Cisco Embedded Automation Systems - EASy
CA Certificate Expiration Notification

January 2010

Objective

Objective

- Problem:

  Customers who have a large number of devices with CA certificates run into issues keeping them current

- Solution:

  This script monitors the certificates and sends both syslog and SNMP warnings

  It runs once a day at a specified time and checks the expiration date of each certificate

  If a certificate will expire within a specified number of days, it sends a notification specifying when the license expires

Overview

Application or Service	This is a self-monitoring and alerting tool.
Technology	Cisco IOS® Crypto images with CA Certificate Authentication
Problem	With a large number of devices having certificates expiring, getting new certificates created and deployed takes a lot of time.
Impact	This tool reduces the number of internal priority cases.
Non-EASy Solution	Manually check devices or use automation tools.
Benefit of EASy Solution	This solution allows devices to do self-monitoring and alerting, which will dramatically reduce the number of cases by dealing with licenses before they expire. Theoretically, this is much more efficient than having to scan all devices periodically looking for expiring or expired licenses.
Category	Network Management – **Security**

CA Certificate Expiration Notification Script

Deploying a New Box

1. A technician deploys a new device and cables it
2. A base configuration is applied to the box manually or through a BOOTP process or Cisco® Networking Services CE deployment, including a list of attributes to poll and copy over the Embedded Event Manager (EEM) policy
3. The policy runs once a day and notifies when any certificates are approaching expiration

CA Certificate Expiration Notification Script

Function of the EEM Policy

1. The EEM policy is called from cron once a day and runs the “show crypto ca certificates” command
2. It scans the output for “end date:” lines
3. It compares each end date to the current time plus a specified number of days
4. If the license is expired or expiring, it generates both a syslog and SNMP message

Setup Procedure for the Device

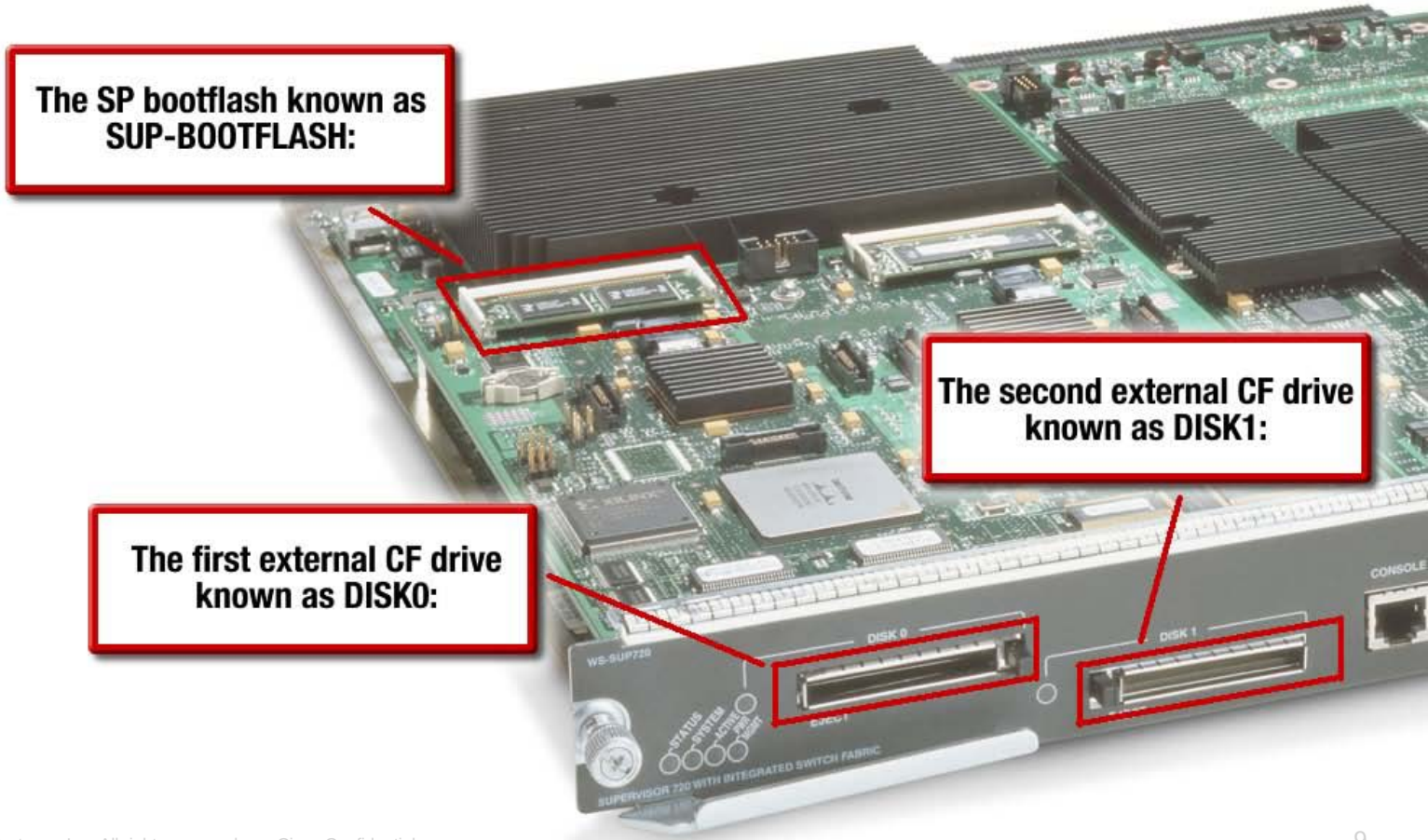
Requirements

- Requires Cisco IOS® Crypto image
- Tested on EEM 2.4

EEM

Where to Store the Tcl Script

- Tcl scripts are typically stored in one of three places: in the switch processor bootflash (known as SUP-BOOTFLASH:) or on one of the compact flash drives on the Supervisor front panel



Installing Using EASy Installer
tm_ccen.tar Package

1. Create the EASy Installer alias if it does not already exist:

   alias exec easy-installer tclsh tftp://*192.168.1.1*/easy-installer.tcl

2. Create a directory to install the package:

   mkdir flash:/EEM

3. Execute EASy Installer:

   easy-installer tftp://*192.168.1.1*/tm_ccen.tar flash:/EEM

- Choose option 2 to configure package parameters
- Choose option 3 to deploy package policies
- For further information on EASy Installer, see:

  http://nm-tac.cisco.com/easy-installer/easy-installer.html

Note: The address in *BLUE* is the address of your TFTP server where the package and the installer are located

EEM

Setting Up the Tcl User Directory

- Create the directory on your device:

  3400# mkdir flash:/EEM

  Create directory filename [EEM]?

  Created dir flash:EEM

  3400#

- Copy over the EEM policy:

  3400# copy tftp://192.168.1.1/tm_ccen.tcl flash:/EEM

- Tell EEM where the user policies are located:

  3400(config)# event manager directory user policy “flash:/EEM"

EEM
Registering the EEM Script

- Make the following configuration:

  Specify the EEM username if TACACS is enabled:

  3400(config)# event manager session cli username “eem_user”

- Set up your environment variables:

  Specify when to run the check:

  3400(config)# event manager environment Poll_Time 0 2 * * *

  Optionally specify the number of days to start warning:

  3400(config)# event manager environment Days_to_Warn 10

- Register the EEM policy:

  3400(config)# event manager policy tm_ccen.tcl type user

Note: The default is to start warning at seven days

EEM

Verifying the Correct Setup

- Check the environment variables:

```
3400# show event manager environment
No.   Name                          Value
1     Days_to_Warn                  10
2     Poll_Time                     0 1 * * *
```

- Check to see if the policy is properly registered:

```
3400# show event manager policy registered
No.  Class   Type  Event Type  Trap  Time Registered            Name
1    script  user    timer cron          Off   Wed Mar 25 10:05:08 2009  tm_ccen.tcl
 name {crontimer2} cron entry {0 1 * * *}
 nice 0 queue-priority normal maxrun 240.000
```

EEM
Output

***Mar 25 18:00:01.598: %HA_EM-6-LOG: tm_ccen.tcl: crypto ca certificate expires on 3400 on 09:11:25 CST Mar 26 2009**

SNMP TRAP:

Version: 1 (0)
Community: public
PDU type: TRAP-V1 (4)
Enterprise: 1.3.6.1.4.1.9.10.91
Agent address: 192.168.243.2 (192.168.243.2)
Trap type: ENTERPRISE SPECIFIC (6)
Specific trap type: 2
Timestamp: 6929300
Object identifier 1: 1.3.6.1.4.1.9.10.91.1.2.3.1.2.18
Value: GAUGE: 24 (0x18)
Object identifier 2: 1.3.6.1.4.1.9.10.91.1.2.3.1.3.18
Value: GAUGE: 0 (0x0)
Object identifier 3: 1.3.6.1.4.1.9.10.91.1.2.3.1.4.18
Value: GAUGE: 0 (0x0)
Object identifier 4: 1.3.6.1.4.1.9.10.91.1.2.3.1.5.18
Value: GAUGE: 0 (0x0)
Object identifier 5: 1.3.6.1.4.1.9.10.91.1.2.3.1.6.18
Value: OCTET STRING: flash:/EEM/tm_ccen.tcl
Object identifier 6: 1.3.6.1.4.1.9.10.91.1.2.3.1.7.18
Value: OCTET STRING: script: tm_ccen.tcl
Object identifier 7: 1.3.6.1.4.1.9.10.91.1.2.3.1.9.18
Value: INTEGER: 0 (0x0)
Object identifier 8: 1.3.6.1.4.1.9.10.91.1.2.3.1.10.18
Value: NOSUCHINSTANCE: no such instance
Object identifier 9: 1.3.6.1.4.1.9.10.91.1.2.3.1.11.18
Value: OCTET STRING: crypto ca certificate expires on 3400 on 09:11:25 CST Mar 26 2009
Object identifier 10: 1.3.6.1.4.1.9.10.91.1.2.3.1.13.18
Value: GAUGE: 0 (0x0)
Object identifier 11: 1.3.6.1.4.1.9.10.91.1.2.3.1.14.18
Value: GAUGE: 0 (0x0)
Object identifier 12: 1.3.6.1.4.1.9.10.91.1.2.3.1.15.18
Value: GAUGE: 0 (0x0)
Object identifier 13: 1.3.6.1.4.1.9.10.91.1.2.3.1.16.18
Value: GAUGE: 0 (0x0)

Detailed Script Analysis

Detailed Script Analysis
tm_ccen.tcl

```
::cisco::eem::event_register_timer cron name crontimer2 cron_entry $Poll_Time maxrun 240
#---------------------------------
# EEM policy that will periodically check for expired crypto keys and send SNMP trap
#
# Copyright (c) February 2009, jepalmer@cisco.com
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
```

- The first line registers the script with the event_register_timer (cron) so the script will know when to run
- All lines beginning with a “#” are comments and are there for informational purposes

Detailed Script Analysis

tm_ccen.tcl

```
if {![info exists Days_to_Warn]}  1
        set days 7
        } else {
        set days $Days_to_Warn
        }

set seconds [expr $days*24*60*60]                ;# Calculate seconds from days  2

set show_crypto_cmd {show crypto ca certificates}  3
```

1. Sets the days to 7 unless an environment variable was set; if the environment is set, it uses that value
2. Converts days to seconds
3. Assigns the show command to a variable

Detailed Script Analysis

tm_ccen.tcl

```
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

set routername [info hostname]
# 1. execute the command
if [catch {cli_open} result] {
    error $result $errorInfo
} else {
    array set cli1 $result
}
if [catch {cli_exec $cli1(fd) "en"} result] {
    error $result $errorInfo
}
```

1. These namespace import commands are required for every Tcl script
2. Saves the routername
3. Opens a TTY to run commands and puts Cisco IOS® Software in enable mode

Detailed Script Analysis

tm_ccen.tcl

```
set time_now [clock seconds]                                                    1
if [catch {cli_exec $cli1(fd) $show_crypto_cmd} result] {                       2
    error $result $errorInfo
} else {
    set cmd_output $result
    # format output: remove trailing router prompt
    set prompt [format "(.*\n)(%s)(\\(config\[^\n\]*\\))?(#|>)" $routername]    3
    if [regexp "[set prompt]" $result dummy cmd_output] {
       # do nothing, match will be in $cmd_output
    } else {
       # did not match router prompt so use original output
       set cmd_output $result
    }
}
```

1. Saves the current time
2. Executes the “show crypto ca certificates” command
3. Strips off the router prompt from the command returned and saves the output in the variable cmd_output

Detailed Script Analysis

tm_ccen.tcl

```
# 2. Scans "show crypto ca certificates" looking for expiring and expired licenses
1 set pat {end[ \t]+date:[ \t]+([0-9]{1,2}[-:/][0-9]{1,2}[-:/][0-9]{1,2}[ \t]+[A-Z]{3}[ \t]+[ADFJMNOS] \
                              [a-z]{2}[ \t]+[0-9]{1,2}[ \t]+[0-9]{1,4})}
  set license_conter 0
  set expired_conter 0
2 foreach line [split $cmd_output \n ] {
    regexp $pat $line _match clock_data
3   if {[info exists clock_data]} {
          set cv [clock scan $clock_data]
          set clock_target [expr ([clock seconds]+$seconds>=$cv) ? 1 : 0]
          set clock_expired [expr ([clock seconds]>=$cv) ? 1 : 0]
```

1. This block sets up the pattern to look for in the output of the show command; it also sets the counters to zero
2. This goes line by line through the show command looking for the pattern
3. If an “end date:” is found, this converts the time to clock ticks (epoch) and checks whether it is expired or will expire in the configured number of days

Detailed Script Analysis
tm_ccen.tcl

```
if { $clock_target==1 } {
     if { $clock_expired==0 } {
          set strdata [format "crypto ca certificate expires on %s on %s" $routername $clock_data]
          set intdata1 0
          incr license_conter
          } else {
          set strdata [format "crypto ca certificate expired on %s on %s" $routername $clock_data]
          set intdata1 1
          incr expired_conter
          }
```

- This block creates the appropriate message if a certificate is expired or about to expire

Detailed Script Analysis
tm_ccen.tcl

```
action_snmp_trap intdata1 $intdata1 strdata $strdata
if {$_cerrno != 0} {
     set result [format "component=%s; subsys err=%s; posix err=%s;\n%s" \
     $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
     error $result
}
action_syslog priority info msg $strdata
if {$_cerrno != 0} {
     set result [format "component=%s; subsys err=%s; posix err=%s;\n%s" \
     $_cerr_sub_num $_cerr_sub_err $_cerr_posix_err $_cerr_str]
     error $result
}
```

- This block generates an SNMP message and a syslog message and does necessary error checking

Detailed Script Analysis
tm_ccen.tcl

```
if {[info exists Debug_File]} {
    # attach output to file
    if [catch {open $Debug_File w+} result] {
        error $result
    }
    set fileD $result
    # save timestamp of command execution
    #      (Format = 00:53:44 PDT Mon May 02 2005)
    set time_now [clock format $time_now -format "%T %Z %a %b %d %Y"]
    puts $fileD "%%% Timestamp = $time_now %%%"
    puts $fileD $cmd_output
    puts $fileD "$license_conter license(s) expiring within $days days"
    puts $fileD "$expired_conter license(s) expired"
    close $fileD
}
```

- This block writes a debug file if the environmental variable is set