Remote Status architecture

Introduction

Syspeace Remote Status is the capability for a Syspeace service to report its status, and for the owner of that Syspeace account to view this status and issue commands. This document will describe the involved parts and provide in-depth details to explain how Remote Status nodes communicate. Reading this document is not necessary to set up and use Remote Status.

Background: Web servers

All servers mentioned are HTTP-with-TLS-encryption ("https") web servers. All speak the HTTP protocol but support different roles by providing web APIs on this server. All communication with the server is encrypted with TLS; this is the same technology powering secure connection to banks, etc.

Background: Asymmetric (public key) and symmetric encryption

(Encryption is a complex and deep subject with detail and nuance; this section is a short primer to explain basic concepts.)

Asymmetric encryption involves generating a pair of very large, mathematically entwined numbers called the private key and the public key. If you have the private key, you can "encrypt" a message to turn it into an unreadable mess, which can be "decrypted" with the public key to turn it into the original message.

You need to know the private key to encrypt a message that can be decrypted with the public key, so being able to produce such a message proves that you have access to the private key. You can also encrypt a message with the public key to produce something only the owner of the private key can decrypt.

When the keys are of sufficient length, attempting to decrypt or encrypt a message without one of the keys is infeasible given current computational limits. Encrypting communications means that effectively no one can impersonate the server on the other end, and that the traffic, if recorded, is not able to be decrypted without knowing the key. (Protocols like TLS are also immune to "playing back" the traffic to perform the same actions, due to unique and time-based factors applied during encryption.)

In practice, keys are not just numbers but a collection of parameters to an equation called a cryptosystem. TLS uses the RSA cryptosystem.

Symmetric encryption is similar to asymmetric encryption but the same key is used in both directions. Usually, asymmetric encryption protects a randomly generated symmetric key that is used to encrypt most of the data, due to weaknesses in asymmetric encryption for data longer than the key length itself. The AES cipher is often used for this purpose.

Background: The Syspeace backend server

The Syspeace backend server is used by Syspeace services to maintain license information. The license key links the Syspeace service to an account and its pool of licenses, and it receives a unique server ID, known to be in this account. Every Syspeace service has daily contact with the backend server to validate its license, and other information is also contributed and received, like the Global Blacklist.

This server is hosted by Treetop Innovation AB, the makers of Syspeace, and running Syspeace is dependent on allowing communication to the backend server.

Syspeace services

Remote Status is built in to the Syspeace service, but is turned off by default. The administrator must turn on Remote Status manually, or enable it through a settings file.

When Remote Status is enabled on a Syspeace service, the service generates an RSA 2048-bit key and an accompanying X.509 certificate. In a message cryptographically signed by the certificate, the public key and the thumbprint of the certificate are sent to the Syspeace backend server, where it is associated with the entry for the computer and the Syspeace account. The private key remains on the computer, stored in the Windows Certificate Store.

Syspeace consoles

Syspeace Remote Status Console (hereafter also "console") is a Windows application that can pair with one or more Syspeace accounts and view the status of the connected Syspeace services.

When the console pairs with a Syspeace account, it generates an RSA 2048-bit key and an accompanying X.509 certificate. The certificate is used to cryptographically sign all communication with the Syspeace backend server. It attempts to authenticate with the Syspeace account with the account name (an email address) and its password, and if successful, the console is associated with the Syspeace account and the thumbprint and public key are stored. The private key remains on the computer, stored in the Windows Certificate Store.

From this point on both the console and the service proceed identically, as "nodes".

Talking to each other via Relay

Each node asks the Syspeace backend server for an authentication token, which verifies its identity as a service or a console connected to the account, and contains the public key, the type of node (service or console), a short time window in which it is to be used, a cryptographic nonce (single-use number), a unique "node" ID and an "account" ID. Neither of the IDs correspond with other actual IDs in use for these purposes.

Each node contacts a Relay Selector server, which is hosted by Treetop (or by someone else, if configured), providing it the node ID and account ID and receives instructions of which Relay server to connect to.

Each node then connects to this Relay server and presents the authentication token. The Relay server validates the authenticity of the token with the Syspeace backend server, which also verifies that it has not previously been used and is being used within the time window. From this point on, the node can receive messages sent through Relay to it specifically or to all nodes of its type ("broadcast"), or send such messages itself.

The messages, as delivered to and received from Relay, contain information about the sender node ID and recipient ID (and recipient types, if the message is a "broadcast" message). The messages are encrypted with a randomly generated symmetric AES key, itself encrypted with the public key of the recipient (so that only the recipient may decrypt it), and cryptographically signed with the private key of the sender, so that Relay can verify its authenticity using the sender's public key.

"Broadcast" messages, sent to multiple nodes of a type, don't correspond to a single recipient public key. Instead, an intra-account key is derived from actual account information not included in the token, available to all nodes but not to the Relay server, and it is used in place of the public key of the recipient when encrypting and decrypting the message.

Relay's role

Relay is both oblivious to the contents of the data and unable to inject "faked" messages. It does not have access to the private keys of any node, so it can neither sign a message nor encrypt its contents. And it does not have access to the actual account information, so it cannot encrypt a broadcast message either. To Relay, the message is a black box, which contents are unknown. This property makes it possible for Relay and Relay Selector servers to be hosted by external parties, including by Syspeace users for their own use.

Aside from the messages from node to node, Relay must deliver some of its own messages directly to nodes. These messages are formatted and handled differently, and are trusted on account of being delivered over a secure connection from Relay, since they all provide information about Relay itself.

In order to facilitate the distribution of the messages, Relay maintains an internal topology of nodes connected to accounts, including information about IDs, node types and public keys. When nodes connect, they receive an initial message about which node IDs are connected, followed by messages when nodes connect and disconnect. Each of these messages also contain tallies of how many nodes of the various types are connected. This information is used to minimize broadcasts; if no nodes of the targeted type are connected, the message is never sent.

HTTP is not a real-time protocol, so messages are fetched by the node by continuously performing so-called "long polling" on an HTTP endpoint - issuing a long-lived request, which either times out or is fulfilled with one or more received messages. When a response is received, a new request is established. This technique is known to survive hardware network filters and proxies well, and disconnections are in many cases detected immediately, as the current long-lived request's connection is severed. If a connection is severed in this way, or if a node has not contacted the Relay server for a short period of time to establish a new polling request, or if a node tells the Relay server that it is shutting down, the node is marked as disconnected and other nodes are notified as per above.

The use of Relay Selector allows for several Relay servers to be used if necessary as the amount of traffic grows, and for accounts to be routed to Relay servers that are geographically close without the Relay connection process changing.

Remote Status messages in practice

Syspeace services and consoles receive lists from the Syspeace backend server of the available Remote Status-enabled nodes, including public key and thumbprint information and their Relay node IDs. For example, a console knows about which services in the Syspeace account have enabled Remote Status, and can list them all. Combined with node connection information, it can issue a message asking for status reports, and correlate the status messages to the right services.

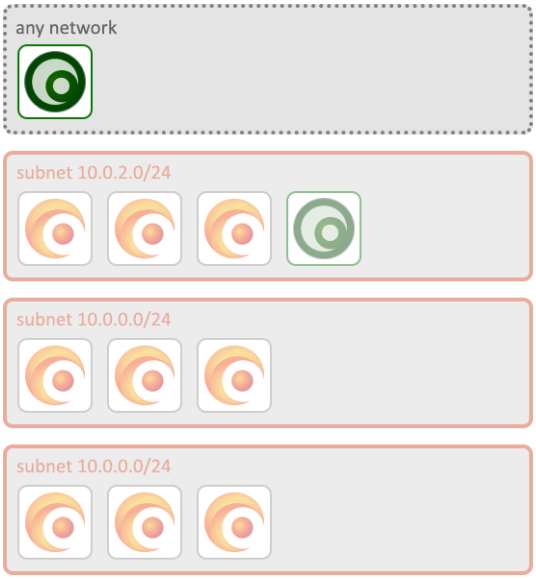
It can also issue commands by sending a message with a request to, for example, export settings to a service's node, and receive the exported settings in a response message.

Status information is sent in chunks, batched into a summary with the initial message. As various information changes, only the changed status information is announced. Large chunks, such as information about a large number of blocks, can be “stubbed out” to a version where only a short description (like: “250 blocks”) and an identifier of the contents are included, and consoles can register their interest to receive the full information (like the complete block information). Console uses this to minimize the amount of information automatically sent, yet receive detailed information when a service is selected.

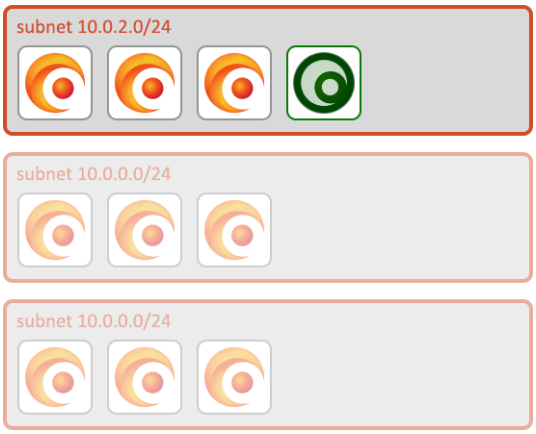
Relay topology

Relay is used to facilitate connections between nodes without line-of-sight. Sending messages via Relay implies a round-trip to the Relay server. The Remote Status Console has a Connection info window through which the round-trip time (measured by the node sending a very short message to itself via Relay and timing the period from it being sent to it being received) can be inspected, as well as the name and certificate of the Relay and Relay Selector servers.

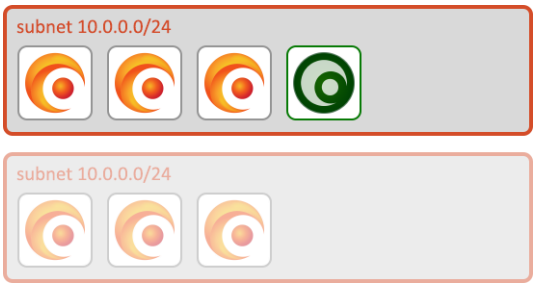
Without Relay, a Remote Status Console would not necessarily be able to reach other nodes.



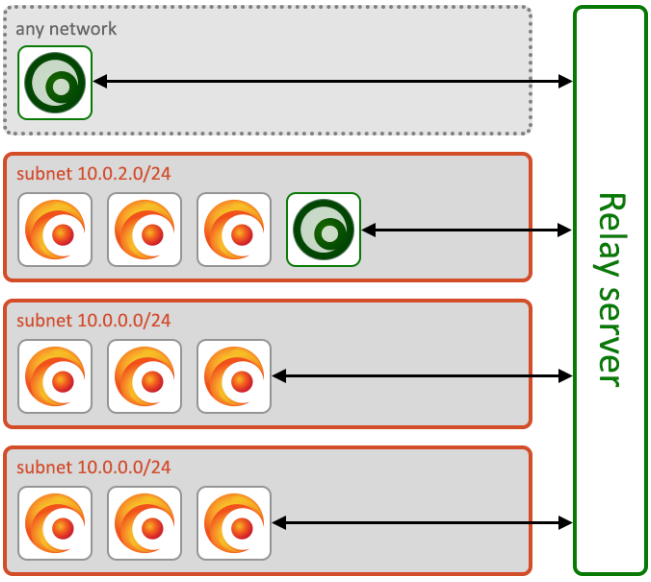
A Console outside of the subnets would not be able to reach any nodes unless they had public IP addresses.



A Console within one subnet might not be able to reach nodes in other subnets.



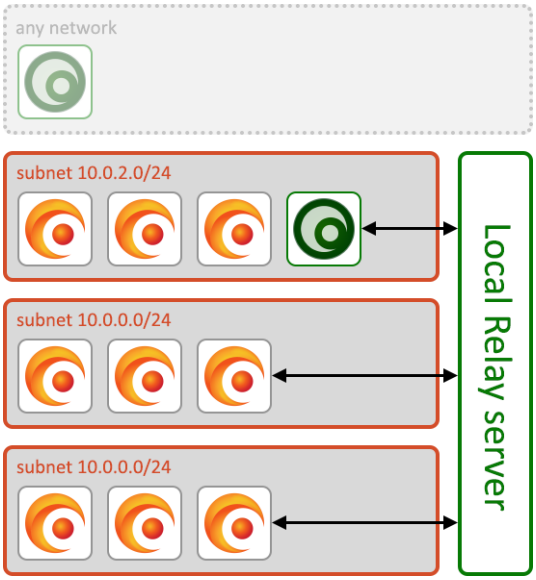
Even if one subnet was reachable from another subnet, not all subnets are, and some subnets use the same IP range. A Console dependent on connecting directly to the nodes would not be able to connect to all of them simultaneously.



With a Relay server, all nodes can talk to each other indirectly. This is the same with a hosted Relay server and the default Relay server. All nodes need to be able to see the Relay server.

If the Relay server is hosted publicly, connecting to it is possible from any network.

The Relay server can be hosted at another location if desired.



With a locally hosted Relay server, all nodes can talk to each other indirectly. All nodes need to be able to see the Relay server, but the Relay server does not need to be exposed publicly.

If the Relay server is hosted locally, connecting to it is only possible from within this network.

The Relay server will still need access to the Syspeace backend server to validate authentication tokens.

Compatibility

Syspeace services running on Windows Server 2003 cannot participate in Syspeace Remote Status due to practical difficulties and incompatibilities encountered during development. In particular, support for some modern encryption standards or practices are absent in Windows Server 2003 and support will not be added by Microsoft. Running Syspeace on Windows Server 2003 remains supported without this functionality, but we always recommend running a version of Windows Server currently supported by Microsoft.