# Distributed Engine Security

## Definitions

Message – representation of the task to be performed by engine. It is JSON serialized object.

Site – secrets are associated with one site. There could be many sites per Secret Server. A secret can only be associated with one site.

Site connector – communication pipeline between Secret Server and engines. There could be many site connectors per Secret Server. A site can only be associated with one site connector.

Engine – engines workers which subscribe to a site connector and process tasks submitted through Secret Server or other engines. There could be many engines per site.

Request Bus – bus used to submit messages from Secret Server or Distributed Engine to site connectors.

Response Bus – bus used to submit task response details from engines to Secret Server.

## Start up

1. Engine restores its identity guid.
   1. The identity guid is generated by each engine when they are first started. They cannot be set through Secret Server.
   2. An identity guid is encrypted as the user running the service.
   3. Running the engine service as a different user post engine installation/activation will cause the engine to lose its active status and will need to be reactivated through Secret Server UI.
2. Engine requests a temporary public/private key combination from Secret Server.
3. Secret Server sends public key to engine
4. Engine encrypts its identity guid with the public key and sends back to Secret Server.
5. Secret server decrypts the request with the private key counterpart.
   1. If the engine is activated, secret server sends configuration information back to engine encrypted with the private key. This configuration contains information about the site that the engine is assigned to, site connector credentials and the symmetric key for the site itself.
   2. If the engine is not activated, it will go into a retry state where every 30 seconds it will repeat the process.
6. The engine decrypts the configuration information and subscribes to the site connector.

## Message submission

1. Secret Server has series of communication buses it uses to post messages to site connectors.
2. When a new task is generated, the message is encrypted with the site symmetric key and is pushed through the respective bus to the site connector.
   1. Messages on the site connector can be seen by administrators but are of little value as they are encrypted.
3. The same applies to a synchronous task, but waits for a response on the bus that the operation succeed of failed.

## Message submission through distributed engine

1. It is common for an engine to take a somewhat undefined task, like network discovery, and as it discovers more work to submit that work back to the site connector.
2. The new messages are encrypted with the site symmetric key and pushed to the site connector.

## Message processing

1. Engine subscribes to the site connector it is associated with.
   1. If there is a message, engine takes the message and has a specified amount of time to process it.
2. The message is decrypted with the symmetric key of the site.
   1. If engine fails or takes too long, the message is returned to the site connector.
   2. If the message has expired and is not marked to be relayed after expiration, the message it thrown away
   3. If the message is corrupted or cannot otherwise be decrypted with the site symmetric key it is thrown away.
3. Every task engine can perform is different and as such as different logging details that it will generate. Log information is stored and sent back asynchronously to Secret Server on an interval.

## Task status reporting

1. Engines have has a response bus which is used to send results of operations directly to Secret Server.
2. The payload is encrypted with the public key from the original handshake with Secret Server.
3. Secret server decrypts the payload with the private key counterpart.

## Engine heart beat

1. On a (configurable) interval, each engine will communicate back to Secret Server to check its status and if there is any updated configuration.
2. If a heartbeat fails, the engine will recycle and restart itself.