MongoDB Security: Making Things Secure by Default

Wed, Aug 9, 2017 11:00 AM - 12:00 PM PDT

Adamo Tonete,
Senior Technical Services Engineer

PERCONA

Recent Security Problems

MONGODB DATABASES HELD UP FOR RANSOM BY MYSTERIOUS ATTACKER

Security Newspaper | January 3, 2017 | Incidents, Malware | No Comments

KNOWLEDGE BELONGS TO THE WORLD

Groups of attackers have adopted a new tactic that involves deleting publicly exposed MongoDB databases and asking for money to restore them. In a matter of days, the number of affected databases has risen from hundreds to more than 10,000.

The issue of misconfigured MongoDB installations, allowing anyone on the internet to access sensitive data, is not new. Researchers have been finding such open databases for years, and the latest estimate puts their number at more than 99,000.

ANALYSIS

MongoDB ransomware attacks and lessons learned

{ me : 'twitter.com/adamotonete' }

Adamo Tonete

I've been working for Percona since late 2015 as a Senior Technical Services Engineer.

Based on São Paulo Brazil

Agenda

- Installing MongoDB;
- Enabling authentication and creating a root and a standard user;
- Default roles;
- Starting a replica-set and a sharded environment with authentication;
- SSL;
- LDAP;
- Audit plugin;
- Store backups safely.

Installing MongoDB

- How to install MongoDB on CentOS 7

```
yum install -y mongodb-org

service mongodb start

$ mongo
> use percona
switched to db percona

> db.foo.insert({x : 1})
WriteResult({ "nInserted" : 1 })
```

Installing MongoDB

By default MongoDB only listens to the 127.0.0.1:27017. We usually change this line to allow external connections.

```
net:
  port: 27017
  bindIp: 127.0.0.1  # Listen to local interface only, comment to listen on all interfaces.
```

Advantage: Everyone can log in

Disadvantage: Everyone can log in. No security or access control, it is not safe at all.

Installing MongoDB

- If it is not a local development with fake data, please make sure that authentication is enabled.
- Try to isolate the access as much as possible, several administrator users is almost the same as no user control.

Enabling authentication

- Creating a root user and restarting the mongod process.

```
mongo
use admin
> db.createUser({user : 'administrator', pwd : '123321', roles : ['root']})
Successfully added user: { "user" : "administrator", "roles" : [ "root" ] }

-- mongod.conf --
#security
security
  authorization : enabled

-- service restart ---

./mongod --auth
```

Enabling authentication

- Checking access:

```
Mongo

> show dbs
2017-04-05T13:39:30.040-0400 E QUERY    [thread1] Error: listDatabases failed:{
     "ok" : 0,
     "errmsg" : "not authorized on admin to execute command { listDatabases: 1.0 }",
     "code" : 13,
     "codeName" : "Unauthorized"
} :
```

Enabling authentication

- Checking access:

```
use admin
db.auth('administrator','123321')
1

mongo -u administrator -p --authenticationDatabase admin
password:


> show dbs
local
percona
```

Default Roles

- All the roles listed below come by default in the MongoDB database server

https://docs.mongodb.com/manual/reference/built-in-roles/

read	readWrite	dbAdmin	dbOwner	userAdmin
clusterAdmin	clusterManager	clusterMonitor	hostManager	backup
restore	readAnyDatabase	readWriteAnyDatabase		userAdminAnyDatabase
dbAdminAnyDatabase		root	__system	

User Defined Roles

- Users can create their own roles based on the default roles or from scratch to give access to specific collections or commands.

- https://www.percona.com/blog/2017/05/17/mongodb-authentication-and-roles-creating-your-first-personalized-role/

Creating a standard user

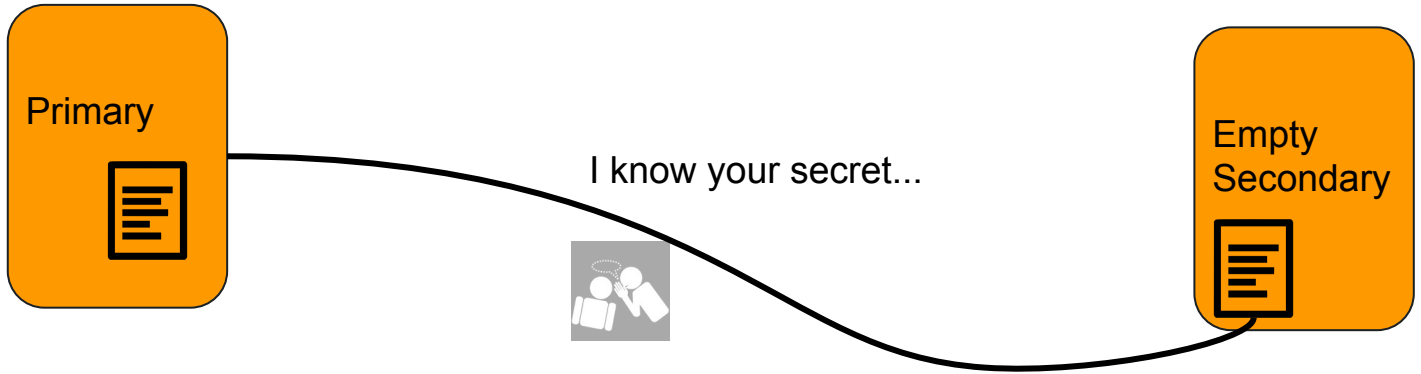
```
$ mongo

db.createUser({user: 'percona_user', pwd: '123', roles : [{ role :'read', db: 'percona'}]})

Successfully added user: {
     "user" : "percona_user",
     "roles" : [
          {
               "role" : "read",
               "db" : "percona"
          }
     ]
}
```

Testing the standard user access

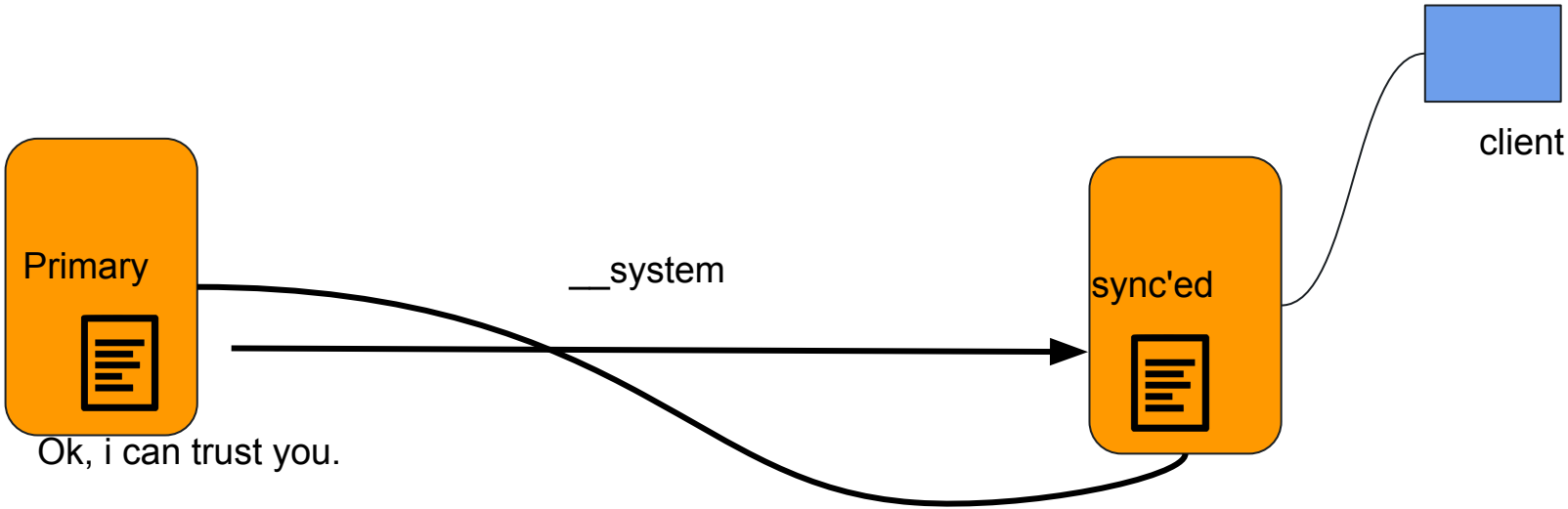
```
$mongo

use admin
switched to db admin
> db.auth('percona_user','123')
1

use percona
db.foo.find()
{ "_id" : ObjectId("58e52660bb89f8c29fb7b886"), "x" : 1 }

> show collections
foo
system.indexes
```

Starting a replica-set using keyfile

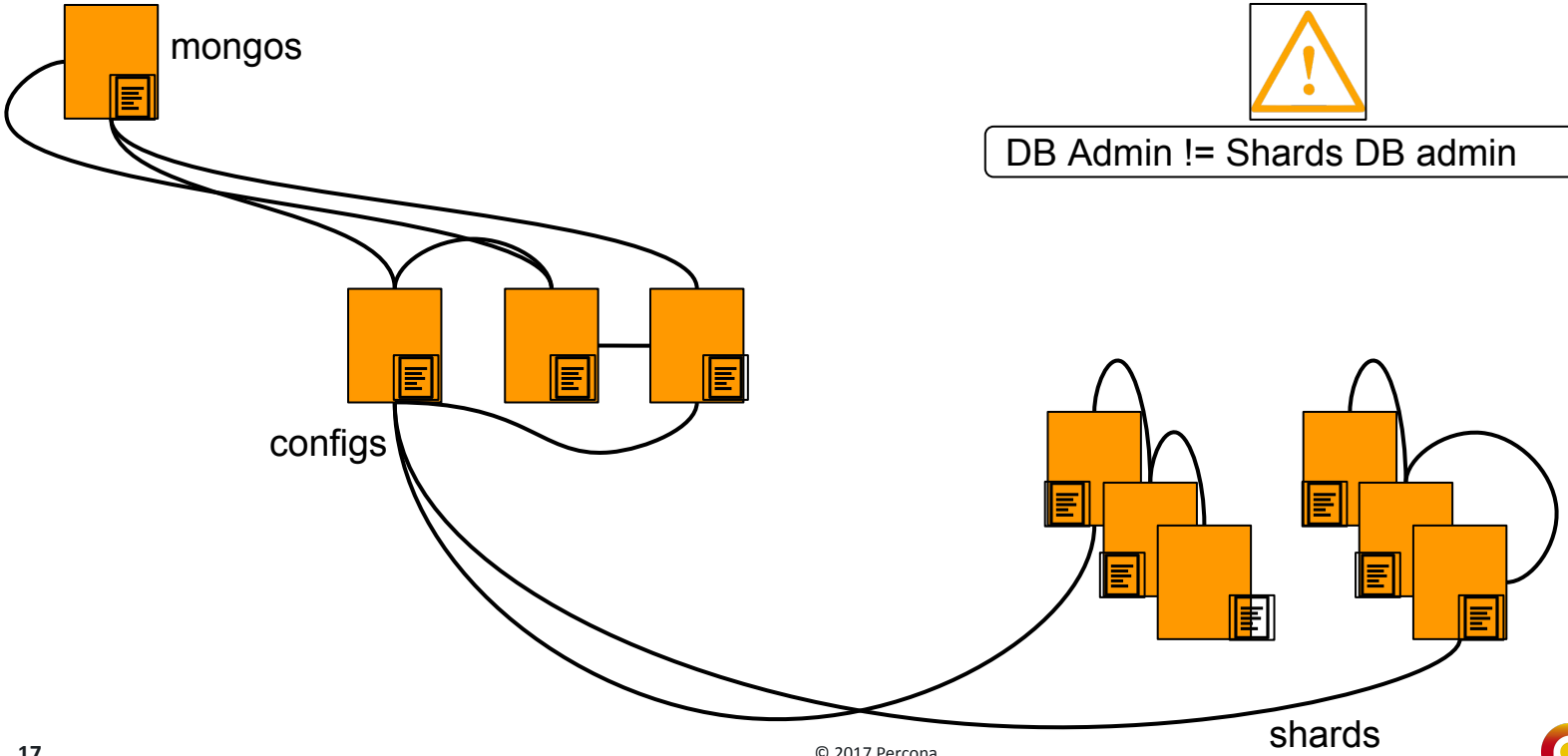
- Pre existing data instance with users in the admin database.



Starting a replica-set with key file



Starting a replicaset/shard using keyfile



Enhanced security

SSL (security socket layer)

- When using keyfiles, the information travels through the wire without any encryption.
- The SSL will encrypt both the communication among the mongod instances and the communication with clients.
- It does requires a CA and valid certificates.

Enhanced security

LDAP (Lightweight Directory Access Protocol)

- LDAP authentication is a free feature in the Percona Server for MongoDB and it allows companies to use a single password server to authenticate the users.

- This is very useful for big companies whose employee users and passwords are already managed by LDAP or Microsoft Active Directory.

Enhanced security

LDAP (Lightweight Directory Access Protocol)

- Instead of keeping the user/pass on the database only the user exists on the database using an external tool to authenticate.

- https://www.percona.com/blog/2017/03/16/percona-server-for-mongodb-dashing-new-ldap-authentication-plugin/

Enhanced security

Audit

- Percona server for mongodb comes with the audit plugin, this plugin is also free on our version it allows you to generate logs from a specific filter.

- https://www.percona.com/doc/percona-server-for-mongodb/LATEST/audit-logging.html

Enhanced security

Backups

- How do you store your backups?
- Do you use at least a password or encryption?
- Have you ever tested your backup?

Live Demo and Questions

DEMO

```
wget https://fastdl.mongodb.org/osx/mongodb-osx-ssl-x86_64-3.4.7.tgz
tar -xvzf mongodb-osx-ssl-x86_64-3.4.7.tgz
mv mongodb-osx-x86_64-3.4.7/ mongo34
cd mongo34/
cd bin/
./mongod --dbpath data --logpath data/log.log --fork --auth
./mongo
use admin
db.createUser({user : 'administrator', pwd : '123321', roles : ['root']})
db.auth('administrator','123321')

use percona
db.foo.insert({x : 1});

./mongo -u administrator -p --authenticationDatabase admin

use percona1
db.foo.insert({y : 1})

use percona2
db.foo.insert({z : 1})
use admin
```

```
db.createUser({user: 'percona_user', pwd: '123', roles : [{ role :'read', db:
'percona'}]})

Test;

db.createRole({
 role : 'read_perconaAndListDBs',
 privileges : [ {resource : {cluster : true}, actions : ["listDatabases"] }],
   roles : [{ role: "read", db: "percona"}]
})

db.revokeRolesFromUser('percona_user', [{ role :'read', db: 'percona'}])
db.grantRolesToUser('percona_user', ["read_perconaAndListDBs"])
```

Get Your Tickets for Percona Live Europe!

Championing Open Source Databases

- MySQL, MongoDB, Open Source Databases
- Time Series Databases, PostgreSQL, RocksDB
- Developers, Business/Case Studies, Operations
- September 25-27th, 2017
- Radisson Blu Royal Hotel, Dublin, Ireland

Advanced Reg Tickets Available Until Sep 5th!

Database Performance Matters