

MonGod_Walkthrough

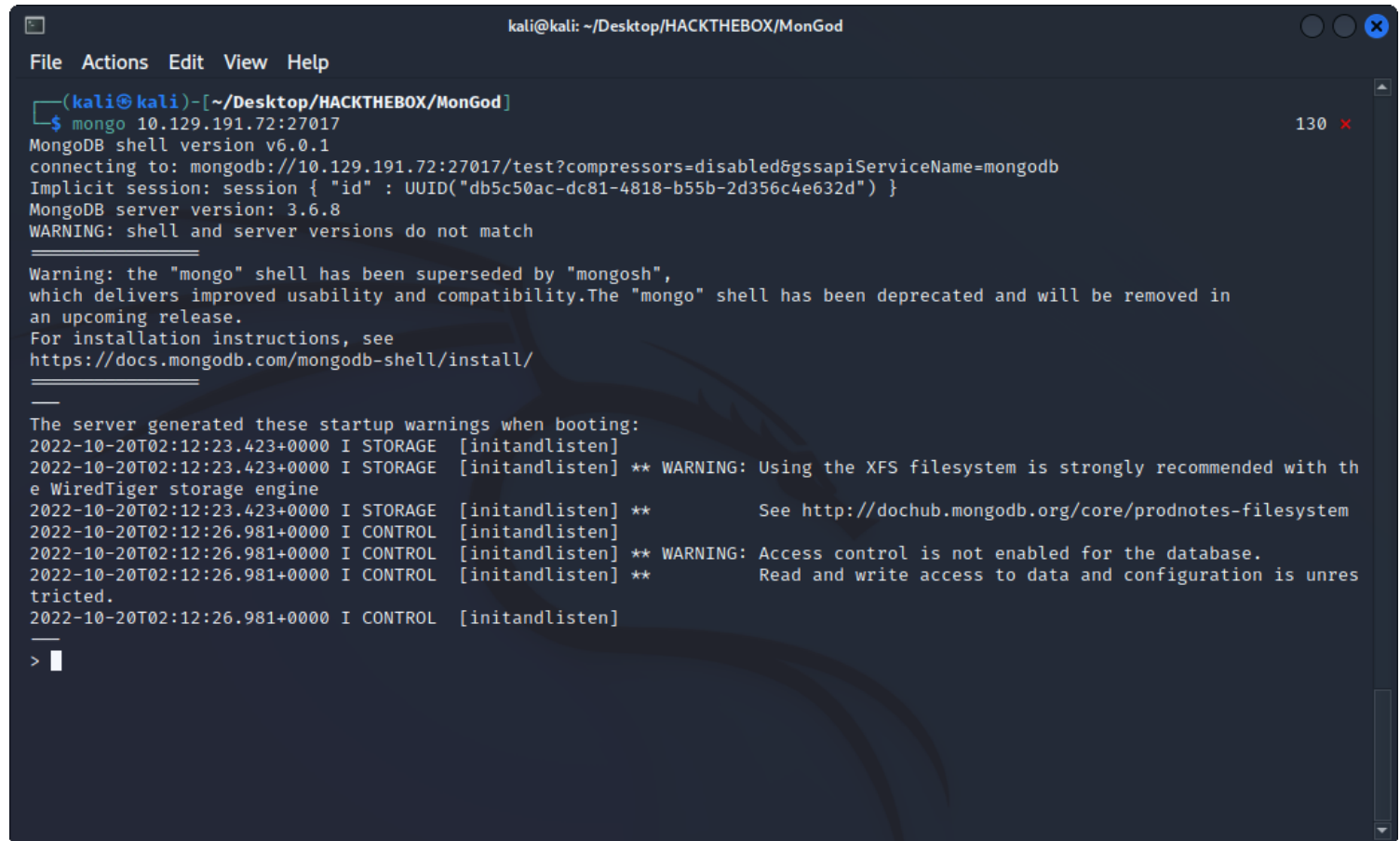
Nmap_Scan

```
nmap -T4 -A -p- 10.129.228.30 > Nmap_Scan
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 15:04 PDT
Warning: 10.129.228.30 giving up on port because retransmission cap hit (2).
Stats: 0:02:39 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 21.84% done; ETC: 15:15 (0:08:43 remaining)
Stats: 0:04:25 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 35.34% done; ETC: 15:16 (0:07:39 remaining)
Stats: 0:13:05 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 15:17 (0:00:00 remaining)
Stats: 0:14:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 15:18 (0:00:00 remaining)
Nmap scan report for 10.129.228.30
Host is up (0.22s latency).
Not shown: 65501 closed tcp ports (reset), 32 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
27017/tcp open  mongod  MongoDB 3.6.8 3.6.8
| mongodb-info:
|   MongoDB Build info
|     versionArray
|       3 = 0
|       2 = 8
|       1 = 6
|       0 = 3
|   storageEngines
|     3 = wiredTiger
|     2 = mmapv1
|     1 = ephemeralForTest
|     0 = devnull
|   gitVersion = 8e540c0b6db93ce994cc548f000900bdc740f80a
|   ok = 1.0
|   allocator = tcmalloc
|   maxBsonObjectSize = 16777216
|   bits = 64
|   debug = false
|   openssl
|     compiled = OpenSSL 1.1.1f 31 Mar 2020
|     running = OpenSSL 1.1.1f 31 Mar 2020
|   buildEnvironment
|     target_arch = x86_64
|     cxxflags = -g -O2 -fdebug-prefix-map=/build/mongodb-FO9rLu/
```

Notes

We can see that ports 22 ssh, 27017 MongoDB are open. lets see how we can connect and interact with this DB.

Mongo_Login

A terminal window titled 'kali@kali: ~/Desktop/HACKTHEBOX/MonGod' showing the process of connecting to a MongoDB instance. The user runs the 'mongo' command with the IP and port. The terminal displays the MongoDB shell version (v6.0.1), connection details, and server version (3.6.8). It also shows a warning about shell versions and a deprecation notice for the 'mongo' shell. Finally, it displays startup warnings from the server, including recommendations for the XFS filesystem and enabling access control.

```
kali@kali: ~/Desktop/HACKTHEBOX/MonGod
File Actions Edit View Help

(kali@kali)-[~/Desktop/HACKTHEBOX/MonGod]
$ mongo 10.129.191.72:27017
MongoDB shell version v6.0.1
connecting to: mongodb://10.129.191.72:27017/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("db5c50ac-dc81-4818-b55b-2d356c4e632d") }
MongoDB server version: 3.6.8
WARNING: shell and server versions do not match

Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/

The server generated these startup warnings when booting:
2022-10-20T02:12:23.423+0000 I STORAGE [initandlisten]
2022-10-20T02:12:23.423+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the
WiredTiger storage engine
2022-10-20T02:12:23.423+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten]
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten]

> 
```

Notes

Kali has mongo already installed, so just use the following to login.
mongo ip add:port Now lets navigate through the DB so we can find our flag.

Mongo_Commands

```
kali@kali: ~/Desktop/HACKTHEBOX/MonGod
File Actions Edit View Help

2022-10-20T02:12:23.423+0000 I STORAGE [initandlisten]
2022-10-20T02:12:23.423+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-10-20T02:12:23.423+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-10-20T02:12:26.981+0000 I CONTROL [initandlisten]

> help
db.help()                help on db methods
db.mycoll.help()          help on collection methods
sh.help()                 sharding helpers
rs.help()                 replica set helpers
help admin                administrative help
help connect              connecting to a db help
help keys                 key shortcuts
help misc                 misc things to know
help mr                   mapreduce

show dbs                  show database names
show collections           show collections in current database
show users                show users in current database
show profile              show most recent system.profile entries with time ≥ 1ms
show logs                 show the accessible logger names
show log [name]           prints out the last segment of log in memory, 'global' is default
use <db_name>             set current database
db.mycoll.find()           list objects in collection mycoll
db.mycoll.find( { a : 1 } ) list objects in mycoll where a = 1
it                         result of the last line evaluated; use to further iterate
DBQuery.shellBatchSize = x set default number of items to display on shell
exit                      quit the mongo shell
```

Notes

Now run the following commands to get your flag. You can get the commands you need from this link. <https://www.educba.com/mongodb-commands/?source=leftnav>

```
show dbs
use sensitive_information
show collections
db.flag.find().pretty()
```

With these commands you can get the root flag.

Root_Flag

```
kali@kali: ~/Desktop/HACKTHEBOX/MonGod
File Actions Edit View Help

help admin      administrative help
help connect    connecting to a db help
help keys       key shortcuts
help misc       misc things to know
help mr         mapreduce

show dbs        show database names
show collections show collections in current database
show users      show users in current database
show profile    show most recent system.profile entries with time ≥ 1ms
show logs       show the accessible logger names
show log [name] prints out the last segment of log in memory, 'global' is default
use <db_name>   set current database
db.mycoll.find() list objects in collection mycoll
db.mycoll.find( { a : 1 } ) list objects in mycoll where a = 1
it              result of the last line evaluated; use to further iterate
DBQuery.shellBatchSize = x set default number of items to display on shell
exit           quit the mongo shell

> show dbs
admin          0.000GB
config         0.000GB
local          0.000GB
sensitive_information 0.000GB
users          0.000GB
> use sensitive_information
switched to db sensitive_information
> show collections
flag
> db.flag.find().pretty()
{
  "_id" : ObjectId("630e3dbcb82540ebbd1748c5"),
  "flag" : "1b6e6fb359e7c40241b6d431427ba6ea"
}
> 
```

Notes

Querying collection is done by find() method.

As find() method will show the findings in a non-structured way, to get the results in a structured pretty() method is used

db.flag.find().pretty()

Thanks for reading I hope you enjoyed .