G.I. Shakulo

Cloud Computing Security in Openstack Architecture: General Overview

Keywords: cloud computing, security, OpenStack.

The subject of article is cloud computing security. Article begins with author analyzing cloud computing advantages and disadvantages, factors of growth, both positive and negative. Among latter, security is deemed one of the most prominent. Furthermore, author takes architecture of OpenStack project as an example for study: describes its essential components and their interconnection. As conclusion, author raises series of questions as possible areas of further research to resolve security concerns, thus making cloud computing more secure technology.

Г.И. Шакуло

ЗАЩИЩЕННОСТЬ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ НА ПРИМЕРЕ АРХИТЕКТУРЫ OPENSTACK: ОБЩИЙ ОБЗОР

На информационном рынке на данный момент сформировался интерес к облачным вычислениям как средству увеличить эффективность финансовых вложений, снизить расходы на ИТ-инфраструктуру, а также повысить удобство использования и доступа к информации и ресурсам. Согласно исследованию компании Parallels, среднегодовой темп роста российского рынка облачных вычислений в период до 2015 г. составит не менее 40%, и к концу периода его объем достигнет 15,8 млрд рублей; в то же время исследовательская компания IDC прогнозирует на ближайшие пять лет по меньшей мере двукратный рост для мирового рынка, причем российский рынок будет одним из наиболее быстро развивающихся.

Однако, несмотря на колоссальные преимущества использования облачных технологий, их применение также имеет ряд недостатков, и первостепенным среди них является нерешенный вопрос безопасности, который по-прежнему останавливает многих потенциальных клиентов облачных услуг. Главной причиной беспокойства является то, что клиент, будь то отдельный пользователь или крупная компания, вынужден перекладывать задачу по обеспечению конфиденциальности, целостности и доступности своей информации на провайдера услуг, при этом фактически теряя контроль над своими данными. В особенности это касается предоставления программного обеспечения как услуги – подвида облачных услуг, который начинает получать широкое распространение.

Многие компании-провайдеры в ответ на растущее недоверие к облачным вычислениям, сформировали собственные технологии защиты данных клиента. Примером может являться технология компании Google – Secure Data Connector, которая позволяет клиенту управлять доступом сотрудников к ресурсам при помощи шифрованного соединения между данными и бизнес-приложениями GoogleApps.

Однако подобные решения, во-первых, обычно достаточно сложны и применимы только к конкретному продукту отдельной компании, во-вторых, провайдеры зачастую ограничивают информацию о системе безопасности, которую используют в своем решении, мотивируя это тем, что таким образом они могут ее скомпрометировать. Эти две причины приводят к тому, что предложение на рынке облачных услуг с точки зрения безопасности разбивается на множество отдельных сегментов, и нет единого подхода к решению проблемы безопасности облачных технологий.

Таким образом, сформировалось противоречие между большим спектром возможностей облачных вычислений (и, соответственно, широким спросом на них) и отсутствием простого и эффективного решения для обеспечения должной защищенности, которая является непременным требованием подавляющего большинства клиентов.

Основными понятиями в данной сфере являются:

облачные вычисления – модель, обеспечивающая повсеместный и удобный сетевой доступ по требованию к разделяемым ресурсам, которые могут быть оперативно предоставлены и освобождены с минимальными затратами на управление или взаимодействием с провайдером услуг (определение Национального института стандартизации и технологий Министерства торговли США (англ. NIST, National Institute of Standards and Technology));

облака – инфраструктуры, обеспечивающие платформу для облачных вычислений.

Существует множество решений, предлагаемых различными компаниями, которые позволяют организациям либо разворачивать облако, предназначенное исключительно для пользования в рамках данной организации, с возможным включением в круг пользователей ее клиентов и подрядчиков (такой подход принято называть «приватным облаком»), либо пользоваться услугами и ресурсами облака внешнего лица, которое предоставляет инфраструктуру, платформу или программное обеспечение широкому кругу клиентов (эту форму принято называть «публичным облаком»). Также существует так называемая «гибридная» форма, которая может объединять в себе различные инфраструктуры облаков как частных, так и публичных.

Рассмотрим архитектуру и безопасность публичного облака – в силу того, что проблемы безопасности приватного облака в большинстве ситуаций являются подмножеством проблем публичного. Это достигается за счет того, что в наиболее распространенной форме приватного облака и клиенты, и инфраструктура облака – серверы, устройства хранения, центры обработки данных и т.д., находятся в юрисдикции владельца, что значительно увеличивает степень контроля последнего за ресурсами и доступом клиентов к ним.

С помощью удаленного соединения, в большинстве случаев защищенного, клиенты подключаются к некоторому порталу, на котором, в зависимости от конкретного решения, могут размещать заказы на предоставление услуг, просматривать статистику, оплачивать услуги и т.д.

На стороне владельца, предоставляющего услуги облачных вычислений, зачастую используется некоторое специализированное промежуточное программное обеспечение, позволяющее проводить мониторинг загрузки текущих ресурсов, балансировку нагрузки, распределение прав доступа и т.д.

Помимо прочего, в большинстве существующих на данный момент облачных решений, между аппаратными ресурсами и программным обеспечением размещается слой виртуализации, позволяя таким образом сглаживать неравномерность нагрузки путем перераспределения виртуальных серверов по реальным, а также с возможностью переноса виртуализованных ресурсов, или так называемой «живой миграции».

Одним из возможных решений проблемы изолированности систем безопасности разных производителей может быть использование облачного решения с открытым кодом. На данный момент самым крупным проектом open-source в области облачных вычислений является OpenStack.

OpenStack – это глобальный проект, объединяющий в себе разработчиков и инженеров в области облачных технологий, направленный на создание повсеместной облач-

ной платформы с открытым кодом, предназначенной для построения приватных и публичных облаков.

Изначально OpenStack был основан компанией Rackspace Hosting совместно с NASA (National Aeronautics and Space Administration, англ. *Национальное управление по воздухоплаванию и исследованию космического пространства*). Теперь – это глобальный проект, в котором участвует множество разработчиков и компаний, включая IBM, HP, Cisco, Dell, Vmware и др.

Технологически OpenStack состоит из нескольких взаимосвязанных проектов, каждый из которых выступает в роли компонента облачной инфраструктуры, отвечающего за определенную задачу. Кратко перечислим компоненты и их функции:

- Compute (кодовое название Nova) – управление виртуальными серверами и выделение ресурсов по требованию.
- Object Store (кодовое название Swift) – объектное хранилище данных: позволяет помещать файлы в хранилище и восстанавливать их (но не монтировать директории подобно файловому серверу).
- Block Storage (кодовое название Cinder) – постоянное блочное хранилище данных для гостевых виртуальных машин.
- Image (кодовое название Glance) – каталог и репозиторий виртуальных дисковых образов, которые в основном используются сервисом Nova.
- Dashboard (кодовое название Horizon) – модульный пользовательский Web-интерфейс для всех служб OpenStack.
  - Identity (кодовое название Keystone) – сервисы аутентификации и авторизации для всех служб OpenStack. Также предоставляет каталог услуг для конкретного облака OpenStack.
  - Network (кодовое название Neutron) – возможности сетевого соединения между сетевыми устройствами, управляемыми другими службами.

В целом, OpenStack спроектирован таким образом, чтобы создавать глубоко масштабируемую облачную операционную среду. Достигается это за счет интеграции служб-компонентов, осуществляемой через общедоступные API (Application Programming Interface). Каждая служба может предоставлять и потреблять API, что позволяет устанавливать взаимосвязь между компонентами и при необходимости заменять компоненты на другие, при условии, что программный интерфейс заменяемого компонента будет оставаться тем же.

Взаимодействие пользователей со службами OpenStack осуществляется либо через общий web-интерфейс Horizon, либо напрямую к каждой службе через соответствующий интерфейс API. Взаимодействие служб между собой осуществляется через API (за исключением случаев, когда необходимы привилегированные команды администратора).

Все службы проходят аутентификацию через общий сервис, предоставляемый службой Keystone.

Остановимся подробнее на службе, отвечающей за управление виртуальными машинами – Nova-compute:

OpenStack Compute состоит из нескольких основных компонентов. Многие из них зачастую выделяют в так называемый «контроллер облака» (англ. *Cloud controller*), который отражает глобальное состояние облака и взаимодействует со всеми остальными компонентами.

- Сервер API выступает в качестве внешнего интерфейса для web-сервисов для контроллера.

- Вычислительные контроллеры предоставляют вычислительные серверные ресурсы.
- Контроллер томов предоставляет быстрое и постоянное блочное хранилище данных для вычислительных серверов.
- Сетевой контроллер создает виртуальные сети, позволяющие вычислительным серверам взаимодействовать между собой и внешней сетью.
- Планировщик выбирает вычислительный контроллер, наиболее подходящий для того, чтобы запустить виртуальную машину.

OpenStack Compute построен на архитектуре, основанной на сообщениях, без разделения ресурсов. Все основные компоненты могут быть запущены на различных серверах. Контроллер облака взаимодействует с внутренним объектным хранилищем через протокол HTTP, но общение с сетевым контроллером, планировщиком и контроллером томов происходит через протокол AMQP (Advanced Message Queue Protocol). Nova использует асинхронные вызовы для того, чтобы избежать блокирования компонентов на время ожидания отклика.

Одним из основных управляющих элементов OpenStack является очередь сообщений. Как только служба получает доступ к очереди, далее авторизационные проверки не производятся.

Важным моментом является то, что OpenStack не имеет средств для обеспечения защищенности на уровне сообщений, например цифровую подпись как средство контроля целостности, поэтому необходимо, по крайней мере, обеспечивать защиту передаваемых сообщений на уровне транспортного канала.

Другим важнейшим компонентом Nova-compute является планировщик. Планировщик – это служба, которая отвечает за выбор того вычислительного ресурса (сервера), на котором будет запущена виртуальная машина. Осуществляется выбор с помошью набора фильтров, в которые входят как простые (например, фильтр ComputeFilter ограничивает диапазон возможных серверов теми, на которых в действительности можно запустить машину), так и более сложные, например, для того чтобы запускать множественные виртуальные машины на одном и том же сервере.

По умолчанию всегда включены ComputeFilter, RamFilter (проверяет доступность оперативной памяти) и AvailabilityZoneFilter (ограничивает выбор *зоной доступности* (см. ниже), указанной при запросе).

Также планировщик поддерживает возможность назначить веса доступным серверам, что, в свою очередь, влияет на выбор сервера, на котором можно развернуть виртуальную машину. Однако единственный критерий для определения веса, на данный момент реализованный в OpenStack, – количество свободной оперативной памяти.

В OpenStack принято несколько понятий, тесно связанных с планировщиком и разделением серверов на группы.

- Зона доступности (англ. *Availability zone*) – логическое разделение ресурсов внутри одного развернутого сервиса Nova. Пользователь может выбирать зону доступности при запросе развертывания новой виртуальной машины. В основном используется для избыточности и физической изоляции от других зон.
- Скопление узлов (англ. *Host aggregate*) – похожее на зону доступности понятие, однако скопления не видимы пользователю и предназначены, в основном, для системного использования планировщиком. Например, одно из основных использований – это ограничение запуска некоторых образов определенным подклассом серверов.

- Ячейка (англ. *Cell*) – предназначена для древовидного распределения облачной инфраструктуры. Каждая ячейка имеет собственный запущенный экземпляр службы Nova-compute, а единый для всех Nova-api расположен в корне дерева. Работа планировщика с ячейками устроена следующим образом: сначала планировщик выбирает ячейку (на данный момент случайным образом, в дальнейшем планируется добавить возможность взвешивания/фильтрования для гибкого выбора), затем управление выбором передается во внутренний планировщик выбранной ячейки.
- Область (англ. *Region*) – подобно ячейкам области предназначены для распределенных инфраструктур, однако, в отличие от ячеек, каждая область имеет свой конечный интерфейс API.

Целью последующих исследований в области безопасности облачных вычислений на примере проекта с открытым кодом OpenStack может быть решение следующих вопросов.

- Существует ли способ, который можно применить для изменения процесса развертывания/остановки виртуальных машин?
- Возможно ли использовать внешний сервис аутентификации, отличный от Keystone?
- Есть ли возможность помечать и ограничивать в применении маршруты передачи данных (для определенных данных использовать строго определенные маршруты)?
- Каким образом в OpenStack возможно встроить выполнение дополнительных действий при сохранении/восстановлении образов виртуальных машин (например, вычисление контрольной суммы)?

После решения данных вопросов можно будет говорить о том, что облачные вычисления на базе OpenStack имеют уровень защищенности и прозрачности происходящих внутри облака процессов, который позволит использовать их в средах со строгими требованиями к защищенности, из-за которых невозможно было ранее использовать облачные вычисления.

СПИСОК ЛИТЕРАТУРЫ:

1. OpenStack Operations Guide // OpenStack Foundation - http://docs.openstack.org/openstack-ops/content/ - 2014.
2. OpenStack Security Guide // OpenStack Foundation - http://docs.openstack.org/security-guide/content/ - 2014.
3. OpenStack Architecture Design Guide // OpenStack Foundation - http://docs.openstack.org/arch-design/content/ - 2014.

REFERENCES:

1. OpenStack Operations Guide // OpenStack Foundation -http://docs.openstack.org/openstack-ops/content/ - 2014.
2. OpenStack Security Guide // OpenStack Foundation -http://docs.openstack.org/security-guide/content/ - 2014.
3. OpenStack Architecture Design Guide // OpenStack Foundation -http://docs.openstack.org/arch-design/content/ - 2014.