP
UDPOptions.DestinationPorts.Values) as ntpdst_ports) | filter(address and pref
= 32) | unique() as NAT, NAT.address = ip)

| unique() | sort(@host, ip, NAT.source)
```

Для серверов OpenVPN, у которых по результатам запроса заполнено значение параметра `NAT.source`, критически важно своевременно и корректно выполнять аудит, чтобы обеспечить эффективную и безопасную маршрутизацию трафика. Рекомендуется присвоить таким серверам OpenVPN, а также серверам, которые обрабатывают поступающий через них трафик, высокую значимость, как описано в разделе «Присвоение значимости активам» Руководства оператора MaxPatrol VM.

Элементы конфигурации OpenVPN

При работе с OpenVPN могут применяться множественные поля (например, порты, маршруты, сертификаты), которые позволяют настроить несколько значений для тех или иных параметров конфигурации.

Запрос для поиска полей, которые не являются множественными:

```
filter(host.softs<OpenVPN>.Configs)
```

```
| select(@host, host.softs<OpenVPN>.Configs.Port as Port,
host.softs<OpenVPN>.Configs.Protocol as Protocol,
host.softs<OpenVPN>.Configs.Active as Active,
compactunique(host.softs<OpenVPN>.Configs.Routes) as Routes,
host.softs<OpenVPN>.Configs.Address as Address,
host.softs<OpenVPN>.Configs.Mode as Mode, host.softs<OpenVPN>.Configs.KeyMode
as KeyMode, host.softs<OpenVPN>.Configs.InterfaceID as InterfaceID,
host.softs<OpenVPN>.Configs.InterfaceType as InterfaceType,
host.softs<OpenVPN>.Configs.TcpServer as TcpServer,
host.softs<OpenVPN>.Configs.Topology as Topology,
host.softs<OpenVPN>.Configs.ClientToClient as ClientToClient,
host.softs<OpenVPN>.Configs.VerifyCert as VerifyCert,
host.softs<OpenVPN>.Configs.ClientConfigDir as ClientConfigDir,
host.softs<OpenVPN>.Configs.DuplicateCN as DuplicateCN,
host.softs<OpenVPN>.Configs.UsernameAsCN as UsernameAsCN,
host.softs<OpenVPN>.Configs.MaxClients as MaxClients)
```

Запрос для поиска клиентов OpenVPN:

```
filter(host.softs<OpenVPN>.configs)

| select(host.softs<OpenVPN>.configs.clients.clientName as client_name,
host.softs<OpenVPN>.configs.clients.Disable,
host.softs<OpenVPN>.configs.clients.,
host.softs<OpenVPN>.configs.clients.Ifconfig.ClientAddress.address,
host.softs<OpenVPN>.configs.clients.Ifconfig.ClientAddress.NetworkID,
host.softs<OpenVPN>.configs.clients.Ifconfig.ClientAddress.Prefix)

| filter(client_name) | unique() | sort(client_name)
```

Запрос для просмотра диапазонов адресов (address pools):

```
filter(host.softs<OpenVPN>.Configs)

| select(@host, host.softs<OpenVPN>.Configs.ConfigPath as ConfigPath,
host.softs<OpenVPN>.Configs.AddressPools.Name,
host.softs<OpenVPN>.Configs.AddressPools.Addresses.Operator,
host.softs<OpenVPN>.Configs.AddressPools.Addresses.Values.Address,
host.softs<OpenVPN>.Configs.AddressPools.Addresses.Values.NetworkID,
host.softs<OpenVPN>.Configs.AddressPools.Addresses.Values.Prefix,
host.softs<OpenVPN>.Configs.AddressPools.Addresses.Values.Wildcard)
```

Запрос для просмотра параметров журналирования (logging settings):

```
filter(host.softs<OpenVPN>.Configs)

| select(@host, host.softs<OpenVPN>.Configs.ConfigPath as ConfigPath,
host.softs<OpenVPN>.Configs.LoggingSettings.LogPath,
host.softs<OpenVPN>.Configs.LoggingSettings.AppendLogPath,
host.softs<OpenVPN>.Configs.LoggingSettings.Verbosity)
```

Запрос для просмотра плагинов (plugins):

```
filter(host.softs<OpenVPN>.Configs)

| select(@host, host.softs<OpenVPN>.Configs.ConfigPath as ConfigPath,
host.softs<OpenVPN>.Configs.Plugins.Path,
host.softs<OpenVPN>.Configs.Plugins.Args)
```

Запрос для просмотра параметров доменной аутентификации пользователей (custom auth, LDAP auth):

```
filter(host.softs<OpenVPN>.Configs)

| select(@host, host.softs<OpenVPN>.Configs.ConfigPath as ConfigPath,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.URL as URL,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.Domain as Domain,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.BindDN as BindDN,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.UserSearch.BaseDN as U_BaseDN,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.UserSearch.RequireGroup as
RequireGroup,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.GroupSearch.BaseDN as
G_BaseDN,
host.softs<OpenVPN>.Configs.CustomAuth<LDAPAuth>.GroupSearch.SearchString as
SearchString)
```

2.4.7. Аудит Kaspersky Security Center

Запрос данных об узлах с Kaspersky Security Center, для которых не проводился аудит

Запрос для поиска узлов с Kaspersky Security Center по портам:

```
select(@Host, host.@id, host.@audittime, Host.Endpoints<TransportEndpoint>,
Host.Endpoints<TransportEndpoint>.Port as Port,
Host.Endpoints<TransportEndpoint>.Protocol as Protocol,
Host.Endpoints<TransportEndpoint>.Status as Status,
Host.Endpoints<TransportEndpoint>.IpAddress as IpAddress)

| filter(IpAddress not in [::/0, 127.0.0.0/8] and Status = "Open" and Port in
[8060, 8061, 8080, 13000, 13291, 13299, 14000, 17000])

| select(@host, host.@audittime, countunique(Port) as cnt, compactunique(Port))

| filter(cnt > 3) | calc(TotalDays(now() - host.@AuditTime) as dur_audit)

| sort(cnt DESC)
```

> **Примечание.** Список портов, используемых Kaspersky Security Center, приведен на сайте support.kaspersky.com.

Если аудит узлов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запросы данных об узлах с Kaspersky Security Center, для которых проводился аудит

Запрос для поиска узлов с Kaspersky Security Center, для которых проводился аудит:

```
filter(host.softs<KasperskySecurityCenter>)

| select(@Host, host.softs<KasperskySecurityCenter>, host.@audittime)  |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(host.@audittime)
```

Если аудит узлов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для поиска узлов с Kaspersky Security Center по серверам баз данных:

```
filter(Host.Softs<SQLInstance>.Databases.DBName = 'KAV' or
Host.Softs<PostgreSQL>.Databases.Name = 'KAV' or
Host.Softs<Software:MySQLCore:Server>.Databases.DBName = 'KAV')

| select(@Host, compactunique(host.hostroles.role),
host.softs<KasperskySecurityCenter>, Host.@audittime) | calc(TotalDays(now() -
host.@AuditTime) as dur_audit)  | sort(host.@audittime)
```

Базы данных Kaspersky Security Center могут быть размещены на тех же серверах, что и приложение Kaspersky Security Center, или на других серверах. Если они размещены на других серверах, вы можете использовать PT NAD и по сетевым обращениям к базам данных найти серверы с приложением Kaspersky Security Center. Для серверов, на которых размещены базы данных Kaspersky Security Center, необходимо регулярно выполнять аудит.

Запрос для оценки качества аудита узлов с Kaspersky Security Center:

```
filter(Host.Softs<KasperskySecurityCenter>)

| select(@Host, compactunique(Host.Softs<KasperskySecurityCenter>.Plugins) as
Plugins, Host.Softs<KasperskySecurityCenter>.Name as Name,
Host.Softs<KasperskySecurityCenter>.Version as Version,
compactunique(Host.Softs<KasperskySecurityCenter>.Licenses) as Licenses,
compactunique(Host.Softs<KasperskySecurityCenter>.HostsInfo) as HostsInfo)
```

Для узлов, для которых в результате запроса значение хотя бы одного из параметров не заполнено, необходимо проверить выполнение задач на аудит (см. раздел 2.4).

В результате аудита создаются активы на основе данных об узлах из БД Kaspersky Security Center.

> **Примечание.** Также для сбора данных об активах вы можете создать (запустить) задачи (см. раздел 2.4) на аудит с профилем KasperskySecurityCenter_<Тип профиля>.

Запрос данных, хранящихся БД Kaspersky Security Center, о защищаемых узлах:

```
filter(Host.Softs<KasperskySecurityCenter>)
```

```
| select(@Host, Host.Softs<KasperskySecurityCenter>,
Host.Softs<KasperskySecurityCenter>.HostsInfo.ID as ID,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Status as Status,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Hostname as Hostname,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Domain as Domain,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Fqdn as Fqdn,
Host.Softs<KasperskySecurityCenter>.HostsInfo.ProtectionState as
ProtectionState)

| filter(ProtectionState and ProtectionState != "Unknown")

| join(select(@host, host.fqdn, host.hostname, host.@AuditTime) as H,
H.host.fqdn = Fqdn or H.host.hostname = Hostname)

| select(@Host, Host.Softs<KasperskySecurityCenter>, ID, Status, Hostname,
Domain, Fqdn, H.@host, H.host.@AuditTime) |  calc(TotalDays(now() -
H.host.@AuditTime) as dur_audit)

| sort(@Host, Host.Softs<KasperskySecurityCenter>, ID)
```

Запрос данных, хранящихся БД Kaspersky Security Center, о защищаемых узлах, для которых не созданы активы или не проводился аудит:

```
filter(Host.Softs<KasperskySecurityCenter>)

| select(@Host, Host.Softs<KasperskySecurityCenter>,
Host.Softs<KasperskySecurityCenter>.HostsInfo.ID as ID,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Status as Status,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Hostname as Hostname,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Domain as Domain,
Host.Softs<KasperskySecurityCenter>.HostsInfo.Fqdn as Fqdn,
Host.Softs<KasperskySecurityCenter>.HostsInfo.ProtectionState as
ProtectionState)

| filter(ProtectionState and ProtectionState != "Unknown")

| join(select(@host, host.fqdn, host.hostname, host.@AuditTime) as H,
H.host.fqdn = Fqdn or H.host.hostname = Hostname)

| select(@Host, Host.Softs<KasperskySecurityCenter>, ID, Status, Hostname,
Domain, Fqdn, H.@host, H.host.@AuditTime) | calc(TotalDays(now() -
H.host.@AuditTime) as dur_audit)

| filter(H.host.@AuditTime = null or H.@host = null)

| sort(@Host, Host.Softs<KasperskySecurityCenter>, ID)
```

Для узлов, данные о которых найдены в результате любого из двух представленных выше запросов, необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

2.4.8. Аудит «1С»

Развертывание системы «1С» осуществляется на кластере серверов (в том числе в случае, если приложение «1С» устанавливается на одном компьютере). Информация о кластере серверов «1С» хранится в БД, аудит которой осуществляется автоматически после аудита главного сервера в кластере, для которого параметр `Host.Softs<ServerOneC>.ClusterCentralServer` имеет значение `true`.

> **Примечание.** Если информация о кластере серверов «1С» указана для сервера, у которого параметр `Host.Softs<ServerOneC>.ClusterCentralServer` имеет значение `false`, то возможен один из вариантов: либо при аудите не отслеживается параметр `Host.Softs<ServerOneC>.Clusters.Servers = localhost`, либо этот кластер серверов «1С» выведен из эксплуатации.
>
> **Внимание!** Запросы, приведенные ниже, применимы только при аудите «1С», установленной на Windows.

Поиск серверов «1С»

Для работы «1С» используются порты 1540, 1541, 1542, 1545, 1560—1591.

Запрос для поиска серверов «1С» по портам:

```
select(@Host, host.@id, host.@audittime, Host.Endpoints<TransportEndpoint>,
Host.Endpoints<TransportEndpoint>.Port as Port,
Host.Endpoints<TransportEndpoint>.Protocol as Protocol,
Host.Endpoints<TransportEndpoint>.Status as Status,
Host.Endpoints<TransportEndpoint>.IpAddress as IpAddress)

| filter(IpAddress not in [::/0, 127.0.0.0/8] and Status = "Open" and (port in
[1540, 1541, 1542, 1545] or (port > 1559 and port < 1592)))

| join(filter(host.softs<serveronec>)

| select(host.@id, host.softs<serveronec>.name as onec)

| filter(onec) | unique() as S, S.host.@id = host.@id)

| select(@Host, compactunique(Port), S.onec, host.@audittime) | unique() |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) | sort(host.@audittime)
```

Если аудит активов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

На отдельных устройствах может быть не установлена «1С», но открыты перечисленные выше порты (например, если это компьютеры разработчиков или прокси-серверы). Для таких устройств тоже необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Чтобы выяснить, на каких устройствах установлена «1С», вы можете выполнить сканирование активов, данные о которых найдены в результате приведенного выше запроса, с профилем Service Discovery (см. раздел 2.2) (в параметрах задачи рекомендуется использовать расширенный список портов, который приведен по умолчанию, или как минимум указать порты `1540-1542/tcp;1545/tcp;1559-1592/tcp;`). После сканирования с профилем Service Discovery вы можете выполнить запрос:

```
filter(host.@pentesttime and
Host.Endpoints<TransportEndpoint>.service<Enterprise1CManagementService>)

| select(@host, host.@audittime) | calc(TotalDays(now() - host.@AuditTime) as
dur_audit) | sort(host.@audittime)
```

Если аудит активов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

После выполнения аудита вы можете выполнить запросы, представленные ниже.

Запрос для поиска серверов «1С»:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>.name,
compactunique(Host.Softs<ServerOneC>.Clusters.Name) as cl_names,
host.@audittime) | unique() | calc(TotalDays(now() - host.@AuditTime) as
dur_audit) | sort(host.@audittime)
```

Если аудит активов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо проверить, что активы не выведены из эксплуатации, и создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для проверки того, включен ли сервис «1С:Enterprise»:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, host.@id, Host.Softs<ServerOneC>.name, host.@audittime) |
unique()

| join(filter(WindowsHost.Softs<ServerOneC>)

| select(WindowsHost.@id, WindowsHost.Services.Name as Name,
WindowsHost.Services.State as State)

| filter(Name like '1C:Enterprise%Server%') as S, S.WindowsHost.@id = host.@id)

| select(@Host, Host.Softs<ServerOneC>.name, compactunique(S.State),
host.@audittime) | calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
sort(host.@audittime)
```

Поиск активов по информации из кластера серверов «1С»

Запрос для поиска активов по информации из кластера серверов «1С»:

```
filter(Host.Softs<ServerOneC>)

| select(@host, host.@Name, host.fqdn, host.@id,
Host.Softs<ServerOneC>.Clusters.Name as cl_name,
Host.Softs<ServerOneC>.Clusters.Servers as Server, Host.Softs<ServerOneC>)

| filter(Server) | calc(if Server
= "localhost" then "host.fqdn" else "Server" as Res_Server)

| join(filter(Host.Softs<ServerOneC>)|select(@host, host.@id, host.hostname,
host.FQDN, compactunique(Host.Softs<ServerOneC>) as onec, host.@audittime) as
H, H.host.hostname = Res_Server or H.host.FQDN = Res_Server)

| select(host.@Name, Host.Softs<ServerOneC>, cl_name, Server, H.@host, H.onec,
H.host.@audittime) | calc(TotalDays(now() - H.host.@AuditTime) as dur_audit) |
sort(host.@Name, Host.Softs<ServerOneC>, cl_name, H.@host)
```

С помощью этого запроса вы можете определить, все ли узлы, данные о которых указаны в кластере серверов «1С», существуют как активы. В запросе используются следующие параметры:

— `host.@Name` — узел, на котором развернут кластер серверов «1С»;

— `Host.Softs<ServerOneC>`, а также `H.onec` — название и версия «1С»;

— `cl_name` — имя кластера серверов «1С»;

— `Server` — имя сервера из кластера по данным «1С»;

— `H.@host` — актив, связанный с кластером серверов «1С» по значению параметра `Hostname` или `FQDN`.

Если значение параметра `H.@host` пусто или аудит активов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для поиска активов, которые не входят в кластеры серверов «1С»:

```
filter(Host.Softs<ServerOneC>)

| select(@host, host.@id, host.hostname, host.fqdn,
compactunique(Host.Softs<ServerOneC>) as onec, host.@audittime) | unique()

| join(filter(Host.Softs<ServerOneC>)

| select(@host, host.@Name, host.@id, Host.Softs<ServerOneC>.Clusters.Name as
name, Host.Softs<ServerOneC>.Clusters.Servers as Server)

| filter(Server) as Cl, Cl.Server = host.hostname or Cl.Server = host.fqdn) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit)

| select(@host, host.@audittime, dur_audit, onec, Cl.name, Cl.Server) |
sort(Cl.name, host.@audittime, @host)
```

С помощью этого запроса вы можете определить, найдена ли в результате аудита информация о кластерах серверов «1С», в которых указаны активы с «1С». Запрос применим, если такие активы существуют и если имеются данные обо всех кластерах серверов «1С». В запросе используются следующие параметры:

— `host.@Name` — актив, на котором найдена «1С»;

— `onec` — список экземпляров «1С» на активе;

— `Cl.name` — имя кластера серверов «1С», в котором по значению параметра `Hostname` или `FQDN` найдена запись об активе.

Если значение параметра `Cl.name` пусто, то возможен один из вариантов: либо в результате аудита не найдены данные о кластере серверов «1С», либо не найден главный сервер кластера, который является источником данных о кластере. Необходимо найти не выявленный кластер серверов «1С». Если найдены выведенные из эксплуатации активы, необходимо удалить их.

Оценка качества аудита серверов «1С»

Оценка качества аудита включает оценку его актуальности и полноты. Для оценки актуальности аудита серверов «1С» вы можете выполнить запросы, представленные выше. Актуальность аудита обеспечена, если значение параметра `dur_audit` не превышает 14, то есть аудит проводился не ранее чем 14 дней назад.

Для оценки полноты аудита вы можете выполнить запросы, представленные ниже.

Запрос 1:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Name as Name,
Host.Softs<ServerOneC>.Version as Version,
Host.Softs<ServerOneC>.ClusterCentralServer as ClusterCentralServer,
compactunique(Host.Softs<ServerOneC>.Clusters.Name) as cl_name,
compactunique(Host.Softs<ServerOneC>.Clusters.Servers) as Servers,
compactunique(Host.Softs<ServerOneC>.Rmngrs.Port) as Rmngrs,
compactunique(Host.Softs<ServerOneC>.Ragents.Port) as Ragents,
compactunique(Host.Softs<ServerOneC>.Rphosts.Port) as Rphosts)

| filter(ClusterCentralServer = true or Servers contains "localhost") |
sort(@Host)
```

Запрос 2:

```
filter(Host.Softs<ServerOneC>.Clusters)

| select(@Host, Host.Softs<ServerOneC>.Clusters.Name as Name,
compactunique(Host.Softs<ServerOneC>.Clusters.InfoBases.Users) as Users)

| filter(Name) | sort(Users)
```

Полнота аудита обеспечена, если в результате выполнения запроса 1 заполнены значения всех параметров, кроме `Ragents` (оно может быть как заполненным, так и пустым), а в результате запроса 2 заполнены все значения параметра `Users` (то есть получены данные о пользователях информационных баз данных).

Параметры серверов «1С»

Запрос для просмотра основных параметров серверов «1С»:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>,
Host.Softs<ServerOneC>.ClusterCentralServer as ClusterCentralServer,
Host.Softs<ServerOneC>.Vendor as Vendor, Host.Softs<ServerOneC>.Name as Name,
Host.Softs<ServerOneC>.InstallPath as InstallPath,
Host.Softs<ServerOneC>.Version as Version, Host.Softs<ServerOneC>.Architecture
as Architecture, Host.Softs<ServerOneC>.OsFamily as OsFamily,
compactunique(Host.Softs<ServerOneC>.Ports) as Ports,
Host.Softs<ServerOneC>.UninstallKey as UninstallKey)
```

Параметр `Host.Softs<ServerOneC>.ClusterCentralServer` определяет, является ли сервер главным в кластере серверов «1С» (в этом случае параметр принимает значение `true`).
В запросах, приведенных ниже, используются данные серверов, которые являются (или ранее являлись) главными в кластере.

Запрос для поиска портов, через которые клиентские приложения подключаются к информационным базам, указанным в кластере серверов «1С»:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, host.@id, Host.Softs<ServerOneC>,
Host.Softs<ServerOneC>.Rmngrs, Host.Softs<ServerOneC>.Rmngrs.ClusterUUID as
ClusterUUID, Host.Softs<ServerOneC>.Rmngrs.Port as Port,
compactunique(Host.Softs<ServerOneC>.Rmngrs.RangePorts.Operator) as Operator,
compactunique(Host.Softs<ServerOneC>.Rmngrs.RangePorts.Values) as Values)

| filter(Port)

| join(filter(Host.Softs<ServerOneC>.Clusters)

| select(Host.Softs<ServerOneC>.Clusters.UUID as UUID,
Host.Softs<ServerOneC>.Clusters.Name as Name, host.@id) as Cl, Cl.host.@id =
host.@id and Cl.UUID = ClusterUUID)

| select(@Host, Host.Softs<ServerOneC>, Port, Operator, Values, Cl.Name) |
unique() | sort(@Host, Port, Host.Softs<ServerOneC>)
```

Результат запроса будет содержать значения параметров:

— `@Host` — актив с главным сервером в кластере серверов «1С»;

— `Host.Softs<ServerOneC>` — название и версия «1С»;

— `Port` — порт подключения;

— `Cl.Name` — имя кластера серверов «1С», к которому осуществляется подключение при обращении к серверу через порт, указанный как значение параметра `Port`.

Запрос для поиска портов, через которые консоль управления кластером серверов «1С» подключается к кластеру:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Ragents.Port as
Port, compactunique(Host.Softs<ServerOneC>.Ragents.RangePorts.Operator) as
Operator, compactunique(Host.Softs<ServerOneC>.Ragents.RangePorts.Values) as
Values)

| filter(Port) | sort(@Host, Port, Host.Softs<ServerOneC>)
```

Примечание. В этом запросе используются данные файлов ragent.exe и rphost.exe.

Результат запроса будет содержать значения параметров:

— `@Host` — актив с главным сервером в кластере серверов «1С»;

— `Host.Softs<ServerOneC>` — название и версия «1С»;

— `Port` — порт подключения;

— `Operator` — параметр, управляющий диапазоном портов процесса rphost.exe;

— `Values` — диапазон портов процесса rphost.exe.

Кластеры серверов «1С» и информационные базы

Кластеры серверов «1С» содержат данные о серверах и информационных базах.

Запрос для поиска серверов «1С»:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Clusters,
Host.Softs<ServerOneC>.Clusters.UUID as UUID,
Host.Softs<ServerOneC>.Clusters.Name as Name,
compactunique(Host.Softs<ServerOneC>.Clusters.Servers) as Servers)

| filter(UUID) | sort(@Host)
```

Запрос для поиска информационных баз:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Clusters as
Clusters, Host.Softs<ServerOneC>.Clusters.InfoBases.UUID as UUID,
Host.Softs<ServerOneC>.Clusters.InfoBases.Name as Name,
Host.Softs<ServerOneC>.Clusters.InfoBases.DBHost as DBHost,
```

```
Host.Softs<ServerOneC>.Clusters.InfoBases.DBName as DBName,
Host.Softs<ServerOneC>.Clusters.InfoBases.DBType as DBType,
Host.Softs<ServerOneC>.Clusters.InfoBases.DBUser as DBUser)

| filter(Clusters) | unique() | sort(@Host, Clusters, Name)
```

Информационные базы содержат базы данных и сведения об узлах, на которых размещены базы данных. На основе этих сведений вы можете проверить, был ли выполнен аудит узлов, с помощью запроса:

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Clusters as
Clusters, Host.Softs<ServerOneC>.Clusters.InfoBases.DBHost as DBHost)

|filter(DBHost and DBHost != "localhost")

|select(@Host, DBHost) | unique()

| join(select(@host, host.hostname, host.@audittime) as DB, DB.host.hostname =
DBHost)

| select(@Host, DBHost, DB.@host, DB.host.@audittime) | calc(TotalDays(now() -
DB.host.@AuditTime) as dur_audit) | sort(DB.host.@AuditTime, DBHost)
```

Если аудит узлов, данные о которых найдены в результате запроса, не проводился вообще (`dur_audit = null`) или проводился больше 14 дней назад (`dur_audit > 14`), то необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Запрос для поиска пользователей, которые могут подключаться к информационным базам (с указанием ролей пользователей):

```
filter(Host.Softs<ServerOneC>)

| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Clusters as
Clusters, Host.Softs<ServerOneC>.Clusters.InfoBases.Users as Users,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.OSUser as OSUser,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.Domain as Domain,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.FullUserName as FullUserName,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.LastChanged as LastChanged,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.IsAdmin as IsAdmin,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.LoginAuth as LoginAuth,
compactunique(Host.Softs<ServerOneC>.Clusters.InfoBases.Users.Roles) as Roles,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.ID as ID,
Host.Softs<ServerOneC>.Clusters.InfoBases.Users.Name as Name)

| filter(Clusters) | sort(@Host, Clusters, Users)
```

Запрос для поиска ролей пользователей:

```
filter(Host.Softs<ServerOneC>)
```

```
| select(@Host, Host.Softs<ServerOneC>, Host.Softs<ServerOneC>.Clusters as
Clusters, Host.Softs<ServerOneC>.Clusters.InfoBases,
Host.Softs<ServerOneC>.Clusters.InfoBases.Roles.Name as Name,
Host.Softs<ServerOneC>.Clusters.InfoBases.Roles.Objects.Name as Objects_name,
compactunique(Host.Softs<ServerOneC>.Clusters.InfoBases.Roles.Objects.Privilege
s) as Privileges)

| filter(Clusters) | sort(@Host, Clusters, Name, Objects_name)
```

2.4.9. Аудит YouTrack

После сканирования активов с профилем Service Discovery (см. раздел 2.2) (с указанием портов 443 и 80) вы можете выполнить запрос для поиска активов с названием `YouTrack`:

```
filter(Host.Endpoints<TransportEndpoint>.Service.SoftBanners.Data = 'YouTrack')

| select(@host, host.@PentestTime, host.@audittime)

| calc(TotalDays(now() - host.@AuditTime) as dur_audit)

| calc(TotalDays(now() - host.@PentestTime) as dur_pentest)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

Для аудита YouTrack необходимо выполнить задачи с профилем Unix Audit (для получения информации об узле) и с профилем Web API Audit (для получения информации о ПО и заполнения значений параметра `host.softs<YoutrackServer>`).

> **Примечание.** Если аудит был проведен, но нет возможности определить, с каким именно профилем, то вы можете просмотреть параметры задач и убедиться, что задачи с обоими профилями выполняются корректно.

Если выполнен аудит с профилем Unix Audit, но не выполнен аудит с профилем Web API Audit, то значение параметра `host.softs<YoutrackServer>` не заполнено, но получена информация о выполняемых в системе процессах. Поэтому вы можете найти активы, на которых выполняются процессы YouTrack (то есть в командной строке (shell) упоминается `youtrack`), с помощью запроса:

```
filter(Computer.Processes[CommandLine like "%youtrack%" and Name not in
["chrome.exe", "firefox.exe", "browser.exe", "msedge.exe"]])

| select(@Computer, Computer.@id, Computer.Processes.Name as Name,
Computer.Processes.Path as Path, Computer.Processes.CommandLine as CommandLine,
Computer.Processes.Owner as Owner, Computer.Processes.PID as PID,
Computer.Processes.ParentPID as ParentPID)

| filter(CommandLine like "%youtrack%")

| join(filter(host.softs<YoutrackServer>)
```

```
| select(host.@id, host.softs<YoutrackServer> as soft) as S, S.host.@id =
Computer.@id)

| select(@Computer, S.soft, Name, Path, CommandLine, Owner, PID, ParentPID) |
sort(@Computer, Name)
```

Для активов с YouTrack, у которых по результатам запроса значение параметра `S.soft` пусто, необходимо создать (запустить) задачи (см. раздел 2.4) на аудит с профилем Web API Audit и настроить их расписание. Когда аудит с профилем Web API Audit будет выполнен, для поиска активов с YouTrack и проверки актуальности выполнения аудита вы можете использовать запрос:

```
filter(host.softs<YoutrackServer>)

| select(@host, host.@PentestTime, host.@audittime)

| calc(TotalDays(now() - host.@AuditTime) as dur_audit)

| calc(TotalDays(now() - host.@PentestTime) as dur_pentest)
```

Для активов, аудит которых не проводился вообще (`dur_audit = null`) или проводился больше месяца назад (`dur_audit > 30`), необходимо создать (запустить) задачи на аудит (см. раздел 2.4) и настроить их расписание.

2.4.10. Оценка качества аудита

Вы можете найти узлы, для которых не был выполнен аудит или аудит которых был выполнен больше месяца назад, а также оценить достаточность информации, полученной по результатам аудита.

Запрос для поиска узлов, для которых не был выполнен аудит:

```
select(@host, Host.@audittime) | filter(Host.@audittime = null)
```

Данные о таких узлах могут быть найдены по разным причинам, например в результате выполнения других задач аудита. Поэтому рекомендуется выполнять аудит в соответствии с порядком следования разделов в этом документе.

Примечание. Параметр `@AuditTime` применим только для узлов.

Запрос для поиска узлов, аудит которых был выполнен больше месяца назад:

```
select(@host, host.@id, compactunique(host.@groups) as groups, Host.@audittime) |
filter(Host.@audittime) | calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
filter(dur_audit > 30) | SORT(dur_audit DESC)
```

Данные о таких узлах могут быть найдены в нескольких случаях:

— **Если задача на аудит узла есть, но не настроено ее расписание.** Для каждого такого узла необходимо найти задачу на аудит и настроить ее расписание. Подробнее в разделе «Поиск и фильтрация задач» Руководства оператора MaxPatrol VM. Рекомендуется осуществлять поиск по IP-адресу узла, а также по параметрам `host.@id` или `host.@groups`. Вы можете фильтровать результаты по этим параметрам.

Примечание. Вы можете запросить отчет обо всех задачах. Подробнее в Руководстве разработчика MaxPatrol VM.

— **Если задача на аудит узла есть, настроено ее расписание, но она завершилась с ошибками.** Задача может завершиться с ошибками по следующим причинам:

- Сетевая недоступность узла в середине маршрута из-за параметров межсетевого экрана. В журналах событий фиксируется информация о невозможности подключения.
- Запрет подключения на самом узле (например, в период аудита был выключен узел или служба, к которой осуществлялось подключение). В журналах событий фиксируется информация о невозможности подключения.
- Некорректно выбранный профиль аудита. Например, для аудита узла с ОС семейства Unix был выбран профиль Windows Audit. В этом случае необходимо определить ОС узла, удалить старый актив, после чего запустить задачу на аудит.
- Учетная запись, используемая для аудита, не существует. В журналах событий фиксируется успешное подключение через порт и ошибка, описывающая некорректные атрибуты аутентификации.
- Учетная запись, используемая для аудита, не имеет нужных прав. В журналах событий фиксируется успешное подключение через порт и ошибка, описывающая, что прав недостаточно.

— **Если задачи на аудит узла нет.** Необходимо создать (запустить) задачу на аудит и настроить ее расписание.

Запрос для поиска узлов, для которых не был выполнен аудит или аудит которых был выполнен больше месяца назад:

```
select(@host, host.@id, compactunique(host.@groups) as groups, Host.@audittime) |
calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
filter( Host.@audittime = null or dur_audit > 30) | calc(if Host.@audittime = null
then "no audit" else "old audit" as for_sorting) | SORT(for_sorting ASC, dur_audit
DESC, @host ASC) | group(for_sorting, COUNT(*))
```

Запрос для поиска узлов, для которых выполнен аудит, но получено недостаточно информации для последующей обработки:

```
select(
    @host, Host.@fulltype, host.OsName, host.OsVersion,
    compactunique(host.Interfaces.L3Settings.Address) as inter,
    compactunique(host.ARPTable) as arp,
    compactunique(host.User.Name) as user,
    compactunique(Host.RoutingTables) as routes,
    host.@audittime,
    compactunique(host.hostroles.role) as roles) |
filter(host.@audittime and (
    Host.@fulltype = "Core.Host" or host.OsName = null
    OR host.OsVersion = null OR host.OsVersion = "None"
    OR inter = null OR arp = null OR routes = null OR (user = null
      AND not roles contains "Domain Controller"))) |
```

```
calc(TotalDays(now() - host.@AuditTime) as dur_audit) |
filter(dur_audit <= 30) | unique() | sort(@host) | group(host.OsName)
```

В этом запросе достаточность информации определяется с помощью критериев, описанных в таблице ниже. Результат запроса будет содержать данные об узлах, для которых не выполнен хотя бы один критерий.

Таблица 2. Критерии для оценки достаточности информации, полученной по результатам аудита

Критерий	Выполнение критерия
`Host.@fulltype != "Core.Host"`	Узлы с классом `Core.Host` отсутствуют. По результатам аудита значение параметра `Host.@fulltype` меняется с `Core.Host` на специфичный класс в зависимости от узла. **Примечание.** Вы можете использовать дополнительный критерий `Host.@fulltype != "Core.Computer"` — это промежуточный класс для объединения атрибутов классов-наследников
`host.OsName and host.OsVersion and host.OsVersion != "None"`	Значения параметров `OsName` (ОС) и `OsVersion` (версия ОС) не пусты
`host.Interfaces.L3Settings.Address and host.ARPTable and Host.RoutingTables`	Заполнены таблицы ARP, сетевые интерфейсы и таблицы маршрутизации. **Примечание.** Выполнение критерия необходимо для построения сетевой топологии
`host.User.Name`	Заполнены данные локальных учетных записей. **Примечание.** Не используется для контроллеров домена

3. Оценка полноты сетевой топологии и проверка достижимости целевых активов

После сканирования активов (см. раздел 2) необходимо оценить полноту сетевой топологии и проверить достижимость целевых активов.

Оценка полноты сетевой топологии

Для оценки полноты сетевой топологии необходимо, используя инструмент анализа топологии MaxPatrol VM, оценить следующие критерии:

— соотношение количества активов в MaxPatrol VM и активов, отображаемых на топологии;

— количество компонент связности;

— количество активов, принадлежащих одной компоненте связности.

Информация о сетевой топологии приведена в разделе «Топология сети. Работа с картой сети» Руководства оператора MaxPatrol VM.

Хорошей считается топология, в которой есть только одна компонента связности, то есть топология не разбита на несколько не связанных между собой графов, и в которой содержится информация обо всех имеющихся активах. Построение топологии обеспечивается за счет полного сканирования всей IT-инфраструктуры и конфигурации активов.

► Чтобы оценить полноту сетевой топологии:

1. В главном меню нажмите .
2. Нажмите **Анализ топологии**.

   Вы можете оценить полноту сетевой топологии, проанализировав на странице **Результаты анализа** значения описанных выше критериев.

Проверка достижимости целевых активов

После оценки сетевой топологии необходимо проверить сетевую достижимость целевых активов, то есть активов, связанных с недопустимыми событиями. Информация о сетевой достижимости активов и инструкции по ее расчету приведены в разделе «Достижимость между активами» Руководства оператора MaxPatrol VM. При расчете достижимости в раскрывающемся списке **Источники** необходимо выбрать значение **Любые активы и сетевые адреса**, в раскрывающемся списке **Цели** — значение **Активы и группы**.

Для сетевых устройств с хорошей достижимостью отображаются правила маршрутизации, преобразования адресов (NAT), списки управления доступом (ACL) и активы из других сетевых сегментов. Если для межсетевого экрана не отображаются какие-либо данные, например списки управления доступом (ACL), то это означает плохую достижимость. В этом случае необходимо проверить полноту информации об активе с помощью PDQL-запросов и повторить сканирование.

positive technologies

Positive Technologies — лидер в области результативной кибербезопасности. Компания является ведущим разработчиком продуктов, решений и сервисов, позволяющих выявлять и предотвращать кибератаки до того, как они причинят неприемлемый ущерб бизнесу и целым отраслям экономики. Наши технологии используют более 4000 организаций по всему миру.
Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (MOEX: POSI), у нее более 205 тысяч акционеров.