Updating the Firmware on a Cisco UCS C-Series Server Using the Non-Interactive HUU (NI-HUU)

Overview

Non Interactive Host upgrade utility or NI-HUU is an application that is used to update firmware on Cisco C-Series servers. With the Multi server NI-HUU, you can update multiple C-Series servers using scripts simultaneously. To use this feature there are tools available for Linux.

Pre-Requisite

Ensure that you have the following installed:

1. Python version 3.x
2. Python-multiprocessing package
3. Pycrypto-2.6

Linux Tool and Commands

This is a python based utility. This utility can be used to update multiple C-Series servers from a Linux host machine simultaneously. The usage of the utility is as follows:

`Usage: update_firmware.py` ***`[options]`***

The parameters for this utility can be given from the command line or in a configuration file.

Table 1: Options

Command	Description
--version	Shows the version number of the program and exit.
-h, --help	Show this help message and exit

Table 2: Single Server Options

Command	Description
-a a.b.c.d, --address=a.b.c.d	CIMC IP address
-u USERNAME, --user=USERNAME	Username of the CIMC admin user
-p PASSWORD, --password=PASSWORD	Password of the CIMC admin user
-q SKIPMEMORYTEST, --skipMemoryTest=Enabled/Disabled	Skip Memory Test Feature can be either Enabled or Disabled
-m ucs-c240-huu-146.iso, --imagefile=ucs-c240-huu-146.iso	HUU iso image file name
-i a.b.c.d, --remoteshareip=a.b.c.d	IP address of the remote share
-d /data/image, --sharedirectory=/data/image	Directory location of the image file in remote share
-t cifs/nfs/www, --sharetype=cifs/nfs/www	Type of remote share
-r REMOTESHAREUSER, --remoteshareuser=REMOTESHAREUSER	Remote share user name
-w REMOTESHAREPASSWORD, --remotesharepassword=REMOTESHAREPASSWORD	Remote share user password
-y COMPONENTLIST, --componentlist=COMPONENTLIST	Component List
-f LOGFILE, --logrecordfile=LOGFILE	Log file name where log data is saved
-b CIMCSECUREBOOT, --cimcsecureboot=CIMCSECUREBOOT	Use CimcSecureBoot. Default is NO. Options yes/no
-k CMCSECUREBOOT, --cmcsecureboot=CMCSECUREBOOT	Use CmcSecureBoot. Default is NO. Options yes/no
-M MOUNTOPTION, --mountOption=MOUNTOPTION	Use mountOption in case of CIFS share to specify the security option
-R REBOOTCIMC, --reboot=REBOOTCIMC	Reboot CIMC before starting update. Options yes/no
-T UPDATETIMEOUT, --timeoutalue=UPDATETIMEOUT	Timeout Value for update

Command	Description
-o UPDATESTOPONERROR, --stopOnError=UPDATESTOPONERROR	Use this option if you want to stop the firmware update once an error is encountered?
-v UPDATEVERIFY, --updateverify=UPDATEVERIFY	Use this option to verify update after reboot
-S USESECURE, --Secure=USESECURE	Use HTTPS. Default is yes. Options yes/no

Table 3: Multiple Server Update Options

Command	Description
-c CONFIGFILE, --configfile=CONFIGFILE	Name of the file with the list of CIMC IP address and other data
-l LOGFILE, --logfile=LOGFILE	Log file name where the log data will be saved
-s USESECURE, --secure=USESECURE	Use HTTPS. Default is yes. Options yes/no
-e INFILE, --encrypt=INFILE	Public key file.
-g, --generatekey	Generate public and private keys
-j, --displayComponentList	Display List of component
-V, --Version	Display version.

Sample Configuration

```
#--------------START CNF-------------------------
#
# Use this flag use_http_secure to toggle betwwen https and http protocol
use_http_secure=yes
# Firmware update should complete within this many minutes. This value will be
# sent along with the firmware update XML request to the CIMC
update_timeout=60
graceful_timeout=3
doForceDown=yes
# Should the firmware update process stop the update once an error is encountered?
update_stop_on_error=no
# Is it required to verify the update by rebooting to the same HUU image after the update
# gets completed?
update_verify=no
# Do you wish to secure Cimc Boot.Use this flag use_cimc_secure.
use_cimc_secure=no
# Do you wish to secure Cmc Boot.Use this flag use_cimc_secure.
use_cmc_secure=no
# Feature is used for skip Memory Test and it reduce the boot time. It support Enabled or
Disabled options.
#skipMemoryTest=Disabled
# List of components to be updated. Check the HUU release note for the list of
# supported components. Multiple components should be comma separated.
update_component=I350
#update_component=9266-8i, BIOS, CIMC, I350
#update_component=all
#update_component=HDD
```

```
#update_type=immediate
#update type can be either delay for a delayed firmware update upon host reboot or immediate,

to start firmware update

#reboot CIMC before Update
reboot_cimc=no
# IP address of the remoted share (cifs/nfs/www) holding the HUU image for booting
# for www share ip address can be given as http://<IPAddr>, https://<IPAddr> or <IPAddr>
remoteshareip=10.104.255.254
# Directory within the share where the HUU image is being kept
sharedirectory=/CIFSShare
# Type of share (nfs/cifs/www)
sharetype=cifs
# Username of the remote share to login to
remoteshareuser=username
# Password corresponding to the remote user
remotesharepassword=password
#Optional mount parameter for CIFS share only. Provide "ntlm,vers=2.0" for CIFS server
version 2.0
(SMB protocol version), default supported version is 3.0
#mountOption=ntlm
#If the running CIMC version is 4.2.2a and above, please provide "ntlmssp or ntlmv2,vers=2.0".
#mountOption=ntlmv2,vers=2.0 or
#mountOption=ntlmssp,vers=2.0

# Password file for remoteshare. If this option is provided, then the above option
(remotesharepassword) should not be given
#remoteshare_passwordfile=/home/arunven/Python_Script/python_script_old/Pyrhon_loop/CRYPTO/remshare.pass

#Common CIMC password --> The password provided below along with CIMC information will be
ignored.
#cimc_password_file=/home/arunven/Python_Script/python_script_old/Pyrhon_loop/CRYPTO/cimc.pass


# Enter the list of CIMC ip addresses where the firmware needs to be updated
address=10.104.255.180, user=cimc_user, password=cimc_password, imagefile=huu.iso

#-------------------------END CNF-------------------------
```

Save this to a file (example config.in) and use the following command:

`./update_firmware.py -c config.in`

Canceling a Delayed Update

The same configuration file, which was used for server firmware update, has to be passed with details of all the servers where the update has to be canceled.

Note Firmware update cancel request is to be sent only in delayed firmware update and when update has not started to avoid corruption of firmware.

`./update_firmware.py cancel -c config.in`

A sample config file multiserver_config is also available in the SVN location.

This utility assumes that the Python interpreter is installed at `/usr/bin/`. In case the Python interpreter is installed at some other location, this utility can also be invoked as follows:

```
/usr/location/python update_firmware.py -c config.in
```

This utility will connect to the CIMC(s) mentioned in the configuration file and boot the host into the mentioned HUU iso. On booting the HUU ISO will detect that a non-Interactive update needs to be done. HUU completes the update and send the results to the CIMC(s), which is responded back to the python utility to be displayed. If a **Verify** option is also mentioned in the Python utility configuration file, the host reboots in HUU and complete the verification.

Encrypting Passwords

Generating Public and Private Keys

This utility allows users to generate encrypted passwords and make use of them. To generated public and private keys use **-g** option.

Example:

```
./update_firmware.py -g
```

This option prompts for a passphrase for the keys. Press **Enter** if you do not want to provide the passphrase. The output of this command are the following two files:

- Private key file—keys.pem
- Public key file—keys.pub

Generating Encrypted Passwords

To generate encrypted passwords use the **-e** option. This also prompts for passphrase. You must enter the passphrase provided during key generation and the TEXT to be encrypted. This TEXT is the password. This command generates a file containing the encrypted password. The parameter for the option **-e** is the public key file.

Example:

```
./update_firmware.py -e keys.pub
```

Encrypted password file—password.key

You must rename it and save it. You need to generate different encrypted password files for Remote Share Password and CIMC passwords, if they are different from each other.

Using the Encrypted Password Files

Only configuration file can make use of these encrypted passwords. There are two options in the configuration file using which you can use to provide the encrypted password files for CIMC and Remote Share passwords.

- remoteshare_passwordfile=<File Path>
- cimc_password_file=<File Path>

Password file for remoteshare—If this option is provided, then the above option should not be given
`remoteshare_passwordfile=/home/arunven/Python_Script/python_script_old/Pyrhon_loop/CRYPTO/remshare.pass`

Common CIMC password—The password provided below is ignored
`cimc_password_file=/home/arunven/Python_Script/python_script_old/Pyrhon_loop/CRYPTO/cimc.pass`

Note Once you use the **cimc_password_file** option all the CIMC(s) mentioned in the configuration use this common file.

When you run the `update_firmware.py` script to start the update, it prompts for the passphrase that you had provided during the key generation.