УДК 004

Особенности контроля взаимодействия клиента VMware vSphere 6.5 с vCenter

П. М. Журов
Московский физико-технический институт (государственный университет),
г. Долгопрудный, Московская обл., Россия

Описана работа по анализу трафика между клиентом VMware vSphere 6.5 и vCenter, выделению на основе этого анализа протокола управления виртуальной инфраструктурой и разработке механизма контроля доступа для этой виртуальной инфраструктуры.

Ключевые слова: разграничение доступа, виртуальная инфраструктура, анализ трафика.

Виртуализация является одной из самых быстроразвивающихся областей в сфере информационных технологий. Преимущественно виртуальные системы призваны заменить классические сервера, поэтому их задачей помимо прочего является и обработка большого количества данных. Как и в любой системе, работающей с данными, в виртуальной инфраструктуре необходимо обеспечить доверенную среду. Одним из важнейших инструментов обеспечения доверенной среды в виртуальной инфраструктуре является разграничение доступа.

При наличии множества других задач, разграничение доступа — одна из самых актуальных проблем в вопросе безопасности виртуальных инфраструктур. Как и во всей сфере IT, в области виртуализации наблюдается рост количества утечек данных. Самые крупные из них [1] связаны именно с неправильным разграничением доступа. Поэтому так важно обеспечить возможность гибкой и удобной настройки прав пользователей.

Лидером в области виртуализации является VMware с продуктом vSphere. Но несмотря на все достоинства данного ПО, в нём есть существенный недостаток с точки зрения безопасности: администратор отвечает как за управление инфраструктурой, так и за назначение прав доступа пользователям к её элементам. Это влечет за собой высокую трудоёмкость решения следующей проблемы: если в системе есть несколько сегментов, то критически возрастает вероятность преднамеренной или случайной утечки данных из одного сегмента в другой [2—4]. Для её решения необходимо средство, позволяющее разграничить доступ к элементам инфраструктуры для всех пользователей, включая администратора.

На рынке существуют всего три продукта с подобным функционалом. Однако один из них [5] не имеет сертификата ФСТЭК (а значит, не может применяться в государственных информационных системах), а второй [6] — не работает напрямую с vSphere (использует сторонний клиент), что вынуждает администратора изменить привычный порядок работы.

Третий комплекс [7] соответствует всей нормативной базе и не меняет порядка работы управляющего персонала. Однако в связи с заявлением VMware о том, что в следующей версии продукта [8] будет использован новый клиент управления на основе HTML5 (влечет за собой изменение принципа взаимодействия веб-клиента с сервером), можно сделать вывод, что существующие решения более не применимы, а значит, проблема остается актуальной. Данную проблему также усугубляет то, что у новой версии нет открытого API, а следовательно, у разработчиков нет возможности быстро переделать существующие решения под нового клиента.

В то же время именно последний продукт — ПАК "Сегмент-В." — представляется наиболее целесообразным для доработки, поскольку в его случае не требуется коренного пересмотра решения. Далее описаны основные аспекты этой доработки, в рамках которой был проведен анализ трафика передаваемого от HTML5-клиента vSphere на vCenter, частично выделен протокол управления, а затем рассмотрены особенности, которые можно использовать для контроля доступа.

Рассмотрение существующей модели разграничения доступа

Прежде всего стоит описать модель разграничения доступа для виртуальной инфраструк-

Журов Павел Михайлович, студент кафедры "Защита информации".
E-mail: pavel.zhurov@gmail.com

Статья поступила в редакцию 13 июня 2018 г.

туры vSphere [9], которая используется в ПАК "Сегмент-В.". Для этого выделим несколько ключевых особенностей данной инфраструктуры, которые эта модель использует.

Самым главным элементом vSphere является центральный сервер виртуальной инфраструктуры — VCSA (vCenter Server Appliance). К нему подключаются различные гипервизоры, на нем создаются хранилища и сети и с помощью него же происходит взаимодействие со всеми этими элементами. Для управления этим сервером используется веб-интерфейс.

Данная модель разграничения доступа использует принцип, схожий с принципом атаки man-in-the-middle: веб-клиент рассматривается в качестве внешнего элемента виртуальной инфраструктуры (ВИ), сам vCenter – внутреннего, а посередине устанавливается прокси-сервер, который анализирует трафик и пропускает команды управления, основываясь на заранее записанных в него политиках доступа. Это позволяет реализовать принцип "разделяй и властвуй": настройками политик доступа занимается администратор безопасности инфраструктуры (АБИ), у которого при этом нет прав системного администратора, и наоборот, системный администратор не имеет возможности настраивать политики доступа.

Реализация на данном прокси-сервере, например, мандатной системы разграничения доступа решит как проблему сегментированности, так и проблему суперпользователя.

Механизм

Рассмотрим, какими качествами должен обладать механизм, который реализует эту модель.

Во-первых, АБИ должен иметь возможность удаленно настраивать политики доступа по защищенному соединению. Помимо удобства это также даст возможность АБИ быстро вносить изменения в политики в критических ситуациях.

Во-вторых, использование данного механизма не должно нарушать привычный порядок работы пользователей инфраструктуры, в частности системного администратора/системных администраторов .

И наконец, крайне желательно, чтобы данный механизм обладал модульностью и масштабируемостью. Под модульностью здесь подразумевается то, что при изменении протокола управления VCSA достаточно будет изменить лишь какую-то часть механизма для продолжения его эксплуатации, а не переделывать весь механизм полностью. Это означает, что каждый модуль должен выполнять строго одну функцию и никак не влиять на работу остальных. Масштабируемость в данном контексте означает возможность беспрепятственно добавлять обработку новых функций VCSA.

Создание механизма

Для создания данного механизма необходимо сначала провести анализ трафика между веб-клиентом vSphere и VCSA. Для этого был создан стенд, содержащий все ключевые элементы виртуальной инфраструктуры: несколько гипервизоров, часть которых объединена в различные кластеры, несколько хранилищ, сетей и пользователей. Все это необходимо, чтобы как можно подробнее изучить протокол управления.

Для анализа данных, передаваемых с клиента на сервер, использовались язык python и библиотека mitmproxy, которая позволяет реализовать прокси-сервер для прослушивания трафика. В vSphere используется протокол HTTPS, который исключает возможность прослушивания трафика. Однако в силу наличия доступа к серверу VCSA, из которого можно взять сертификат и ключ TLS, данное ограничение не помешало реализации.

Новый HTML5-клиент использует для управления протокол JSON RPC, в котором клиент передает команды в формате JSON и в таком же виде получает их от сервера. Эта информация использовалась в дальнейшем для обработки запросов.

Далее, путем воспроизведения действий пользователя в ВИ и журналирования трафика, проходящего через прокси-сервер, были выделены следующие особенности протокола управления.

- Операции, в которых происходят изменения в элементах инфраструктуры (например, изменение настроек виртуальной машины), всегда выполняются с помощью POST-запросов и однозначно идентифицируются по ключам JSON-словаря.
- Операции, направленные на получение информации об инфраструктуре, всегда выполняются с помощью GET-запроса и однозначно идентифицируются по своему URL.
- Информация о пользователе, совершившем действие, всегда хранится в заголовке запроса в открытом виде (даже не в виде идентификатора сессии!).
- Информацию об объекте, с которым работает пользователь, можно получить двумя путями:
  - в большинстве случаев id объекта содержится в URL запроса;
  - иногда, например когда объектов несколько, информация о них содержится в теле запроса.

Эти особенности позволяют однозначно идентифицировать из любого запроса объект, субъект и операцию.

С использованием данной информации и описанных требований к механизму была разработана его архитектура. В первую очередь было принято решение разделить прокси-сервер на модули, каждый из которых выполняет одну строго определенную задачу. Взаимодействие между модулями должно происходить так, чтобы при замене одного модуля не приходилось изменять другой. Получившаяся архитектура представлена на рисунке.

Принцип работы прокси-сервера следующий. Inspector перехватывает запрос и передает его в Handler (обработчик). Тот, пользуясь описанными особенностями протокола управления, получает из запроса данные, необходимые для инициализации объектов, пользователя и операции, затем инициализирует их с помощью модуля Adapter и передает возвращенные адаптером объекты в модуль Logic, который принимает решение о допустимости данной операции. Именно это решение возвращается из Handler'а в Inspector. Под инициализацией здесь подразумевается создание объектов (программных), которые содержат пользователя/элемент ВИ/операцию и все соответствующие ему/ей атрибуты безопасности. Рассмотрим каждый модуль немного подробнее.

Модуль Inspector отвечает за перехват трафика. В нем подключается библиотека mitmproxy, с её помощью перехватывается запрос, который передается в модуль Handler, а последний возвращает в ответ "да/нет" (пропустить запрос или нет). Если запрос заблокирован (Handler вернул нет), Inspector заменяет ответ сервера на ответ с ошибкой.

Модуль Handler является связующим звеном между остальными модулями. Он отвечает за получение необходимой информации из запроса, а именно информации о пользователе, операции, виде операции и объектах, участвующих в операции (их может и не быть).

Модуль Adapter использует локальную базу данных, в которую записаны политики доступа и данные, полученные из Handler. Он инициализирует объекты и пользователя с соответствующими им атрибутами безопасности (уровни и метки доступа, а также список допустимых операций для пользователя) и операцию. В конце адаптер возвращает инициализированные объекты в Handler.

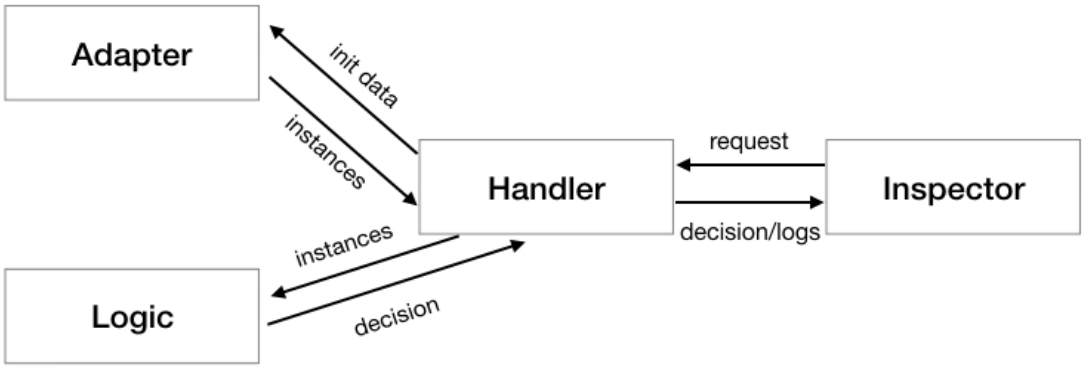
Модуль Logic получает от Handler объекты, пользователя и операцию со всеми соответствующими атрибутами и принимает решение о предоставлении/не предоставлении доступа данному пользователю к данной операции с данными объектами.

Анализ полученного механизма

В первую очередь возникает вопрос к архитектуре — зачем передавать все данные об объектах через Handler, когда можно напрямую передавать их из SQL adapter в Logic, а затем сразу возвращать решение в Inspector.

Это обусловлено тем, что операции бывают нескольких видов: без объектов, операции с одним объектом и с несколькими объектами. В зависимости от вида операции данные, передаваемые в модуль Logic, различаются и сделать передачу данных из Adaprter'а в Logic напрямую — значит добавить в адаптер функцию разделения вида операций, что нарушает принцип модульности (данный модуль должен отвечать только за взаимодействие с базой данных).

Также может возникнуть еще один вопрос — как Handler определяет тип операции, и как он выполняет требование масштабируемости, если при добавлении нового функционала приходится менять и сам Handler.



Архитектура прокси-сервера

Это реализовано с помощью файла operation keys. Указанный файл включает в себя словари, которые содержат:

- Соответствия между ключами из словаря POST-запроса и операциями.
- Соответствия между URL и операциями.
- Список операций, получаемых несколькими путями (например, изменение параметров сети может быть различным: от изменения физического адаптера до изменения названия порт-группы. Все эти операции являются различными с точки зрения управления, но одной и той же операцией — "изменение конфигурации сети" — с точки зрения разграничения доступа).
- Названия операций с несколькими объектами в качестве ключей и путями к получению данных объектов из тела запроса в качестве значений.
- Набор операций, в которых не используются объекты.

При обработке запроса Handler пользуется данным файлом для выделения нужной информации из запроса (название операции и id объекта/объектов). При изменении запроса для какой-либо операции достаточно поменять соответствующую запись в одном из словарей указанного файла, а при добавлении какой-либо операции – добавить соответствующую запись.

Заключение

Проблема разграничения доступа в виртуальных инфраструктурах является очень актуальной. Существует работающая теоретическая модель, которая позволяет решить эту проблему. Однако реализация данной модели оказалась недостаточно гибкой для того, чтобы сохранить работоспособность после изменений протокола управления vCenter. Автором показано одно из возможных решений проблемы сохранения работоспособности в случае изменений в виртуальной инфраструктуре, а также выделены особенности этого решения, которые позволяют успешно его реализовать.

Литература

1. Топ-5 крупнейших утечек с начала года. Kaspersky Lab [Электронный ресурс]. Режим доступа: https://www.kaspersky.ru/blog/data-leaks-2017/18993/ (дата обращения: 12.04.2018).
2. Приказ № 17. "Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах". Утв. ФСТЭК России 11.02.2013.
3. Приказ № 21. "Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных". Утв. ФСТЭК России 18.02.2013.
4. *Угаров Д. В., Постоев Д. А.* Проблемы реализации разграничения доступа к функциям управления виртуальных сред // Вопросы защиты информации. 2016. № 3 (114). С. 34—35.
5. ESXi Security [Электронный ресурс]. Режим доступа https://www.hytrust.com/solutions/private-cloud-controls/esxi/ (дата обращения: 12.04.2018).
6. vGate [Электронный ресурс]. Режим доступа: https://www.securitycode.ru/products/vgate/ (дата обращения: 12.04.2018).
7. ПАК Сегмент-В. [Электронный ресурс]. Режим доступа: http://www.accord.ru/segment-v.html/ (дата обращения: 12.04.2018).
8. Goodbye, vSphere Web Client! [Электронный ресурс]. Режим доступа: https://blogs.vmware.com/vsphere/2017/08/goodbye-vsphere-web-client.html (дата обращения: 12.04.2018).
9. *Конявская С. В., Угаров Д. В., Постоев Д. А.* Инструмент контроля доступа к средствам управления виртуальной инфраструктурой [Электронный ресурс]. Режим доступа: http://www.okbsapr.ru/konyavskaya_2016_2.html (дата обращения: 12.04.2018).

Features of monitoring the interaction of the client VMware vSphere 6.5 with vCenter

P. M. Zhurov

Moscow Institute of Physics and Technology (State University), Dolgoprudny, Moscow region, Russia

This article describes the work to analyze traffic between the VMware vSphere 6.5 client and vCenter, determination, based on this analysis, the virtual infrastructure management protocol and developing an access control mechanism for this virtual infrastructure.

Keywords: access delineation, virtual infrastructure, traffic analysis.

Bibliography — 9 references. *Received June 13, 2018*