CS6501: A1-3 Cryptography Cont.

Access Netlab using an Internet browser – https://netlab2.weltec.ac.nz.

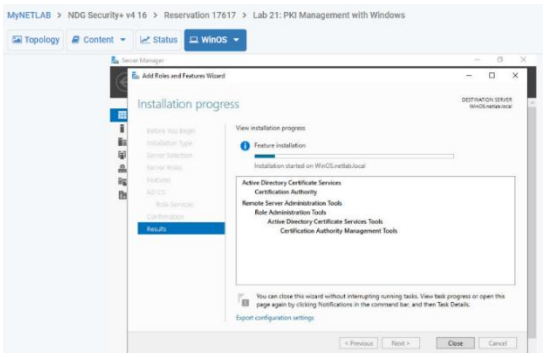
Part 1:

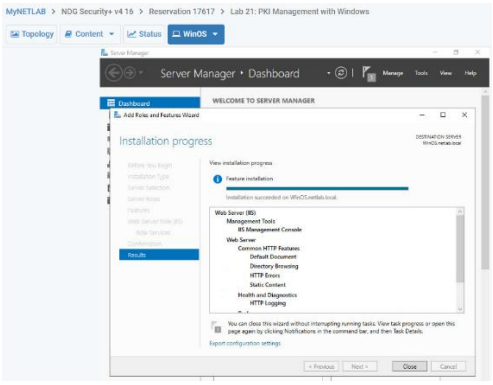
Please complete “PKI Management with Windows” from the NDG Security+ V4 series.

The deliverable for is a document that shows the occasional screenshot (which includes either your student ID and name or your Netlab username) and a **brief narrative explaining what you were doing and the resultant output**. This indicates to me that you have attempted the lab and I can confirm this by viewing the reservation logs on the system. The narrative explaining the screenshot indicates to me whether or not you understand what it is that is captured in the screenshot.

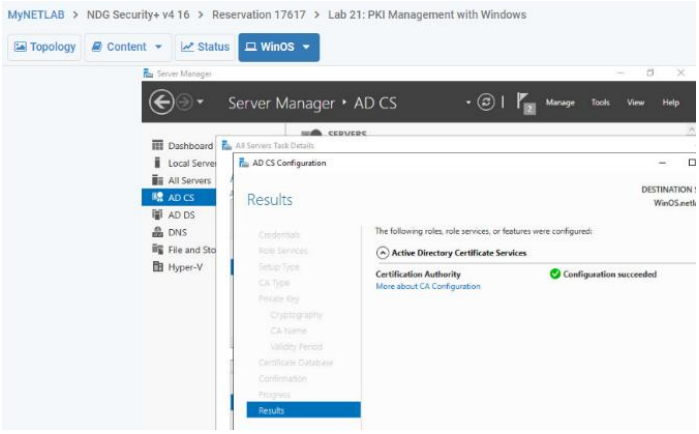
Please submit the document via Moodle Dropbox when you have completed the exercise.

We are to implement a public key infrastructure. First part was to add ADCS role to the window server (what we are operating from). We were able to achieve this by using the Server Manager application. This enabled us to add roles and features for the ADCS. We need to make sure Certification Authority was checked for the Role Services for ADCS. This now meant window server had ADCS role/Certification authority capabilities. AD CS Server installed.





We could now configure and customize the Certificate Authority on the AD CS server (what we previously set up). It appears as a server on sever manager window. We action configuration, checkbox for CA (Certification Authority), Standalone CA (members/group domain operate offline), Root CA (top of PKI hierarchy issue own self-signed certificate), Create new Private Key (CA) must have a private key, Cryptography, CA Name -default. Validity period – set to 30 days (CA expiration date). CA configuration succeeded. This means CA has been created for AD CS.



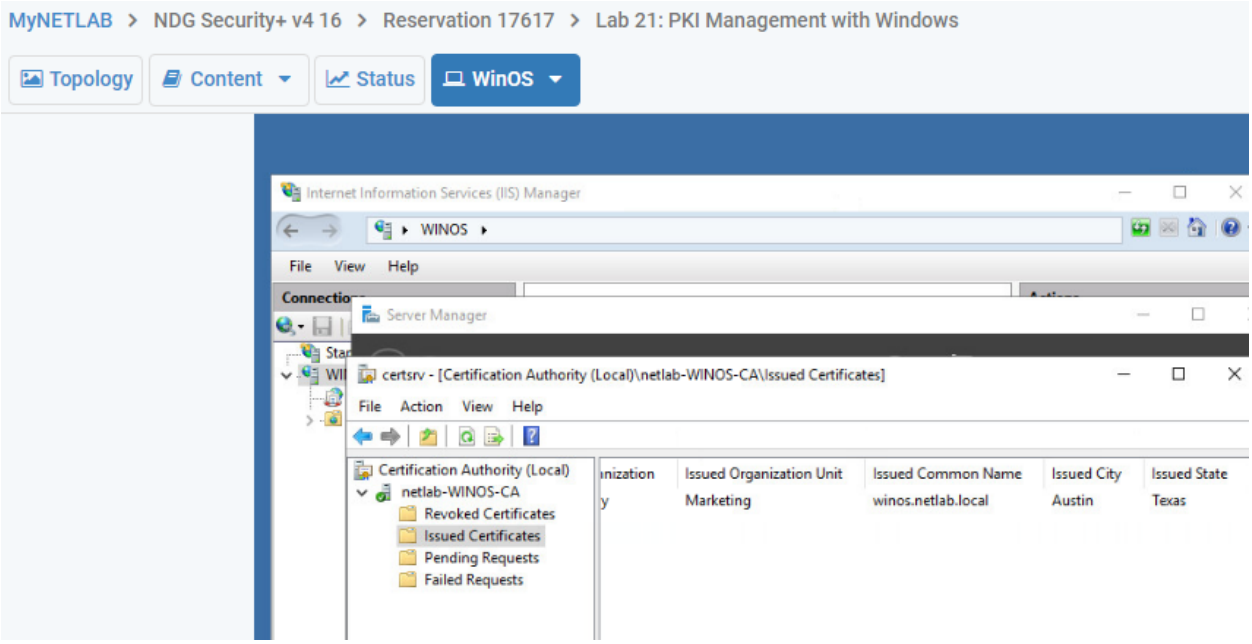
The Netab server will indicate certificate authority with a green tick. What this means is being a CA, you can create, sign and revoke digital certificates that bind keys to user identities.

Since we are now CA a Certificate Authority – we could issue a certificate.

Add Web Server (IIS – Internet Information Service) roles to our working server. Web server role lets you share information over the internet. We install this with configurations.

We can now see IIS in our desktop server. Under IIS Manager window – we can create Domain Certificate. We can create a certificate. Include server who will specify the online certification and the name. A certificate is now issues from WINOS server. This can be confirmed by checking the Server Manager window, tools and CA – issued certificates.

One was issued to XYZ Security in this case. A public key infrastructure has been commenced.



Part 2:

Please complete the "*File Integrity*" lab exercise from Labtainers Cyber Exercises. This is a slightly different environment. It uses an Ubuntu virtual machine in which docker containers run.

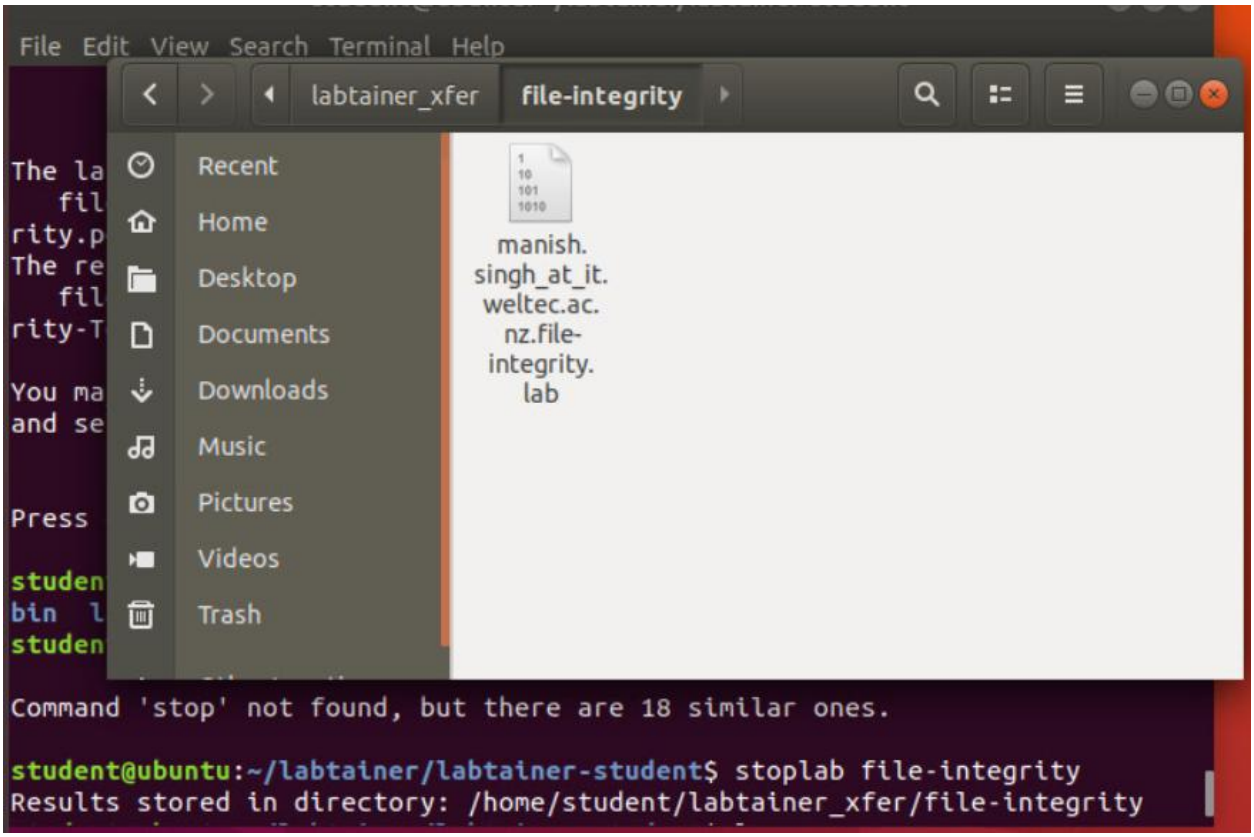
You will need to resize the desktop in the virtual machine. Go to Settings -> Devices -> Displays and change the resolution. I suggest 1280x960. You may need to drag the settings window so that you can click the Apply button.

Please be patient once you have started the lab exercise "labtainer file-integrity". It will be pulling down and starting docker containers. Enter your @weltec.ac.nz email address when prompted for an email address.

To start the lab, type *labtainer file-integrity*.

When you have completed the lab, type *stoplab file-integrity*

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. (See screenshot below)



The deliverable for Part 2 is the submission of the document in the labtainer_xfer/file-integrity folder (there will be shortcut to the folder labtainer_xfer stored in the Desktop) to Moodle. This Netlab virtual machine is connected to the Internet (you can email).

File Integrity

We learn how to use commands for system integrity. A way we can see files being modified, deleted or added. There are some additional tools that help us.

So we create a sha1sum tempfile. Sha1sum command can verify the integrity of the file by producing a sha1 160bit hash value. We calculate a new digest and receive the digest values.

We redirect the output to a file called hashes.txt

We modify the content and check the changes of the file

```
[Joe@file-integrity ~]$ su
Password:
[root@file-integrity Joe]# sha1sum tempfile
da39a3ee5e6b4b0d3255bfef95601890afd80709  tempfile
[root@file-integrity Joe]# sha1sum tempfile > hashes.txt
[root@file-integrity Joe]# sha1sum tempfile
da39a3ee5e6b4b0d3255bfef95601890afd80709  tempfile
[root@file-integrity Joe]# sha1sum --check hashes.txt
tempfile: OK
[root@file-integrity Joe]# sha1sum /usr/bin/* > hashes.txt
[root@file-integrity Joe]# sha1sum /usr/sbin/* >> hashes.txt
[root@file-integrity Joe]# sha1sum /etc/* >> hashes.txt
```

Sha1sum –check hashes.txt checks whether if the files has changed.
We calculate digests for many critical files by executing commands

To determine how many files were hashed: wc -l hashes.txt

Sha1sum –check hashes.txt
(if any failures the above command will display it, or if nay of files change the above command will display it)

This command will help if files have been modified or deleted.

We have a find command, that will display files and directories in a hierarchy. This helps to see if files have been added. Below we redirect files and save these.

```
[root@file-integrity Joe]# touch /usr/bin/dummyfile
[root@file-integrity Joe]#
[root@file-integrity Joe]# find /usr/bin -print > tempfile
[root@file-integrity Joe]# find /usr/sbin -print >> tempfile
[root@file-integrity Joe]# find /etc -print >> tempfile
[root@file-integrity Joe]# diff myfiles tempfile
```

We create a dummy file in the bin directory

Then issue the diff myfiles tempfile command to identify the add file which gives us the below result

```
[root@file-integrity Joe]# diff myfiles tempfile
447a448
> /usr/bin/dummyfile
```

We learn the AIDE command (Advanced Intrusion Detection Environment (AIDE) is a open source integrity product. Saves a lot more than it digests.

We drop caches to free up memory. A command is entered to tell OS to drop all caches.

```
[root@file-integrity Joe]# sysctl -w vm.drop_caches=2
vm.drop_caches = 2
[root@file-integrity Joe]# date ; aide --init ; date
Sun Mar 24 00:19:16 UTC 2024
```

We build the database entering the default AIDE configuration.

```
[root@file-integrity Joe]# date ; aide --init ; date
Sun Mar 24 00:12:57 UTC 2024
```

We change the AIDE configuration to enable two digests are saved for any file instead of one SHA256, We modify the config to enable two by adding sha512 and then clearing the caches again
Then rebuild the data bases using modified configuration (AIDE). It will share how long it took to build date base.

Multiple commands can be put on one command line and separating using a semi colon.

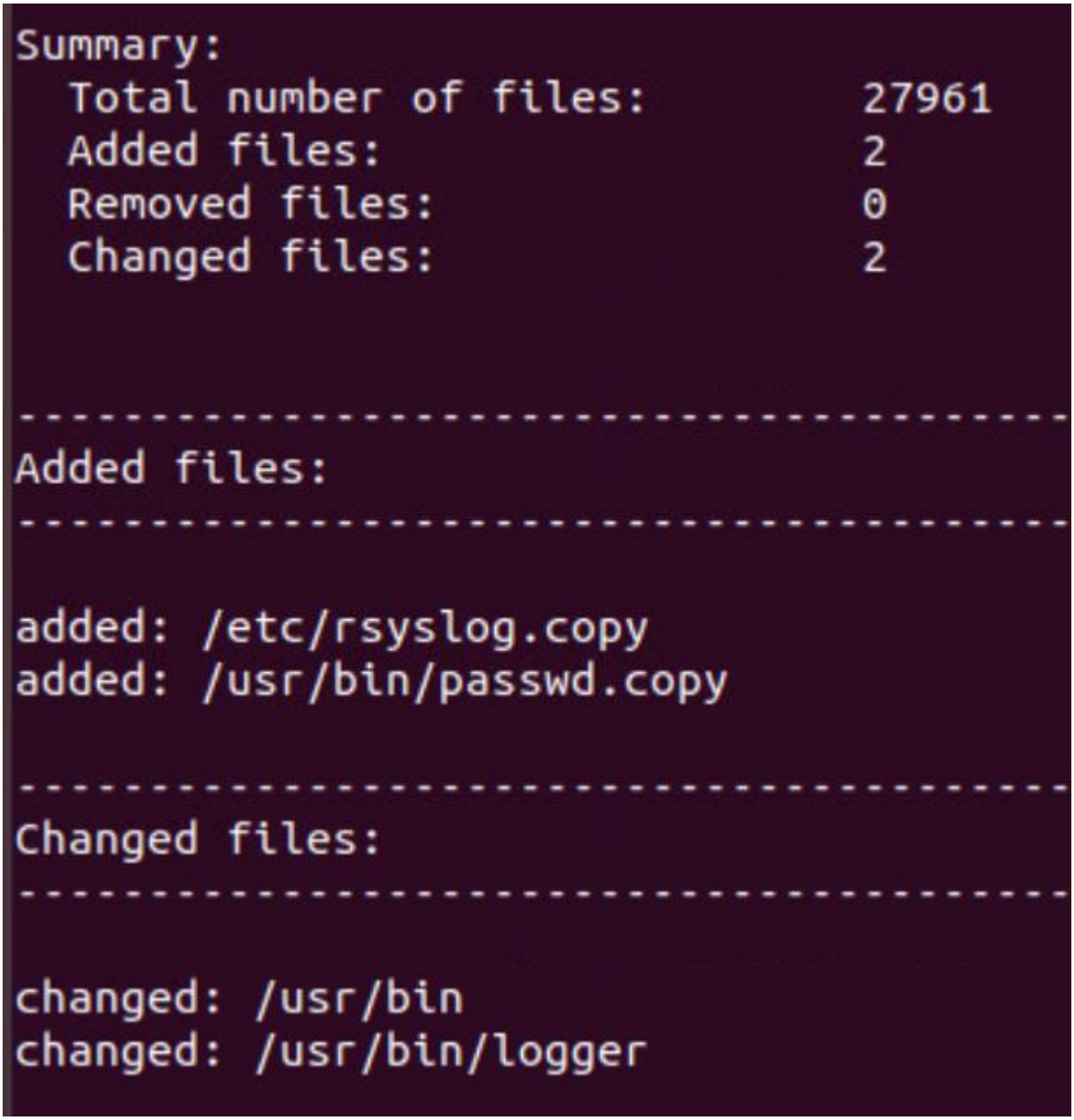
The shell will execute the first command, date command has completed it will excuse the aide command, and then the date command. We will see how long it takes the aide command to execute.

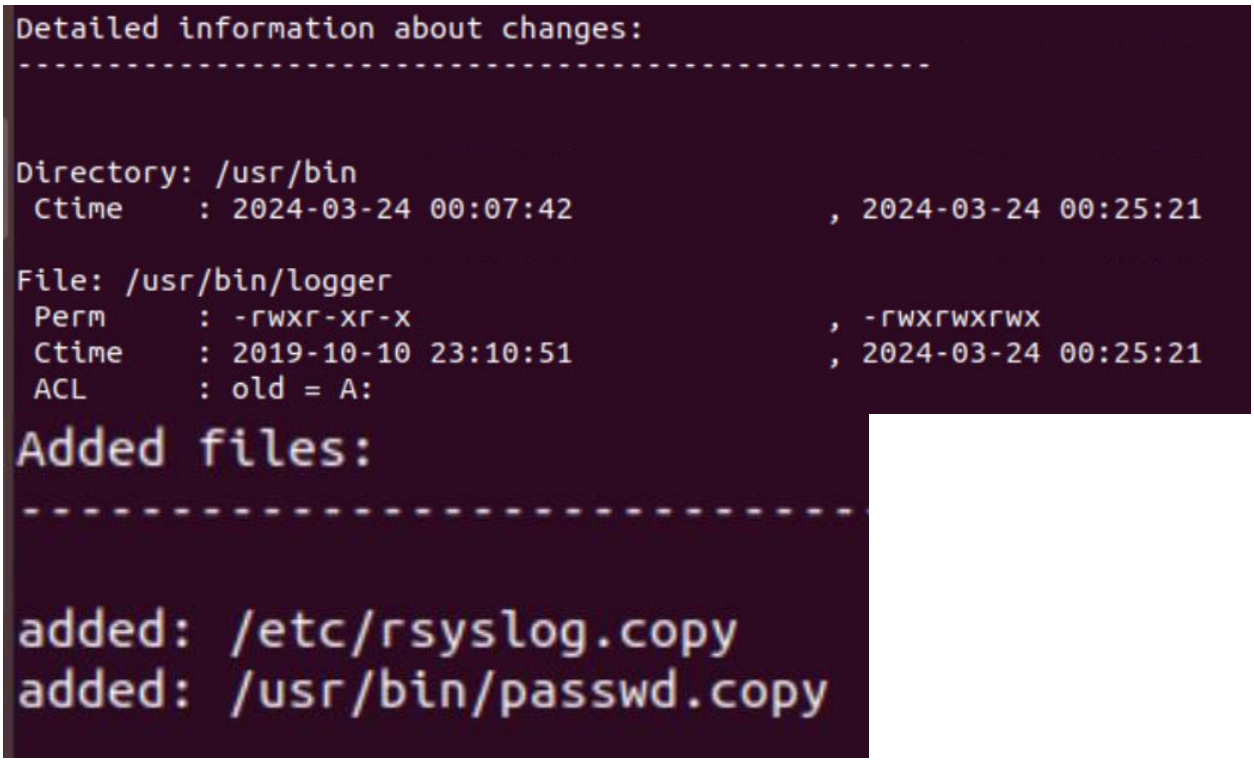
```
[root@file-integrity Joe]# cp /etc/rsyslog.conf /etc/rsyslog.copy
[root@file-integrity Joe]# echo "# another comment" >> /etc/rsyslog.conf
[root@file-integrity Joe]# cp /usr/bin/passwd /usr/bin/passwd.copy
[root@file-integrity Joe]# echo " " /usr/bin/passwd
  /usr/bin/passwd
[root@file-integrity Joe]# chmod ugo=rwx /bin/logger
```

```
[root@file-integrity Joe]# chmod ugo=rwx /bin/logger
[root@file-integrity Joe]# cd /var/lib/aide
[root@file-integrity aide]# mv aide.db.new.gz aide.db.gz
[root@file-integrity aide]# aiide --check
```

Other labtainer exercises that you could try that relate to the cryptography lecture are:

One-Way Hash Function and MAC
Exploring Symmetric Key Encryption Modes.




Detailed information about changes:
Directory: /usr/bin
Ctime : 2024-03-24 00:07:42 , 2024-03-24 00:25:21
File: /usr/bin/logger
Perm : -rwxr-xr-x , -rwxrwxrwx
Ctime : 2019-10-10 23:10:51 , 2024-03-24 00:25:21
ACL : old = A:
Added files:
added: /etc/rsyslog.copy
added: /usr/bin/passwd.copy