INCMS™ Secure Communications

Table of Contents

1 Secure Communications

It is important that any customer who runs a CMS is very confident that modules within the CMS do not compromise security. Even if a security breach only reveals content that will eventually be published to the public web site, the adverse publicity associated of a security breach of any kind is undesirable. For this reason, it is important that customers understand how the Sitemorse **INCMS™** works with respect to the flow of messages and how these are initiated.

The only change that might be required is to allow an outbound connection to the Sitemorse server on the Sitemorse server port. By default, this is 5372.

2 Protocol messages

In the follow diagrams and notes, error handling and some of the message detail has been omitted for clarity. It is important to note that the default for all communications is an encrypted SSL connection.

2.1 Authentication

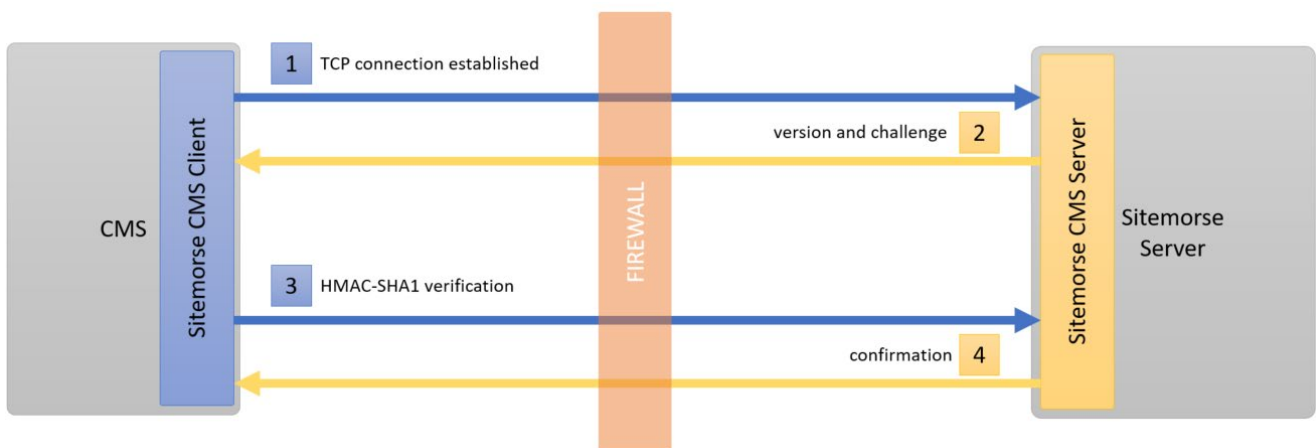The following diagram summarises the authentication stage message flow.

Figure 1 - Authentication Messages

1. The Sitemorse CMS Client initiates a TCP connection from within the customer CMS and behind the customer's firewall
2. The Sitemorse Server responds with a version number and an opaque challenge string of 8 to 64 characters
3. The client uses parts of the key combined with the challenge to create an HMAC-SHA1 string
4. The server confirms the HMAC-SHA1 string is correct

As only the client and the server have a copy of the license key, only the true client and server can generate the same HMAC-SHA1 string. The content manager is not involved in this process. There is no need for the content manager to authenticate with Sitemorse to conduct a test. No additional passwords need be remembered. If the content manager is authenticated with the CMS, and they can perform a Sitemorse test.

2.2 Test Request

Once the authentication phase is complete, the client can send the content for testing. The following diagram shows how the client initiates the test.
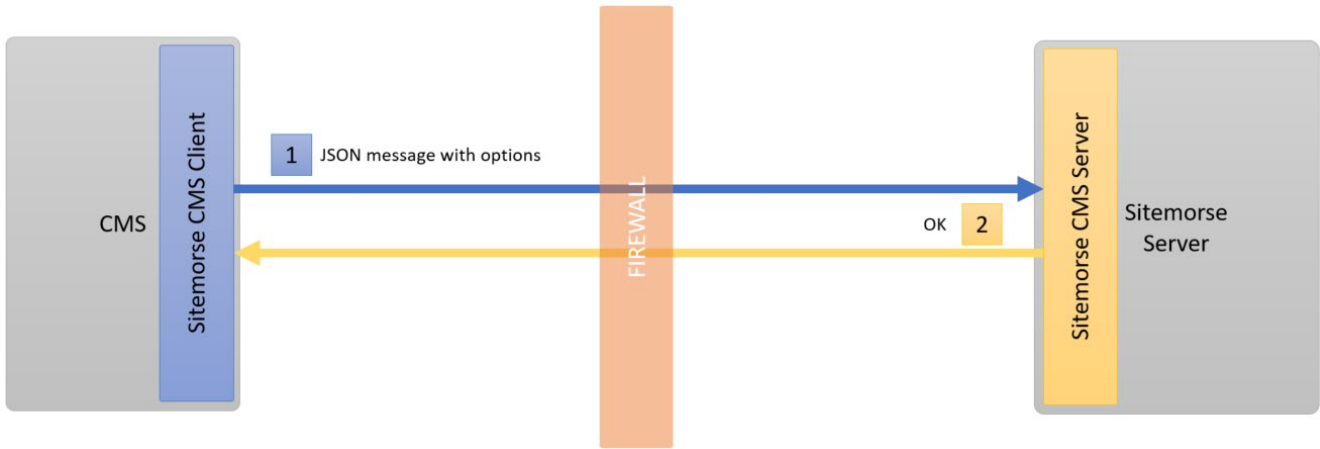
Figure 2 - Test Request Messages

1. The client creates a JSON message containing the information about what is to be tested. This message includes the URL, hostnames, how the result is to be viewed and many more optional parameters. The client sends the length of the message and the message
2. The server responds with OK

The client message contains a list of hostnames that are behind the firewall. This tells the Sitemorse server which items need to be retrieved through the client and which can be retrieved from a public site. For example, the content being tested might be in the CMS but links to images or CSS might be on the public web site.

2.3   HTTP Proxy

After the test request, the client enters the HTTP proxy stage. The server needs to request the HTML to be tested and then any images, CSS, JavaScript, etc. The client now acts as a proxy for the server to allow the server to obtain all the required items. The client can reject any request that falls outside the pre-agreed hostname, port number, etc. Although the server is requesting content, the client still maintains the required level of security.
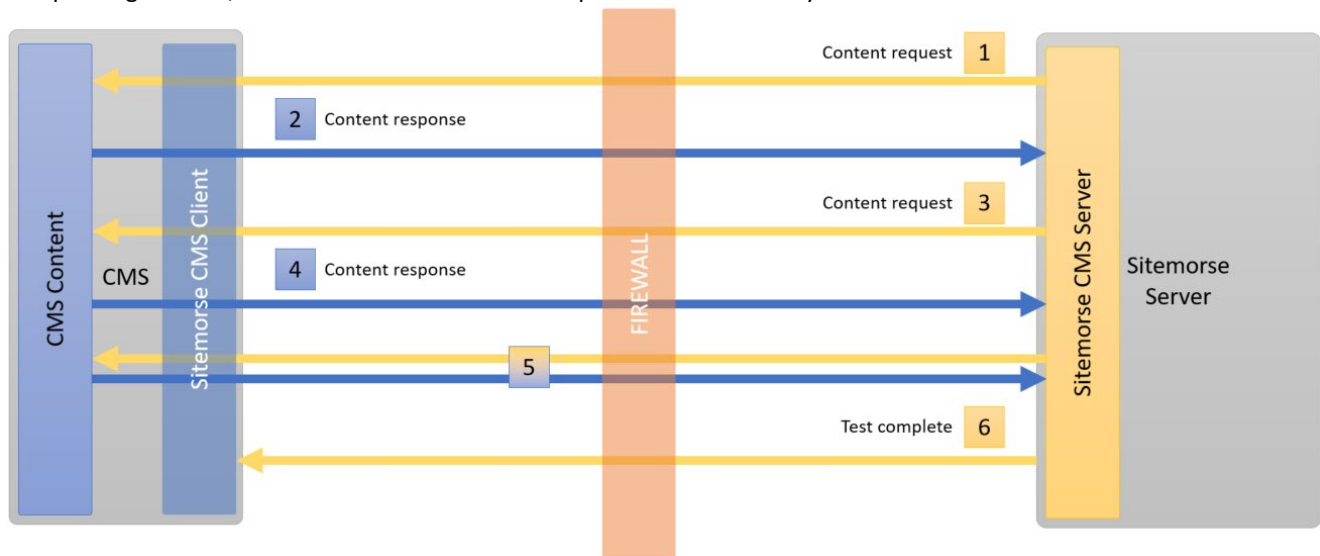

Figure 3 - HTTP Proxy Messages

1. The server sends a request for content
2. The client obtains the content locally and responds
3. The server sends a request for content
4. The client obtains the content locally and responds
5. The request/response process is carried out until all the required items have been obtained by the server
6. The server sends a test complete message with the URL of the page that contains the test results and information about any errors on the page

Note that often links for images and CSS might be from public servers and these are not shown in the diagram for clarity.

The status of the errors on the page returned in the last step above, can be used to update content lists within the CMS. For example, the following screen shot shows the Drupal CMS content search UI with an added column to show the result of the last assessment done.

Figure 4 - Drupal UI with Sitemorse Assessment Status

3 Summary

From the communications diagrams in this document, it is quite clear that the Sitemorse server never initiates a connection to the CMS behind the firewall. An example to prove this, is to configure a test CMS on the localhost domain and try a test. The test will work perfectly even though localhost is not a location the Sitemorse server can ever access directly.

4 Document Control

Amendment history

Date Issued	Version	Reasons for change
7 July 2020	1.0	Document creation

Distribution list

Name	Organisation/role	Relevant sections

Related documentation

Reference No	Title	Author	Version and date

Disclaimer