

MongoDB Authentication (VM, PC, SCA)

Thank you for your interest in authenticated scanning! When you configure and use authentication, you get a more in-depth assessment of your hosts, the most accurate results and fewer false positives. This document provides tips and best practices for setting up MongoDB authentication for compliance scans.

A few things to consider

Which technologies are supported?

MongoDB 3.x, MongoDB 4.x

Why should I use authentication?

With authentication we can remotely log in to each target system with credentials that you provide, and because we're logged in we can do more thorough testing. This will give you better visibility into each system's security posture. Is it required? Yes, required for compliance scans.

Are my credentials safe?

Yes, credentials are exclusively used for READ ACCESS ONLY to your system. The service does not modify or write anything on the device in any way. Credentials are securely handled by the service and are only used for the duration of the scan.

What are the steps?

First, set up a MongoDB user account and privileges (on target hosts) for authenticated scanning. Then, using Qualys Policy Compliance, complete these steps: 1) Add a MongoDB authentication record, 2) Launch a compliance scan, and 3) Run the Authentication Report to view the authentication status (Passed or Failed) for each scanned host.

MongoDB Credentials

We've provided a set of scripts below to help you set up an account and privileges which must exist prior to running scans. Note - These scripts require a super-user account which has privilege to createRole, createUser and grantRole. For example, accounts with userAdmin or dbOwner role.

Please run the scripts provided, in the order shown. The role and scan account needs to be created in the admin database to run successfully.

1) Create a Role for the Scan Account within the 'MongoDB' Database

This script creates a role for the user account to be used for scanning. It also grants privileges to the role needed for successful authentication and compliance scanning. We recommend creating a role called `qualys_Role` and provide a password before running the script.

```
use admin
db.createRole(
  {
    role: "qualys_Role",
    privileges: [
      { resource: { db: "", collection: "" }, actions: [ "viewRole", "viewUser" ]},
      { resource: { "cluster" : true }, actions: [ "getCmdLineOpts" ]},
      { resource: { db: "admin", collection: "system.users" }, actions: [ "find" ]},
      { resource: { db: "admin", collection: "system.roles" }, actions: [ "find" ]}
    ],
    roles: []
  }
)
```

2) Create a User Account

This script creates a user account to be used for scanning. Please provide a password before running the script. The script also grants the role created in Step 1 (`qualys_Role`) to the account.

We recommend you create an account called `qualys_scan` and provide a password before running the script.

```
use admin
db.createUser(
  {
    user: "qualys_scan",
    pwd: "<password>",
    roles: [ "qualys_Role" ]
  }
)
```

If a user identified by its X509 subject is created for scanning, please grant the role created in Step 1 (`qualys_Role`) to the user account.

3) Verify Privileges on the Scan Account

Verify that the `qualys_scan` account has all the privileges in the admin database to run a successful compliance scan. Log into the instance using the “`qualys_scan`” account, then run the following queries to see if access is available to the account.

```
3a)
use admin
db.runCommand({getCmdLineOpts:1})
```

Sample Expected Output:

```
{
  "argv" : [
    "/usr/bin/mongod",
    "--config",
    "/etc/mongodbl.conf"
  ],
  "parsed" : {
    "config" : "/etc/mongodbl.conf",
    "net" : {
```

```

        "port" : 27017
      }
    },
    "security" : {
      "authorization" : "enabled"
    },
    "storage" : {
      "dbPath" : "/usr/local/mongodb1/data",
      "journal" : {
        "enabled" : true
      }
    },
    "systemLog" : {
      "destination" : "file",
      "logAppend" : true,
      "path" : "/var/log/mongodb1.log",
      "quiet" : false
    }
  },
  "ok" : 1
}

```

3b)

```

use admin
db.runCommand({"find":"system.users","filter":{},limit:1,"projection":{"user":1,"_id":0}})

```

Sample Expected Output:

```

{
  "cursor" : {
    "firstBatch" : [
      {
        "user" : "qualys_scan"
      }
    ],
    "id" : NumberLong(0),
    "ns" : "admin.system.users"
  },
  "ok" : 1
}

```

3c)

```

use admin
db.runCommand({"find":"system.roles","filter":{},limit:1,"projection":{"role":1,"_id":0}})

```

Sample Expected Output:

```

{
  "cursor" : {
    "firstBatch" : [
      {
        "role" : "qualys_Role"
      }
    ],
    "id" : NumberLong(0),
    "ns" : "admin.system.roles"
  },
  "ok" : 1
}

```

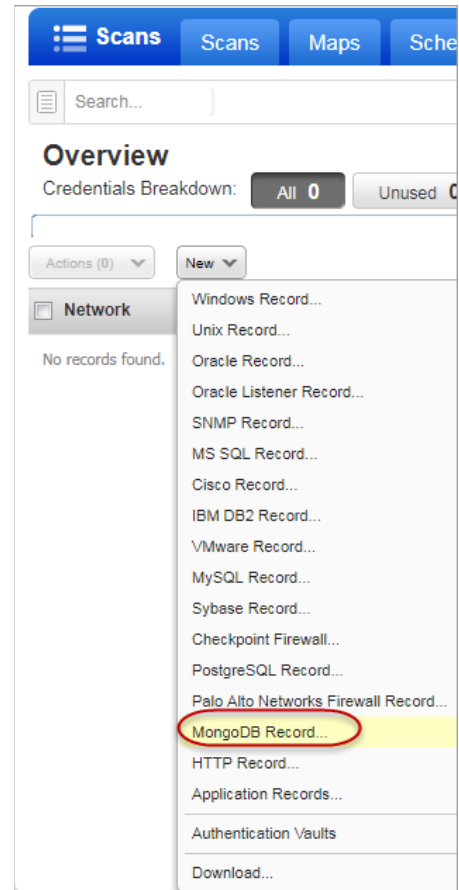
Did you get different results? Contact your MongoDB DBA to ensure that privileges are set up correctly.

MongoDB Authentication Records

You'll need to create a separate authentication record for each MongoDB instance to be scanned. During scanning we'll authenticate to one or more MongoDB instances on a host using all the MongoDB authentication records in your account.

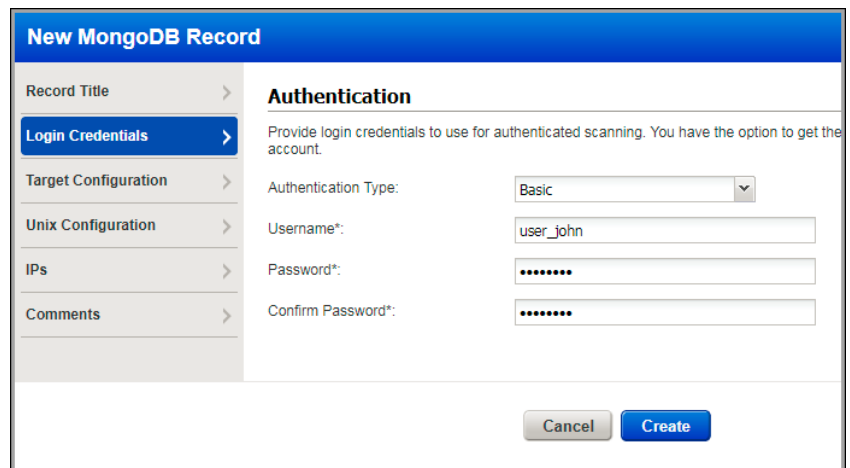
Where do I create records?

Go to Scans > Authentication > New > MongoDB Record.



Your login credentials

Enter the login credentials (user name, password) our service will use to log in to Unix hosts at scan time.

A screenshot of a web form titled 'New MongoDB Record'. The form has a sidebar on the left with navigation links: 'Record Title', 'Login Credentials' (selected), 'Target Configuration', 'Unix Configuration', 'IPs', and 'Comments'. The main content area is titled 'Authentication' and contains the following fields: 'Authentication Type:' with a dropdown menu set to 'Basic'; 'Username*:' with a text input field containing 'user_john'; 'Password*:' with a password input field showing six dots; and 'Confirm Password*:' with a password input field showing six dots. At the bottom right, there are two buttons: 'Cancel' and 'Create'.

Can I access a password in a vault?

Yes. We support integration with multiple third party password vaults. Go to Scans > Authentication > New > Authentication Vaults and tell us about your vault system. Then choose “Authentication Vault” in your record and select your vault name. At scan time, we’ll authenticate to hosts using the account name in your record and the password we find in your vault.

The screenshot shows the 'New MongoDB Record' form with the 'Authentication' section selected. The form includes a sidebar with options: Record Title, Login Credentials, Target Configuration, Unix Configuration, IPs, and Comments. The main content area is titled 'Authentication' and contains the following fields: 'Authentication Type' (set to 'Vault based'), 'Username*' (set to 'user_john'), 'Vault Type' (a dropdown menu with 'CA Access Control' selected), 'Vault Record*' (a dropdown menu), 'End Point Name*', 'End Point Type*', and 'End Point Container*'. A note at the top of the section reads: 'Provide login credentials to use for authenticated scanning. You have the option to get the log account.'

Using private keys

For MongoDB authentication key authentication is supported. You can define private keys in MongoDB authentication records.

What database information is required?

Tell us the database name to authenticate to and the port the database is running on (or use the default database name and port).

The screenshot shows the 'New MongoDB Record' form with the 'Target Configuration' section selected. The sidebar is the same as in the previous screenshot. The main content area is titled 'Target Configuration' and contains the following fields: 'Database Name*' (set to 'admin', with an example 'admin(default)') and 'Port*' (set to '27017', with an example '27017(default)'). A note at the top of the section reads: 'Tell us the user account to use for authentication, the database instance you want to authenticate to, and the port where the database is installed.'

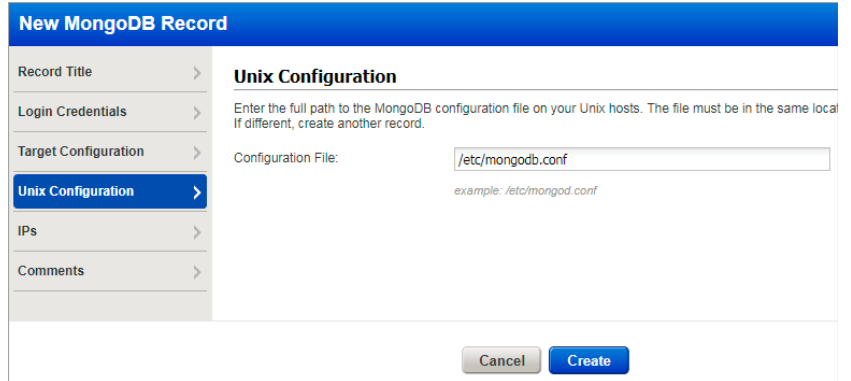
Should I use SSL?

Using SSL provides a secure connection to your database. By selecting “SSL Verify”, and if your database server supports SSL, you will be requesting a SSL secured link. The server SSL certificate verification is also enforced. By default, this option is set to false.

The screenshot shows the 'New MongoDB Record' form with the 'Target Configuration' section selected. The sidebar is the same as in the previous screenshots. The main content area is titled 'Target Configuration' and contains the following fields: 'Database Name*' (set to 'admin', with an example 'admin(default)'), 'Port*' (set to '27017', with an example '27017(default)'), and 'SSL Verify*' (a checkbox that is checked, with a note: 'Select this option to verify that the server's SSL certificate is valid and trusted.'). Below this is a 'Hosts*' field with a text area containing '192.168.3.20-192.168.3.25' and an example: 'host domain1, host domain2, mimongodb32e.s2012r2.mlqa.rdlab.qualys.com'. At the bottom of the form are 'Cancel' and 'Create' buttons.

Your MongoDB configuration file

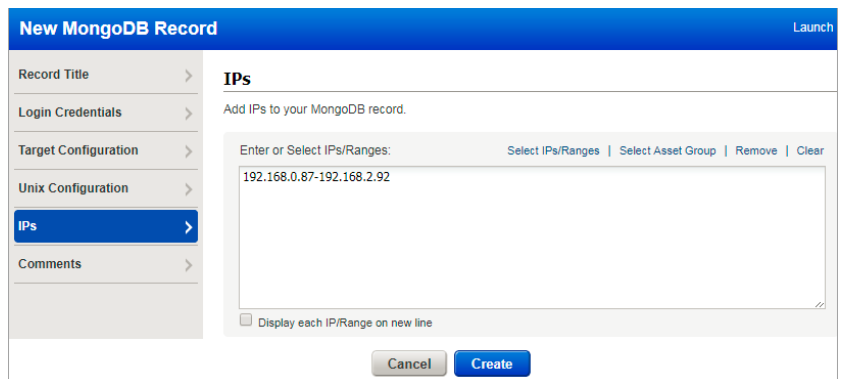
It is essential, though not required, that you provide the location of the MongoDB configuration file within the authentication record. This file is required for certain checks. For Unix, this file helps us gather the information needed to provide the information you are looking for.



The screenshot shows the 'New MongoDB Record' form with the 'Unix Configuration' tab selected. The left sidebar contains a menu with 'Unix Configuration' highlighted. The main content area has a title 'Unix Configuration' and a description: 'Enter the full path to the MongoDB configuration file on your Unix hosts. The file must be in the same local If different, create another record.' Below this is a text input field for 'Configuration File' containing '/etc/mongodb.conf', with a small example note below it: 'example: /etc/mongod.conf'. At the bottom right are 'Cancel' and 'Create' buttons.

Add IPs to the record

Select the IP addresses for the MongoDB databases that the scanning engine should log into using the provided credentials.



The screenshot shows the 'New MongoDB Record' form with the 'IPs' tab selected. The left sidebar contains a menu with 'IPs' highlighted. The main content area has a title 'IPs' and a description: 'Add IPs to your MongoDB record.' Below this is a text input field for 'Enter or Select IPs/Ranges:' containing '192.168.0.87-192.168.2.92'. Above the input field are links for 'Select IPs/Ranges', 'Select Asset Group', 'Remove', and 'Clear'. Below the input field is a checkbox labeled 'Display each IP/Range on new line'. At the bottom right are 'Cancel' and 'Create' buttons.

Last updated: June 14, 2018