DataStax Enterprise Authentication (PC)

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 DataStax Enterprise authentication for compliance scans.

A few things to consider

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, it's required for compliance scans.

Are my credentials safe?

Yes, credentials are exclusively used for READ access 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?

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

DataStax 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. These scripts require a super-user account which has privilege to create user. For example, Cassandra account or accounts with administrative privilege.

Please run the scripts provided, in the order shown.

1) Create a scan 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 to the account. Tip – We recommend creating an account called qualys_scan.

```
CREATE ROLE IF NOT EXISTS qualys_scan WITH SUPERUSER = FALSE
AND PASSWORD = '[enter password here]' AND LOGIN = TRUE;
```


1) Create a scan 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 to the account. Tip – We recommend creating an account called qualys_scan.

```
CREATE ROLE IF NOT EXISTS qualys_scan WITH SUPERUSER = FALSE
AND PASSWORD = '[enter password here]' AND LOGIN = TRUE;
```


2) Create a role for the scan account

This script creates a role, called QUALYS_ROLE, for the user account.

```
CREATE ROLE IF NOT EXISTS qualys_role;
```

3) Grant the role to the scan account

This script grants the role, called QUALYS_ROLE, to the user account, called QUALYS_SCAN.

```
GRANT qualys_role TO qualys_scan;
```

4) Grant privileges to the scan account via role

This script grants privileges to the user account to be used for scanning. The following privileges are required for successful authentication and compliance scanning.

```
GRANT SELECT ON SYSTEM_AUTH.ROLES TO qualys_role;
GRANT SELECT ON SYSTEM_AUTH.ROLE_PERMISSIONS TO qualys_role;
GRANT SELECT ON SYSTEM_AUTH.ROLE_MEMBERS TO qualys_role;
GRANT EXECUTE ON INTERNAL SCHEME TO qualys_role;
```

5) Verify Privileges on the Scan Account

Verify that the qualys_scan account has all the privileges in the database in order 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.

5a) This command checks if the role is granted to the scan account.

```
SELECT ROLE, MEMBER FROM SYSTEM_AUTH.ROLE_MEMBERS WHERE
ROLE='qualys_role' AND MEMBER='qualys_scan';
```

Sample Expected Output:

```
 role        | member
-------------+-------------
 qualys_role | qualys_scan
```

5b) This set of commands checks if the role and scan account exists, and if the privileges are granted correctly.

```
SELECT ROLE, COUNT(*) FROM SYSTEM_AUTH.ROLES WHERE
ROLE='qualys_scan';
SELECT ROLE, COUNT(*) FROM SYSTEM_AUTH.ROLES WHERE
ROLE='qualys_role';
SELECT ROLE, COUNT(*) FROM SYSTEM_AUTH.ROLE_PERMISSIONS WHERE
ROLE='qualys_role' AND
RESOURCE='data/system_auth/role_members';
SELECT ROLE, COUNT(*) FROM SYSTEM_AUTH.ROLE_PERMISSIONS WHERE
ROLE='qualys_role' AND
RESOURCE='data/system_auth/role_permissions';
SELECT ROLE, COUNT(*) FROM SYSTEM_AUTH.ROLE_PERMISSIONS WHERE
ROLE='qualys_role' AND RESOURCE='data/system_auth/roles';
SELECT ROLE, COUNT(*) FROM SYSTEM_AUTH.ROLE_PERMISSIONS WHERE
ROLE='qualys_role' AND
RESOURCE='authentication_schemes/INTERNAL';
```

Sample Expected Output:

```
 role        | count
-------------+-------
 qualys_scan |     1
 qualys_role |     1
 qualys_role |     1
 qualys_role |     1
 qualys_role |     1
 qualys_role |     1
```

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

Last updated: March 28, 2025