**Cryptography, TLS attacks**

**Task 1: Becoming a Certificate Authority (CA)**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  cp /usr/lib/ssl/openssl.cnf ./myCA\_openssl.cnf  mkdir demoCA  mkdir demoCA/certs demoCA/crl demoCA/newcerts demoCA/private  touch demoCA/index.txt  echo 1000 > demoCA/serial  openssl req -x509 -newkey rsa:4096 -keyout demoCA/private/ca.key -out demoCA/certs/ca.crt -days 3650 -sha256 -config myCA\_openssl.cnf -subj "/C=US/O=netsec Inc./CN=www.netsec.com" -passout pass:dees  **Screenshots:**      **Observations & Explanations:**  In this task, we become the certificate Authority (CA) by signing it ourselves (root certificate). To accompolish this we must first go copy default configuration file for openssl which basically is a file that can generate a certificate. After that we create the needed directories and files. We create a serial number file which is initialized with 1000 and an index.txt which will be our database.  Now the main part, with big openssl command we create a private key (using rsa) and create a self-signed certification. The big command basically means:   * Creates a **4096-bit RSA private key** and a **self-signed certificate for a CA** (**x509**), valid for 10 years, with **SHA-256 encryption**, using custom configurations and secured by a passphrase which is **dees**.   The last command basically displays the details for the newly generated certificate.  What part of the certificate indicates this is a CA’s certificate?    What part of the certificate indicates this is a self-signed certificate?  the subject is the same as the issuer    and we can also verify by checking if the subject key identifier is equal to the key identifier of the certificate    we can look at the decoded content of the RSA key by running the command:  openssl rsa -in ca.key -text -noout  this allows us to identify the elements e, d, n, p and q: |

**Task 2: Generating a Certificate Request for Your Web Server**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  openssl genpkey -algorithm RSA -out server.key -pkeyopt rsa\_keygen\_bits:2048  openssl req -newkey rsa:2048 -sha256 \  -keyout server.key -out server.csr \  -subj "/C=US/O=netsec Inc./CN=www.netsec.com" \  -passout pass:dees \  -addext "subjectAltName = DNS:www.netsec.com, DNS:www.netsecA.com, DNS:www.netsecB.com"  **Screenshots:**      **Observations & Explanations:**  In this task, create a certificate signing request (CSR) for a server. To do this, we need to create a RSA private key for the server. This key will be used make a new CSR. When you create this, you also have a Subject Alternate Name (SAN) which allows the certificate to be used for multiple domain name. We did not use the CA from task 1 in this task. |

**Task 3: Generating a Certificate for your server**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  openssl ca -config /home/seed/Desktop/netSec/A4/Labsetup/Labsetup/myCA\_openssl.cnf -policy policy\_anything -md sha256 -days 3650 -in **/home/seed/Desktop/netSec/A4/Labsetup/Labsetup/server.csr** -out **/home/seed/Desktop/netSec/A4/Labsetup/Labsetup/server.crt** -batch -cert **/home/seed/Desktop/netSec/A4/Labsetup/Labsetup/demoCA/certs/ca.crt** -keyfile **/home/seed/Desktop/netSec/A4/Labsetup/Labsetup/demoCA/private/ca.key**  **Screenshots:**        Proof of new signed cert in “index.txt”:    **Observations & Explanations:**  In Task 3, we signed the CSR generated in Task 2 with your CA from Task 1 to issue a server certificate. Sounded quite simple but we got into some issues. Some issues that we encountered in this task:   * We did not uncomment the two lines in the screenshot, but we got another error after uncommenting the 2 lines. * simply stating the name of the file was not enough and caused errors like “no such file in directory”, so we used the absolute paths to make it work which fixed the problem. |

**Task 4: Deploying Certificate in an Apache-Based HTTPS Website**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  openssl req -newkey rsa:2048 -sha256 \    -keyout server.key -out server.csr \    -subj "/CN=www.netsec.com/O=netsec Inc./C=US" \    -passout pass:dees \    -addext "subjectAltName = DNS:www.netsec.com, DNS:www.netsecA.com, DNS:www.netsecB.com"  cp /usr/lib/ssl/openssl.cnf /home/seed/Desktop/netsec/pki/pki/  mv openssl.cnf myCA\_openssl.cnf  a2enmod ssl  a2ensite netsec\_apache\_ssl  a2ensite example\_apache\_ssl  a2ensite examplea\_apache\_ssl  service apache2 restart  openssl req -newkey rsa:2048 -sha256 \    -keyout server.key -out server.csr \    -subj "/CN=www.examplea.com/O=examplea Inc./C=US" \    -passout pass:dees \    -addext "subjectAltName = DNS:[www.examplea.com](http://www.examplea.com)"  openssl req -newkey rsa:2048 -sha256 \    -keyout server.key -out server.csr \    -subj "/CN=www.amazon.com/O=amazon Inc./C=US" \    -passout pass:dees \    -addext "subjectAltName = DNS:[www.amazon.com](http://www.amazon.com)"  **Screenshots:**    <VirtualHost \*:443>      DocumentRoot /var/www/netsec      ServerName www.netsec.com      ServerAlias www.netsecA.com      ServerAlias www.netsecB.com      DirectoryIndex index.html      SSLEngine On      SSLCertificateFile /certs/server.crt      SSLCertificateKeyFile /certs/server.key  </VirtualHost>  <VirtualHost \*:80>      DocumentRoot /var/www/netsec      ServerName www.netsec.com      DirectoryIndex index\_red.html  </VirtualHost>  # Set the following gloal entry to suppress an annoying warning message  ServerName localhost      **Observations & Explanations:**  Make sure DNS entries for all the domains that'll be used  Start by making an apache server conf page for each to serve the specific domain in the container.  cd /etc/apache2/sites-available/  Make a separate file for each domain, for task 4 - use this for netsec\_apache\_ssl.conf  Next, we must import the server keys and sever certificate from task 3. We cp server.crt server.key [volumes] from the folder that contains the files. Then in the container we mv /volume/server\* /certs  You must also create a files folder for the apache to use. We can just rename the bank32 folder in /var/www by mv /var/www/bank32 /var/www/netsec   if the user tries to access http:[www.netsec.com](http://www.netsec.com), he will be successful but https:[www.netsec.com](http://www.netsec.com) won't work as we haven’t yet added the CA which signed the server cert to the authorized certs. Import ca.crt here. |

**Task 5: Launching a Man-In-The-Middle Attack**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  Index.html  <html>  <head>  <title>Hello welcome to example.com</title>  </head>  <body>  <h1>We are the real example.com, believe us totally!</h1>  </body>  </html>  Conf file for apache server  <VirtualHost \*:443>  DocumentRoot /var/www/example  ServerName www.example.com  DirectoryIndex index.html  SSLEngine On  SSLCertificateFile /certs/server.crt  SSLCertificateKeyFile /certs/server.key  </VirtualHost>  <VirtualHost \*:80>  DocumentRoot /var/www/example  ServerName www.example.com  DirectoryIndex index.html  </VirtualHost>  # Set the following global entry to suppress an annoying warning message  ServerName localhost  Refreshing code  a2enmod ssl  a2ensite netsec\_apache\_ssl  a2ensite example\_apache\_ssl  a2ensite examplea\_apache\_ssl  service apache2 restart  **Screenshots:**      **A screenshot of a computer  Description automatically generatedObservations & Explanations:**  In task 5, we pretend to be a domain with forged certs, while tasks 5 and 6 are supposed to be on the same domain to illustrate the difference. However, we used "example" and "exampleA.com" to differentiate, as it simplifies the tasks. ExampleA.com is simply a duplicate of Example.com, created specifically for task 6.  Start by making the index.html file your domain will lead to. The first code is for that. We created it in /var/www/example. You can edit this html file to anything, perhaps something more lookalike of an actual page. Next, make an apache server conf page for each to serve the specific domain in the container.  cd /etc/apache2/sites-available/  Make a separate file for each domain, example\_apache\_ssl.conf  This should be the conf file (we can  Now refresh the Apache and restart the server.  We notice this using http, we load the page properly but with https we are unable to do so. This is because https requires the correct cert, thus the cert created for netsec.com will not work for example.com and it raises a warning. |

**Task 6: Launching a Man-In-The-Middle Attack with a Compromised CA**

You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Please also list the important code snippets followed by explanation. Simply attaching code without any explanation will not receive credits. In addition, answer any questions if any.

|  |
| --- |
| **Code Snippet:**  Conf file updated  <VirtualHost \*:443>  DocumentRoot /var/www/example  ServerName www.examplea.com  DirectoryIndex index.html  SSLEngine On  SSLCertificateFile /certs/forged/server.crt  SSLCertificateKeyFile /certs/forged/server.key  </VirtualHost>  <VirtualHost \*:80>  DocumentRoot /var/www/example  ServerName www.examplea.com  DirectoryIndex index.html  </VirtualHost>  # Set the following gloal entry to suppress an annoying warning message  ServerName localhost  Refreshing code  a2enmod ssl  a2ensite netsec\_apache\_ssl  a2ensite example\_apache\_ssl  a2ensite examplea\_apache\_ssl  service apache2 restart  Illegitimate key generation  Key request:  openssl req -newkey rsa:2048 -sha256 \    -keyout server.key -out server.csr \    -subj "/CN=www.examplea.com/O=examplea Inc./C=US" \    -passout pass:dees \    -addext "subjectAltName = DNS:[www.examplea.com](http://www.examplea.com)"  Key signing  Use Task 3 code  **Screenshots:**      **Observations & Explanations:**  Here, instead of continuing with Example.com, I'll repeat the process for ExampleA.com. Firstly, assuming the keys for the forged website ExampleA.com are generated on the container, we must have a PKI folder. This PKI folder contains ca.key and ca.crt (in our case, obtained by sending through volumes). At this point, we have a configuration file for ExampleA.com. We just need to add the forged certificates to the directory as listed.  Using the command, self-request a certificate for "ExampleA.com" and then sign it with the stolen CA key. Then copy this into the appropriate directory, matching with the cert directory in the configuration file. After this, simply restart the server.  Now with https, we do not get a warning. This is because the certificate is seems legitimate as it is signed by the ca.key, which we obtained wrongly and generated by ourselves. |