# Tier 0 - Redeemer

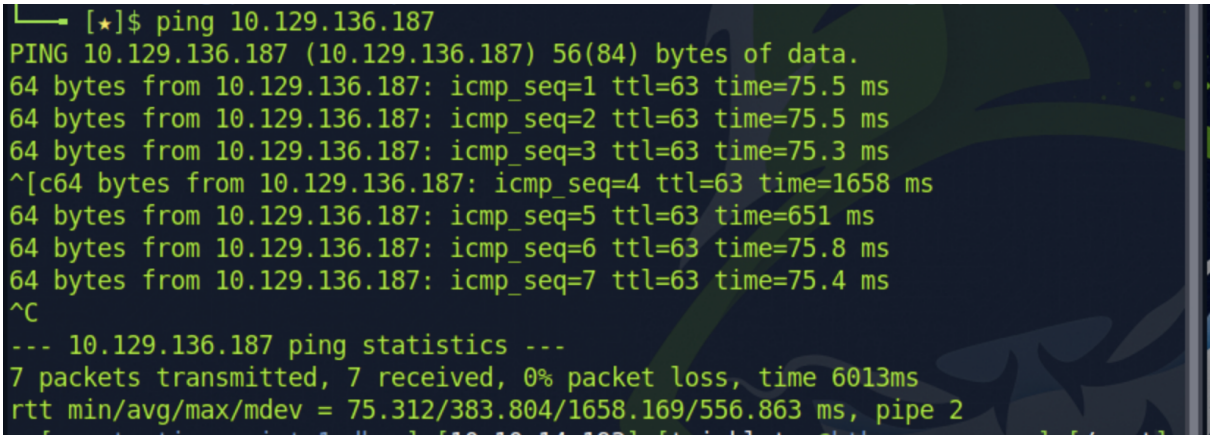
In this lab, the primary focus is on remotely enumerating a Redis server and subsequently extracting its database to obtain the flag. Throughout this exercise, we become familiar with the redis-cli command line utility, which facilitates interaction with the Redis service. Additionally, we gain insights into fundamental redis-cli commands utilized for interacting with the Redis server and its key-value database.

## Enumeration

Using the ping command we will check if the target is reachable and responsive.

```
ping {target_IP}

control+C to cancel
```



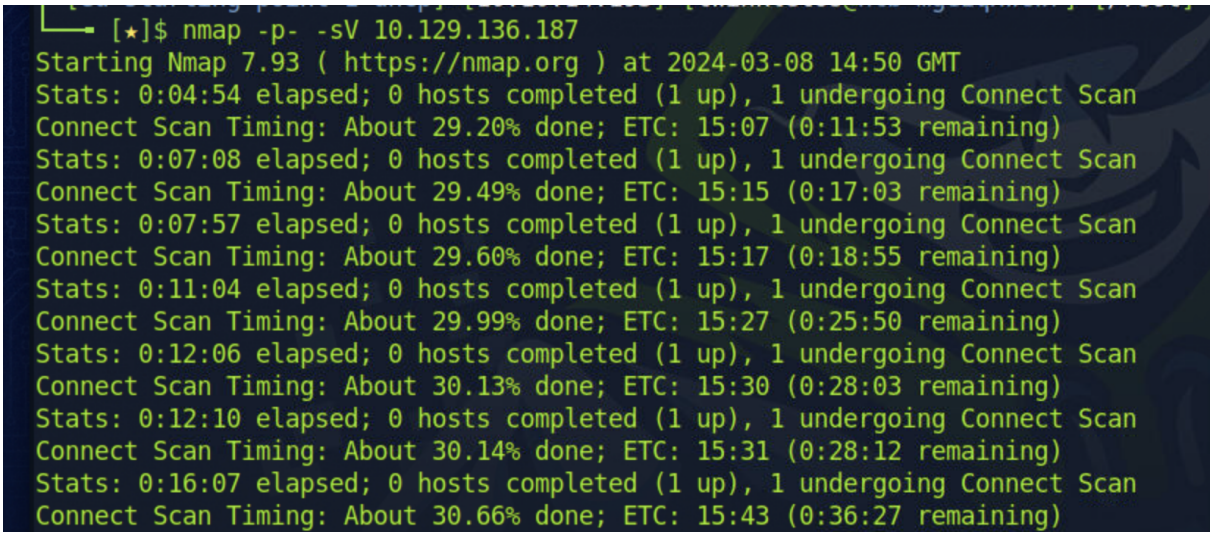

we can see a good connection. lets move onto the next step.

Using nmap we will see what ports are open and versions

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

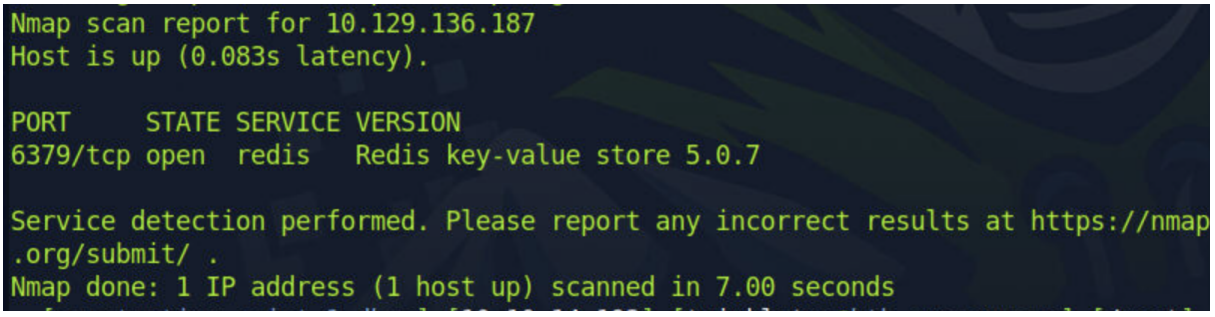
-sV Attempts to determine the version of the service running on port

-p scan ports all ports




Hitting enter gives us an update on the ports scanned and ETCs.

After the scan is complete a similar result should appear.




We can see port 6397 is open running Redis server.

## What is Redis?

Redis, short for Remote Dictionary Server, is a versatile open-source NoSQL database, cache, and message broker. It stores data in a dictionary format with key-value pairs, making it efficient for quick data retrieval. Redis is commonly utilized for temporary data storage requiring rapid access. Additionally, it backs up data to hard drives to ensure consistency.

The database resides in the server's RAM for swift data access. Periodically, Redis also saves the database contents to disk to create backups, safeguarding against potential failures.

## Installing redis-cli

To install Redis cli utility we will use:

```
sudo apt install redis-tools
```

Now that  have updated the Redit cli ulility we can now interact with the Redis server.

Redis help page can be found at:

```
redis-cli --help
```

```
└─╼ [★]$ redis-cli --help
redis-cli 7.0.10

Usage: redis-cli [OPTIONS] [cmd [arg [arg ...]]]
  -h <hostname>      Server hostname (default: 127.0.0.1).
  -p <port>          Server port (default: 6379).
  -s <socket>        Server socket (overrides hostname and port).
  -a <password>      Password to use when connecting to the server.
                     You can also use the REDISCLI_AUTH environment
                     variable to pass this password more safely
                     (if both are used, this argument takes precedence).
  --user <username>  Used to send ACL style 'AUTH username pass'. Needs -a.
  --pass <password>  Alias of -a for consistency with the new --user option.
  --askpass          Force user to input password with mask from STDIN.
                     If this argument is used, '-a' and REDISCLI_AUTH
                     environment variable will be ignored.
  -u <uri>           Server URI.
  -r <repeat>        Execute specified command N times.
  -i <interval>      When -r is used, waits <interval> seconds per command.
                     It is possible to specify sub-second times like -i 0.1.
                     This interval is also used in --scan and --stat per cycle.
                     and in --bigkeys, --memkeys, and --hotkeys per 100 cycles.
  -n <db>            Database number.
  -2                 Start session in RESP2 protocol mode.
  -3                 Start session in RESP3 protocol mode.
  -x                 Read last argument from STDIN (see example below).
  -X                 Read <tag> argument from STDIN (see example below).
  -d <delimiter>     Delimiter between response bulks for raw formatting (defaul
t: \n).
  -D <delimiter>     Delimiter between responses for raw formatting (default: \n
```

Using the following command we can connect specifically to a hostname.

```
redis-cli -h {target_IP}
```

-h <hostname>

```
└─╼ [★]$ redis-cli -h 10.129.136.187
10.129.136.187:6379>
```

We have successfully connected to the redis server indicated by the prompt above.

Another fundamental Redis enumeration command is "info," which provides comprehensive information and statistics regarding the Redis server. However,

due to the extensive output generated by this command, it may contain a large amount of data. For now lets focus on the server and keyspace section.

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

```
# Keyspace
db0:keys=4,expires=0,avg_ttl=0
10.129.136.187:6379>
```

The keyspace section provides statistics on the main dictionary of each database. The statistics include the number of keys, and the number of keys with an expiration.

We know only one database exists with the index db0.

Using the select command we will select this database.

```
select 0
```

```
10.129.136.187:6379> select 0
OK
```

We can list all the keys present in the database using the command :

```
keys *
```

```
10.129.136.187:6379> keys *
1) "temp"
2) "flag"
3) "stor"
4) "numb"
```

Using the get command followed by the keynote we can view the values stored for a corresponding key.

```
get <key>
```


Under the flag key we can see the stored flag.