Branch: master

Find file Copy path

cs-35l / assignment8 / log.txt

prithvikannan passing more test cases

93ef48b 3 days ago

1 contributor

Raw Blame History

143 lines (121 sloc) 4.81 KB

```
beaglebone setup:
ssh root@192.168.7.2

connmanctl
enable wifi
scan wifi
services
agent on
connect wifi_2cf7f106a0ab_4352333736302d77696669_managed_psk
    3760ClassNet
quit

wlan0     Link encap:Ethernet  HWaddr 2c:f7:f1:06:a0:ab
          inet addr:10.97.85.28  Bcast:10.97.85.255  Mask:255.255.255.0
          inet6 addr: fe80::2ef7:f1ff:fe06:a0ab/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST DYNAMIC  MTU:1500  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:246 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13936 (13.6 KiB)  TX bytes:55663 (54.3 KiB)
sudo apt-get update
sudo apt-get install xauth
sudo apt-get install xvfb

nano /etc/ssh/ssh_config
    had to change to ForwardX11
nano /etc/ssh/sshd_config

sudo apt-get install firefox-esr-l10n-en-gb


on the server:

ssh-keygen
    hit enter
    Generating public/private rsa key pair.
    Enter file in which to save the key (/root/.ssh/id_rsa):
    Created directory '/root/.ssh'.
    Enter passphrase (empty for no passphrase):
    Enter same passphrase again:
    Your identification has been saved in /root/.ssh/id_rsa.
    Your public key has been saved in /root/.ssh/id_rsa.pub.
    The key fingerprint is:
    8a:02:4b:31:21:07:e2:c3:b6:0f:b1:5a:e0:b8:01:f3 root@beaglebone
    The key's randomart image is:
    +---[RSA 2048]----+
    |+o.              |
    |=..              |
    |+O               |
    |=+B              |
    |+*E     S        |
```

```
    |o*o  . .         |
    |+ ... .          |
    |   .             |
    |                 |
    +-----------------+

sudo useradd -d /home/harsh -m harsh
sudo passwd harsh
    set password to 'harsh'
cd /home/harsh
sudo mkdir .ssh
sudo chown -R harsh .ssh
sudo chmod 700 .ssh

I thought I was going to be working with Harsh, so I made an account under his
name on my beaglebone. However, Harsh was unable to make it to the lab so I
ended up working with Anirudh Mani instead.

I told my partner, Anirudh, to use harsh@10.97.85.28 to log into my beaglebone
using the password 'harsh' to first verify that the account worked.

on the client:

ssh-keygen
    Your identification has been saved in /root/.ssh/id_rsa.
    Your public key has been saved in /root/.ssh/id_rsa.pub.
    The key fingerprint is:
    14:a1:ec:dd:04:a6:1e:8b:8a:b9:5f:8c:a1:1c:ab:04 root@beaglebone
    The key's randomart image is:
    +---[RSA 2048]----+
    |        =.       |
    |     . + o       |
    |      = . .      |
    |     + = o       |
    |E.. . + S .      |
    |o+o=             |
    |++o o            |
    |o. .             |
    |o..              |
    +-----------------+

ssh-copy-id -i prithvi@10.97.85.36
eval $(ssh-agent)
ssh-add
ssh prithvi@10.97.85.36
nano oooie.txt
    I was able to log into Anirudh's beaglebone using the account he made for
    me without the password since I had saved the key within the ssh-agent.
    When I was logged in, I created a file called 'oooie.txt' and wrote the
    message 'i got in without a password'. Anirudh could see this message on his
    beaglebone, so we know it worked.

ssh -X prithvi@10.97.85.36
    Now I logged back into Anirudh's beaglebone using -X forwarding and I was
    able to open firefox and see the window appear on my screen.

HOMEWORK:

ssh root@192.168.7.2
scp root@192.168.7.2:/sys/bus/i2c/devices/0-0050/eeprom ~/Desktop
    Ran this command to grab the eeprom file onto my local desktop. Then I moved
    the eeprom to my linux server.

gpg2 --gen-key
    I ran this originally on server 3 but got the entropy bug so after checking
    a piazza post where it said how to check the entropy of the system:
```

```
    cat /proc/sys/kernel/random/entropy_avail, I ended up using server 10.

    gpg: key 86BE3653 marked as ultimately trusted
    public and secret key created and signed.
    gpg: checking the trustdb
    gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
    gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
    pub   2048R/86BE3653 2019-11-25
        Key fingerprint = 6C5A 9851 80CE E5B1 95B6  6931 96C2 C939 86BE 3653
    uid                  Prithvi Kannan (making a key using gpg2) <prithvi.kannan@gmail.com>
    sub   2048R/AFED4338 2019-11-25
gpg2 --clearsign eeprom
gpg2 --detach-sign  eeprom

mkdir -m go-rwx .gnupg
gpg2 --homedir .gnupg --import hw-pubkey.asc
gpg2 --homedir .gnupg --verify eeprom.sig eeprom
    200 gpg: Signature made Mon 25 Nov 2019 01:33:09 PM PST using RSA key ID 140714F2
    < lgpg: Good signature from "Prithvi Kannan (test key) <prithvi.kannan@gmail.com>"
    gpg: WARNING: This key is not certified with a trusted signature!
    gpg:          There is no indication that the signature belongs to the owner.
    Primary key fingerprint: 0468 D47C E605 87D7 AD4C  4DA8 4DB5 769D 1407 14F2

awk '200 < length' log.txt hw.txt


```