Host Security and Remote Connections - LAB 2

1. Write a script to ‘rsync’ some webpages from your ‘testing server’ (on your local system) to a ‘production server’. The ‘production server’ can be a (different) virtual machine on your laptop.  
   Test by creating some new pages; verify that they get sync’ed.  
   Set the rsync up so that when pages don’t exist any longer on the source system, then these pages will be deleted automatically on the remote system. Add some logging about the deleted files. Always verify with the ‘--dry-run’ option before taking into production.

Document the steps:

| > First I installed ‘openssh-server’ on both machines.  $ sudo apt install openssh-server  > Then I generated ssh keys on both machines and saved them  in /home/<USER>/.ssh/key\_for\_remote\_connection  $ ssh-keygen -t ed25519  > Afterwards I stored the key of the remote server on the local server so that I could establish connections without having to enter a password every single time.      > Note: It’s also useful to enable ssh in firewall (or deny if necessary) for the remote machine.    > Then I created directories on both machines.  $ mkdir ~/lab\_2\_server  > And added some dummy files on my local ‘server’.  $ touch home.html index.html login.html  > And then I created an rsync\_script.sh file which will run every 10 seconds with the help of crontab.  “””  #!/bin/bash  # Week 11  # Exercise: Secure Connections - Lab 2 - Rsync  # Author: Rodzers Usackis  # ACS201  help\_message="  Description: This script rsync's files between the local and remote server.  Usage: **'$0 [OPTIONS]'**  OPTIONS:  -h, **--help** Display these instructions  no option Execute this script  "  commandline\_input=**$1**  # Assignment only asked to create a script for rsync and not automating the process of setting up the ssh keys...  **execute\_script**() {  if [ **"$commandline\_input"** = **""** ]  then  **printf** **"\n%s\n\n%s\n\n\n"** **"--- Files on local server before rsync ---"** **"$(ls -l ~/lab\_2\_server/)"**  **printf** **"\n%s\n\n%s\n\n\n"** **"--- Files on remote server before rsync ---"** **"$(rsync server@192.168.122.107:~/lab\_2\_**>  **printf** **"\n%s\n\n"** **"--- Executing rsync command ---"**  rsync -av **--delete** ~/lab\_2\_server/ server@192.168.122.107:~/lab\_2\_server/  **printf** **"\n%s\n\n%s\n\n\n"** **"--- Files on remote server after rsync ---"** **"$(rsync server@192.168.122.107:~/lab\_2\_s**>  elif [[ **$commandline\_input** == **-h** || **$commandline\_input** == **--help** ]]  then  **printf** **"%s\n"** **"$help\_message"**  exit 0  else  **printf** **"\n%s\n%s\n\n"** **"Not sure what you were trying to do..."** **"Type $0 -h for help."**  fi  }  execute\_script  “””    And an example of files being removed from remote if not present in local… |
| --- |