ufw-refresh.ros

Nicholas Huber

December 1, 2022

Contents

1 Refresh the IPs in the UFW by hostnames

1.1 Roswell Header stuff

```
#!/bin/sh
#|-*- mode:lisp -*-|#
#|
exec ros -Q -- $0 "$@"                  ; ; ; ; ; ; ; ; ; ;
|#
```

1.2 Info Block

```
;;;; UFW refresh
;;;  The purpose of this script is to check refresh which hosts are allowed to access a
```

```
;;;  by checking their current IPs, using hostnames, and checking them against IPs curr
;;;  in UFW.
;;;  Script must be run as ROOT user

;;; *TODO* modify to allow for arbitrary number of hosts
;;; *TODO* remove host specific functions
```

1.3 Roswell init stuff

voodoo magic:

```
(progn ;;init forms
  (ros:ensure-asdf)
  #+quicklisp(ql:quickload '() :silent t)
  )

(defpackage :ros.script.ufw-refresh.3876675733
  (:use :cl))
(in-package :ros.script.ufw-refresh.3876675733)
```

1.4 Hostnames and ports

Lists of the hostnames and their ports to check:

```
(defparameter *hostname* "jumpserver")
(defparameter *port* "22")
```

1.5 Get Current IPs

Functions for getting the current IPs based off of hostname `get-current-ip` takes hostname and calls `GETENT AHOSTS` and parses the result to get its current ip

```
(defun get-current-ip (hostname)
 "Get the current IP of a host using the hostname by calling GETENT"
  (car (uiop:split-string (uiop:run-program (uiop:strcat "/usr/bin/getent ahosts " host

(defun get-jumpserver-ip ()
 "Get the current IP of the JUMPSERVER"
  (get-current-ip *hostname*))
```

1.6 Get Old IPs

Functions for grabbing the IPs currently in UFW's rules `get-old-ip` calls `UFW STATUS` to get the IPs currently in the rules

```
(defun get-old-ip ()
 "Get the old IP of a host by taking the last element of a list of the output of UFW\'s
  (cdr (uiop:split-string (uiop:run-program "/usr/sbin/ufw status"))))
```

1.7 Delete Old Rules

Functions to delete old UFW rules `delete-old-rule` takes an IP and a Port and calls UFW with `UFW DLETE` to remove that rule based on the specific IP and Port

```
(defun delete-old-rule (ip port)
 "Delete the old rule for an allowing an IP"
  (uiop:run-program (uiop:strcat "/usr/sbin/ufw delete allow from " ip " to any port "

(defun delete-old-jumpserver ()
 "Delete the old rule allowing JUMPSERVER access"
  (delete-old-rule (get-jumpserver-ip) *port*))
```

1.8 Define New Rules

Functions used to define new UFW rules `add-new-rule` takes an IP and Port and calles `UFW INSERT` to insert a new rule allowing that IP on that Port

```
(defun add-new-rule (ip port)
 "Add a new rule allowing current IP access"
  (uiop:run-program (uiop:strcat"/usr/sbin/ufw insert 1 allow from " ip " to any port "

(defun add-new-jumpserver ()
 "Add a new rule allowing JUMPSERVER\'s new IP access"
  (add-new-rule (get-jumpserver-ip) *port*))
```

1.9 Replace IPs

Functions used to delete old UFW rules and define new ones `replace-ip` takes the old IP and new IP and calls the respective functions to delete and add new rules **TODO**: generalize `replace-ip` **TODO**: possbily replace rules regardless of whether or not they match?

```
(defun replace-ip (new-ip old-ip)
 "Replace old IP by calling DELETE-OLD-JUMPSERVER and ADD-NEW-JUMPSERVER"
  (cond ((string= new-ip old-ip)
         (progn
           (delete-old-jumpserver)
           (add-new-jumpserver)))))

(defun replace-jumpserver ()
 "Replace JUMPSERVER\'s IP by calling REPLACE-IP with GET-OLD-IP and GET-JUMPSERVER-IP
  (replace-ip (get-old-ip) (get-jumpserver-ip)))
```

1.10 Main Function

Main function to run program

```
(defun main (&rest argv)
  (declare (ignorable argv))
  (replace-jumpserver))
```

1.11 Roswell Footer Info

```
;;; vim: set ft=lisp lisp:
```