МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра технологий обработки и защиты информации

Реферат
Уязвимости облачной инфраструктуры

Направление 10.03.01 Информационная безопасность

Технологии обработки и защиты информации

Зав. кафедрой _____ *д.т.н., профессор, А.А. Сирота*

Обучающийся _____ *С.А. Барышников, 4 курс, д/о*

Руководитель _____ *к.т.н., доцент, М.А. Дрюченко*

Воронеж 2023

Оглавление

Введение 3

1. Уязвимость Google Диска……3

2. DDoS-атаки, незащищенные интерфейсы и API 5

3. Потеря данных 6

4. Кража данных..... 7

5. Пример взлома облака в Azure и AWS..... 8

Введение

В нынешнее время, понятие "Облако" знакомо почти каждому. Облачное хранилище – это модель облачных вычислений, предусматривающая хранение данных в Интернете с помощью поставщика облачных вычислительных ресурсов, который предоставляет хранилище данных, как сервис и обеспечивает управление им.

Облачные хранилища расположены в ЦОД, что позволяет каждому пользователю имеющему доступ к интернету пользоваться хранилищем. Функционал данных "Облаков" очень разнообразный. Не только пользователи могут использовать хранилище, для личных целей, но и небольшие компании, которые не в силах организовать собственную структуру. Но чем больше данных начинают хранить пользователи, тем чаще злоумышленники стараются украсть информацию из сервисов. Поэтому необходимо периодически проверять целостность безопасности облачного хранилища.

1. Уязвимость Google Диска

Уязвимость в функции «Управление версиями» может использоваться для осуществления эффективных фишинговых атак.

В облачном сервисе Google Диск обнаружена уязвимость, которой могут воспользоваться злоумышленники для подмены легитимных документов или изображений их вредоносными версиями, что открывает возможность для осуществления целевых фишинговых атак.

Проблема кроется в функции «Управление версиями» (manage versions), позволяющей пользователям загружать и управлять различными версиями файла, а также в том, как интерфейс предлагает новую версию файла.

Теоретически, функциональность «Управление версиями» должна предоставлять пользователям возможность обновить старый файл новой версией, имеющей то же расширение, но в действительности, сервис позволяет загружать новую версию хранящегося в облаке файла с любым расширением. Таким образом, злоумышленник может подменить любой файл вредоносным вариантом, причем при предпросмотре он будет казаться совершенно безобидным.

Google разрешает изменять версию файла, но не проверяет, тот ли это тип файла и имеет ли он то же расширение.

2. DDoS-атаки, незащищенные интерфейсы и API

На облако могут быть предприняты атаки типа «отказ в обслуживании», которые вызывают перегрузку инфраструктуры, заставляя задействовать огромный объем системных ресурсов и не давая заказчикам пользоваться этой услугой. Внимание прессы чаще всего привлекают распределенные, или DDoS-атаки, но есть и другие типы DoS-атак, которые могут блокировать облачные вычисления, пишет CSA. К примеру, злоумышленники могут запустить асимметричные DoS-атаки прикладного уровня, используя уязвимости в Web-серверах, базах данных или других облачных ресурсах, чтобы завалить приложение с очень малой полезной нагрузкой.

Слабые интерфейсы ПО или Application Programming Interface (API), используемые заказчиками для управления и взаимодействия с облачными услугами, подвергают организацию целому ряду угроз, пишет CSA. Эти интерфейсы должны быть правильно спроектированы и обязательно включать аутентификацию, управление доступом и шифрование, чтобы обеспечить необходимую защиту и готовность облачных услуг.

CSA добавляет также, что организации и сторонние подрядчики часто используют облачные интерфейсы для предоставления дополнительных услуг, что делает их более сложными и увеличивает риск, поскольку может потребоваться, чтобы заказчик сообщил свои регистрационные данные такому подрядчику для упрощения предоставления услуг.

3. Потеря данных

Данные, хранящиеся в облаке, могут быть украдены злоумышленниками или потеряны по другой причине, пишет CSA. Если поставщик облачных услуг не внедрит должные меры резервного копирования, данные случайно может удалить сам провайдер или они пострадают при пожаре или стихийном бедствии. С другой стороны, заказчик, который шифрует данные до того, как выгрузить их в облако, вдруг потерявший шифровальный ключ, также утратит свои данные, добавляет CSA.

Опасение обосновано, но проблем можно избежать резервным копированием. Компании, которые заботятся о клиентах и о репутации, ежедневно и не менее двух раз автоматически копируют базу данных. Таким образом, если пользователь обратиться в техподдержку с сообщением о случайно удаленных, но важных файлах, их можно будет восстановить.

Такая проблема также должна решаться превентивно, со стороны пользователя, и относится к вопросу инструктажа и компьютерной грамотности коллег, а также ограничением прав доступа к изменению и удалению файлов.

4. Кража данных

Кража конфиденциальной корпоративной информации - всегда страшит организации при любой ИТ-инфраструктуре, но облачная модель открывает «новые, значительные магистрали атак», указывает CSA. «Если база данных облака с множественной арендой не продумана должным образом, то изъян в приложении одного клиента может открыть взломщикам доступ к данным не только этого клиента, но и всех остальных пользователей облака», - предупреждает CSA.

У любого «облака» есть несколько уровней защиты, каждый из которых защищает информацию от разного типа «покушений».

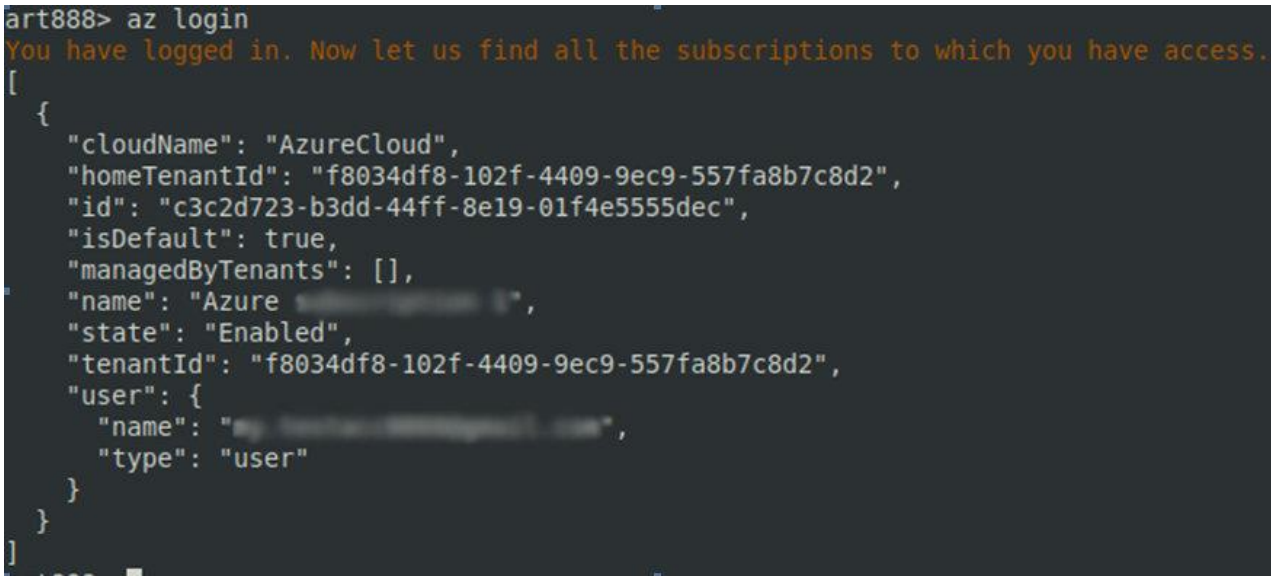
Так, например, физическая защита сервера. Здесь речь идет даже не о взломе, а о воровстве или порче носителей информации. Вынести сервер из помещения может быть тяжело в прямом смысле этого слова. Кроме этого, любая уважающая себя компания хранит информацию в дата-центрах с охраной, видеонаблюдением и ограничением доступа не только посторонним, но и большинству сотрудников компании. Так что вероятность того, что злоумышленник просто придет и заберет информацию, близка нулю.

Подобно тому как опытный путешественник, опасаясь ограблений, не хранит все деньги и ценности в одном месте, SaaS-компании не держат всю информацию на одном сервере. Так, взлом, даже если он произойдет, становится куда менее болезненным. Чем он грозит пользователю? Практически, ничем. Как показывает практика, чаще всего при взломе сервера воруют базу email-адресов. Это значит, что пользователь получит на почтовый ящик долю спама. И всё.

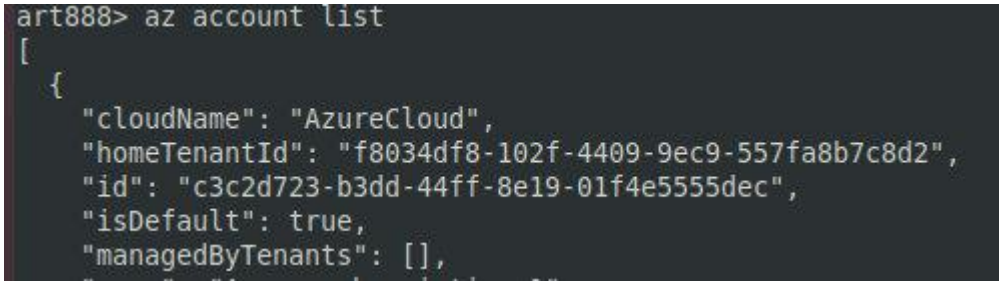
Второй уровень защиты «облаков» – это защита в процессе передачи данных. SaaS-компании шифруют весь траффик с помощью https-протокола с использованием SSL-сертификата. Так данные будут в безопасности от попыток анализаторов траффика перехватить их.

5. Пример взлома облака в Azure и AWS
Сбор данных с помощью утилиты Az

Azure CLI — это набор команд для создания ресурсов Azure и управления ими. Azure CLI доступен в различных службах Azure и предназначен для быстрой работы с ними. Для входа в профиль Azure по умолчанию используется команда az login.

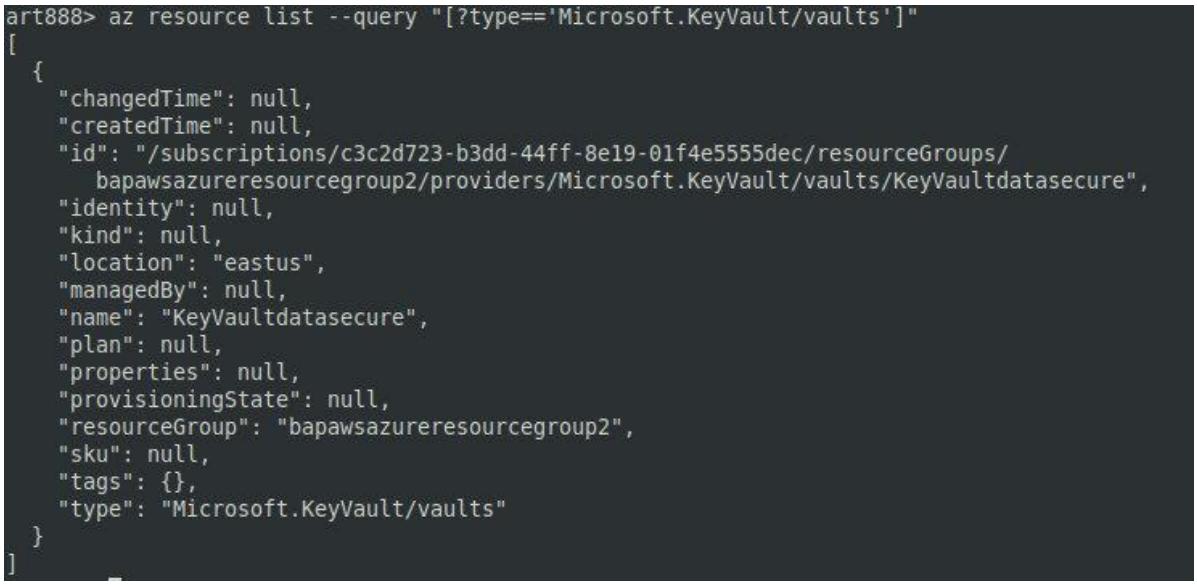


Как оказалось, в нашем случае линукс-сервер имеет доступ к инфраструктуре Azure. Какую полезную информацию из этого можно извлечь? Для начала с помощью команды az account list получим перечень подписок авторизированного пользователя.



С помощью директивы az account show --query "id" можно получить ID учетной записи. А команда - az resource list --query "[?type=='Microsoft.KeyVault/vaults']" позволяет узнать данные о Key Vault. Key Vault — это служба, которая помогает хранить ключи в аппаратных модулях безопасности (HSM), зашифровывая ключи и небольшие секретные данные, например пароли. Очевидно, что Key Vault должен быть правильно

настроен, иначе может произойти нечто подобное тому, что показано на скриншоте ниже.



А вот еще несколько команд, позволяющих выудить из Azure нужные взломщику сведения:

- az resource show --id /subscriptions/… | grep -E enablePurgeProtection|enableSoftDelete — проверить, можно ли восстановить Key Vault;
- az keyvault secret list --vault-name name --query [*].[{"name":attributes.name},{"enabled":attributes.enabled},{"expires":attributes.expires}] — проверить, когда секретный ключ Key Vault истекает;
- az keyvault secret list --vault-name KeyVaultdatasecure --query '[].id' — получить URL для Key Vault;
- az keyvault secret show --id — получить данные, хранящиеся в Key Vault;
- az network nsg list --query [*].[name,securityRules] — получить данные о политике безопасности для сети Azure.

```
art888> az keyvault secret show --id https://keyvaultdatasecure.vault.azure.net/secrets/ExamplePassword
{
  "attributes": {
    "created": "2020-06-19T15:48:58+00:00",
    "enabled": true,
    "expires": "2022-06-19T15:23:38+00:00",
    "notBefore": "2020-06-19T15:48:47+00:00",
    "recoveryLevel": "Recoverable+Purgeable",
    "updated": "2020-06-19T15:48:58+00:00"
  },
  "contentType": null,
  "id": "https://keyvaultdatasecure.vault.azure.net/secrets/ExamplePassword/7d3cfcd4b1ad4d11aa6b567fc939993e",
  "kid": null,
  "managed": null,
  "name": "ExamplePassword",
  "tags": null,
  "value": "hVFkk965BuUv"
}
```

С помощью приведенных выше команд можно увидеть детали настроек политик безопасности для сети, к примеру название политики, группы, конфигурации. Обрати внимание на
теги access, destinationPortRange, protocol и direction. Они показывают, что на сервере разрешены внешние подключения. Установка удаленного доступа к C&C значительно облегчает задачу атакующему и повышает шансы остаться незамеченным.

Сбор данных с помощью утилиты AWS

AWS CLI — это единый инструмент для управления сервисами AWS. Загружаешь всего одно средство — и можешь контролировать множество сервисов AWS из командной строки и автоматизировать их с помощью скриптов.

Если утилита AWS установлена на скомпрометированной машине, можно проверить, сконфигурирован ли AWS-профиль. Конфигурационные данные к AWS на компьютерах под управлением Linux хранятся в файле ~/.aws/credentials, а в Windows — в C:\Users\USERNAME\.aws\credentials. Этот файл может содержать данные к учетной записи AWS (access key ID, secret access key и session token). Полученную таким образом учетную запить можно использовать для

удаленного доступа в будущем.

С помощью следующих команд AWS CLI мы получим важную информацию о развернутой в сети облачной инфраструктуре:

- aws sts get-caller-identity — получить данные об используемой учетной записи;
- aws iam list-users — перечислить всех IAM-пользователей;
- aws s3 ls — перечислить все доступные AWS S3;
- aws lambda list --functions — перечислить все lambda-функции;
- aws lambda get-function --function-name [function_name] — собрать дополнительную информацию по lambda-переменным, локации и так далее;
- aws ec2 describe-instances — перечислить все доступные виртуальные машины;
- aws deploy list-applications — перечислить все доступные веб-сервисы;
- aws rds describe-db-instances — показать все доступные базы данных RDS.

Существуют и другие методы сбора информации на скомпрометированной системе. К примеру, можно воспользоваться командой history и посмотреть, какие команды выполнялись за последнее время на этой машине.