DATASTAX

DATASTAX

Securing Cassandra & DataStax Enterprise

Johnny Miller, Solutions Architect

@CyanMiller

www.linkedin.com/in/johnnymiller

What is security?

DATASTAX

Security is about the protection of assets

- Protective measures:
  - Prevention
  - Detection
  - Reaction
- What if your asset is information?
  - Information can be stolen - but you still have it…
  - Confidential information may be copied and sold - but the theft might not be detected…
  - The criminals may be on the other side of the world…

What is security?

Features

- Confidentiality
  - prevention of unauthorized disclosure of information
- Integrity
  - prevention of erroneous modification of information
- Availability
  - prevention of unauthorized withholding of information
- Non-repudiation
  - can you prove I did that?
- Authentication
  - verifying that users are who they claim to be
- Authorization
  - determine whether a user should have access to resources
- Accountability
  - actions can be traced back to the responsible party

Reality Check

Data Breaches

- Heartland Payment Systems, 2008-2009: 130 million records compromised
- Target Stores, 2013: 110 million records compromised
- Sony online entertainment services, 2011: 102 million records compromised
- National Archive and Records Administration, 2008: 76 million records compromised
- Anthem, 2015: 69 million to 80 million records compromised
- Epsilon, 2011: 60 million to 250 million records compromised
- Home Depot, 2014: 56 million payment cards compromised
- Evernote, 2013: More than 50 million records compromised
- Living Social, 2013: More than 50 million records compromised
- TJX Companies Inc., 2006-2007: At least 46 million records compromised
- Sony Pictures Entertainment, 2014: Company's inner workings completely exposed
- Plus many, many more that we know and don't know about…..

What are my requirements?

If you don't know, then ask

What do I have to do?

- Legal and regulatory requirements
  - Data Protection
  - PCI
  - HIPAA/HITECH
  - Etc…
- Enterprise policy
- Etc…

What should I do?

- Design and develop with security in mind
- Security in depth/layers
- OWASP
- Don't ever say "No one cares about this data"

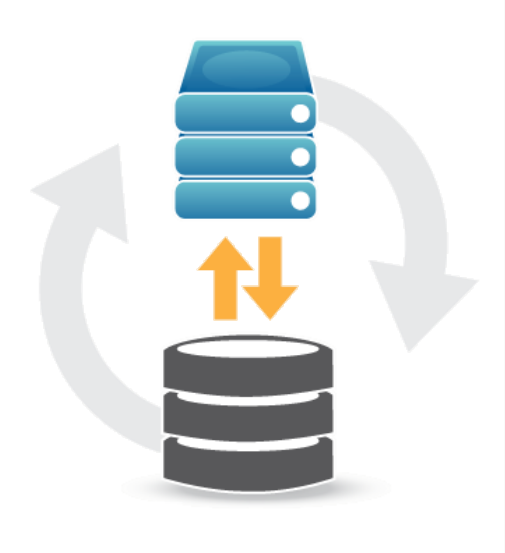
Backups

Information Assurance

DATASTAX

Ensuring data is not lost when critical issues arrive!

- `nodetool snapshot etc…`
- OpsCenter Backup Service
  - create scheduled or one-off backups
    - data and commit logs
  - custom scripts that can be run before or after a backup
  - can backup to local server and/or S3
  - automatically compresses backup files to save storage
  - retention policies on backups
  - full, table-level, or point-in-time restores for a cluster
  - All functions available via REST API or via user interface
  - Plus lots, lots more…..



Access Controls

Authentication

DEFAULT IS OFF

Turn on authentication straight away!

- Internal
  - Update the `authenticator` property in the **`cassandra.yaml`** to:

    `authenticator: PasswordAuthenticator`
  - Stores usernames and bcrypt-hashed passwords in the **`system_auth.credentials`** table
- External
  - LDAP v3, Kerberos, Active Directory, Oracle Directory Server Enterprise Edition
  - Update **`cassandra.yaml`** authenticator e.g.

`authenticator: com.datastax.bdp.cassandra.auth.LdapAuthenticator`

- Update **`dse.yaml`** to configure the external provider details e.g.

```
ldap_options:
    server_host: somehost
    server_port: 389
    search_dn: cn=Admin
    search_password: secret
    use_ssl: false
    use_tls: false
    etc…….
```

Access Controls

Authorization

- Permissions to access all keyspaces, a named keyspace, or a table can be granted to a user
  - GRANT `e.g. GRANT SELECT ON ALL KEYSPACES TO johnny;`
  - LIST PERMISSIONS `e.g. LIST ALL PERMISSIONS OF johnny;`
  - REVOKE `e.g. REVOKE SELECT ON someks.sometable FROM johnny;`

Permission	CQL Statement
ALL	All statements
ALTER	ALTER KEYSPACE, ALTER TABLE, CREATE INDEX, DROP INDEX
AUTHORIZE	GRANT, REVOKE
CREATE	CREATE KEYSPACE, CREATE TABLE
DROP	DROP KEYSPACE, DROP TABLE
MODIFY	INSERT, DELETE, UPDATE, TRUNCATE
SELECT	SELECT

- Update the authorizer property in the **`cassandra.yaml`** to:

```
authorizer: CassandraAuthorizer
```

Access Controls

Once Enabled

- A default superuser is created (cassandra/cassandra)

  - Default users are bad - create a new super user, change the cassandra users password to something huge (and forget it) and take away the cassandra users superuser status

- Increase the replication factor for the **`system_auth`** keyspace to N (number of nodes)
- Adjust the validity period for permissions caching by setting the **`permissions_validity_in_ms`** option in the **`cassandra.yaml`** (default is 2s)
- Update application code to use credentials e.g

```
Cluster.Builder clusterBulder = Cluster.builder()
                                .withCredentials(userName, password);
```

- Best Practices:
  - Don't share credentials i.e. named user access only and per application user accounts
  - Don't over grant permissions i.e. if you only need read access, only grant read access
  - Use a file to store credentials via **`~/.cassandra/cqlshrc`** and **`~/.dserc`** files i.e. don't pass credentials in on the CLI via cqlsh
  - Use strong passwords

Access Controls

Enabling without downtime

- The `TransitionalAuthenticator` and `TransitionalAuthorizer` included with DataStax Enterprise allow internal authentication and authorization to be enabled without downtime or modification to client code or configuration.
- Procedure:
  1. Update the `cassandra.yaml` file
     - Set the `authenticator` to `com.datastax.bdp.cassandra.auth.TransitionalAuthenticator`.
     - Set the `authorizer` to `com.datastax.bdp.cassandra.auth.TransitionalAuthorizer`.
  2. Rolling restart
  3. Run a repair on the `system_auth` keyspace
  4. Login via cqlsh and the default superuser – create all the logins and permissions you need
  5. Update your applications to start using the credentials
  6. Update the `cassandra.yaml` to use the `authenticator` and `authorizer` you want and perform a rolling restart
  7. Remove the default superuser and create your new one.

Data Auditing

DataStax Enterprise

- DataStax Enterprise comes with the ability to audit all Cassandra access across the cluster.

Permission	CQL Statement
ADMIN	Logs describe schema versions, cluster name, version, ring, and other administration events.
AUTH	Logs login events.
DML	Logs insert, update, delete and other DML events.
DDL	Logs object and user create, alter, drop, and other DDL events.
DCL	Logs grant, revoke, create user, drop user, and list users events.
QUERY	Logs all queries.

- Auditing can be limited to specific keyspaces only
- Audit logs can be written to filesystem log files (default) using log4j, or to a Cassandra table (`dse_audit.audit_log`) – use log4j appenders to ship the logs off server
- To enable, update the `audit_logging` section in the `dse.yaml`
- Auditing can also be enabled for the DSE Search via the `filter-mapping` element of the Solr `web.xml`

Data Auditing

Information included

Field	Decription
host	dse node address
source	client address
user	authenticated user
timestamp	system time of log event
category	DML/DDL/QUERY for example
Type	API level operation
batch	batch id
ks	Keyspace
cf	column family
operation	textual description

e.g:

host:ip-10-85-22-245.ec2.internal/10.85.22.245|source:/127.0.0.1|user:johnny
|timestamp:1370537557052|category:DDL|type:ADD_KS
|ks:test|operation:create keyspace test with replication = {'class':'NetworkTopologyStrategy', 'Analytics': 1};

Encrypting Data

In-flight

- All data in-flight can be encrypted using SSL
- All nodes must have all the relevant SSL certificates for all nodes.
- Client-to-node encryption
  - Enable via the `client_encryption_options` in the `cassandra.yaml`
  - Specify the path to your `.keystore` and `.truststore` files and the password used when generating them
- Node-to-node encryption
  - Enable via the `encryption_options` in the `cassandra.yaml`
  - Specify the path to your `.keystore` and `.truststore` files and the password used when generating them
  - You can enable SSL for:
    - None
    - All – all traffic is encrypted
    - DC – only encrypt inter-dc traffic
    - Rack – encrypt all inter-rack traffic

Use a separate network for Client-to-node and Node-to-node communication

- Make sure .keystore is readable only by the DSE daemon and not by any user of the system.
- Make sure everyone is communicating over the SSL ports

Encrypting Data

In-flight

- Applications need to be updated to use SSL also e.g:

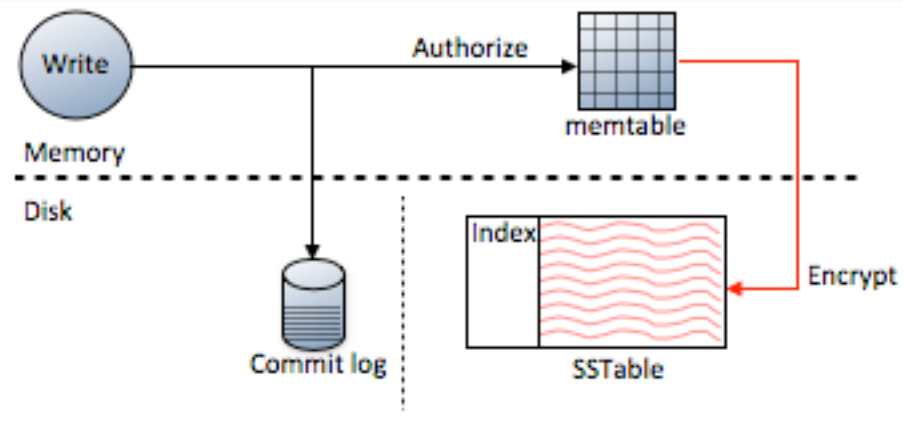
```
SSLContext sslContext =
  SSLFactory.createSSLContext(encryptionOptions, true);
SSLOptions sslOptions = new SSLOptions(sslContext,
encryptionOptions.cipher_suites);
clusterBuilder.withSSL(sslOptions);
```

- Further reading: http://www.datastax.com/dev/blog/accessing-secure-dse-clusters-with-cql-native-protocol

- Example code: https://github.com/PatrickCallaghan/datastax-ssl-example

Encrypting Data

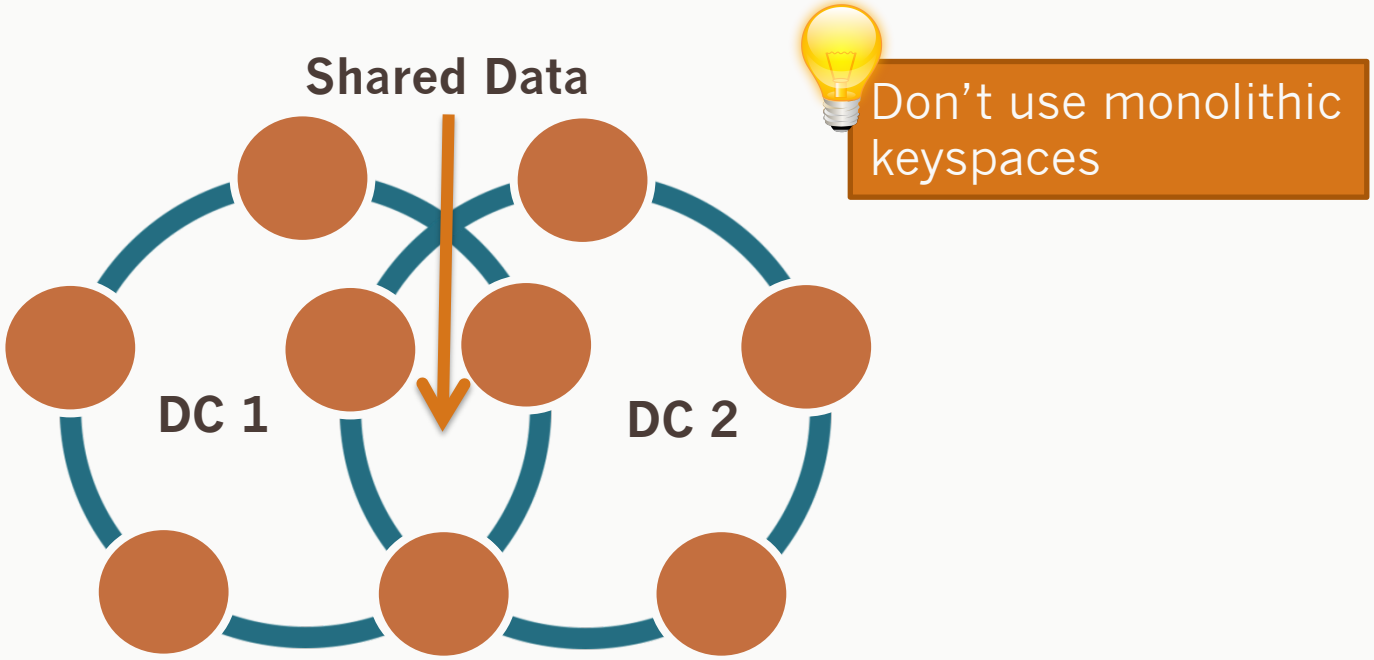
At rest – Transparent data encryption



- SSTables can be encrypted at rest via DataStax Enterprise
- Note – this is only the SSTables. If you need commit log encryption, you should look at using an OS-level encrypted file system.
- The OS needs to be hardened as an invasion of the local file system will invalidate the encryption as the encryption keys are stored locally
- Use the dse command to generate a system key for encryption e.g. `dsetool createsystemkey 'AES/ECB/PKCS5Padding' 128 system_key`
  - This is inserted into the **`dse_system.encrypted_keys`** table and written to a location on disk e.g. /etc/dse/conf (or specifiy `system_key_directory` in `dse.yaml`) – make sure the permissions on the directory are correct! This key will need to be copied to each node in the cluster.
  - The entire cluster uses the system key to decrypt SSTables for operations such as repairs
  - You also use the system key during upgrading and restoring SSTables that might have been corrupted
  - When enabling on existing cluster, ensure that the user encrypting data has been granted ALTER permission on the table containing the data to be encrypted. E.g.
    ALTER TABLE users WITH compression = { 'sstable_compression' : 'EncryptingSnappyCompressor', 'cipher_algorithm' : 'AES/ECB/PKCS5Padding', 'secret_key_strength' : 128, 'chunk_length_kb' : 128 };
  - Rewrite the sstables i.e nodetool upgradesstables --include-all-sstable

Data Replication Security

- A popular feature from a data security perspective is the ability to control at a keyspace/schema level which data centres data should be replicated to.
- What this means is that in a multi-data centre (both physical and virtual) cluster you can ensure that data is not shipped anywhere it shouldn't be and access to that data can be controlled.
- This is very simple to set-up and is extremely useful when you need to share some of your data, but not all of you data or if you have requirements around where your data is permitted to reside.

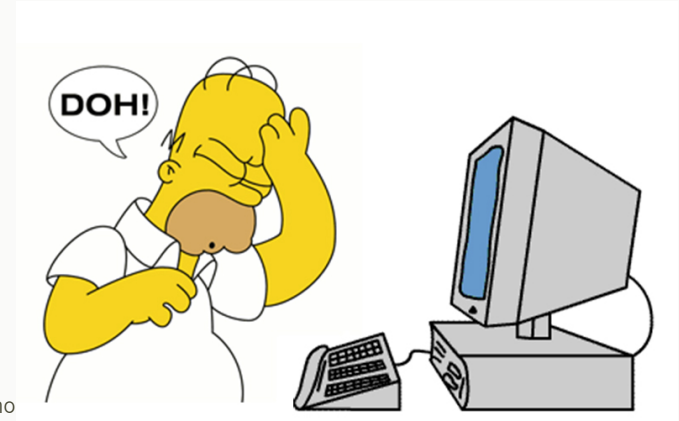


Secure OpsCenter

DATASTAX

- OpsCenter is a cluster administration tool for DataStax Enterprise
- Supports:
  - Role-based security controls (enterprise only)
  - SSL between OpsCenter server and agents running on nodes
  - HTTPs for web and API access

Secure OpsCenter – if it gets compromised….



Security Features

Cassandra or DataStax Enterprise

DATASTAX

Cassandra	DataStax Enterprise
Internal Authentication • Simple to implement & easy to understand • Internal validation of authorized • No learning curve	External Authentication • External validation of authorized users • Leverages Kerberos & LDAP) • Single sign-on to all data domains
Object Permission Management • Deep control over who can add/change/delete/ read data • Uses familiar GRANT/REVOKE from relational world • No learning curve	Transparent Data Encryption • Protects sensitive data at rest (only sstables) • No changes needed at application level • Encrypt both Cassandra and Hadoop data
Client to Node Encryption • Ensures data cannot be captured/stolen in route to a server • Data is safe both in flight from/to a database and on the database • Complete coverage is ensured	Data Auditing • Audit trail of all accesses and changes • Control to audit only what's needed • Uses log4j interface to ensure performance & efficient audit operations

DataStax and Vormetric

Partnership

- We've seen a rising number of customers in heavily-regulated industries e.g healthcare, Finance, Retail, Education etc…
- Vormetric is the industry leader in data security solutions that span physical, virtual, cloud and big data environments from both internal and external threats.
- Vormetric with DataStax Enterprise enhances our at-rest encryption:
  - A single place for managing policies
  - Protection of data sources, DSE environment and analytic reports through data at rest encryption – encrypts the entire disk
  - Dashboards and reports
  - Application layer encryption
  - Key management
  - And lots more….

Read more: http://bit.ly/dse_pci

Quote

"The mantra of any good security engineer is: 'Security is a not a product, but a process.' It's more than designing strong cryptography into a system; it's designing the entire system such that all security measures, including cryptography, work together."

Bruce Schneier
Cryptographer, Computer Security and Privacy Specialist
https://schneier.com
@schneierblog

Read this: *http://bit.ly/dse_security*

Thank You