Tech Tip: Saving Your iControl Changes II: Encrypted Configurations

Don MacVittie, 2008-20-03

While saving the configuration changes you've made is important stuff, our last trip into this realm only provided a solution for unencrypted configurations. If your configuration files are stored encrypted on the BIG-IP, then you'll need a different set of routines to deal with the encryption.

The encryption routines are just a subset of the core configuration file manipulate routines that we looked at in the last installment of this series. For background it might be worthwhile to read the first article in this series, particularly if the concept of saving iControl configuration changes is new to you.

Note that we do not develop code here – the code presented in the previous Tech Tip on this topic can be slightly modified to call the same routines with encrypted inserted into the name and the passphrase parameter handled, so we are not re-presenting them here. All of these routines, like all of the unencrypted routines, are members of the iControl.System.ConfigSync class.

Some of the core calls we covered last time work on both encrypted and unencrypted configuration files. These are the routines that deal with file-level manipulation and do not need to actually open the file. They are

delete_configuration()
download_configuration()
upload_configuration()

When dealing with calls that must encrypt or decrypt the configuration file, you'll have to provide the pass phrase, so we have a separate set of routines that have the word 'encrypted' inserted into them:

install_encrypted_configuration()
save_encrypted_configuration()
save_partial_encrypted_configuration()

To load a different configuration than the one currently loaded, in an encrypted environment you should call install_encrypted_configuration()

Passing in the name of the configuration file to be loaded and the passphrase to decrypt that file.
This decrypts the file, loads it into memory, and makes it the running configuration. This is particularly useful if you are testing changes and have to stop for one reason or another – it allows you to save the test configuration to a given name, and then reload it when you can test again – like in non-peak hours.

To save the current running configuration in an encrypted format, you can call
save_encrypted_configuration()
which takes the filename to save the configuration as and passphrase to use to encrypt the contents of the file. Note (as indicated in the Wiki documentation) that this routine saves the entire configuration, not just the bits saved in the .conf files.

To save portions of the running configuration in an encrypted format, you'll want to call
save_partial_encrypted_configuration()
Which takes the filename and passphrase along with two arrays that indicate which parts of the configuration to explicitly save and which to excluded from saving.

Note that features to exclude is limited to private keys, and features to include is limited to Enterprise Manager ISO disks… Optional items that you can decide what to do with at save time.

With that you should have enough to get up and running with encrypted configurations.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com