How-To instruction on product deployment and operation for Cloud Providers

Kaspersky Security for Virtualization Light Agent

Table of Contents

1 Kaspersky Security для виртуальных сред Легкий агент 3

2 Глоссарий 4

3 Архитектура программы 5
3.1 Комплект поставки 5
3.2 Компоненты программы 5
3.3 Аппаратные и программные требования KSVLA 6
3.4 Сетевое взаимодействие 7

4 Рекомендуемый порядок развертывания защиты 9

5 Сценарии использования KSVLA в режиме multitenancy 10
5.1 "Полный клиент" 10
5.2 "Упрощенный клиент" 10

6 Сценарии развертывания решения KSVLA 12

7 Сценарий "Полный клиент" 22
7.1 Создание клиента 22
7.2 Получение информации о клиенте 23
7.3 Удаление клиента 24
7.4 Регистрация виртуальной машины 24
7.5 Получение информации о защищаемых машинах клиента 24
7.6 Отмена регистрации виртуальной машины 25
7.7 Включение защиты клиента 25
7.8 Выключение защиты клиента 25
7.9 Работа с отчетами 25

8 Сценарий "Упрощенный клиент" 27
8.1 Создание клиента 27
8.2 Включение защиты клиента 27
8.3 Другие запросы 27

9 Примеры развертывания KSVLA в режиме multitenancy на пользовательских инфраструктурах 28
9.1 Сценарий "Полный клиент" 28
9.2 Сценарий "Упрощенный клиент" 29

1 Kaspersky Security для виртуальных сред Легкий агент

Программа Kaspersky Security для виртуальных сред Легкий агент (далее также "решение", "программа", "KSVLA") представляет собой интегрированное решение, обеспечивающее комплексную защиту виртуальных машин от различных видов информационных угроз, сетевых и мошеннических атак. Решение содержит новейшие технологии и обеспечивает многоуровневую защиту виртуальных и гибридных инфраструктур. Kaspersky Security для виртуальных сред Легкий агент позволяет достичь высокой степени защиты и консолидации виртуальных машин в виртуальной среде, поддерживает последние технические решения крупных поставщиков решений виртуализации и облачных решений: VMware vSphere, Microsoft Hyper-V, Citrix Hypervisor, KVM, OpenStack, Clouds и т.д.

В состав решения KSVLA входит выделенная виртуальная машина (SVM), на которой размещаются базы данных вредоносного ПО. SVM проверяет на наличие вирусов и других вредоносных программ фрагменты файлов, которые присылают на проверку Легкие агенты, установленные на виртуальных машинах. Благодаря интеллектуальной оптимизации проверки, например, использованию общего кэша и устранению избыточной информации, SVM сокращает объем обрабатываемых данных и количество выполняемых действий, значительно снижая число операций ввода-вывода в секунду, циклов ЦП, потребности в памяти и дисковом пространстве. Это помогает достичь высоких коэффициентов консолидации, что делает вложения в проекты виртуализации более рентабельными.

Управление программой происходит средствами Kaspersky Security Center (KSC), с помощью плагинов управления программой.

Для получения информации о виртуальной среде и получения виртуальными машинами адреса SVM используется компонент программы Сервер интеграции.

Для взаимодействия со службами Лаборатории Касперского в рамках лицензирования, использования KSN, обновления баз и модулей программ используется KSC в роли прокси-сервера. Такая архитектура позволяет создать изоляцию виртуальной среды от внешних сетей.

Подробнее:

https://www.kaspersky.ru/small-to-medium-business-security/virtualization-hybrid-cloud

https://www.kaspersky.ru/small-to-medium-business-security/virtualization-light-agent

https://support.kaspersky.com/KSVLA/5.2/ru-RU/145134.htm

2 Глоссарий

KSC (Kaspersky Security Center) – инструмент для централизованного управления комплексной системой защиты.

SVM – Secure virtual machine, виртуальная машина защиты. Виртуальная машина на гипервизоре, на которой установлен Сервер защиты, компонент программы KSVLA.

Легкий агент – компонент программы KSVLA. Устанавливается на каждую виртуальную машину, которую требуется защищать.

Сервер интеграции – компонент программы KSVLA. Осуществляет взаимодействие между компонентами программы KSVLA и виртуальной инфраструктурой.

Консоль Сервера интеграции – компонент программы KSVLA. Графический интерфейс для управления Сервером интеграции.

Mmc-плагин управления – компонент программы KSVLA, предоставляющий интерфейс для управления программой KSVLA через Консоль администрирования KSC.

Веб-плагин управления – компонент программы KSVLA, предоставляющий интерфейс для управления программой KSVLA через KSC Web Console.

Multitenancy – режим работы программы, при котором один экземпляр программы, установленный провайдером, обслуживает множество клиентов.

vKSC (Виртуальный Сервер администрирования) – компонент программы KSC, предназначенный для управления сетью организации-клиента.

3 Архитектура программы

3.1 Комплект поставки

В комплект поставки входят:

- Файл ksvla-components_X.X.X.X_mlg.exe для запуска мастера установки mmc-плагинов управления (для управления программой KSVLA через Консоль администрирования KSC), Сервера интеграции и Консоли Сервера интеграции. Мастер также используется для распаковки файлов для установки Легкого агента для Windows и Легкого агента для Linux.
- Образы SVM (виртуальной машины защиты) с установленным Сервером защиты и прочим ПО, необходимым для работы SVM. Поставляются в виде нескольких архивов для разных типов гипервизоров. В каждом архиве находится файл описания образа ksvla-svm_manifest_X.X.X.X.xml для Мастера управления SVM и сам образ в нужном для гипервизора формате:
    - ksvla-svm_microsoft-hyper-v_X.X.X.X_mlg.vhdx для развертывания на Microsoft Windows Server (Hyper-V).
    - ksvla-svm_citrix-hypervisor_X.X.X.X_mlg.xva для развертывания на Citrix Hypervisor.
    - ksvla-svm_vmware-vsphere_X.X.X.X_mlg.ova для развертывания на VMware ESXi.
    - ksvla-svm_kvm_based_X.X.X.X_mlg.qcow2 для развертывания на KVM, Proxmox VE, Р-Виртуализация, HUAWEI FusionCompute CNA, Nutanix AHV.
- MIB-файл KSVLA-MIB.txt, который может быть использован для получения сведений о состоянии SVM с помощью системы SNMP-мониторинга.
- Архивы для установки веб-плагинов управления (для управления программой KSVLA через KSC Web Console):
    - ksvla-web_plugin_wla_X.X.X.X_mlg.zip для установки веб-плагина Легкого агента для Windows.
    - ksvla-web_plugin_lla_X.X.X.X_mlg.zip для установки веб-плагина Легкого агента для Linux.
    - ksvla-web_plugin_svm_X.X.X.X_mlg.zip для установки веб-плагина Сервера защиты.

3.2 Компоненты программы

SVM

Компонент KSVLA Сервер защиты поставляется в виде образа SVM (виртуальная машина защиты, secure virtual machine). В зависимости от используемой платформы виртуализации необходимо выбрать образ, соответствующий платформе. Сервер защиты, установленный на SVM, выполняет следующие функции:

- Проверяет на наличие угроз файлы, которые присылают Легкие агенты;
- Получает обновления баз и модулей программы из хранилища Сервера администрирования Kaspersky Security Center;
- Осуществляет работу с лицензионными ключами и контроль лицензионных ограничений.

Легкий агент

Компонент KSVLA Легкий агент устанавливается на каждую виртуальную машину, которую необходимо защитить. Легкий агент выполняет следующие функции:

- Защищает виртуальную машину, на которой он установлен, от вирусов и других угроз.
- Контролирует работу программ и устройств на защищенной виртуальной машине, а также отслеживает изменения в операционной системе виртуальной машины.

Сервер интеграции

Компонент KSVLA Сервер интеграции осуществляет взаимодействие между компонентами KSVLA и виртуальной инфраструктурой. Сервер интеграции используется для выполнения следующих задач:

- Развертывание, удаление, изменение конфигурации SVM.
- Получение информации о защищаемой инфраструктуре и передача этой информации SVM.
- Получение Легкими агентами списка доступных для подключения SVM.
- Развертывание и использование KSVLA в режиме multitenancy.

В зависимости от сценария для управления Сервером интеграции можно использовать REST API или Консоль Сервера интеграции.

Плагины управления (mmc- и веб-плагины)

Плагины управления обеспечивают интерфейс для управления решением KSVLA через Kaspersky Security Center. Вы можете использовать Консоль администрирования KSC или KSC Web Console.

Подробнее:

https://support.kaspersky.com/ksvla/5.2/ru-RU/67380.htm

3.3 Аппаратные и программные требования KSVLA

Аппаратные и программные требования KSVLA подробно описаны в справке: https://support.kaspersky.com/KSVLA/5.2/ru-RU/64743.htm

Аппаратные и программные требования KSC подробно описаны в справке: https://support.kaspersky.com/KSC/13.2/ru-RU/96255.htm

Легкий агент для Windows в большинстве сценариев потребляет 200–300 МБ оперативной памяти. Легкий агент для Linux в большинстве сценариев потребляет 150–200 МБ оперативной памяти.

В случае запуска задач сканирования по требованию (в том числе, по расписанию) количество потребляемой памяти может увеличиваться, но память освобождается по завершении сканирования. Также в случае обнаружения вредоносной активности или попыток шифрования данных вредоносными программами количество потребляемой памяти также может быть увеличено.

Алгоритмы решения KSVLA по оптимизации запусков задач сканирования и обновления в автоматическом режиме распределяют нагрузку по всем Легким агентам, предотвращая штормы исполнения задач.

Подробнее:

https://support.kaspersky.com/KSVLA/5.2/ru-RU/67380.htm

https://support.kaspersky.com/KSVLA/5.2/ru-RU/145134.htm (KSVLA)

https://support.kaspersky.com/KSC/13.2/ru-RU/3396.htm (KSC)

3.4 Сетевое взаимодействие

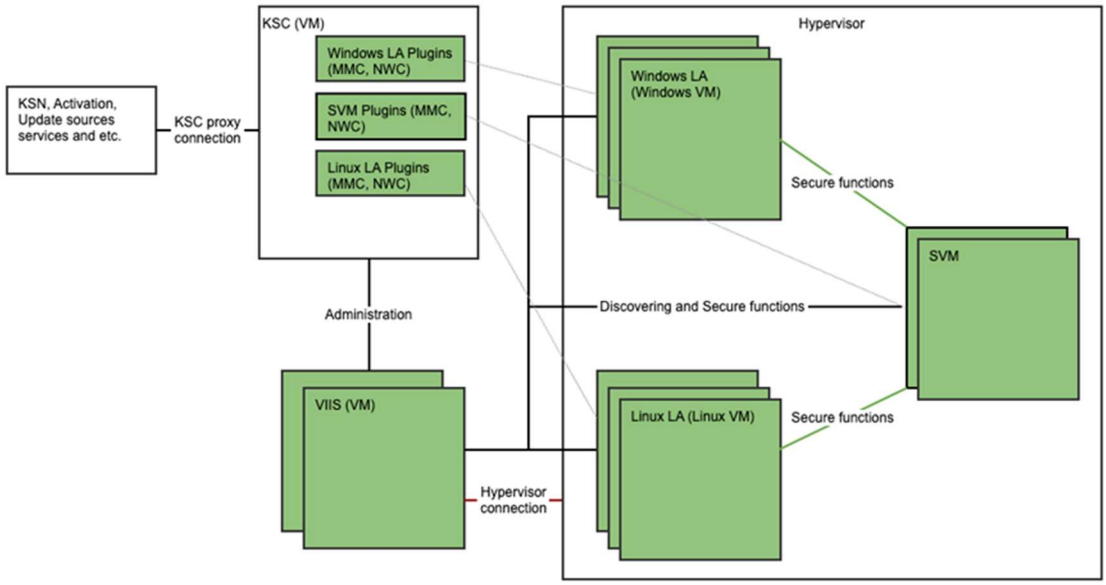
Для полноценной работы решения необходимо обеспечить следующие сетевые взаимодействия:

- Взаимодействие SVM с защищаемыми этой SVM виртуальными машинами.
- Взаимодействие SVM и Легких агентов с Сервером интеграции.
- Взаимодействие SVM и Легких агентов с KSC.
- Для использования решения в режиме multitenancy – взаимодействие Сервера интеграции с KSC.
- Взаимодействие Сервера интеграции с гипервизорами или средствами управления виртуальными средами.

Настройка портов для работы KSVLA подробна описана в справке: https://support.kaspersky.com/KSVLA/5.2/ru-RU/133882.htm

Настройка портов для работы KSC подробна описана в справке: https://support.kaspersky.com/KSC/13.2/ru-RU/158830.htm

Учетные записи для установки и работы решения подробно описаны в справке: https://support.kaspersky.com/KSVLA/5.2/ru-RU/85889.htm



Изоляция виртуальной среды от внешних сетей:

Для взаимодействия со службами Лаборатории Касперского в рамках лицензирования, использования KSN, обновления баз и модулей программ используется KSC в роли прокси-сервера. Такая архитектура позволяет создать изоляцию виртуальной среды от внешних сетей:

- Чтобы предотвратить обращение KSC к серверам активации, необходимо использовать для активации KSVLA файлы ключа, а не коды активации.
- Чтобы предотвратить обращение KSC к серверам KSN (см. подробнее https://www.kaspersky.ru/ksn), необходимо использовать Kaspersky Private Security Network (KPSN, см. подробнее https://www.kaspersky.ru/enterprise-security/private-security-network). Также можно выключить использование KSN.
- Чтобы предотвратить обращение KSC к серверам обновления, необходимо использовать утилиту Kaspersky Updater Utility (KUU, см. подробнее https://support.kaspersky.ru/updater4). Утилита для загрузки обновлений баз и модулей из указанного источника обновлений, может быть использована на

отдельно стоящем узле, а KSC может быть настроен на использование этого узла, как источника обновления.

Шифрование соединений между Агентами и SVM:

Если необходимо обеспечить безопасность каналов передачи данных между SVM и Легкими агентами средствами решения KSVLA, можно настроить защиту соединения с помощью SSL-шифрования. Рекомендуется учесть, что защита соединения с помощью шифрования может снижать производительность работы решения KSVLA.

Подробнее:

https://support.kaspersky.com/KSVLA/5.2/ru-RU/102192.htm

4 Рекомендуемый порядок развертывания защиты

1. Установить Kaspersky Security Center (см. подробнее: https://support.kaspersky.com/KSC/13.2/ru-RU/171268.htm). Сервер администрирования должен иметь доступ ко всем защищаемым виртуальным машинам для развертывания Легких агентов и управления ими. Сервер администрирования можно установить как на отдельную виртуальную машину, так и на физический сервер. Подробнее см. в справке KSC: https://support.kaspersky.com/KSC/13.2/ru-RU/3396.htm
2. Установить компоненты управления KSVLA: mmc-плагины управления (плагин для Сервера защиты, плагин для Легкого агента для Windows, плагин для Легкого агента для Linux), Сервер интеграции и Консоль Сервера интеграции. Сервер интеграции может быть установлен отдельно от KSC (как на физическую, так и на виртуальную машину). Установка выполняется с помощью мастера. Мастер установки компонентов управления также создает и распространяет на защищаемые виртуальные машины инсталляционные пакеты Легкого агента для операционных систем Windows и Linux.
3. Создать политики KSC для Легких агентов (https://support.kaspersky.com/KSVLA/5.2/ru-RU/74263.htm, https://support.kaspersky.com/KSVLA/5.2/ru-RU/103975.htm) и SVM (https://support.kaspersky.com/KSVLA/5.2/ru-RU/129817.htm).
4. Развернуть необходимое количество SVM (варианты развертывания в зависимости от базовых сценариев инфраструктуры описаны ниже). В процессе развертывания необходимо настроить сетевые параметры развертывания. Подключение SVM к сети должно обеспечивать базовые потребности Сервера защиты:
    - соединение с Kaspersky Security Center (для получения обновлений, политик, задач, пересылки событий);
    - соединение с Сервером интеграции;
    - соединение со всеми защищаемыми виртуальными машинами.
5. Создать задачу активации для SVM (https://support.kaspersky.com/KSVLA/5.2/ru-RU/86947.htm).
6. Установить Легкие агенты на защищаемые виртуальные машины. Если вы настроили автоматическое перемещение виртуальных машин с установленными Легкими агентами, виртуальные машины после установки Легких агентов будут перемещены в группу администрирования KSC с созданной на предыдущих шагах политикой KSC. Политика KSC распространит необходимые параметры для соединения Легкого агента с SVM. В результате Легкий агент при подключении к SVM автоматически получит обновления, информацию о лицензии и некоторые параметры защиты.

5 Сценарии использования KSVLA в режиме multitenancy

В сценариях работы в режиме multitenancy организация-провайдер предоставляет услуги по виртуализации для организаций-клиентов. Провайдер хочет обеспечить защиту виртуальных машин своих клиентов, но установка отдельных KSC, Сервера интеграции, SVM для каждого клиента его не устраивает из-за избыточного потребления ресурсов. Решение KSVLA предлагает собственные подходы к автоматизации развертывания и управления защитой клиентов в режиме multitenancy.

Предусмотрены следующие сценарии использования программы в режиме multitenancy.

5.1 "Полный клиент"

"Полный клиент" – основной сценарий для работы в режиме multitenancy. В этом сценарии структура защиты клиентов создается с помощью REST API запросов к Серверу интеграции. Сценарий построен на предоставлении каждому клиенту Виртуального сервера администрирования KSC (далее также "vKSC", см. подробнее https://support.kaspersky.com/KSC/13.2/ru-RU/92246.htm).

Благодаря использованию vKSC сценарий "Полный клиент" обладает рядом преимуществ:

- Происходит экономия ресурсов клиента, поскольку нет необходимости устанавливать каждому клиенту новый KSC и Сервер интеграции.
- Клиент получает возможность использовать схожую с обычным KSC функциональность. Каждый vKSC имеет собственную группу нераспределенных устройств, собственные наборы отчетов, выборки устройств и события, инсталляционные пакеты, правила перемещения и т. д.
- Клиенты изолированы друг от друга, имеют возможность работы только в рамках своего vKSC.

В рамках данного сценария:

- Провайдер создает для каждого из клиентов собственный изолированный vKSC.
- Провайдер имеет возможность включать и отключать защиту своих клиентов с помощью одного запроса к REST API Сервера интеграции. Например, провайдер может отключать защиту в случае неоплаты клиентом услуг защиты.
- Клиент работает со своими виртуальными машинами в рамках своего vKSC, имеет возможность управлять защитой своих виртуальных машин с помощью политик для Легкого агента.
- Провайдер может получать отчеты для мониторинга использования защиты виртуальных машин своих клиентов. Эти отчеты могут быть использованы для биллинга клиентов.

5.2 "Упрощенный клиент"

Этот сценарий используется, если у провайдера уже настроена работа программы в режиме multitenancy и провайдер не хочет в ней что-то менять, но хочет добавить возможность получать отчеты о защите виртуальных машин своих клиентов. Инфраструктура у провайдера может быть организована на основе vKSC или на основе групп администрирования.

В рамках данного сценария:

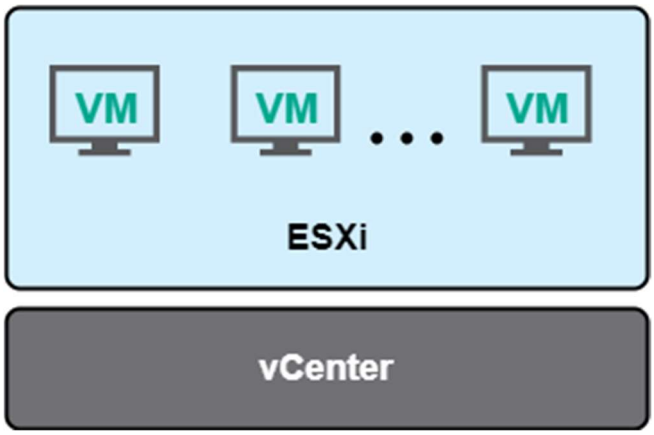
- Провайдер регистрирует клиентов и их виртуальные машины на Сервере интеграции**.**
- В случае неоплаты услуг защиты провайдер может отключить защиту клиентов, но в этом сценарии это делается не автоматически:
    - Провайдер отключает защиту клиента без использования REST API запросов, например, через политики KSC.
    - Провайдер вызывает REST API запрос к Серверу интеграции, сообщая, что защита клиента отключена.
    - Сервер интеграции перестает мониторить защиту всех зарегистрированных машин этого клиента и добавлять информацию о защите клиента в отчеты.
- Провайдер может получать отчеты для мониторинга использования защиты виртуальных машин своих клиентов. Эти отчеты могут быть использованы для биллинга клиентов.

6 Сценарии развертывания решения KSVLA

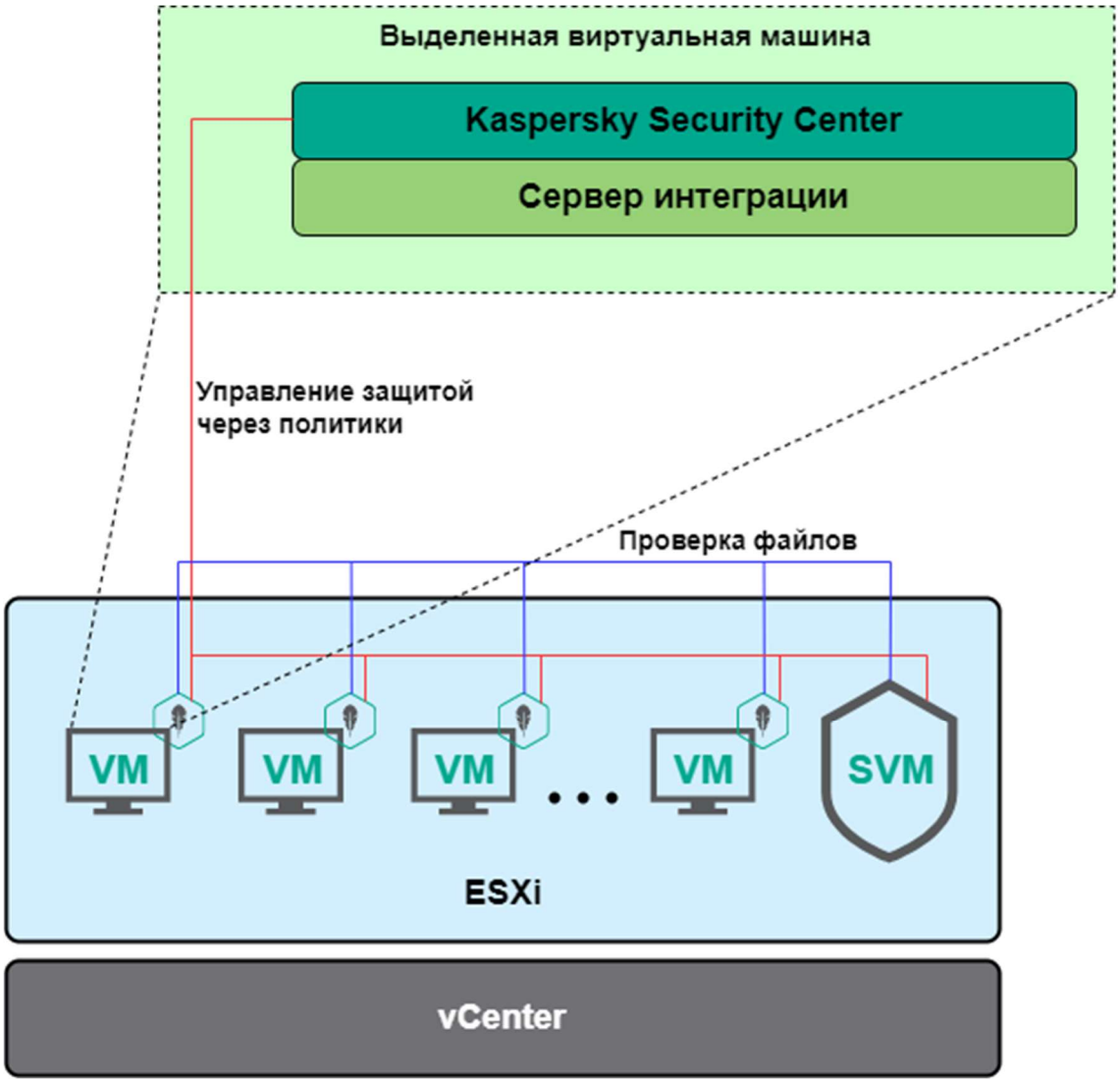
Рассмотрим сценарии развертывания на примерах различных инфраструктуры пользователя. Для примера возьмем инфраструктуру на платформе VMware vSphere с сервером управления виртуальной инфраструктурой vCenter.

Базовый сценарий

Базовым является вариант, когда пользователь имеет один гипервизор ESXi в vCenter.



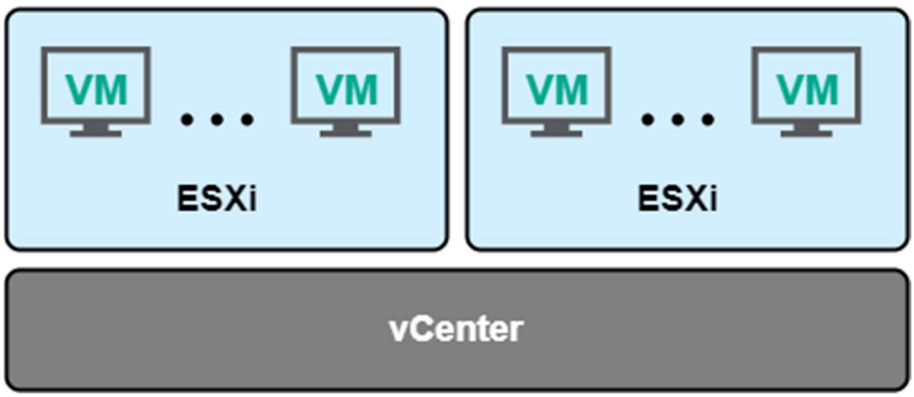
В этом случае пользователь выделяет одну виртуальную машину, которая будет использоваться для управления защитой, и устанавливает на эту машину KSC и Сервер интеграции. На гипервизоре пользователь разворачивает SVM, исходя из базовых расчетов 80 Легких агентов к 1 SVM (*см. примечание ниже). На всех машинах, которые необходимо защитить, устанавливает Легкий агент.



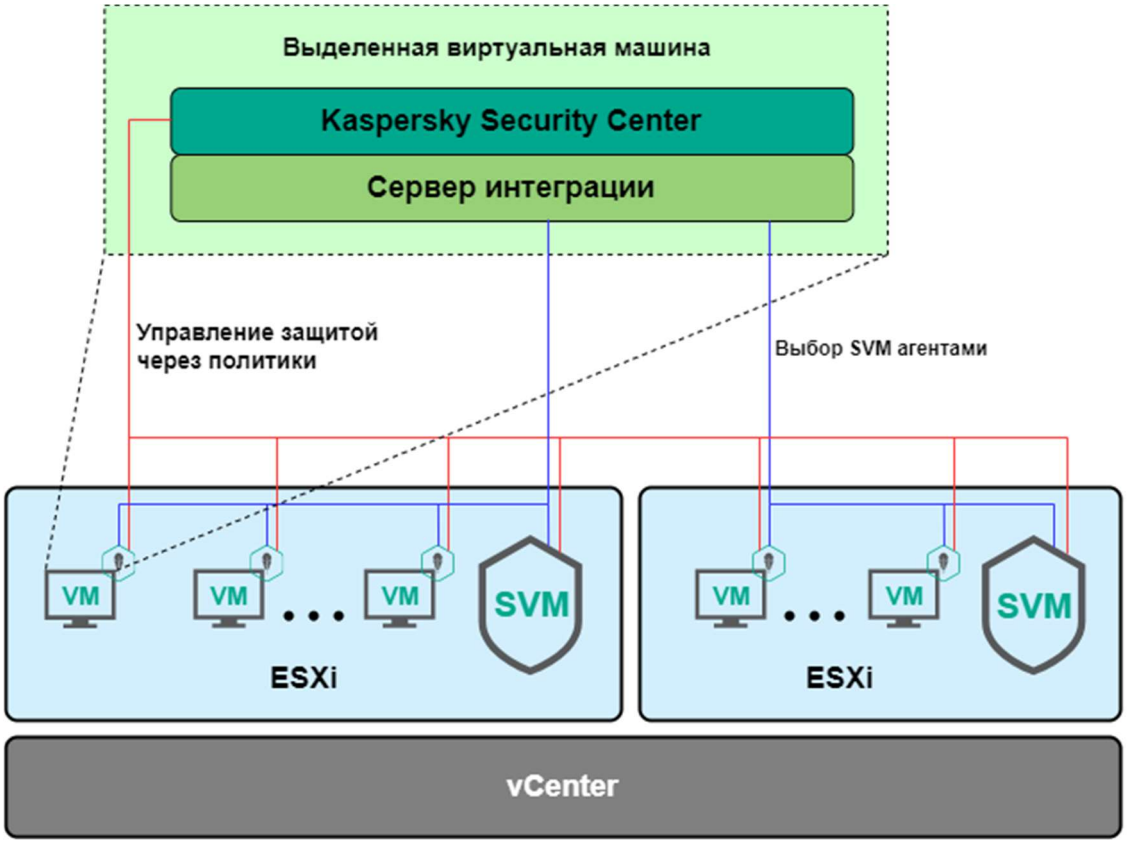
*Примечание. Важно понимать, что "базовый расчет" количества Легких агентов на SVM может колебаться от "160 к 1" до "40 к 1", в зависимости от активности и однотипности виртуальных машин под защитой конкретной SVM. Расчет "80 к 1" подходит для подавляющего большинства клиентов, хотя есть клиенты и с "200 к 1" без видимых проблем по имеющимся артефактам мониторинга.

Стандартный сценарий

Наиболее частый пример пользовательской инфраструктуры – когда у пользователя есть несколько гипервизоров, а общее количество виртуальных машин меньше 10000.

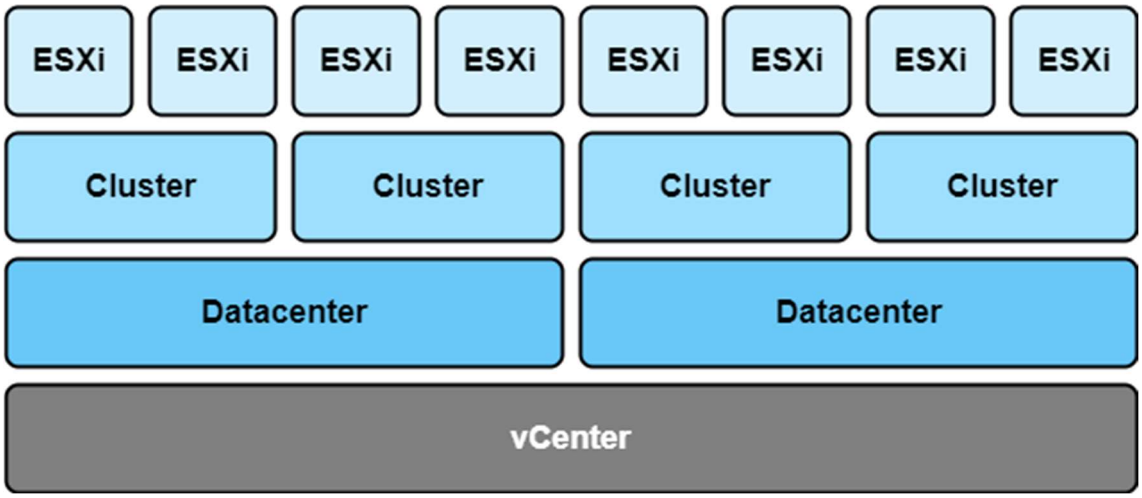


В этом случае, пользователь также выделяет одну виртуальную машину на одном из гипервизоров и устанавливает на нее KSC и Сервер интеграции. На каждом гипервизоре пользователь разворачивает SVM, на виртуальные машины устанавливает Легкий агент. В этом сценарии будет целесообразно использовать стандартный алгоритм выбора SVM. Когда используется стандартный алгоритм, Легкий агент выбирает для подключения SVM, которая развернута на том же гипервизоре. Если на гипервизоре нет доступных для подключения SVM, Легкий агент выбирает SVM, к которой подключено наименьшее количество Легких агентов, независимо от расположения SVM.

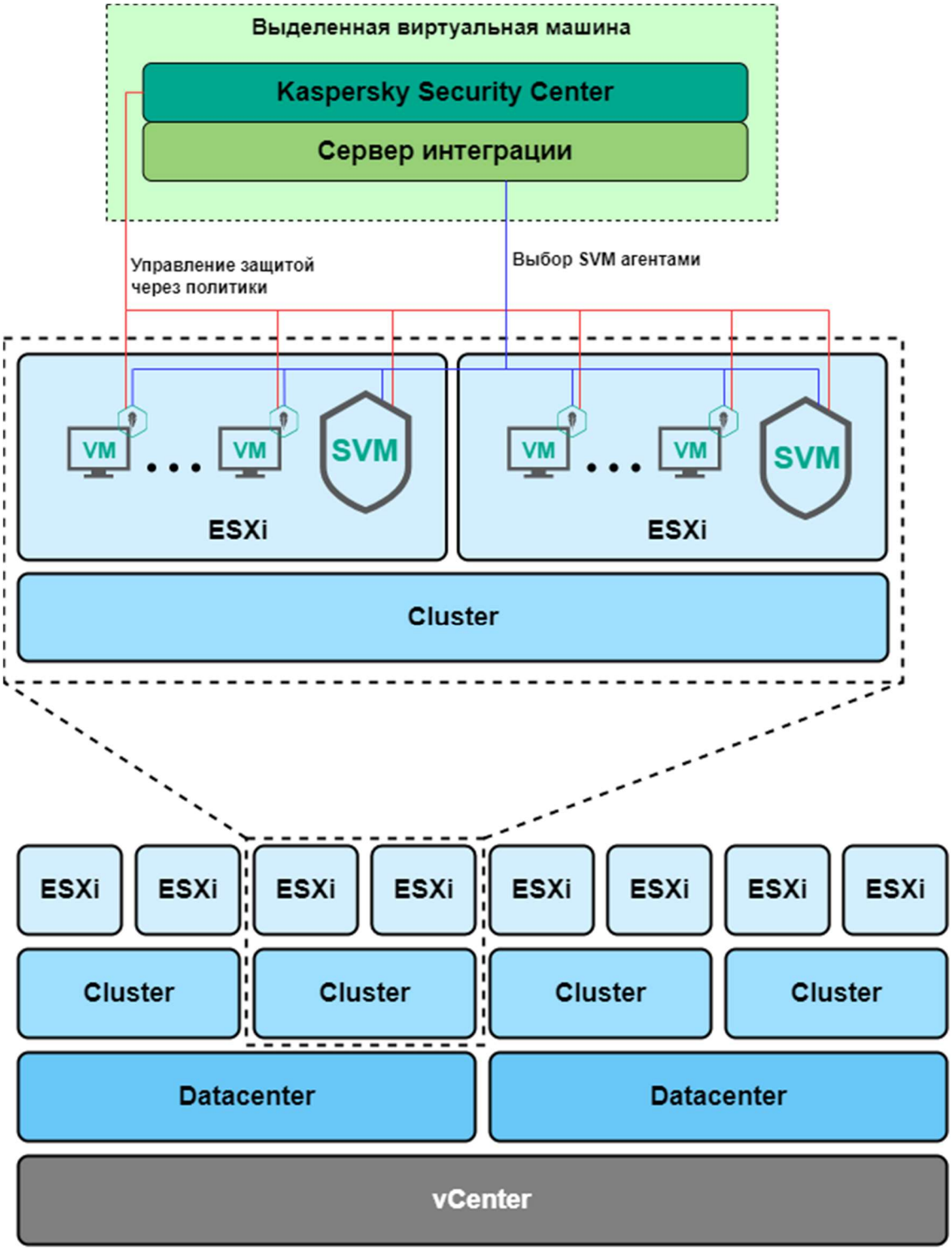


Сценарий больших инфраструктур

Рассмотрим более сложный сценарий кластеризации, или сценарий больших инфраструктур (от 10000 виртуальных машин). В данном случае у пользователя есть большое количество гипервизоров, которые объединены в кластеры, которые, в свою очередь, могут быть объединены в Datacenter.



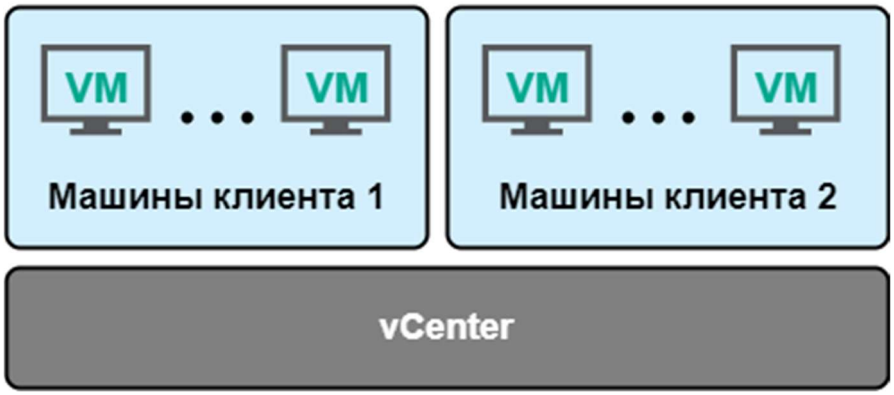
При такой инфраструктуре пользователю следует использовать расширенный алгоритм выбора SVM. Если применяется этот алгоритм, пользователь может указать в политике Легкого агента, каким образом Легкие агенты должны учитывать расположение SVM. Например, Легкий агент может подключаться к SVM, развернутой на том же кластере гипервизоров.



Расширенный алгоритм выбора SVM позволяет сбалансировать нагрузку на SVM. Например, если на одном из ESXi мало виртуальных машин и мощности SVM простаивают, SVM может защищать виртуальные машины на другом ESXi, расположенном на том же кластере, снимая нагрузку с другой SVM.

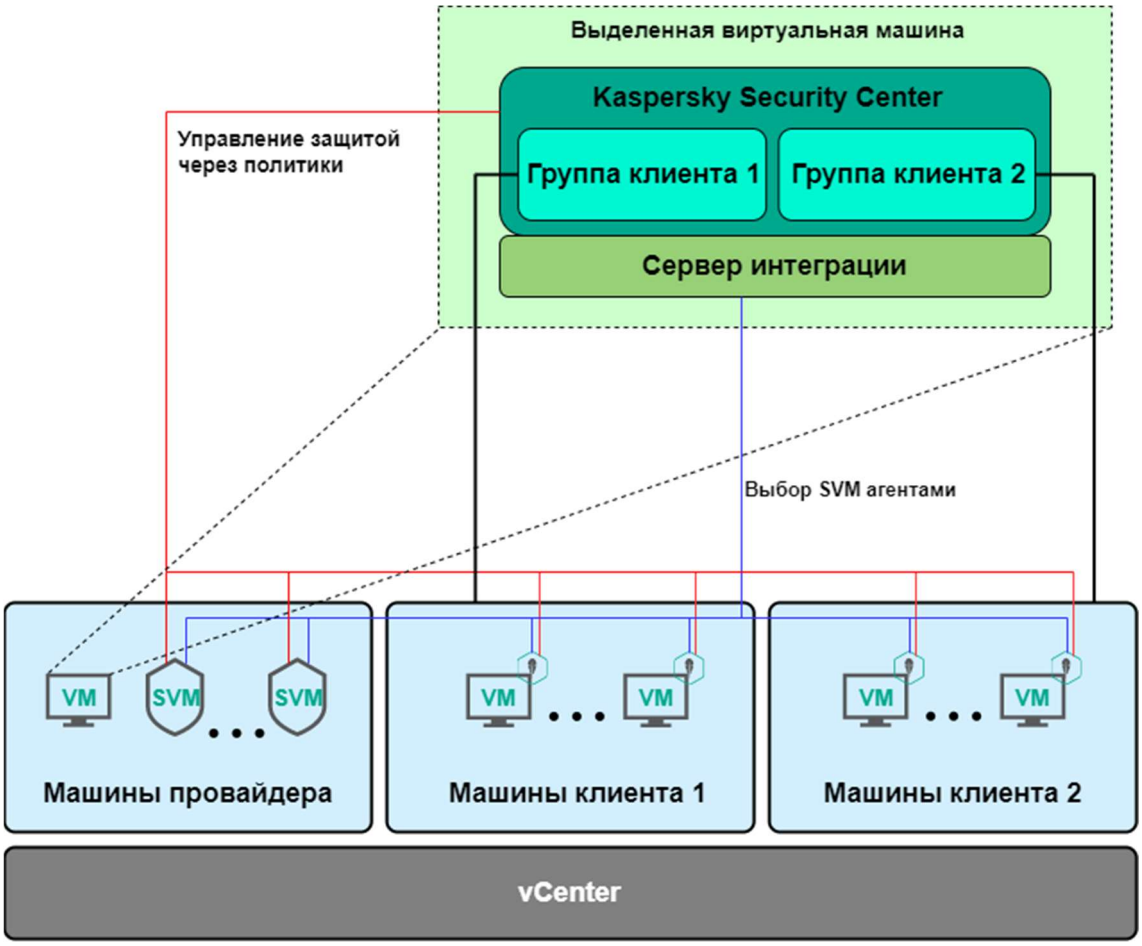
Сценарий провайдера услуг (Saas provider)

Рассмотрим сценарий, когда пользователь является провайдером услуг виртуализации на своем vCenter в формате Software as a service. Провайдер продает нескольким клиентам доступ к виртуальным машинам, развернутым на его vCenter. При этом виртуальные машины клиента могут быть развернуты на разных ESXi, клиент может вообще ничего не знать про инфраструктуру vCenter.

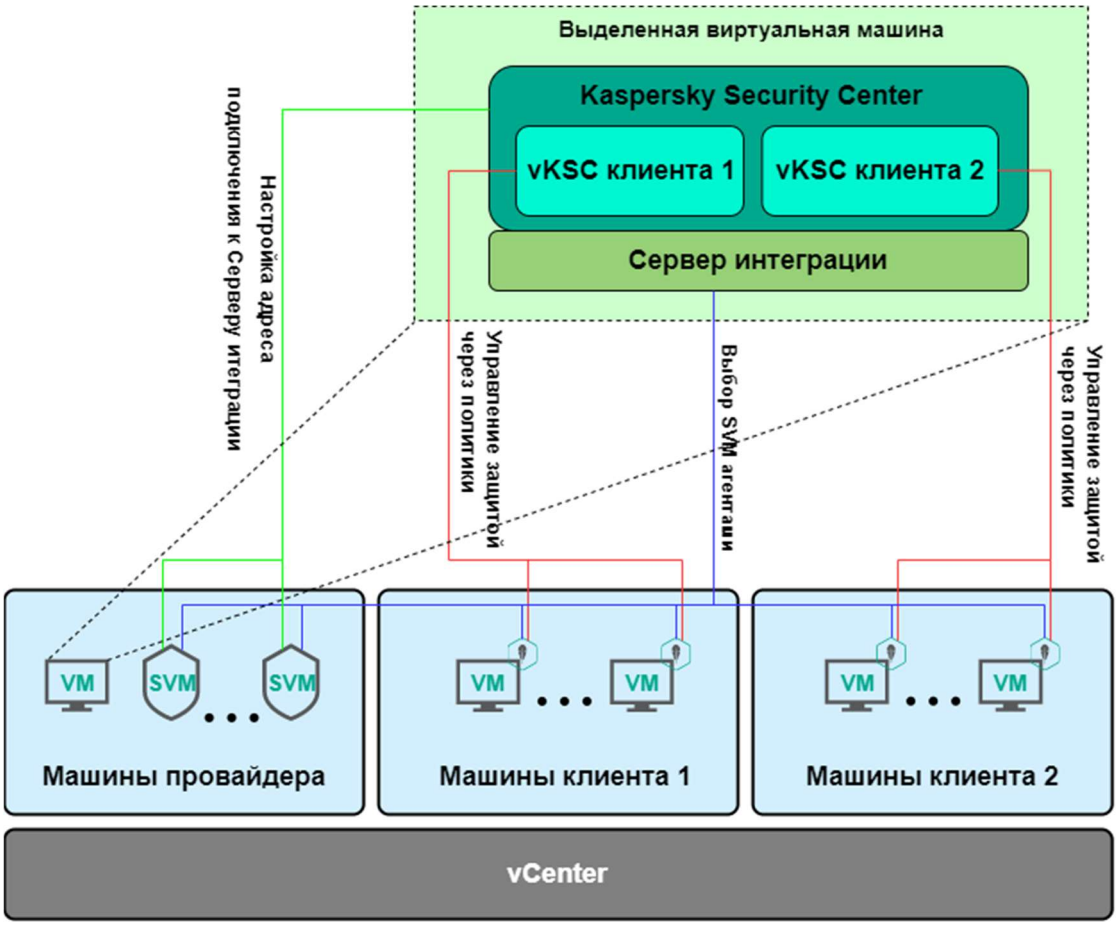


Провайдер хочет обеспечить защиту виртуальных машин своих клиентов с помощью KSVLA. В зависимости от особенностей и потребностей клиентов это может быть сделано разными способами.

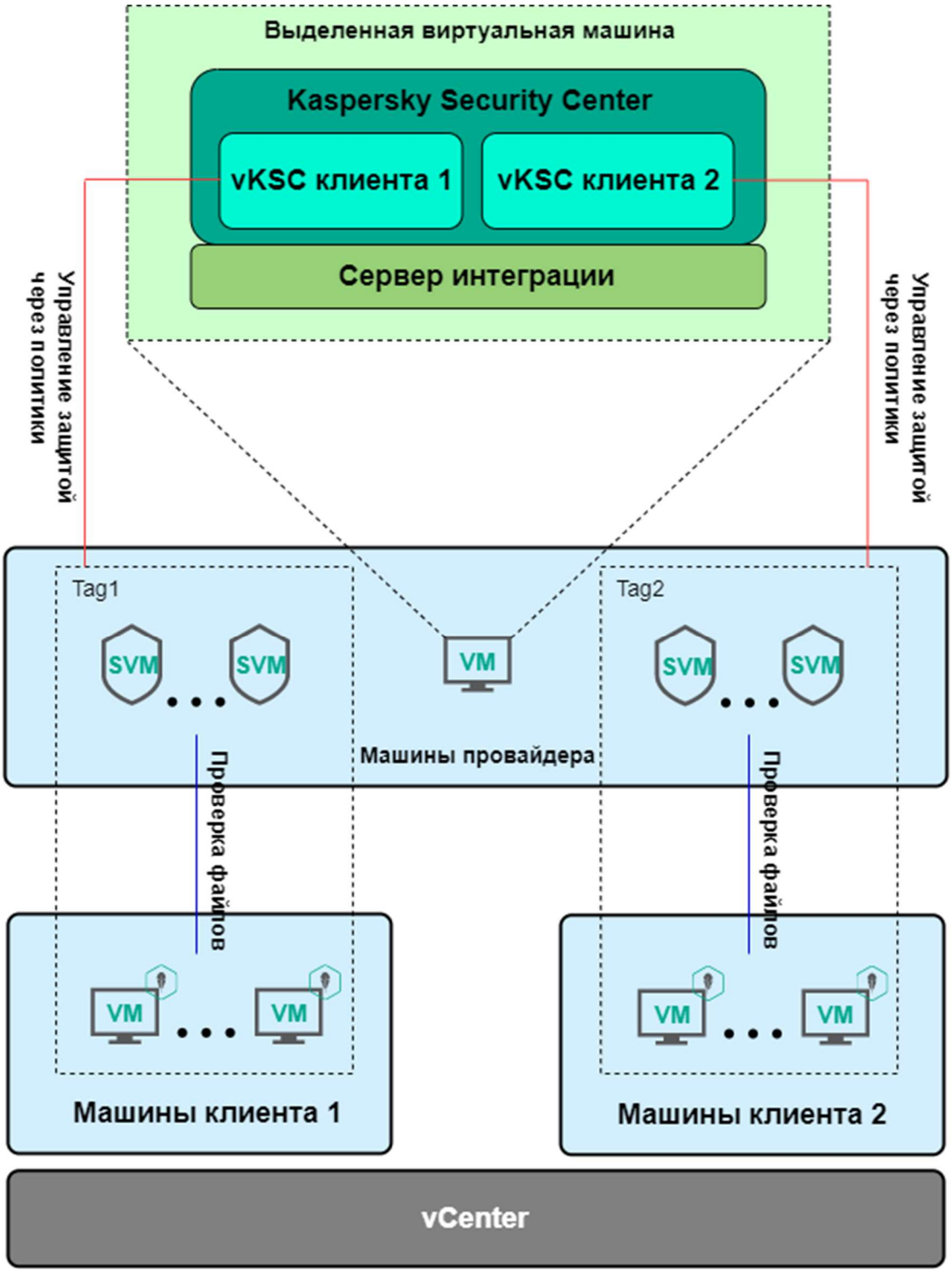
- Самый простой вариант – провайдер полностью самостоятельно управляет защитой своих клиентов. Провайдер разворачивает KSC и Сервер интеграции на одной из машин, к которой имеет доступ только он. В KSC для каждого клиента создается отдельная сервисная группа, в которой будут регистрироваться виртуальные машины этого клиента. Провайдер устанавливает Легкие агенты на машины клиентов и разворачивает необходимое количество SVM. SVM могут быть развернуты как на каждом гипервизоре, где есть виртуальные машины, так и на отдельном гипервизоре. Если провайдер берет оплату за защиту виртуальных машин клиентов и хочет получать отчеты о том сколько времени защищалась каждая из машин клиента, провайдер может использовать KSVLA в режиме multitenancy по сценарию "Упрощенный клиент".



- Если клиент хочет иметь возможность настраивать защиту своих машин, провайдер может предоставить каждому клиенту свой vKSC. Чтобы автоматизировать создание vKSC для каждого клиента, провайдер может использовать KSVLA в режиме multitenancy по сценарию "Полный клиент". В этом случае провайдер также устанавливает KSC, Сервер интеграции, разворачивает SVM, а установку Легких агентов уже может осуществлять сам клиент через свой vKSC. Как и в предыдущем варианте, провайдер имеет возможность получать отчеты о защите своих клиентов.



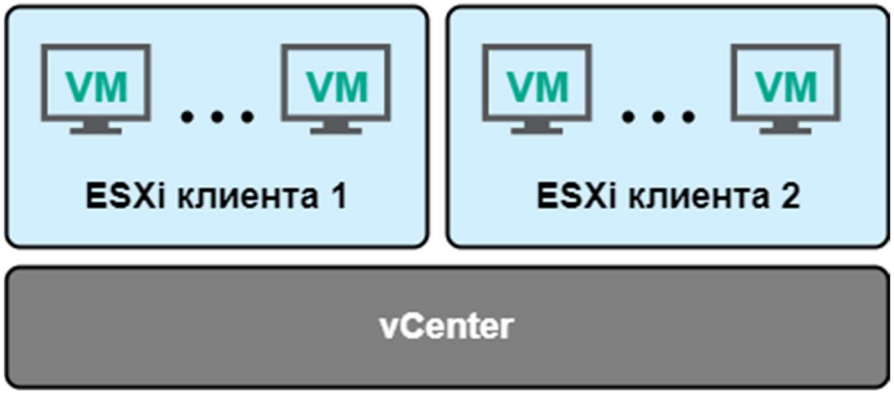
В варианте с предоставлением vKSC клиенту также можно предоставлять набор SVM, которые будут защищать только виртуальные машины этого клиента. Для этого провайдер должен использовать теги, которые задаются в политиках для SVM и Легких агентов. Легкий агент будет подключаться только к тем SVM, которым назначен такой же тег, как и у Легкого агента.



Важно различать теги, которые используются в KSVLA от тегов устройств KSC (см. подробнее https://support.kaspersky.com/KSC/13.2/ru-RU/166115.htm). Теги в KSVLA задаются только в политиках для SVM и Легкого агента.

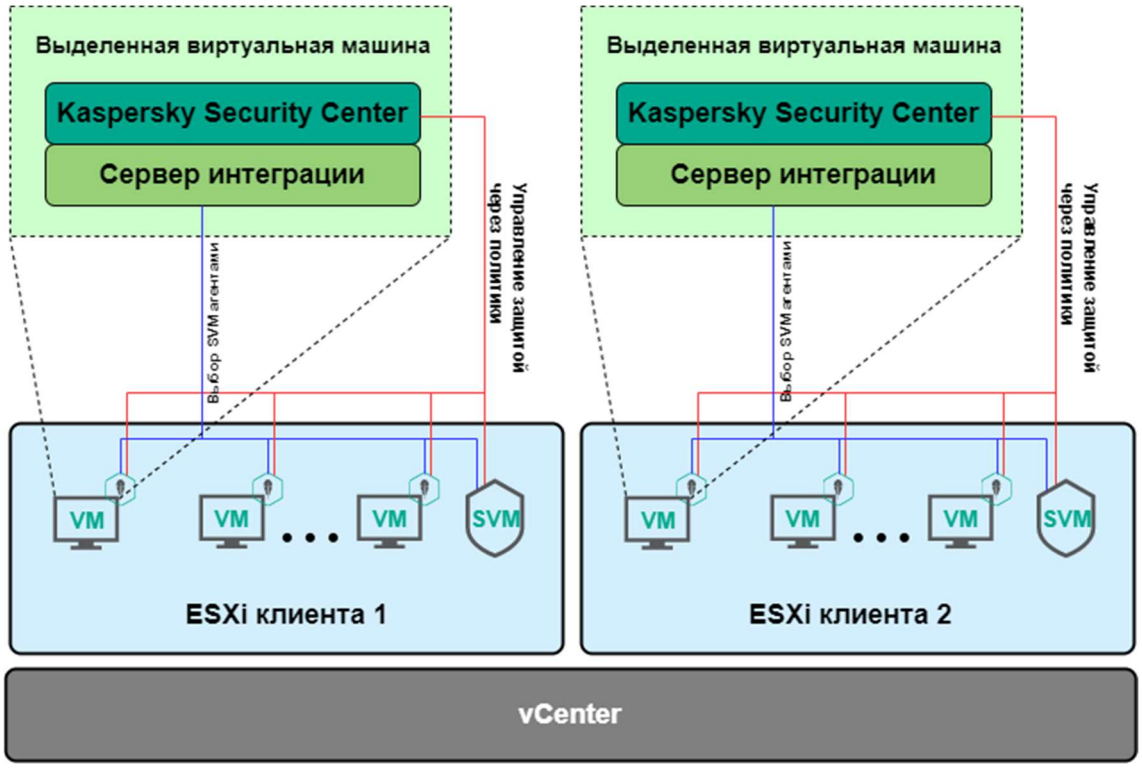
Сценарий провайдера услуг (Iaas provider)

Пользователь может также быть провайдером услуг виртуализации в формате Infrastructure as a service. В этом случае он не просто передает клиенту доступ к виртуальным машинам, а предоставляет клиенту один или несколько ESXi, на которых клиент может самостоятельно развернуть виртуальные машины.

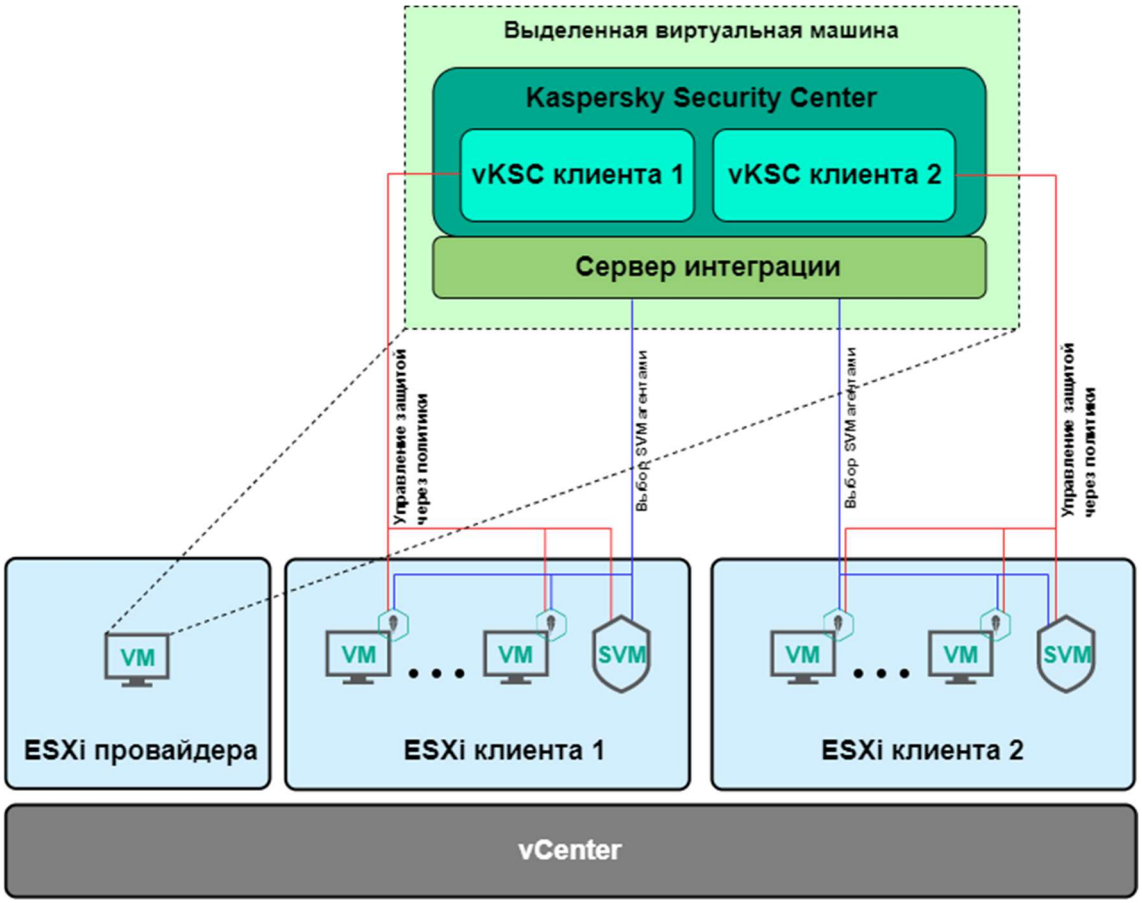


В этом сценарии у провайдера также есть несколько способов обеспечить защиту своих клиентов.

Поскольку клиенты полностью владеют своими ESXi, клиенты могут самостоятельно развернуть свой KSC, Сервер интеграции и развернуть свои SVM без использования режима multitenancy в KSVLA.



Если провайдер хочет продавать услуги защиты виртуальных машин клиентов, он может использовать KSVLA в режиме multitenancy по сценарию "Полный клиент". В этом сценарии выдать каждому клиенту свой набор SVM будет проще, достаточно будет настроить расширенный алгоритм выбора SVM. В таком варианте экономятся ресурсы клиента, поскольку клиент не будет устанавливать свой KSC. Также все защищаемые машины клиента будут взаимодействовать с SVM только в рамках сети клиента, что позволяет не включать шифрование соединения между Легкими агентами и SVM.


Выделенная виртуальная машина
Kaspersky Security Center
vKSC клиента 1
vKSC клиента 2
Сервер интеграции
Управление защитой через политики
Выбор SVM агентами
Выбор SVM агентами
Управление защитой через политики
VM
VM
VM
SVM
VM
VM
SVM
ESXi провайдера
ESXi клиента 1
ESXi клиента 2
vCenter

7 Сценарий "Полный клиент"

Использование KSVLA в режиме multitenancy по сценарию "Полный клиент" предполагает создание структуры защиты клиентов с помощью REST API Сервера интеграции. При создании провайдером полного клиента Сервер интеграции создает сервисную группу с указанным именем клиента. В этой сервисной группе для каждого клиента Сервер интеграции создает Виртуальный сервер администрирования (vKSC), а также учетную запись, с помощью которой клиент сможет подключаться к своему vKSC.

Управление защитой клиента осуществляется различными способами, в зависимости от версии программы.

- В программе версии 5.2 защитой клиента управляет Сервер интеграции. Когда Легкие агенты, установленные на виртуальных машинах клиента, обращаются с Серверу интеграции за списком доступных SVM, Сервер интеграции проверяет состояние защиты клиента:
    - Если защита клиента включена, то Легкому агенту в ответ возвращается список доступных для подключения SVM. Легкий агент подключается к одной из них, и защита виртуальной машины начинает работать.
    - Если защита клиента выключена, то Легкому агенту в ответ возвращается несуществующая, фиктивная SVM. Легкий агент не может к ней подключится, и защита виртуальной машины не работает.
- В программе версии 5.1 защитой управляют политики включения и выключения защиты, см. подробнее: https://support.kaspersky.com/ksvla/5.1/ru-RU/199684.htm.

Рассмотрим более подробно запросы к REST API Сервера интеграции, которые используются для создания структуры защиты клиентов.

7.1 Создание клиента

Для создания полного клиента используется следующий запрос:

POST /api/2.0/virtualization/tenants

```
<tenant>
    <!-- название клиента будет использоваться для vKSC и группы -->
    <name>{название}</name>
    <description>{описание}</description>
    <userData>
        <!-- более подробное описание клиента -->
        <![CDATA[{дополнительные сведения}]]>
    </userData>
    <preferredViisAddress>{IP-адрес Сервера
интеграции}</preferredViisAddress>
    <type>Complete</type>
    <vKsc>
        <!-- пользователь клиента для подключения к vKSC -->
        <user>
            <name>{имя администратора}</name>
            <!-- пароль должен быть зашифрован в Base64 -->
            <password>{пароль администратора}</password>
        </user>
    </vKsc>
</tenant>
```

Создание клиента – операция относительно долгая, поэтому она выполняется асинхронно и в ответ на запрос пользователь получает xml с созданной задачей.

```
<task id={идентификатор} created={дата и время создания}
stateChanged={дата и время изменения состояния} changed={дата и время
изменения прогресса}>
    <!-- например Created, Running, Completed, Failed -->
    <state>{состояние}</state>
    <!-- в данном случае CreateTenant -->
    <type>{тип}</type>
    <!-- например CreateVKsc, CreateVKscUser, CreateDbEntries -->
    <stage>{этап}</stage>
    <progress>{процент выполнения}</progress>
    <!-- содержит результат задачи, если она завершилась успешно -->
    <result>{результат}</result>
    <!-- содержит описание ошибки, если задача завершилась с ошибкой -->
    <error>{сообщение об ошибке}</error>
</task>
```

По идентификатору задачи можно запрашивать текущее состояние задачи и ждать ее окончания с помощью запроса

GET /api/2.0/virtualization/tasks/{идентификатор задачи}

После выполнения задачи в поле result будет содержаться информация о созданном клиенте:

```
<task id={идентификатор} created={дата и время создания}
stateChanged={дата и время изменения состояния} changed={дата и время
изменения прогресса}>
    <state>Completed</state>
    <type>CreateTenant</type>
    <progress>100</progress>
    <!-- содержит результат задачи, если она завершилась успешно -->
    <result>
        <tenant id="{идентификатор клиента}" created="{дата и время
создания}" updated="{дата и время обновления}">
            <name>{название клиента}</name>
            <vKsc id="{идентификатор vKSC}">
                <user>
                    <name>{имя администратора}</name>
                </user>
            </vKsc>
            <status>Inactive</status>
        </tenant>
    </result>
</task>
```

7.2 Получение информации о клиенте

Получать информацию об отдельном клиенте по идентификатору клиента, так и обо всех клиентах сразу с помощью запросов:

GET /api/2.0/virtualization/tenants/{идентификатор клиента}

GET /api/2.0/virtualization/tenants

В ответ возвращается та информация, которая указана провайдером при создании клиента:

```
<tenant id="{идентификатор}" created="{дата и время создания}"
updated="{дата и время обновления}">
    <name>{название клиента}</name>
    <description>{описание}</description>
    <userData>
        <![CDATA[{дополнительные сведения}]]>
    </userData>
    <type>Complete</type>
    <status>{статус (Active/Inactive)}</status>
    <vKsc id="{идентификатор vKSC}">
        <user>
            <name>{имя администратора}</name>
        </user>
    </vKsc>
</tenant>
```

Клиент создается со статусом "неактивный", поэтому изначально в поле status указано Inactive.

7.3 Удаление клиента

Для удаления клиента используется следующий запрос:

DELETE /api/2.0/virtualization/tenants/{идентификатор клиента}?removeTenantArtifacts={true|false}

С помощью параметра removeTenantArtifacts можно указать, нужно ли удалять различные артефакты клиента в KSC: группу клиента, vKSC, учетную запись для подключения к vKSC.

7.4 Регистрация виртуальной машины

Для получения отчетов о защите своих клиентов, провайдер должен зарегистрировать виртуальные машины клиентов на Сервере интеграции с помощью следующего запроса, в который необходимо передать bios id машины:

POST /api/2.0/virtualization/tenants/{идентификатор клиента}/vms/register

```
<vm biosId="{идентификатор BIOS ID}">
     <name>{имя}</name>
     <userData><![CDATA[{дополнительные сведения}]]></userData>
</vm>
```

7.5 Получение информации о защищаемых машинах клиента

Для получения информации о виртуальных машинах, зарегистрированных для клиента, используется следующий запрос.

GET /api/2.0/virtualization/tenants/{идентификатор клиента}/vms

В ответ на запрос возвращается список машин клиента:

```
<vms>
    <vm id="{идентификатор в базе}" biosId="{идентификатор BIOS ID}">
        <name>{имя}</name>
        <userData><![CDATA[{дополнительные сведения}]]></userData>
    </vm>
<vms>
```

7.6 Отмена регистрации виртуальной машины

Для отмены регистрации виртуальной машины клиента используется следующий запрос, в который необходимо передать идентификаторы клиента и машины:

POST /api/2.0/virtualization/tenants/{идентификатор клиента}/vms/unregister?vmId={идентификатор}

7.7 Включение защиты клиента

Для включения защиты необходимо активировать клиента с помощью следующего запроса:

POST /api/2.0/virtualization/tenants/{идентификатор клиента}/activate

В ответ на запрос возвращается задача с типом ChangeTenantActivation.

При включении защиты клиента происходят следующие действия:

- Активируются политики включения защиты, которые применяются для Легких агентов версии 5.1.
- Статус клиента в базе Сервера интеграции меняется на Active.
- Сервер интеграции начинает возвращать Легким агентам версии 5.2 список доступных для подключения SVM.

7.8 Выключение защиты клиента

Для выключения защиты необходимо деактивировать клиента с помощью запроса:

POST /api/2.0/virtualization/tenants/{идентификатор клиента}/deactivate

В ответ на запрос возвращается задача с типом ChangeTenantActivation.

При выключении защиты клиента происходят следующие действия:

- Активируются политики выключения защиты, которые применяются для Легких агентов версии 5.1.
- Статус клиента в базе Сервера интеграции меняется на Inactive.
- Сервер интеграции начинает возвращать Легким агентам версии 5.2 адрес фиктивной SVM.

После выключения защиты, провайдеру рекомендуется изолировать машины клиентов от SVM путем настройки параметров сети, чтобы клиент не мог несанкционировано подключать Легкие агенты к SVM.

7.9 Работа с отчетами

Для получения отчета используется следующий запрос:

POST /api/2.0/virtualization/reports/tenants?tenantId={идентификатор клиента}&from={дата и время}&to={дата и время}

В адресе запроса можно указать следующие необязательные параметры:

- tenantId – идентификатор клиента, для которого необходимо построить отчет.
- параметры from, to – временной промежуток, за который необходимо построить отчет. Время необходимо указывать в UTC формате, например: 2022-08-11T11:54:20.

В ответ на запрос возвращается задача, после авершения которой в поле result будет содержаться идентификатор отчета. По этому идентификатору можно получить сам отчет с помощью следующего запроса:

GET /api/2.0/virtualization/reports/tenants/{идентификатор отчета}

Ответ приходит в csv формате со следующими столбцами:

- идентификатор клиента;
- название клиента;
- идентификатор виртуальной машины;
- имя виртуальной машины;
- дата и время включения защиты;
- дата и время выключения защиты.

В отчете указываются в течение каких периодов защищались виртуальные машины клиентов. В первых двух столбцах указывается информация о клиенте, в следующих двух – информация о виртуальной машине, в последних двух – период времени, в течении которого эта машина была защищена. Время для всех периодов указывается в UTC формате.

8 Сценарий "Упрощенный клиент"

Использование KSVLA в режиме multitenancy по сценарию "Упрощенный клиент" предполагает, что у провайдера уже настроена какая-то структура защиты клиента в режиме multitenancy. Структура защиты может быть организована как с помощью групп администрирования, в которых располагаются машины клиентов, так и с помощью vKSC. Сценарий применяется для автоматизации получения провайдером отчетов о защите виртуальных машин клиентов.

Как и в сценарии "Полный клиент", провайдеру необходимо зарегистрировать своих клиентов и их виртуальные машины на Сервере интеграции, но в этом сценарии на KSC не будет создаваться никаких сущностей, информация о клиентах будет добавлена только в базу Сервера интеграции.

Для работы этого сценария этого применяются следующие запросы.

8.1 Создание клиента

Для создания упрощенного клиента используется такой же запрос, как для создания полного клиента. Отличие в том, что теле запроса указывается тип клиента Simple, а также не указывается пользователь для vKSC.

POST /api/2.0/virtualization/tenants

```
<tenant>
    <!-- название клиента, которое будет храниться в базе -->
    <name>{название}</name>
    <description>{описание}</description>
    <userData>
        <![CDATA[{дополнительные сведения}]]>
    </userData>
    <type>Simple</type>
</tenant>
```

8.2 Включение защиты клиента

Для включения защиты клиента используется следующий запрос:

POST /api/2.0/virtualization/tenants/{идентификатор клиента}/activate

В ответ на запрос возвращается задача с типом ChangeTenantActivation.

При включении защиты упрощенного клиента меняется только статус защиты клиента в базе Сервера интеграции (на Active). После смены статуса информация о защите клиента начинает добавляться в отчеты о защите клиентов.

8.3 Другие запросы

Другие запросы для работы с упрощенным клиентом, например, регистрация виртуальной машины, получение информации о клиенте, получение отчетов и т.д, не отличаются от запросов для работы с полным клиентом, описанных выше.

9 Примеры развертывания KSVLA в режиме multitenancy на пользовательских инфраструктурах

Рассмотрим примеры развертывания KSVLA в режиме multitenancy в инфраструктуре на платформе VMware vSphere.

9.1 Сценарий "Полный клиент"

Рассмотрим пример пользовательской инфраструктуры провайдера, на которой еще не установлено решение KSVLA.

Провайдер предоставляет услуги по виртуализации на своем vCenter, на котором развернуты виртуальные машины для его клиентов. Провайдер хочет защитить виртуальные машины клиентов, используя решение KSVLA, при этом дав возможность клиентам управлять защитой.

1. Администратор провайдера начинает развертывание с установки Kaspersky Security Center, который будет общим для всех клиентов.

Сервер администрирования KSC должен иметь доступ ко всем защищаемым виртуальным машинам для развертывания Легких агентов и управления ими. Сервер администрирования можно установить как на отдельную виртуальную машину, так и на физический сервер. Подробнее в справке KSC: https://support.kaspersky.com/KSC/13.2/ru-RU/3396.htm

2. Администратор провайдера устанавливает общие для всех компоненты управления KSVLA.

Компоненты управления устанавливаются с помощью мастера установки компонентов – файла ksvla-components_X.X.X.X_mlg.exe, который входит в комплект поставки. По завершении работы мастера будут установлены Сервер интеграции, Консоль Сервера интеграции и mmc-плагины для Сервера защиты и Легкого агента:

- Kaspersky Security для виртуальных сред <номер версии> Легкий агент – Сервер защиты;
- Kaspersky Security для виртуальных сред <номер версии> Легкий агент для Linux;
- Kaspersky Security для виртуальных сред <номер версии> Легкий агент для Windows;

3. Если провайдер хочет получать отчеты о защите клиентов, администратор провайдера включает на Сервере интеграции сбор информации о защите клиентов. Для этого администратор устанавливает значение true для параметра EnableTenantsProtectionReports в конфигурационном файле Сервера интеграции (%ProgramFiles(x86)%\Kaspersky Lab\Kaspersky VIISLA\viislaservice.exe.config) и перезагружает Сервер интеграции, перезагрузив службу VIISLA.

4. Администратор провайдера подключает Сервер интеграции к KSC с помощью Консоли Сервера интеграции. Для этого используется раздел "Параметры подключения к Kaspersky Security Center", где необходимо указать адрес, логин, пароль для подключения к KSC.

5. Администратор провайдера добавляет своих клиентов в базу Сервера интеграции.

Для создания клиентов он использует REST API запрос к Серверу интеграции, описанный выше (POST /api/2.0/virtualization/tenants). После добавления клиентов, администратор провайдера передает доступ к созданным vKSC (адрес и логин/пароль учетной записи) администраторам клиентов. Адрес vKSC представляет собой адрес KSC + "/" + название vKSC.

6. Администраторы клиентов устанавливают на свои виртуальные машины Агент администрирования KSC (см. подробнее https://support.kaspersky.com/KSC/13.2/ru-RU/3305.htm).

Агент администрирования необходим для подключения виртуальной машины к KSC (в данном случае vKSC клиента). Для того, чтобы Агент администрирования подключился к нужному vKSC, необходимо при установке в поле “Адрес сервера” указать адрес нужного vKSC. После установки Агентов администрирования виртуальные машины появятся в списке управляемых устройств на vKSC.

7. Администратор провайдера передает на vKSC клиентов инсталляционные пакеты для установки Легких агентов с помощью задачи распространения инсталляционных пакетов (см. подробнее https://support.kaspersky.com/KSC/13.2/ru-RU/6383.htm). Администраторы клиентов в своем vKSC в папке Управляемые устройства выбирают машины, на которых необходимо установить Легкий агент, и запускают на них переданный инсталляционный пакет, выбирая пункт “Установить программу”.

8. Администратор провайдера добавляет машины, на которых были установлены Легкие агенты, в базу Сервера интеграции с помощью REST API запроса, описанного выше.

9. Администратор провайдера устанавливает необходимое количество SVM на ESXi, например, исходя из базовых расчетов 80 Легких агентов к 1 SVM. Компонент SVM поставляется в виде образа. Развертывание SVM осуществляется с помощью мастера управления SVM, который запускается в Консоли Сервера интеграции. В процессе развертывания администратор подключает vCenter к Серверу интеграции, выбирает гипервизоры, на которых необходимо развернуть SVM, настраивает параметры виртуальной машины (сетевые адаптеры и хранилища), а также параметры подключения SVM к Серверу интеграции и Серверу администрирования.

10. Администратор провайдера создает политики для SVM и Легких агентов. В политике для Легких агентов выбирает алгоритм выбора SVM Легкими агентами:

- Стандартный алгоритм выбора SVM: Легкий агент пытается подключиться к SVM, развернутой на том же гипервизоре, на котором работает Легкий агент.
- Расширенный алгоритм выбора SVM: пользователь может указать с помощью ползунка "Расположение SVM", каким образом Легкие агенты должны учитывать расположение SVM при выборе SVM для подключения. Например, Легкий агент может подключаться к SVM, развернутой на том же кластере гипервизоров или в том же Datacenter, что и Легкий агент.

11. Администратор провайдера включает защиту клиентов с помощью REST API запроса к серверу интеграции, описанного выше. При необходимости, например, в случае неоплаты клиентом услуг защиты, провайдер имеет возможность отключить защиту клиента.

Подобную схему следует масштабировать по достижении, примерно, 50000 виртуальных машин в виртуальной инфраструктуре провайдера – как ограничение на стабильную обработку запросов клиентов Сервером интеграции. В данный момент также имеются ограничения на количество Виртуальных серверов администрирования со стороны KSC – 500 vKSC, а также на количество защищаемых машин в KSC – 15000 машин. Таким образом, рекомендуется масштабировать схему в целом по достижении 15000 машин у клиентов в группе.

9.2 Сценарий "Упрощенный клиент"

Теперь рассмотрим пример пользовательской инфраструктуры, в которой уже настроена структура защиты клиентов в режиме multitenancy.

Провайдер предоставляет услуги по виртуализации на своем vCenter и у провайдера уже установлено решение KSVLA: установлены KSC, Сервер интеграции, развернуты SVM, на машинах клиента установлен Легкий агент. В KSC провайдер сгруппировал машины клиента по сервисным группам и самостоятельно управляет защитой клиентов.

Провайдер хочет автоматизировать выставление счетов клиентам за услуги защиты, для этого ему необходимо знать периоды, в которые была защищена каждая из машин клиента.

1. Администратор провайдера включает на Сервере интеграции сбор информации о защите клиентов. Для этого администратор устанавливает значение true для параметра EnableTenantsProtectionReports в конфигурационном файле Сервера интеграции (%ProgramFiles(x86)%\Kaspersky Lab\Kaspersky VIISLA\viislaservice.exe.config) и перезагружает Сервер интеграции, перезагрузив службу VIISLA.

2. Администратор провайдера заносит информацию о своих клиентах в базу Сервера интеграции, используя REST API запрос, описанный выше. В случае отказа клиентов от услуг защиты / покупки услуг защиты новыми клиентами администратор провайдера также пользуется REST API запросами, удаляя и добавляя клиентов.

3. Администратор провайдера заносит информацию о виртуальных машинах клиентов в базу Сервера интеграции, используя REST API запрос, описанный выше. В случае прекращения клиентом использования машин / получения новых машин клиентами администратор провайдера также пользуется REST API запросами, регистрируя и отменяя регистрацию машин клиента.

4. Администратор провайдера управляет защитой клиентов. В случае неоплаты услуг защиты провайдер может отключить защиту клиентов, но в этом сценарии это делается не автоматически:

- Провайдер отключает защиту клиента без использования REST API запросов, например, через политики KSC.
- Провайдер вызывает REST API запрос к Серверу интеграции, сообщая, что защита клиента отключена.
- Сервер интеграции перестает мониторить защиту всех зарегистрированных машин этого клиента и добавлять информацию о защите клиента в отчеты.

5. Когда необходимо выставить счет клиентам за услуги защиты, администратор провайдера запрашивает у Сервера интеграции отчет, указывая идентификатор клиента, по которому необходимо построить отчет, а также период, за который необходимо его построить, например, за месяц. На основе полученных данных провайдер понимает, какое количество времени защищалась каждая из машин клиента, и имеет возможность посчитать, сколько должен заплатить клиент по тарифу провайдера.