Crosswork Situation Manager Deployment

Security Hardening Instructions

Overview

This document describes the post-deployment steps required to further secure the Crosswork Situation Manager runtime environment.

Operating system

Cisco Crosswork Situation Manager does not ship nor manage the Operating System layer but does recommend that all customers use their Platform Engineering team's official, fully-patched version version 7 of either RedHat or CentOS Linux. The ongoing security patching and monitoring of the Operating System layer should be managed by customer systems administrators following the same standards and processes as for all the other version 7 RedHat or CentOS Linux instances within their environment.

Securing Your Deployment

The following steps provide the core "checklist" for securing Crosswork Situation Manager deployments:

1. Follow all the applicable steps as documented in Secure Your Installationstarting on page 111 of the Cisco Crosswork Situation Manager 8.0.x Implementer Guide including:
   a. Apply Valid SSL Certificates (P. 111)
   b. Encrypt Database Communications (P. 112)
   c. Establish Trust for the MySQL Certificate (P. 112)
   d. Configure Cisco Crosswork Situation Manager to use SSL for Database Communications (P. 112)
   e. Moog Encryptor (P. 113)
   f. Encrypt a password (P. 113)
   g. Configure Cisco Crosswork Situation Manager to use encrypted passwords (P. 114)
   h. Change the location of the encryption key (P. 114)
   i. Configure External Authentication (P. 115)
   j. Configure Single Sign-On with SAML (P. 115)
   k. Configure your SAML IdP (P. 115)
2. Configure local firewalls on each Crosswork Situation Manager server such that all ports are denied by default from all sources/locations except where explicitly specified in the **Distributed HA system Firewall** section on Page 53 of the Cisco Crosswork Situation Manager 8.0.x Implementer Guide.
   See **Fully Distributed HA Installation** on Page 31 of the Cisco Crosswork Situation Manager 8.0.x Implementer Guide for the full installation steps for a fully distributed system running with HA.
   These are the only source/destination/port combinations that should be allowed:

Source	Destination	Ports	Bi-directional
UI 1, UI 2	Core 1, Core 2	3309, 5672, 9200	-
UI 1, UI 2	RedServ	5672, 9200	-
UI 1, UI 2	DB 1, DB 2, DB 3	3306, 3309, 9198	-
Core 1	Core 2	5701, 9300, 4369, 5672	Yes
Core 1, Core 2	RedServ	5701, 9300, 4369, 5672	Yes
Core 1	Core 2, RedServ	25672	
Core 1	Core 1, RedServ	25672	
RedServ	Core 1, Core 2	25672	
Core 1, Core 2	DB 1, DB 2, DB 3	3306, 9198	-
LAM 1, LAM 2	Core 1, Core 2, RedServ	5672	-

Source	Destination	Ports	Bi-directional
LAM 1, LAM 2	DB 1, DB 2, DB 3	3306, 9198	-
DB 1	DB 2, DB 3	3306, 4567, 4444, 5468	Yes

If any of the default ports are changed then substitute it in the tables above. The ports are responsible for the following:

9200	Used for inbound Elastic Search REST API
9300	Used for Elastic nodes communication within a cluster
5672	Access to mooms bus (RabbitMQ)
15672	Access to mooms (RabbitMQ) console
4369	Required for mooms (RabbitMQ) cluster
5701	Required for Hazelcast cluster
8091	Access the Hazelcast cluster info via Hazelcast's
3309	Used for initializing UI servers
3306	Regular MySQL port
4567	For group communication in Percona XtraDB Cluster
4444	For State Snapshot Transfer in Percona XtraDB Cluster
4568	For Incremental State Transfer in Percona XtraDB Cluster
9198	Allows HAProxy to check the node's Percona XtraDB Cluster status via http
25672	Used for inter-node and CLI tools communication

3. If possible use separate network interfaces for intra-Moogsoft-component communication.
   e.g. eth0 for "regular" traffic and eth1 for intra-Moogsoft-component communication
4. See ***Fully Distributed HA Installation*** on Page 31 of the Cisco Crosswork Situation Manager 8.0.x Implementer Guide for the full installation steps for a fully distributed system running with HA.
5. Configure user authentication to only use SAML2 Single-Sign-On (SSO) for all users of the system as documented in the section ***Configure Single Sign-On with SAML*** on Page 115 of the Cisco Crosswork Situation Manager 8.0.x Implementer Guide.

NOTE
To ensure that Moogsoft-side authentication is disabled, delete or comment out any realm type in $MOOGSOFT_HOME/config/security.conf that has realType = "DB" (example shown below)

```
#  "DB realm" : {
#     "realmType": "DB"
#}
```

NOTE

- In SAML2 SSO deployments, the Moogsoft system is the Service Provider (SP) and the customer's existing authentication system is the Identity Provider (IdP).
- This allows the customer to fully define, manage and enforce the password policy used to access Moogsoft (in the same manner as for all other applications across the enterprise)
- With the customer (as IdP) handling all of the authentication, the customer can define, manage and enforce anti-brute-force mechanisms (in the same manner as for all other applications across the enterprise)
- By design, the SAML2 workflows (Security Assertion Markup Language) is a process where the IdP "asserts" that at a given point in time, using a specified mechanism(s), a user has been authenticated. The IdP redirects the user back to theService Provider (SP) i.e. Moogsoft with a digitally (cryptographically) signed and encrypted SAMLResponse XML document that the SP validates and upon successful validation then issues a session token which allows the user to continue to use the SP system without having to re-authenticate for every transaction. Once the session token expires, the SP automatically redirects the user back to the IdP to re-authenticate.

  If the user changes their password, the old password will never be able to be used for IdP authentication, not for Moog nor for any other SSO application in the customer's environment.

  The very next time a Moogsoft user needs to authenticate (i.e. when their session token expires) they must authenticate with the IdP using their new password.

6. The Moogsoft REST API (referred to as the "Graze API") is an optional component meant for convenience of configuration automation. It is not required for day-to-day operations of the product. For most users who do not require the Graze API, the security hardening best practice recommendation is to disable the Graze API altogether. This can be achieved simply by not enabling any users with the graze permission.

   For those with need of Graze API:

   The security hardening best practice is to only allow calling the API from the local machine and only via a service account.

   NOTE

   The Graze API methods support sending auth tokens either in the querystring of HTTP GET requests or in the body of HTTP POST requests. The security hardening best practice is to only ever use the HTTP POST request approach when sending auth tokens.

   Example always call the API in this manner:

   ```
   curl -X POST -u graze:graze -k -v "https://localhost/graze/v1/
   addService" -H "Content-Type: application/json; charset=UTF-8" -d
   '{"name" : "New Service 1", "description" : "This is my description
   12345"}'
   ```

   and not this manner:

   ```
   curl -X GET -u graze:graze -k -v "https://localhost/graze/v1/addService?
   name=value&description=value" -H "Content-Type: application/json;
   charset=UTF-8"
   ```

For the few (optional) Moogsoft (Graze) API methods that return sensitive data, only ever call them from the local Moogsoft server (i.e. not over the network).
While the entire call/response to/from these API methods is fully encrypted with Transport Layer Security (TLS), the endpoint of the API call can "see" the contents (by design) and therefore such API methods should only ever be called "locally"
Here are two examples of such API endpoints:
https://<server>/integrations/api/v1/integrations
https://<server>/graze/v1/getSecurityRealms?auth_token=
https://<>/graze/v1/authenticate?username=&amp;password=
https://<>/graze/v1/getUsers?auth_token=
Additionally, to minimize visibility of any error messages returned by Moogsoft API calls, Moogsoft recommends restricting calling API methods "locally" only (vs over a network)
An example of an API endpoint that displays verbose error information:
https://<>/graze/v1/application.wadl
Local only-use of Moogsoft API methods can be enforced by removing the "/graze" entry from the nginx config file (/etc/nginx/conf.d/moog-ssl.conf)
Remove this stanza from the nginx config file:

```
location /graze {
    proxy_pass http://localhost:8085/graze;
    include conf.d/moog-api-headers.conf;
}
```

Then only call the API via the http://localhost:8085/graze URL.

7. Set the contents of the $MOOGSOFT_HOME/ui/web.conf file to an "empty" JSON value of:{}

   NOTE
   (Note that the existing content shipped with Crosswork Situation Manager is merely example and not needed for system operation)

8. Be sure to only use encrypted credentials in all Crosswork Situation Manager config files
   Example, use `encrypted_password` instead of `password:#"password"        : "Abacus","encrypted_password" : "eY388HQJZG/caUnVb4hImm4gIOpb4rwpF4="`
   See the ***Moog Encryptor*** section on Page 113 of the Cisco Crosswork Situation Manager 8.0.x Implementer Guide for usage information
9. Change the default password for Moogsoft components:
   -- Initial Moogsoft "admin/admin" UI account
   -- Base component default accounts: RabbitMQ, MySQL, Elasticsearch