Assignment 4: Remote Acquisition

Pawan Chandra, Aditya Subedi, EJ Truscott, Josh Zurbuchen

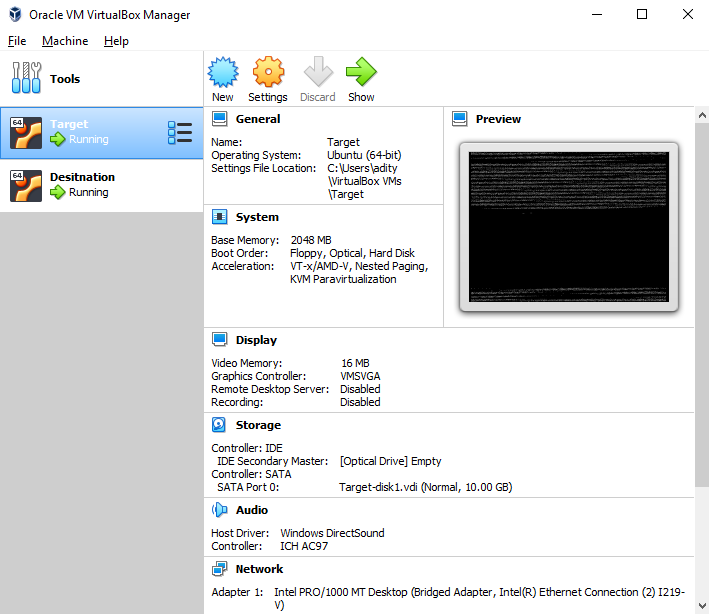
California State University, Sacramento

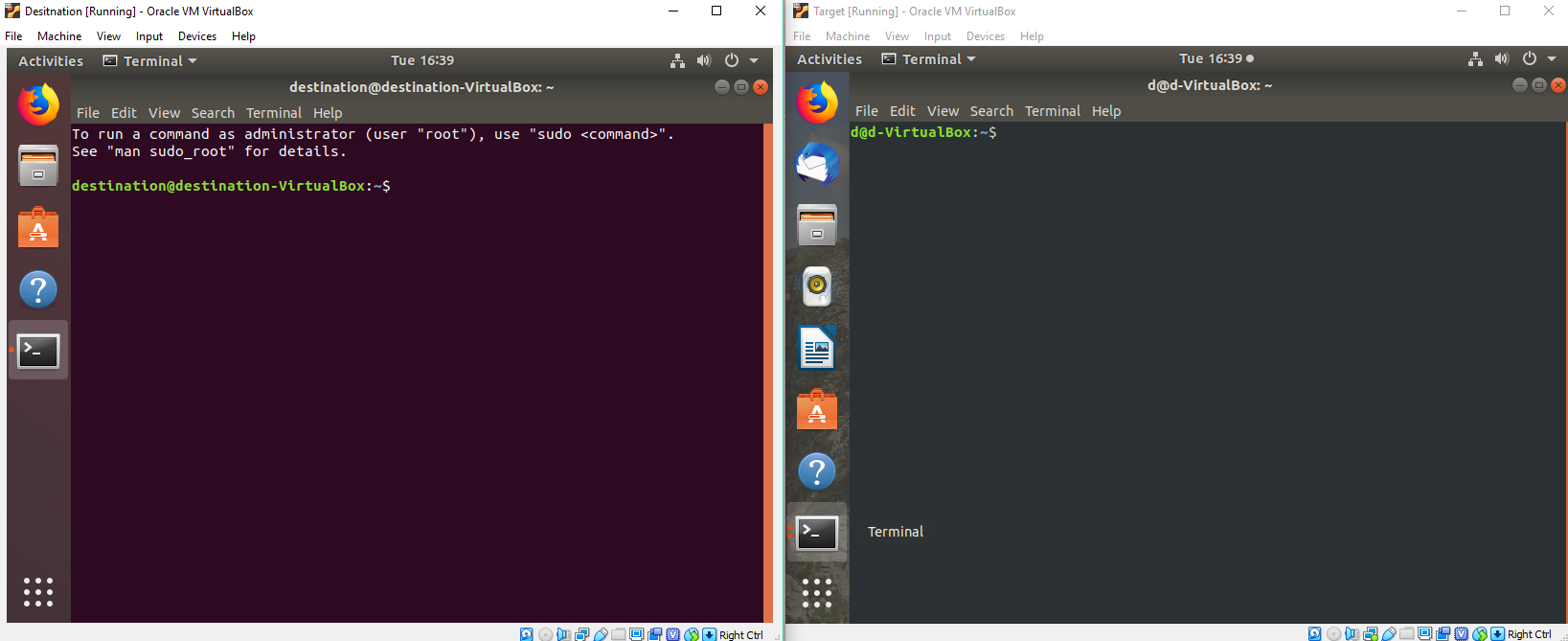
CSC 153, Section 2

Assignment 4: Remote Acquisition

**Imaging from a Live Source**

1. Here we have created two VMs in VirtualBox: Destination and Target. They are both running Ubuntu (64-bit) OS.

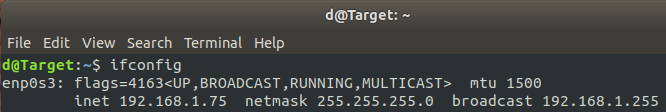




2) We obtained the IPs of both machines by using ifconfig. Also, we were able to make the machines talk to each other (sending packets of data) by pinging the IP addresses.

**Targets IP:**

* 192.168.1.75



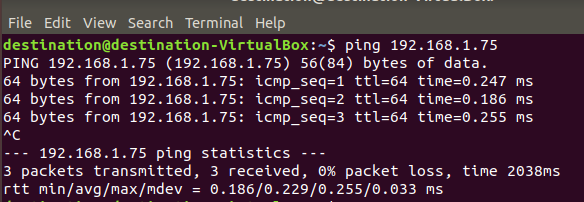
**Destination IP:**

* 192.168.1.99



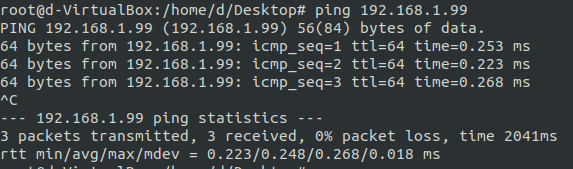
**Ping Target:**

* ping 192.168.1.75



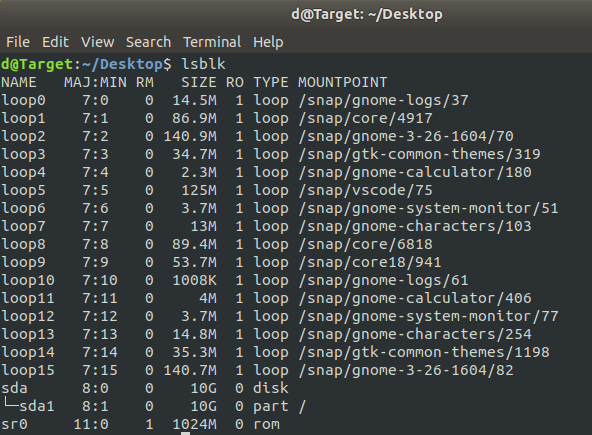
**Ping Destination:**

* ping 192.168.1.99



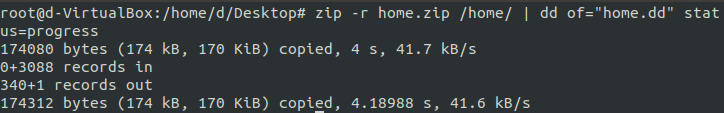
**Information about target drive**

* **lsblk** lists information about all available or the specified block devices.



**Imaging target:**

* zip -r home.zip /home/ | dd of="home.dd" status=progress
  + zip -r : recurse into directories
  + status=progress : output progress while making an image with dd
  + of: output file.



**Encrypting the image:**

* gpg is a encryption and signing tool
  + -c : Encrypt with a symmetric cipher using a passphrase. The default symmetric cipher used is CAST5
  + Passphrase used: E^/rd(~8$KLCe-}`



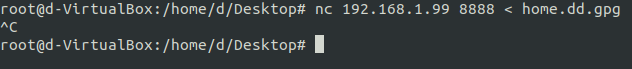
**Obtaining original hash of the image**

* md5sum - compute and check MD5 message digest
* sha1sum - compute and check SHA1 message digest



**Perform remote acquisitions [sending it to destination machine]**

* nc 192.168.1.99 8888 < home.dd.gpg



**Perform remote acquisitions [receive image from destination machine]**



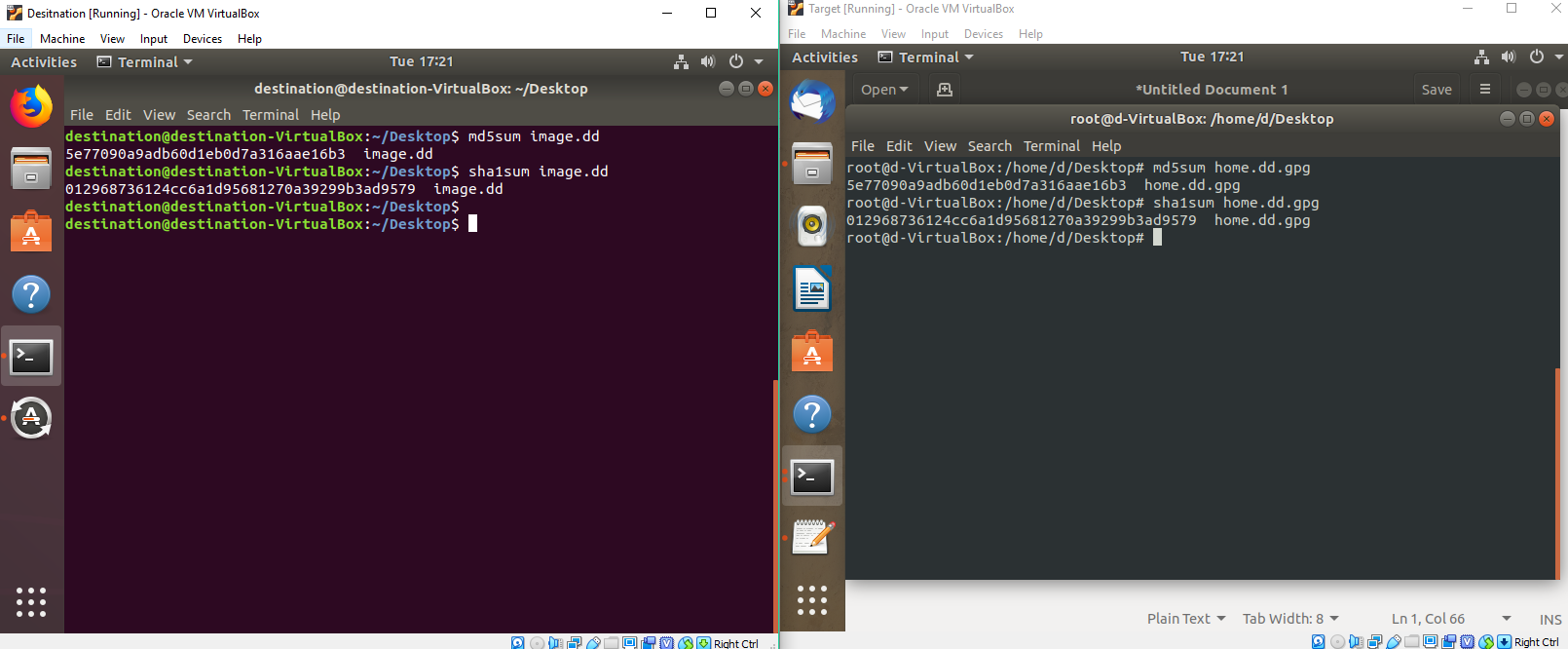
**Hash obtained image from the target machine**

****

****

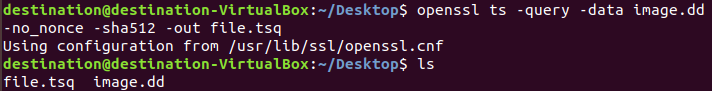
**Check if the both of hashes match**

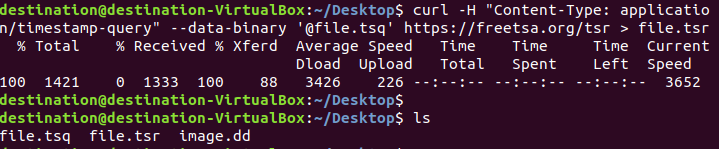
* They both have the same hashes

****

**Adding a trusted timestamp**

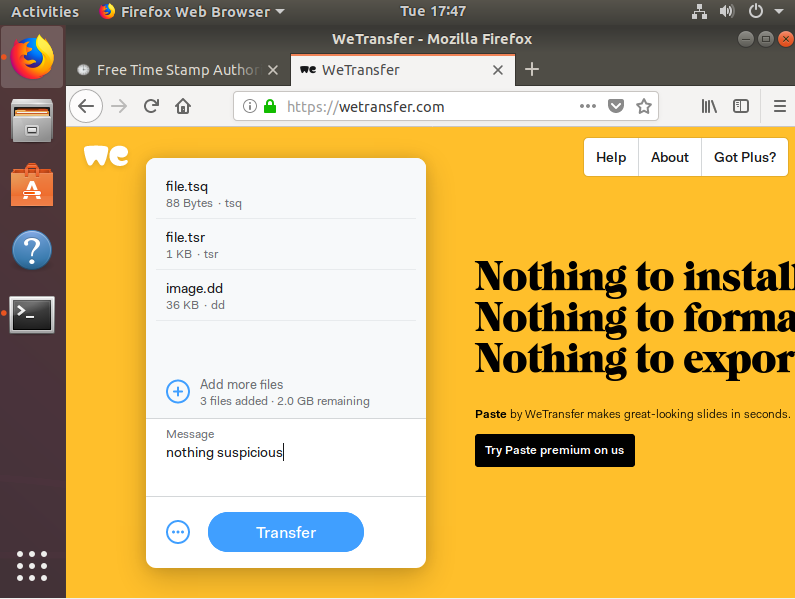
* openssl ts -query -data image.dd -no\_nonce -sha512 -out file.tsq
* curl -H "Content-Type: application/timestamp-query" --data-binary '@file.tsq' https://freetsa.org/tsr > file.tsr
* Used <https://freetsa.org/> to add a trusted timestamp

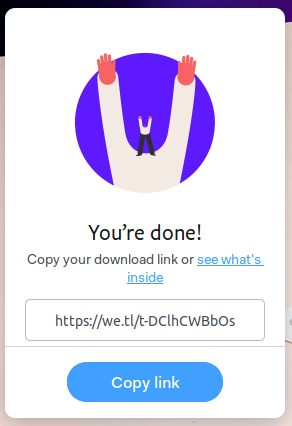




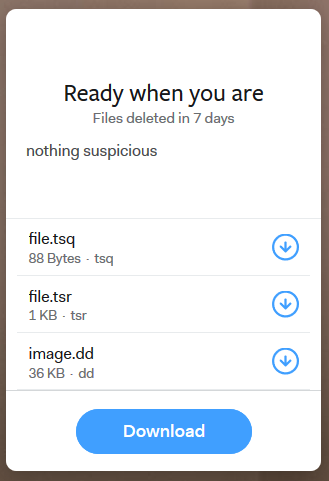
**Sending image over the internet to another computer in a different room**

* Used WeTransfer to send image file and timestamp files.
  + WeTransfer is a cloud-based computer file transfer service.
  + It generated a link a to download





**Downloading image that was sent over the internet on a different computer**





**Verifying download image hash**

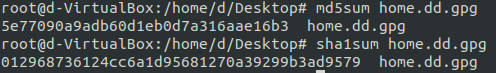
* The hash matches with the original hash

Downloaded Image



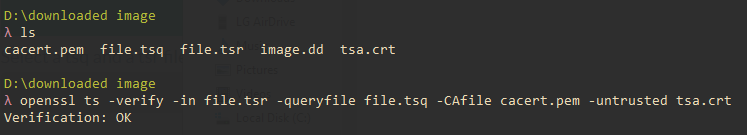


Original Image



**Verifying timestamp**

* Timestamp is verified
* Tsa.crt is Freetsa TSA Certificate which is downloaded from their website
* Cacert.pem is Freetsa CA Certificate which is downloaded from their website



**Conclusion:**

We setup two ubuntu machine one the target and the other destination. On the target machine we created an image of the home folder and then encrypted it. We generated the hash to check later and then sent it to destination machine using NCAT. On the destination machine we made sure the hashes matched and generated a timestamp. We then sent the image and the timestamp files to another computer in a different room. We again made sure the hashes matched which it did and made sure the time stamp was verified. We learned the importance of hashing files every time the chain of custody changed because it is easy to modify the files which will lead to a different hash. One problem we faced was how to defend the authenticity and integrity of files by using timestamp. We weren't sure how this process was done but eventually found a good online time stamping service.