

SUNY POLYTECHNIC INSTITUTE

NCS 450

NETWORK SECURITY

Lab 4: Local Users, SSH, NTP, and Syslog

Author:

Jess YANARELLA

Tony PEREZ

Professor:

Ronny BULL

March 2015

1 Introduction:

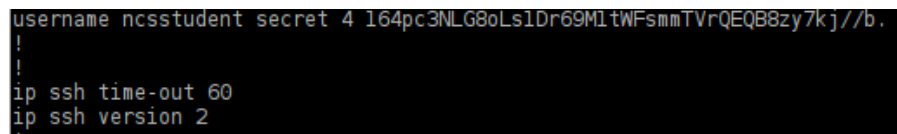
In this lab, we are creating a user on the router to be used for sshing. We need to enable SSH on the router by creating public/private keys. The POD PC will be used to connect to the router over the LAN. After we have a successful connection, the syslog server can be configured. It will be using the server from CS departmen and the main purpose of this is to log files in one place including user login, configuration changes, etc.

2 Diagram:

3 Local Users:

First, we need to create a local user account on the router. This is the secure way of to password protect an account.

```
(config)# username ncsstudent secret dh1240
```



```
username ncsstudent secret 4 164pc3NLG8oLs1Dr69M1tWFsmmTVrQEQB8zy7kj//b.
!
!
ip ssh time-out 60
ip ssh version 2
```

Figure 1: Local User

4 Enabling SSH Access:

In this section, we are creating a public/private key to be used so we can SSH to the router. We are changing some of the defaults by configuring the modulus size of 2048.

```
(config)# crypto key generate rsa
```

Next, we need to enable version 2 of SSH becuae it is more secure.

```
(config)# ip ssh version 2
```

We want the SSH connection to disconnect during authentication if no response is received from a client within 60 seconds.

```
(config)# ip ssh time-out 60
```

We also want it allow at most 3 retries when authenticating.

```
(config)# ip ssh authentication-retries 3
```

We now can configure a VTY line, which makes it require SSH only and not Telnet and SSH. SSH is more secure then Telnet. The only way to be able to SSH to the router is using the user we created in the first section.

```
(config)# line vty 0 4
(config)# login local
(config)# transport input ssh
```

5 Network Time Protocol:

We need to set up and verify NTP on our router and to do this, we are given the CS Department's NTP server IP address.

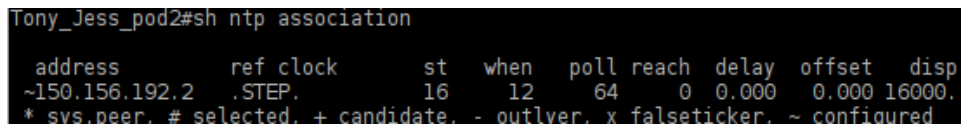
```
(config)# ntp peer 150.156.192.2
```

To set the clock so it displays the right time in our area.

```
(config)# clock timezone utc -5 0
```

Now that the NTP is configured, we can easily view the status of it. Before we do anything, we need to check that our router is able to communicate with the CS NTP server.

```
(config)# sh ntp association
```

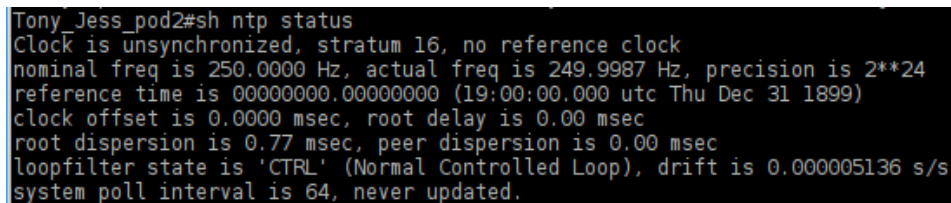


```
Tony_Jess_pod2#sh ntp association
address      ref clock    st  when  poll reach  delay  offset  disp
~150.156.192.2 .STEP.      16   12    64    0 0.000  0.000 16000.
* svs.peer, # selected, + candidate, - outlver, x falseticker, ~ configured
```

Figure 2: NTP Association

To show the status of the NTP synchronization

```
(config)# sh ntp status
```



```
Tony_Jess_pod2#sh ntp status
Clock is unsynchronized, stratum 16, no reference clock
nominal freq is 250.0000 Hz, actual freq is 249.9987 Hz, precision is 2**24
reference time is 00000000.00000000 (19:00:00.000 utc Thu Dec 31 1899)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.77 msec, peer dispersion is 0.00 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000005136 s/s
system poll interval is 64, never updated.
```

Figure 3: NTP Status

6 Logging to a Syslog Server:

While typing commands on the router, sometimes we are interrupted by status messages. These can be stopped by disabling the logging console.

```
(config)# no logging console
```

The Syslog server was created for this reason to accept and log all the information on a remote machine. The server runs the syslog service and maintains the log file. Now we can give the syslog server an IP address

```
(config)# logging 10.103.0.25
```

Next, we specify the syslog facility to use

```
(config)# logging facility local6
```

We can also log any successful and unsuccessful login attempts to the device.

```
(config)# login on-success log
```

```
(config)# login on-failure log
```

We also want to log all changes made to the configuration. This needs to be done in archive mode. To enter archive mode, just type archive when in config mode.

```
(config)# archive
```

```
(config-archive)# log config
```

The commands to actually enable logging and notify the syslog server of the logs

```
(config)# logging enable
```

```
(config)# notify syslog
```

To instruct it to not record keystrokes for password entries

```
(config)# sh logging
```

```
Tony_Jess_pod2#sh logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

Console logging: disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled
Buffer logging:  level debugging, 43 messages logged, xml disabled,
                  filtering disabled
Exception Logging: size (4096 bytes)
Count and timestamp logging messages: disabled
Persistent logging: disabled
Trap logging: level informational, 45 message lines logged
Logging to 10.103.0.25 (udp port 514, audit disabled,
link up),
6 message lines logged,
0 message lines rate-limited,
0 message lines dropped-by-MD,
xml disabled, sequence number disabled
filtering disabled
Logging Source-Interface:      VRF Name:

Log Buffer (8192 bytes):
Jan  2 12:00:02.599: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL: Module name = c1900 Next reboot level = ipbasek9 and Licen
= ipbasek9
Feb 23 21:54:43.111: %IFMGR-7-NO_IFINDEX_FILE: Unable to open nvram:/ifIndex-table No such file or directory
Feb 23 21:54:51.279: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
Feb 23 21:54:51.283: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
Feb 23 21:54:52.279: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
Feb 23 21:54:52.283: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
Feb 23 21:54:57.031: %USBFLASH-5-CHANGE: usbflash0 has been inserted!
```

Figure 4: Syslog

To show all records in the config log

```
(config)# show archive log config all
```

7 Basic Access Configuration:

The first part in this section wants us to set up a name for our router and configure DNS, which we did in previous labs. We saved our configuration and brought it back in from the

```
Tony_Jess_pod2#show archive log config all
idx  sess      user@line  Logged command
  1    1      console@console | logging enable
  2    1      console@console | notify syslog
  3    1      console@console | hidekeys
  4    1      console@console | exit
  5    1      console@console | exit
```

Figure 5: Archive Log

TFTP server.

Since we set up SSH earlier in the lab, we can now test it to make sure it's working. We are going to perform the SSH connection to the router from our POD PC.

```
# ssh ncsstudent@10.103.5.65
```

It prompted us to type in the password and after entering the credentials, the connection was successful.

```
(brain:~) yanarej> ssh ncsstudent@10.103.5.65
The authenticity of host '10.103.5.65 (10.103.5.65)' can't be established.
RSA key fingerprint is 4f:44:3a:d6:e2:aa:9a:62:4a:51:11:f3:32:d6:b8:de.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.103.5.65' (RSA) to the list of known hosts.
Password:
```

Figure 6: SSH

To test that the syslog server is working, we must connect to atlantis and then look at the /etc/syslog.conf file. The actual log file is located in /var/log/cisco.

```

yanarej@atlantis:~>cat /etc/syslog.conf
# $FreeBSD: src/etc/syslog.conf,v 1.30.4.3 2012/11/17 11:36:10 svnexp Exp $
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manpage.
*.err;kern.warning;auth.notice;mail.crit                /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.*                                                /var/log/security
auth.info;authpriv.info                                  /var/log/auth.log
mail.info                                                 /var/log/maillog
lpr.info                                                  /var/log/lpd-errs
ftp.info                                                  /var/log/xferlog
cron.*                                                    /var/log/cron
*.=debug                                                  /var/log/debug.log
*.emerg                                                  *
local6.*                                                  /var/log/cisco
# uncomment this to log all writes to /dev/console to /var/log/console.log
# touch /var/log/console.log and chmod it to mode 600 before it will work
#console.info                                             /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600 before it will work
#*. *                                                    /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*. *                                                    @loghost
#*. *                                                    @logbox
# uncomment these if you're running inn
# news.crit                                               /var/log/news/news.crit
# news.err                                               /var/log/news/news.err
# news.notice                                            /var/log/news/news.notice
!ppp
*. *                                                    /var/log/ppp.log
!*

```

Figure 7: Syslog Config File

Using the grep command with our router's hostname, we were able to find any changes made.

```
# cat /var/log/cisco | grep 'pod2r'
```

To watch the log file in real time

```
# tail -f /var/log/cisco
```

```

Feb 25 12:04:21 pod7r.cs.sunyit.edu 52: Feb 25 16:04:48.087: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:exit
Feb 25 12:04:21 pod7r.cs.sunyit.edu 53: Feb 25 16:04:49.027: %SYS-5-CONFIG I: Configured from console by console
Feb 25 12:07:33 pod4r.cs.sunyit.edu 43: Feb 25 16:07:59.952: %SYS-6-TTY_EXPIRE_TIMER: (exec timer expired, tty 0 (0.0.0.0)), user
Feb 25 12:09:47 pod1r.cs.sunyit.edu 38: Feb 25 16:10:13.814: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:!exec: enable
Feb 25 12:09:49 pod1r.cs.sunyit.edu 39: Feb 25 16:10:16.722: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:!exec: enable
Feb 25 12:10:11 pod1r.cs.sunyit.edu 40: Feb 25 16:10:38.323: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:hostname router
Feb 25 12:10:21 pod1r.cs.sunyit.edu 41: Feb 25 16:10:48.335: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:hostname router_pod_1
Feb 25 12:12:22 pod1r.cs.sunyit.edu 42: Feb 25 16:12:49.256: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:hostname haha
Feb 25 12:12:35 pod1r.cs.sunyit.edu 43: Feb 25 16:13:02.188: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:hostname router_pod_1
Feb 25 12:13:36 pod2r.cs.sunyit.edu 48: Feb 25 16:14:03.372: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:!exec: enable
Feb 25 12:14:43 pod2r.cs.sunyit.edu 49: Feb 25 16:15:10.589: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:interface GigabitEthernet0/0
Feb 25 12:14:53 pod2r.cs.sunyit.edu 50: Feb 25 16:15:20.041: %SYS-5-CONFIG I: Configured from console by console
Feb 25 12:14:57 pod4r.cs.sunyit.edu 44: Feb 25 16:15:24.343: %PARSER-5-CFGLG_LOGGEDCMD: User:console logged command:!exec: enable

```

Figure 8: Log Output