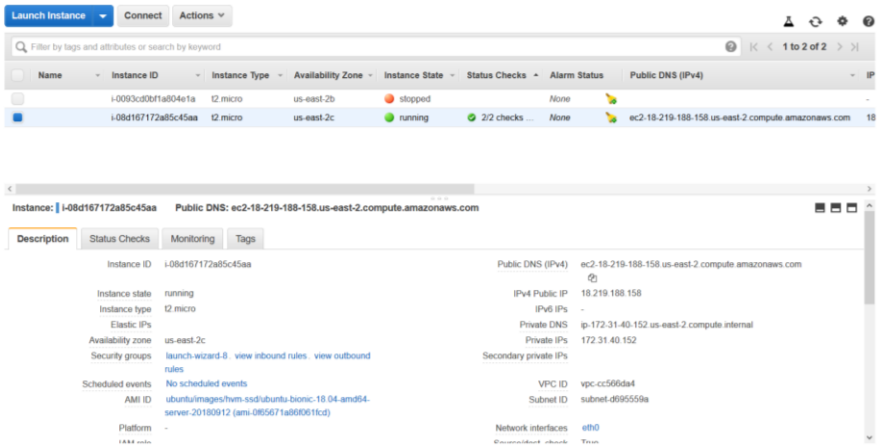
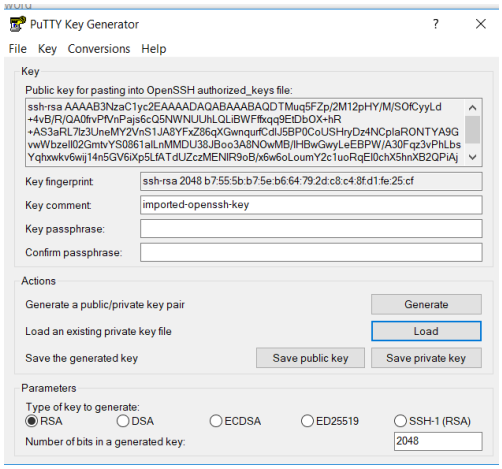
ITIS 6240
Parag Mhatre
Project 1
Oct 21st, 2018

I used a AWS instance of Ubuntu 16.04 LTS - 64-bit Linux server. Following are the steps that I undertook to perform the Project and successfully setup a Apache server with strong Cryptography.

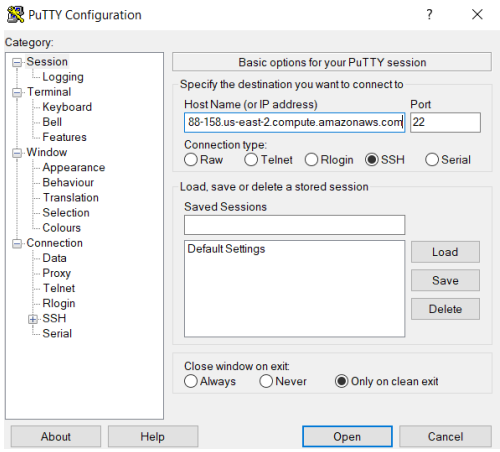
1. I spun up a Ubuntu 64-bit machine on AWS and opened up it's ports for HTTPS traffic.



2. I used PuTTYgen to convert the "keypair.pem" key that I got from AWS to Public and Private key files for the Server.

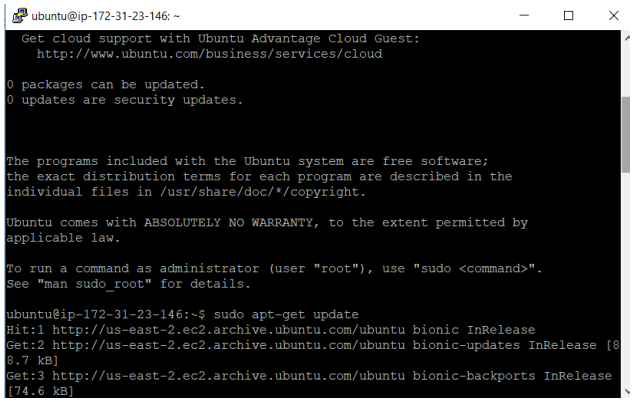


3. Using the Public Address of the Instance and ".pvk" key file, I connected to it using PuTTY.



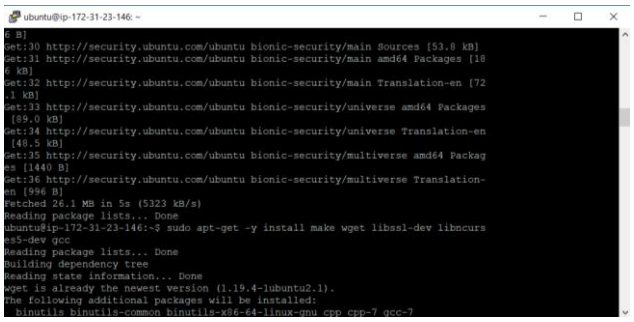
4. After Connecting to the Remote Machine, the first thing I did was, to update its application libraries. To do this, I ran the command, on the instance through SSL command line:

sudo apt-get update



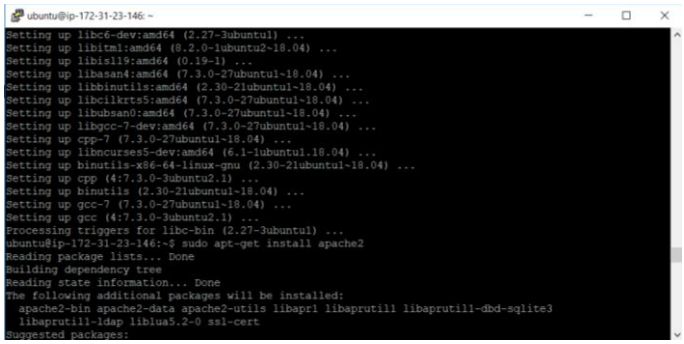
5. After updating the library, I installed other packages as given in the instructions.

Sudo apt-get -y install make wget libssl-dev libncurses5-dev gcc



6. Then I installed Apache2 on the server. To do this, I entered,

Sudo apt-get install apache2



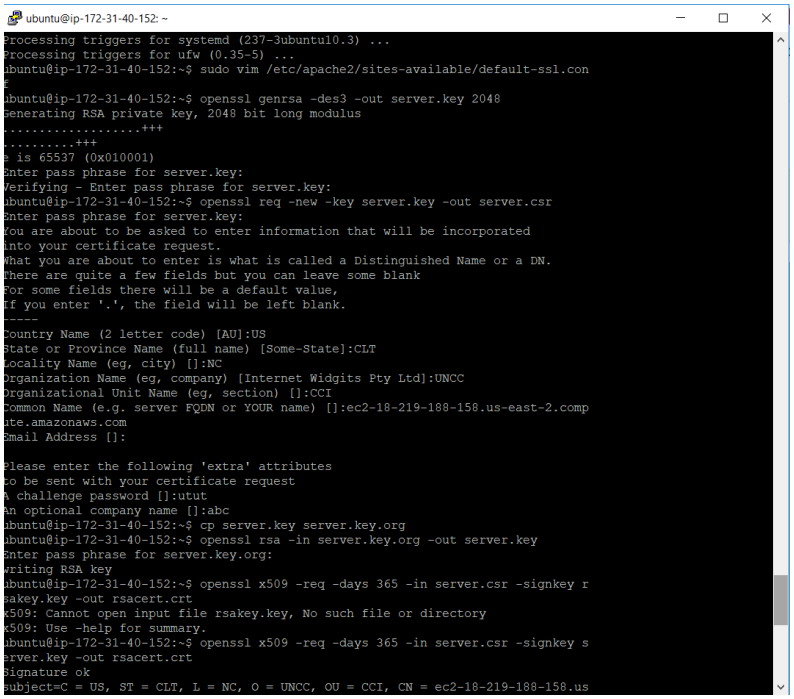
7. Now, to create and install the SSL certificate and key, I used the following commands:

Openssl genrsa -des3 -out server.key 2048

Openssl req -new -key server.key -out server.csr

Openssl rsa -in server.key.org -out server.key

Openssl x509 -req -days 365 -in server.csr -signkey rsakey.key -out rsacert.crt



8. After that, I ran some commands to enable the settings to load in the Apache service and restarted it with the modifications.

Sudo a2enmod ssl

Sudo a2enmod headers

Systemctl restart apache2

```
ubuntu@ip-172-31-40-152:~$ sudo service apache2 restart
ubuntu@ip-172-31-40-152:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
ubuntu@ip-172-31-40-152:~$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

9. Now, I modified the default-ssl.conf file located in /etc/apache2/sites-available/ and added some lines to the text.

Sudo vim /etc/apache2/sites-available/default-ssl.conf

ServerName ec2-18-219-188-158.us-east-2.compute.amazonaws.com

SSLCertificateFile rsacert.crt

SSLCertificateKeyFile rsakey.key

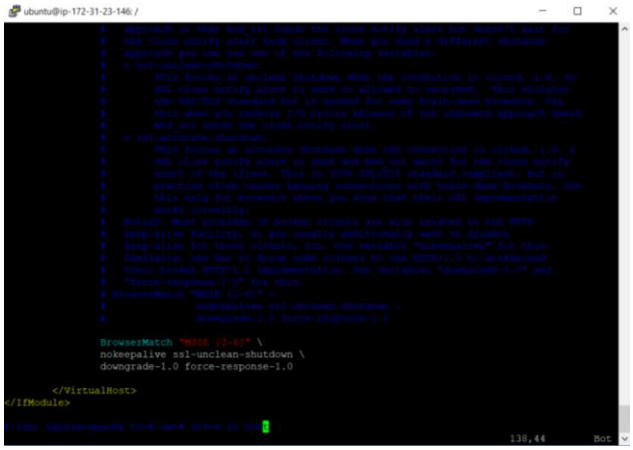
```
ubuntu@ip-172-31-40-152: ~
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost
                ServerName ec2-18-219-188-158.us-east-2.compute.amazonaws.com
                DocumentRoot /var/www/html

                # Available loglevels: trace8, ..., trace1, debug, info, notice,
 warn,
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For examp
le the
                # following line enables the CGI configuration for this host onl
y
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on

                #   A self-signed (snakeoil) certificate can be created by insta
lling
                #   the ssl-cert package. See
                #   /usr/share/doc/apache2/README.Debian.gz for more info.
                #   If both key and certificate are stored in the same file, onl
y the
                #   SSLCertificateFile directive is needed.
                SSLCertificateFile      rsacert.crt
                SSLCertificateKeyFile rsakey.key

                #   Server Certificate Chain:
                #   Point SSLCertificateChainFile at a file containing the
                #   concatenation of PEM encoded CA certificates which form the
                                                                  1,1           Top
```

Furthermore:

*BrowserMatch "SIE [2-6]" *

*Nokeepalive ssl-unclean-shutdown *

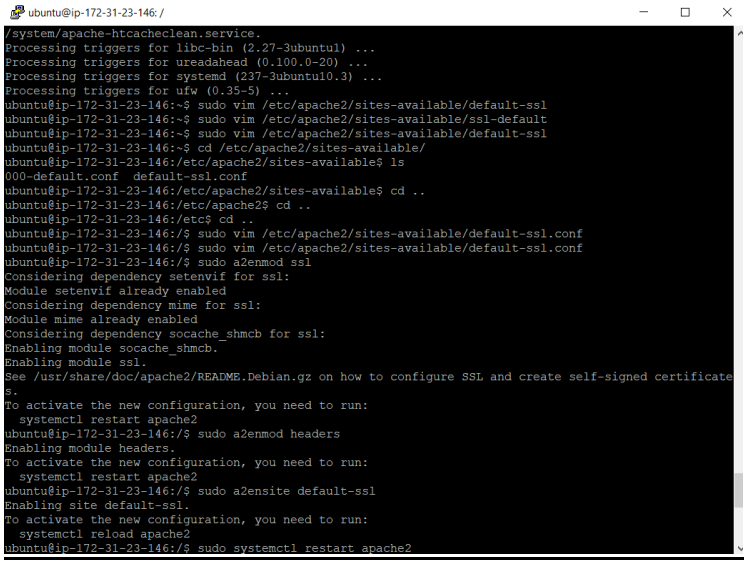
Downgrade-1.0 force-response-1.0



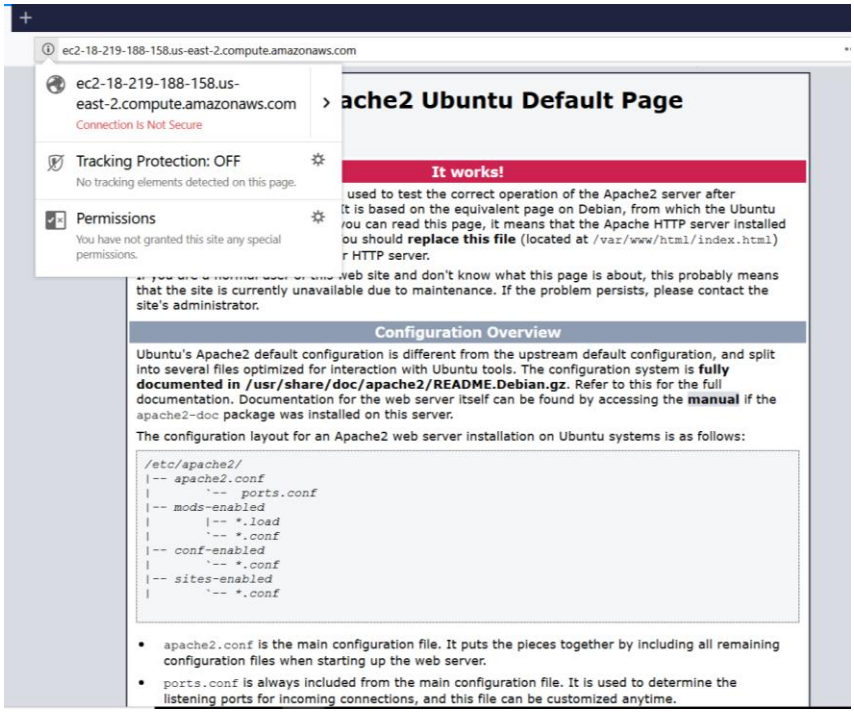
10. After modifying the "default-ssl.conf" file, I loaded it into the apache and restarted the service.

Sudo a2ensite default-ssl

Sudo systemctl restart apache2



11. Now after that process, I checked if the server was accessible through the browser. I opened it up using Mozilla Firefox

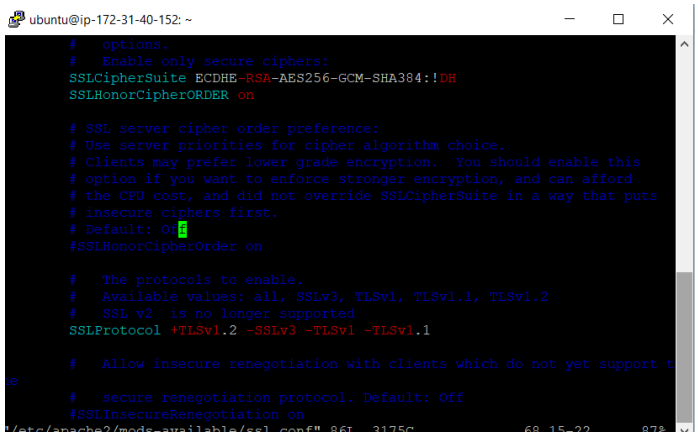


12. Now to change the SSL Configuration, I accessed the “ssl.conf” file from /etc/apache2/mods-available/ssl.conf and I added the following lines and restarted the “apache2” service.

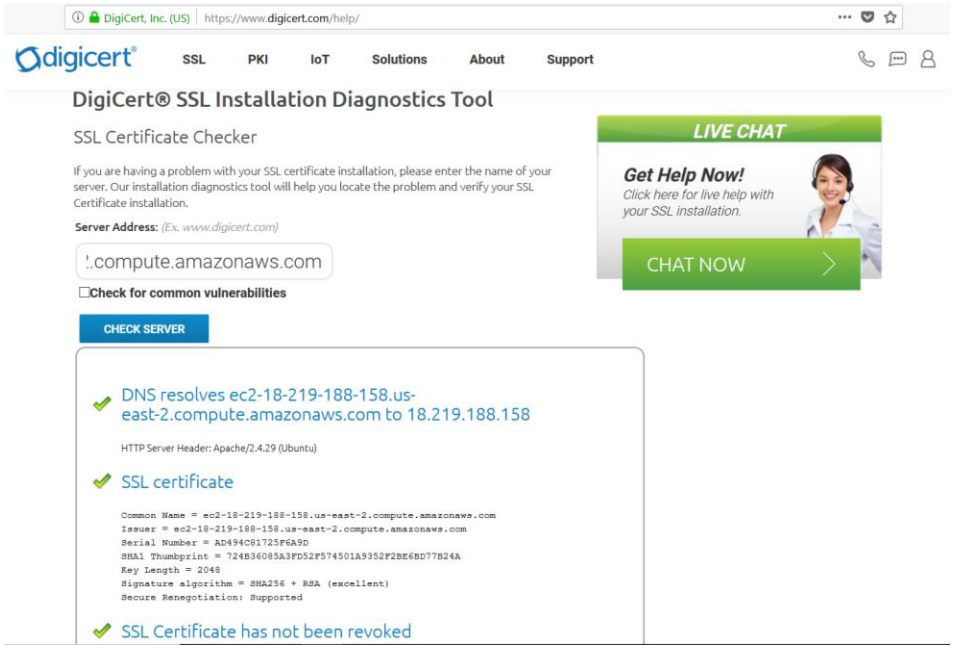
SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:!DH

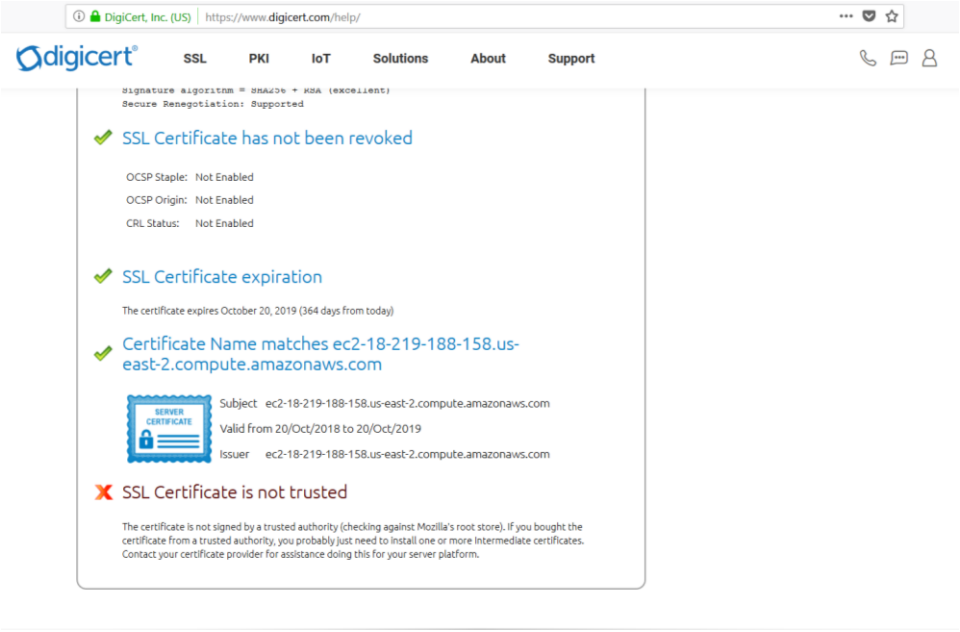
SSLHonorCipherORDER on

SSLProtocol +TLSv1.2 -TLSv1 -TLSv1.1 -SSLv3

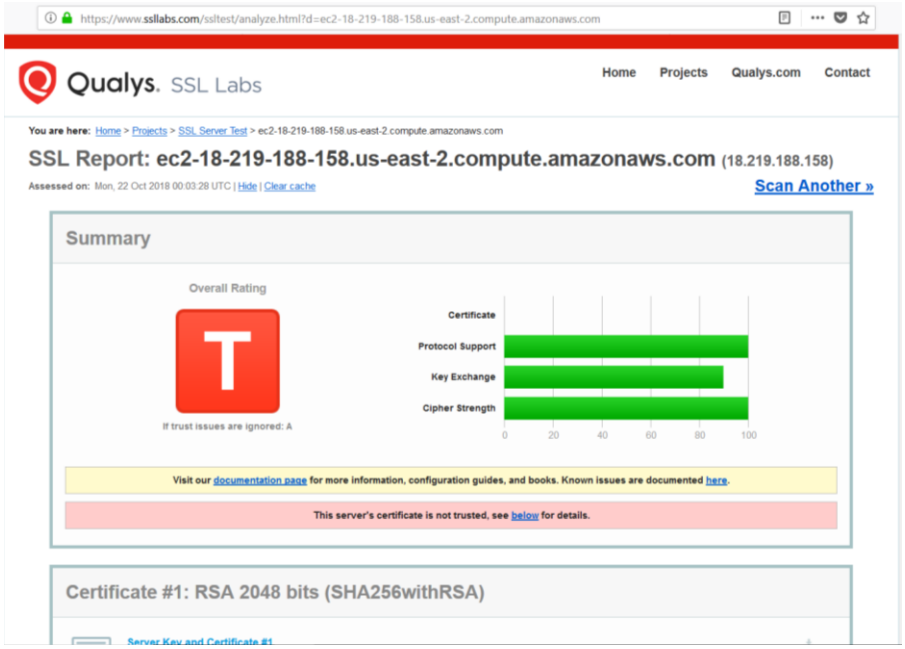


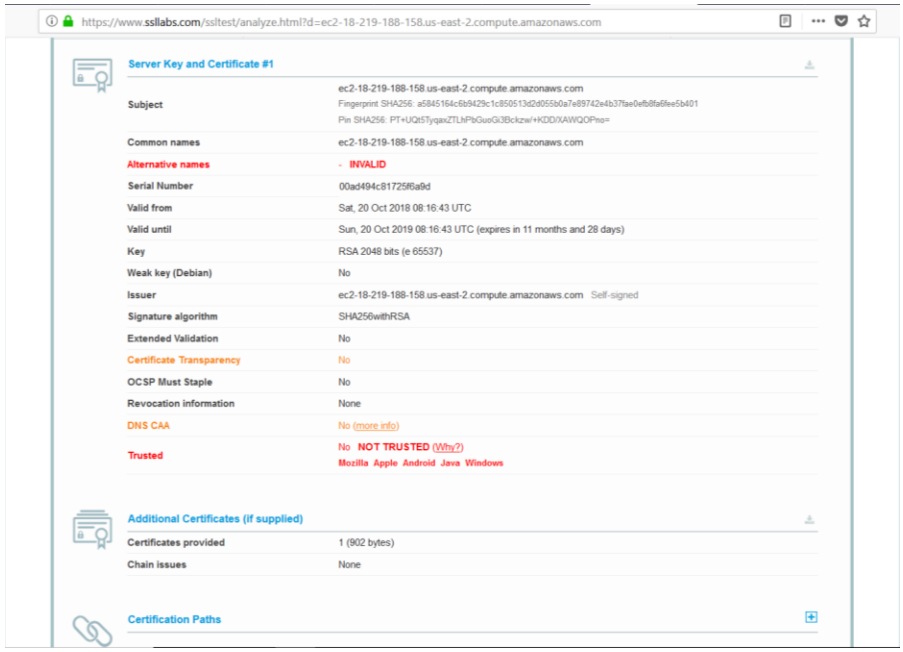
13. Now, I ran tests on DigiCert.com and the screenshots of the results are attached below:





14. Furthermore, I ran tests on SSLLabs.com and the results are given below(including the list of ciphers on the server):





https://www.ssllabs.com/ssltest/analyze.html?d=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

Certification Paths

Click here to expand

Configuration

Protocols

TLS 1.3	No
TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

For TLS 1.3 tests, we only support RFC 8446.

Cipher Suites

 # TLS 1.2 (we could not determine if the server has a preference)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH x25519 (eq. 3072 bits RSA) FS 256

Handshake Simulation

Android 4.4.2	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Android 5.0.0	Server sent fatal alert: handshake_failure			
Android 6.0	Server sent fatal alert: handshake_failure			
Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS

https://www.ssllabs.com/ssltest/analyze.html?d=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
BingPreview Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Chrome 49 / XP SP3	Server sent fatal alert: handshake_failure			
Chrome 69 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Chrome 70 / Win 10	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	Server sent fatal alert: handshake_failure			
Firefox 47 / Win 7 R	Server sent fatal alert: handshake_failure			
Firefox 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Googlebot Feb 2018	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
IE 11 / Win 7 R	Server sent fatal alert: handshake_failure			
IE 11 / Win 8.1 R	Server sent fatal alert: handshake_failure			
IE 11 / Win Phone 8.1 R	Server sent fatal alert: handshake_failure			
IE 11 / Win Phone 8.1 Update R	Server sent fatal alert: handshake_failure			
IE 11 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Java 8u161	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.0.1l R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
OpenSSL 1.0.2e R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 6 / iOS 6.0.1	Server sent fatal alert: handshake_failure			
Safari 7 / iOS 7.1 R	Server sent fatal alert: handshake_failure			
Safari 7 / OS X 10.9 R	Server sent fatal alert: handshake_failure			
Safari 8 / iOS 8.4 R	Server sent fatal alert: handshake_failure			
Safari 8 / OS X 10.10 R	Server sent fatal alert: handshake_failure			
Safari 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1 FS

https://www.ssllabs.com/ssltest/analyze.html?d=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Safari 10 / iOS 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Safari 10 / OS X 10.12 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Apple ATS 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
Yahoo Slurp Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS
YandexBot Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDH secp256r1	FS

 # Not simulated clients (Protocol mismatch)

Click here to expand

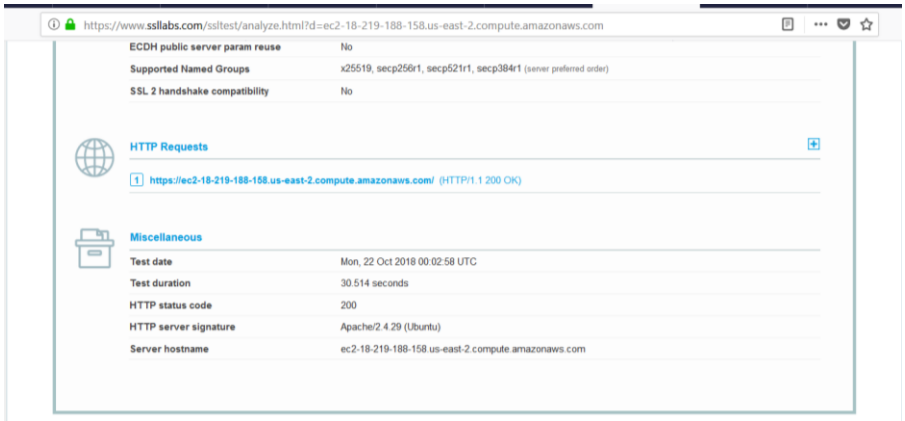
(1) Clients that do not support Forward Secrecy (FS) are excluded when determining support for it.
(2) No support for virtual SSL hosting (SNI). Connects to the default site if the server uses SNI.
(3) Only first connection attempt simulated. Browsers sometimes retry with a lower protocol version.
(R) Denotes a reference browser or client, with which we expect better effective security.
(All) We use defaults, but some platforms do not use their best protocols and features (e.g., Java 6 & 7, older IE).
(All) Certificate trust is not checked in handshake simulation, we only perform TLS handshake.

Protocol Details

DROWN	No, server keys and hostname not seen elsewhere with SSLv2 (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Downgrade attack prevention	Unknown (requires support for at least two protocols, excl. SSL2)
SSL/TLS compression	No

https://www.ssllabs.com/ssltest/analyze.html?d=ec2-18-219-188-158.us-east-2.compute.amazonaws.com

Downgrade attack prevention	Unknown (requires support for at least two protocols, excl. SSL2)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	Yes (with most browsers) ROBUST (more info)
ALPN	Yes http/1.1
NPN	No
Session resumption (caching)	Yes
Session resumption (tickets)	Yes
OCSP stapling	No
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	No (more info)
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No, DHE suites not supported
DH public server param (Ys) reuse	No, DHE suites not supported
ECDH public server param reuse	No
Supported Named Groups	x25519, secp256r1, secp521r1, secp384r1 (server preferred order)



15. Then I connected to the server using my SSL Command Line and the results of the operation are given below:

openssl s_client connect ec2-18-219-188-158.us-east-2.compute.amazonaws.com:443