Искупитель

Подготовил: dotguy

Введение

Базы данных представляют собой набор организованной информации, к которой можно легко получить доступ, управлять и обновлять. В большинстве сред системы баз данных очень важны, поскольку они передают информацию, связанную с вашими транзакциями продаж, запасами продуктов, профилями клиентов и маркетинговой деятельностью.

Существуют различные типы баз данных, и одна из них — Redis, которая представляет собой базу данных «в памяти». Базы данных в памяти — это те, которые в основном полагаются на основную память для хранения данных (это означает, что база данных управляется в оперативной памяти системы); в отличие от баз данных, которые хранят данные на диске или SSD. Поскольку первичная память значительно быстрее, чем вторичная память, время извлечения данных в случае баз данных «в памяти» очень мало, что обеспечивает очень эффективное и минимальное время отклика.

Базы данных в памяти, такие как Redis, обычно используются для кэширования данных, которые часто запрашиваются для быстрого поиска. Например, если есть веб-сайт, который возвращает некоторые цены на главной странице сайта. Веб-сайт может быть написан так, чтобы сначала проверить, есть ли нужные цены в Redis, а если нет, то проверить традиционную базу данных (например, MySQL или MongoDB). Когда значение загружается из базы данных, оно затем сохраняется в Redis в течение более короткого периода времени (секунды, минуты или часы) для обработки любых аналогичных запросов, поступающих в течение этого периода времени. Для сайта с большим трафиком эта конфигурация обеспечивает гораздо более быстрое извлечение большинства запросов, сохраняя при этом стабильное долгосрочное хранение в основной базе данных.

В этом лабораторном занятии основное внимание уделяется удаленному перечислению сервера Redis, а затем созданию дампа его базы данных для получения флага. В этом процессе мы узнаем об использовании `Redis-Cli` утилита командной строки, которая помогает взаимодействовать с сервисом Redis. Мы также узнаем о некоторых основных командах redis-cli, которые используются для взаимодействия с сервером Redis и базой данных «ключ-значение».

Давайте теперь погрузимся прямо в лабораторию.

перечисление

Чтобы проверить подключение и доступность цели, мы можем запустить `пинг` команда с IP адрес целевой машины. После двух успешных ответов мы можем прервать команду ping, так как качество связи нас устраивает. Нам не всегда нужно долго запускать команды. Иногда получение фрагмента результата или обзора вместо подробного отчета является более эффективным по времени, чем альтернатива.

```
$ ping {target_IP}

PING {target_IP} ({target_IP}) 56(84) bytes of data.
64 bytes from {target_IP}: icmp_seq=1 ttl=63 time=16.3 ms
64 bytes from {target_IP}: icmp_seq=2 ttl=63 time=15.7 ms
^C
--- {target_IP} ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
rtt min/avg/max/mdev = 15.740/16.043/16.346/0.303 ms
```

Сразу после этого мы можем провести предварительное сканирование цели. Используя переключатель `-sV` с соответствующим определения версии службы, мы сканируем IP-адрес на наличие открытых портов и служб.

> - sV: исследовать открытые порты для определения информации о сервисе/версии.

```
$ nmap -p- -sV {target_IP}

Starting Nmap 7.92 ( https://nmap.org )
Nmap scan report for {target_IP}
Host is up (0.22s latency).
Not shown: 65534 closed tcp ports (conn-refused)

PORT     STATE SERVICE
6379/tcp open  redis   Redis key-value store 5.0.7
```

Из результатов сканирования мы можем сделать вывод, что на целевой машине открыт только один порт, т. е. `порт 6379` который работает сервер Redis.

Что такое Редис?

Редис (**RE**соринка**DI**словарь**C**erver) — это расширенное хранилище данных NoSQL с открытым исходным кодом, используемое в качестве базы данных, кэша и брокера сообщений. Данные хранятся в формате словаря с парами ключ-значение. Обычно он используется для краткосрочного хранения данных, которые требуют быстрого поиска. Redis выполняет резервное копирование данных на жесткие диски для обеспечения согласованности.

Сервер

Redis работает как серверное программное обеспечение, поэтому его основные функции находятся в его серверном компоненте. Сервер прослушивает подключения от клиентов программно или через интерфейс командной строки.

CLI

Интерфейс командной строки (CLI) — это мощный инструмент, который дает вам полный доступ к данным Redis и его функциям, если вы разрабатываете программное обеспечение или инструмент, который должен с ним взаимодействовать.

База данных

База данных хранится в оперативной памяти сервера для обеспечения быстрого доступа к данным. Redis также записывает содержимое базы данных на диск с различными интервалами, чтобы сохранить его в качестве резервной копии на случай сбоя.

УстановкаRedis-Cli

Теперь, чтобы иметь возможность удаленно взаимодействовать с сервером Redis, нам `Redis-Cli` полезность. Это нужно загрузить его, используя следующую команду:

```
судоapt установить Redis-инструменты
```

Кроме того, мы также можем подключиться к серверу Redis с помощью утилиты netcat, но мы будем использовать `редис-Кли` в этой записи, как более удобно использовать.

Перечисление сервера Redis

После успешной установки `Redis-Cli` утилиты, давайте просмотрим ее справочную страницу, введя `редис-кли --помощь` команду в нашем терминале, чтобы получить список всех возможных переключателей для инструмента и их описание.

```
редис-кли --помощь

Использование: redis-cli[ОПЦИИ ][команда [аргумент [аргумент ...]]]
  - ч<имя хоста>      Имя хоста сервера (по умолчанию: 127.0.0.1). Порт
  - п<порт>           сервера (по умолчанию: 6379).
  - с<сокет>          Сокет сервера (переопределяет имя хоста и порт). Пароль
  - а<пароль>         для использования при подключении к серверу.
  - р<повтор>         Выполнить указанную команду N раз.
  - я<интервал>       Когда используется -r, ждет<интервал>секунд на команду. Можно
                      указать время меньше секунды, например -i 0.1. Номер базы данных.
  - н<дБ>
  - Икс               Прочитать последний аргумент из STDIN.
  - д<разделитель>    Многокомпонентный разделитель для необработанного форматирования (по умолчанию: \n).
  - с                 Включите режим кластера (следуйте перенаправлениям -ASK и -MOVED). Использовать
  - - сырой           необработанное форматирование для ответов (по умолчанию, если STDOUT не является tty).

  ---не сырой         Принудительно отформатировать вывод, даже если STDOUT не является tty.
  --CSV               Вывод в формате CSV.
  - - стат            Распечатать скользящую статистику о сервере: память, клиенты, ...
```

- - задержка	Войдите в специальный режим непрерывной выборки задержки. Аналогично --
- - история задержки	latency, но отслеживание изменений задержки с течением времени. Интервал времени по умолчанию составляет 15 секунд. Измените его с помощью -i.
- - латентность-расстояние	Показывает задержку в виде спектра, требуется xterm 256 цветов. Интервал времени по умолчанию составляет 1 секунду. Измените его с помощью -i. Смоделируйте
- - lru-тест<клавиши>	рабочую нагрузку кэша с распределением 80-20. Имитация ведомого,
- - раб	показывающего команды, полученные от ведущего. Перенесите дамп RDB с
- - РДБ<имя файла>	удаленного сервера в локальный файл. Перенесите необработанный протокол Redis
- - трубка	со стандартного ввода на сервер.
- - тайм-аут трубы<н>	В режиме --pipe прерывание с ошибкой после отправки всех данных. ответ не получен в течение<н>секунды. Тайм-аут по умолчанию: 30. Используйте 0, чтобы ждать вечно.
-- большие ключи	Пример ключей Redis, ищущих большие ключи.
- - сканирование	Перечислите все ключи с помощью команды SCAN.
- - шаблон<pat>	Полезно с --scan для указания шаблона SCAN.
- - внутренняя задержка<сек>	Запустите тест для измерения внутренней задержки системы. Тест будет выполняться в течение указанного количества секунд.
- - оценка<файл>	Отправьте команду EVAL с помощью сценария Lua по адресу<файл>.
- - помощь	Выведите эту справку и выйдите.
- - версия	Выходная версия и выход.

В нашем случае нам нужно будет использовать только следующий переключатель для указания хоста, к которому нам нужно подключиться:

```
- ч<имя хоста>: укажите имя хоста цели для подключения
```

Давайте подключимся к серверу Redis, используя следующую команду:

```
Redis-Cli-час{целевой_IP}
```

После успешного подключения к серверу Redis мы должны увидеть приглашение в терминале, как показано на изображении выше.
Одна из основных команд перечисления Redis: Информация который возвращает информацию и статистику о Редис-сервер. Поскольку вывод этой команды довольно длинный, я вырезал менее важную информацию:

```
10.10.10.127:6379> info

# Server
redis_version:5.0.7
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:66bd629f924ac924

[** SNIP **]

# Keyspace
db0:keys=4,expires=0,avg_ttl=0
```

The **ключевое пространство**Раздел предоставляет статистику по основному словарю каждой базы данных. Статистика включает количество ключей и количество ключей с истечением срока действия.

В нашем случае под `Ключевое пространство` мы видим, что существует только одна база данных с индексом. `0`

Давайте выберем эту логическую базу данных Redis, используя `Выбрать` команда, за которой следует порядковый номер базу данных, которую необходимо выбрать:

```
выберите 0
```

```
10.10.10.127:6379> select 0
OK
```

Кроме того, мы можем перечислить все ключи, присутствующие в базе данных, с помощью команды:

```
ключи *
```

```
10.10.10.127:6379> keys *
1) "temp"
2) "stor"
3) "numb"
4) "flag"
```

Наконец, мы можем просмотреть значения, сохраненные для соответствующего ключа, используя ключевую `получить` команда, за которой следует пометку:

```
получить<ключ>
```

```
10.10.10.127:6379> get temp
"1c98492cd337252698d0c5f631dfb7ae"

10.10.10.127:6379> get stor
"e80d635f95686148284526e1980740f8"

10.10.10.127:6379> get numb
"bb2c8a7506ee45cc981eb88bb81dddab"

10.10.10.127:6379> get flag
"03e1d2b376c37ab3f5319922053953eb"
```

Поздравляем! Мы успешно получили значение флага из базы данных Redis.
