SSL Vulnerabilities

Secure Sockets Layer (SSL) generally aims to host a secure communication protocol between the customer and the web page that we encounter on shopping sites and in order to protect credit card information, password entries and user data. The SSL certificate includes the following features:

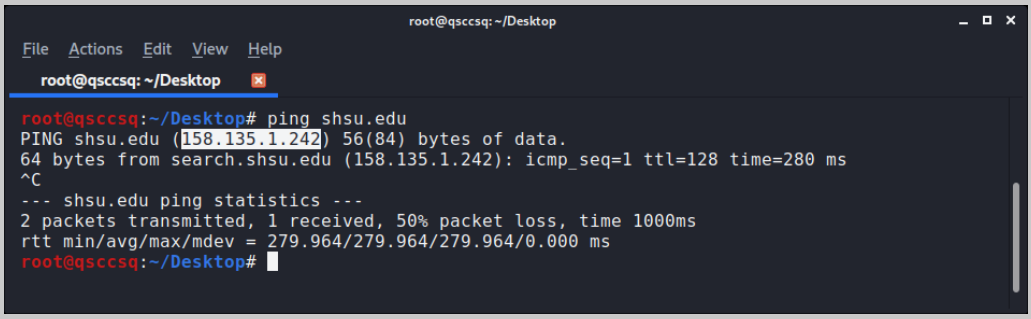
- Title of the certificate holding institution
- Certificate serial number and expiration date
- Certificate holder's public key
- Electronic Certificate Service Provider signature

SSL, which was first announced by Netscape in 1994, was naturally implemented by companies that provide paid services. However, thanks to the new method called OpenSSL, with the increase of open source initiatives, each user was able to generate his own SSL certificate. By 2014, it was determined that 66% of web pages that have SSL certificate on the internet use OpenSSL. However, on the same dates, a bug that appeared in OpenSSL version 1.0 greatly affected the security of web pages. The vulnerability called HeartBleed leaked 64-bit encrypted data. As a result of the man-in-the-Middle attack, it was possible to access customer information on any web page. Taking advantage of SSL weaknesses is still among the methods preferred by hackers today. The document will show you how to perform SSL scans of the target site by a hacker. It should be noted that an SSL attack method called CRIME at the Black Hat conference held in 2013 still threatens millions of web pages in 2020. However, the method is kept confidential by engineers who carried out the CRIME attack.

Conference: https://www.youtube.com/watch?v=e3hOJfrSD9g

Detecting the HeartBleed Vulnerability

Step 1: Determine the target's IP address;



Step 2: Use NMAP to detecting HeartBleed Vulnerable on the target;

It could take a while. You can use TAB key to check percentage of processes.

```
root@qsccsq:~# nmap -d --script ssl-heartbleed --script-args vulns.showall -sV 158.135.1.242
```

```
Nmap scan report for search.shsu.edu (158.135.1.242)
Host is up, received reset ttl 128 (0.021s latency).
Scanned at 2020-07-02 17:51:02 EDT for 242s
Not shown: 998 filtered ports
Reason: 998 no-responses
PORT     STATE SERVICE      REASON         VERSION
80/tcp   open  http-proxy   syn-ack ttl 128 F5 BIG-IP load balancer http proxy
| http-server-header: BigIP
443/tcp open  ssl/https? syn-ack ttl 128
| ssl-heartbleed:
|   NOT VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for stealing infor
mation intended to be protected by SSL/TLS encryption.
|     State: NOT VULNERABLE
|     References:
|       http://www.openssl.org/news/secadv_20140407.txt
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
|_      http://cvedetails.com/cve/2014-0160/
Service Info: Device: load balancer
Final times for host: srtt: 21079 rttvar: 39650  to: 179679

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 17:55
Completed NSE at 17:55, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 17:55
Completed NSE at 17:55, 0.00s elapsed
Read from /usr/bin/../share/nmap: nmap-payloads nmap-service-probes nmap-services.
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 241.51 seconds
           Raw packets sent: 3042 (133.664KB) | Rcvd: 152 (6.088KB)
root@qsccsq:~#
```

Detecting Encryption Issues of Target's SSL Certificate

Step 1: Installation of SSLYZE

```
root@qsccsq:~# apt-get install sslyze
Reading package lists... Done
Building dependency tree
Reading state information... Done
sslyze is already the newest version (3.0.7-0kali1).
0 upgraded, 0 newly installed, 0 to remove and 662 not upgraded.
root@qsccsq:~#
```

Step 2: Use SSLYZE to detecting Encryption Issues on the target;

```
root@qsccsq: ~
File  Actions  Edit  View  Help
root@qsccsq: ~

root@qsccsq:~# sslyze --regular shsu.edu

 CHECKING HOST(S) AVAILABILITY
 -----------------------------

   shsu.edu:443                       => 158.135.1.242
```

```
root@qsccsq: ~
File  Actions  Edit  View  Help
root@qsccsq: ~

 SCAN RESULTS FOR SHSU.EDU:443 - 158.135.1.242
 ---------------------------------------------

 * Deflate Compression:
                                        OK - Compression disabled

 * Session Renegotiation:
      Client-initiated Renegotiation:   VULNERABLE - Server honors client-initiated renegotiations
      Secure Renegotiation:             OK - Supported

 * OpenSSL Heartbleed:
                                        OK - Not vulnerable to Heartbleed

 * Certificates Information:
      Hostname sent for SNI:            shsu.edu
      Number of certificates detected:  1


    Certificate #0 ( _RSAPublicKey )
      SHA1 Fingerprint:                 872a4c821216195c461212f9a01009a2bbfb273f
      Common Name:                      *.shsu.edu
      Issuer:                           GlobalSign Organization Validation CA - SHA256 - G2
      Serial Number:                    278543173050067057682589052 64
      Not Before:                       2018-07-06
      Not After:                        2020-08-17
      Public Key Algorithm:             _RSAPublicKey
      Signature Algorithm:              sha256
      Key Size:                         2048
      Exponent:                         65537
      DNS Subject Alternative Names:    ['*.shsu.edu', 'shsu.edu']
```

```
root@qsccsq: ~
File  Actions  Edit  View  Help
root@qsccsq: ~

    Certificate #0 - Trust
      Hostname Validation:              OK - Certificate matches server hostname
      Android CA Store (9.0.0_r9):      OK - Certificate is trusted
      Apple CA Store (iOS 13, iPadOS 13, macOS 10.15, watchOS 6, and tvOS 13):OK - Certificate is trusted
      Java CA Store (jdk-13.0.2):       OK - Certificate is trusted
      Mozilla CA Store (2019-11-28):    OK - Certificate is trusted
      Windows CA Store (2020-05-04):    OK - Certificate is trusted
      Symantec 2018 Deprecation:        OK - Not a Symantec-issued certificate
      Received Chain:                   *.shsu.edu --> GlobalSign Organization Validation CA - SHA256 - G2
      Verified Chain:                   *.shsu.edu --> GlobalSign Organization Validation CA - SHA256 - G2 --> GlobalSign Root CA
      Received Chain Contains Anchor:   OK - Anchor certificate not sent
      Received Chain Order:             OK - Order is valid
      Verified Chain contains SHA1:     OK - No SHA1-signed certificate in the verified certificate chain

    Certificate #0 - Extensions
      OCSP Must-Staple:                 NOT SUPPORTED - Extension not found
      Certificate Transparency:         OK - 3 SCTs included

    Certificate #0 - OCSP Stapling
                                        NOT SUPPORTED - Server did not send back an OCSP response

 * SSL 2.0 Cipher suites:
    Attempted to connect using 7 cipher suites; the server rejected all cipher suites.

 * OpenSSL CCS Injection:
                                        OK - Not vulnerable to OpenSSL CCS injection

 * TLS 1.0 Cipher suites:
    Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * TLS 1.3 Cipher suites:
    Attempted to connect using 5 cipher suites; the server rejected all cipher suites.
```

```
root@qsccsq: ~
File  Actions  Edit  View  Help
root@qsccsq: ~

 * ROBOT Attack:
                                          OK - Not vulnerable, RSA cipher suites not supported.

 * TLS 1.2 Session Resumption Support:
      With Session IDs: OK - Supported (5 successful resumptions out of 5 attempts).
      With TLS Tickets: NOT SUPPORTED - Server did not return a TLS ticket.

 * Downgrade Attacks:
       TLS_FALLBACK_SCSV:                 OK - Supported

 * TLS 1.1 Cipher suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * SSL 3.0 Cipher suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * TLS 1.2 Cipher suites:
     Attempted to connect using 158 cipher suites.

     The server accepted the following 14 cipher suites:
        TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384            256       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384            256       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA               256       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256            128       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256            128       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA               128       ECDH: prime256v1 (256 bits)
        TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA            256       DH (1024 bits)
        TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA            128       DH (1024 bits)
        TLS_DHE_RSA_WITH_AES_256_GCM_SHA384              256       DH (1024 bits)
        TLS_DHE_RSA_WITH_AES_256_CBC_SHA256              256       DH (1024 bits)
        TLS_DHE_RSA_WITH_AES_256_CBC_SHA                 256       DH (1024 bits)
        TLS_DHE_RSA_WITH_AES_128_GCM_SHA256              128       DH (1024 bits)
```

Detecting the Common SSL Vulnerability with TestSSL

Step 1: Installation of TestSSL

```
root@qsccsq: ~/Desktop
File  Actions  Edit  View  Help
root@qsccsq: ~/Desktop
root@qsccsq:~/Desktop# git clone --depth 1 https://github.com/drwetter/testssl.sh.git
Cloning into 'testssl.sh'...
remote: Enumerating objects: 84, done.
remote: Counting objects: 100% (84/84), done.
remote: Compressing objects: 100% (81/81), done.
remote: Total 84 (delta 12), reused 17 (delta 2), pack-reused 0
Receiving objects: 100% (84/84), 8.51 MiB | 516.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
root@qsccsq:~/Desktop#
```

Step 2: Use TestSSL to detecting Common SSL Issues on the target;

```
root@qsccsq: ~/Desktop/testssl.sh
File  Actions  Edit  View  Help
root@qsccsq: ~...top/testssl.sh
root@qsccsq:~/Desktop# cd testssl.sh/
root@qsccsq:~/Desktop/testssl.sh# ./testssl.sh shsu.edu
```

```
root@qsccsq: ~/Desktop/testssl.sh
File  Actions  Edit  View  Help
root@qsccsq: ~...top/testssl.sh

root@qsccsq:~/Desktop# cd testssl.sh/
root@qsccsq:~/Desktop/testssl.sh# ./testssl.sh shsu.edu

###########################################################
    testssl.sh       3.1dev from https://testssl.sh/dev/
    (9122ffe 2020-06-26 10:02:23 -- )

      This program is free software. Distribution and
             modification under GPLv2 permitted.
      USAGE w/o ANY WARRANTY. USE IT AT YOUR OWN RISK!

       Please file bugs @ https://testssl.sh/bugs/

###########################################################

 Using "OpenSSL 1.0.2-chacha (1.0.2k-dev)" [~179 ciphers]
 on qsccsq:./bin/openssl.Linux.x86_64
 (built: "Jan 18 17:12:17 2019", platform: "linux-x86_64")


 Start 2020-07-06 21:52:00        -->> 158.135.1.242:443 (shsu.edu) <<--

 Further IP addresses:   2620:7e:c080::1f2
 rDNS (158.135.1.242):   mydegree.shsu.edu. massemail.shsu.edu. irm.shsu.edu. shsuphysicians.com. search.shsu.edu.
                         thetexasreview.org. irb.shsu.edu. www.shsu.edu.
 Service detected:       Couldn't determine what's running on port 443, assuming no HTTP service => skipping all HTTP checks


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3
```

```
root@qsccsq: ~/Desktop/testssl.sh
File  Actions  Edit  View  Help
root@qsccsq: ~...top/testssl.sh

 ALPN/HTTP2 not offered

 Testing cipher categories

 NULL ciphers (no encryption)                      not offered (OK)
 Anonymous NULL Ciphers (no authentication)        not offered (OK)
 Export ciphers (w/o ADH+NULL)                     not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)      not offered (OK)
 Triple DES Ciphers / IDEA                         not offered
 Obsoleted CBC ciphers (AES, ARIA etc.)            offered
 Strong encryption (AEAD ciphers) with no FS       not offered
 Forward Secrecy strong encryption (AEAD ciphers)  offered (OK)


 Testing server's cipher preferences

 Has server cipher order?     yes (OK)
 Negotiated protocol          TLSv1.2
 Negotiated cipher            ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Cipher per protocol

Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits     Cipher Suite Name (IANA/RFC)
-----------------------------------------------------------------------------------------------------------
SSLv2
 -
SSLv3
 -
TLSv1
 -
TLSv1.1
 -
TLSv1.2 (server order)
```

root@qsccsq: ~/Desktop/testssl.sh

File Actions Edit View Help

root@qsccsq: ~...top/testssl.sh

```
TLSv1.1
 -
TLSv1.2 (server order)
 xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 256   AESGCM      128      TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
 xc013   ECDHE-RSA-AES128-SHA              ECDH 256   AES         128      TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
 xc027   ECDHE-RSA-AES128-SHA256           ECDH 256   AES         128      TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
 xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 256   AESGCM      256      TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
 xc014   ECDHE-RSA-AES256-SHA              ECDH 256   AES         256      TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
 xc028   ECDHE-RSA-AES256-SHA384           ECDH 256   AES         256      TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
 x9e     DHE-RSA-AES128-GCM-SHA256         DH 1024    AESGCM      128      TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
 x33     DHE-RSA-AES128-SHA                DH 1024    AES         128      TLS_DHE_RSA_WITH_AES_128_CBC_SHA
 x67     DHE-RSA-AES128-SHA256             DH 1024    AES         128      TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
 x9f     DHE-RSA-AES256-GCM-SHA384         DH 1024    AESGCM      256      TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
 x39     DHE-RSA-AES256-SHA                DH 1024    AES         256      TLS_DHE_RSA_WITH_AES_256_CBC_SHA
 x6b     DHE-RSA-AES256-SHA256             DH 1024    AES         256      TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
 x45     DHE-RSA-CAMELLIA128-SHA           DH 1024    Camellia    128      TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
 x88     DHE-RSA-CAMELLIA256-SHA           DH 1024    Camellia    256      TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
TLSv1.3
 -


 Testing robust forward secrecy (FS) -- omitting Null Authentication/Encryption, 3DES, RC4

 FS is offered (OK)          ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA384 ECDHE-RSA-AES256-SHA DHE-RSA-AES256-GCM-SHA384
                             DHE-RSA-AES256-SHA256 DHE-RSA-AES256-SHA DHE-RSA-CAMELLIA256-SHA ECDHE-RSA-AES128-GCM-SHA256
                             ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA DHE-RSA-AES128-GCM-SHA256 DHE-RSA-AES128-SHA256
                             DHE-RSA-AES128-SHA DHE-RSA-CAMELLIA128-SHA
 Elliptic curves offered:    prime256v1 secp384r1 X25519
 DH group offered:           Unknown DH group (1024 bits)

 Testing server defaults (Server Hello)

 TLS extensions (standard)   "renegotiation info/#65281" "EC point formats/#11" "extended master secret/#23"
```

root@qsccsq: ~/Desktop/testssl.sh

File Actions Edit View Help

root@qsccsq: ~...top/testssl.sh

```
 Testing server defaults (Server Hello)

 TLS extensions (standard)   "renegotiation info/#65281" "EC point formats/#11" "extended master secret/#23"
 Session Ticket RFC 5077 hint no -- no lifetime advertised
 SSL Session ID support      yes
 Session Resumption          Tickets no, ID: yes
 TLS clock skew              Random values, no fingerprinting possible
 Signature Algorithm         SHA256 with RSA
 Server key size             RSA 2048 bits (exponent is 65537)
 Server key usage            Digital Signature, Key Encipherment
 Server extended key usage   TLS Web Server Authentication, TLS Web Client Authentication
 Serial / Fingerprints       5A008D1EC26AD40EFF48A4B0 / SHA1 872A4C821216195C461212F9A01009A2BBFB273F
                             SHA256 4B7C5AFCE414759089D507E46D6D1E80F02E7A7B60B2B559C56CA255E6079CC9
 Common Name (CN)            *.shsu.edu
 subjectAltName (SAN)        *.shsu.edu shsu.edu
 Issuer                      GlobalSign Organization Validation CA - SHA256 - G2 (GlobalSign nv-sa from BE)
 Trust (hostname)            Ok via SAN (same w/o SNI)
 Chain of trust              Ok
 EV cert (experimental)      no
 ETS/"eTLS", visibility info not present
 Certificate Validity (UTC)  expires < 60 days (41) (2018-07-06 10:01 --> 2020-08-17 16:47)
 # of certificates provided  2
 Certificate Revocation List http://crl.globalsign.com/gs/gsorganizationvalsha2g2.crl
 OCSP URI                    http://ocsp2.globalsign.com/gsorganizationvalsha2g2
 OCSP stapling               not offered
 OCSP must staple extension  --
 DNS CAA RR (experimental)   not offered
 Certificate Transparency    yes (certificate extension)


 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)          not vulnerable (OK), no heartbeat extension
```

root@qsccsq: ~/Desktop/testssl.sh

File Actions Edit View Help

root@qsccsq: ~...top/testssl.sh

```
 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  --   (applicable only for HTTPS)
 ROBOT                                     Server does not support any cipher suites that use RSA key transport
 Secure Renegotiation (RFC 5746)           supported (OK)
 Secure Client-Initiated Renegotiation     VULNERABLE (NOT ok), potential DoS threat
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK) (not using HTTP anyway)
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK), no SSLv3 support
 TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible (OK), no protocol below TLS 1.2 offered
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate elsewhere with SSLv2 enabled services
                                           https://censys.io/ipv4?q=4B7C5AFCE414759089D507E46D6D1E80F02E7A7B60B2B559C56CA255E6079CC9
could help you to find out
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT ciphers
                                           But: Unknown DH group (1024 bits)
 BEAST (CVE-2011-3389)                     not vulnerable (OK), no SSL3 or TLS1
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Check patches
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)

Could not determine the protocol, only simulating generic clients.

 Running client simulations via sockets

 Android 4.4.2                TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 5.0.0                TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 6.0                  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 7.0 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 8.1 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 9.0 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
```

root@qsccsq: ~/Desktop/testssl.sh

File Actions Edit View Help

root@qsccsq: ~...top/testssl.sh

```
 Running client simulations via sockets

 Android 4.4.2                TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 5.0.0                TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 6.0                  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 7.0 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 8.1 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 9.0 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Android 10.0 (native)        TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Chrome 74 (Win 10)           TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Chrome 79 (Win 10)           TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Firefox 66 (Win 8.1/10)      TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Firefox 71 (Win 10)          TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 IE 6 XP                      No connection
 IE 8 Win 7                   No connection
 IE 8 XP                      No connection
 IE 11 Win 7                  TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win 8.1                TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win Phone 8.1          TLSv1.2 ECDHE-RSA-AES128-SHA, 256 bit ECDH (P-256)
 IE 11 Win 10                 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Edge 15 Win 10               TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Edge 17 (Win 10)             TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Opera 66 (Win 10)            TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 9 iOS 9               TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 9 OS X 10.11          TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 10 OS X 10.12         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 12.1 (iOS 12.2)       TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 13.0 (macOS 10.14.6)  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Apple ATS 9 iOS 9            TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Java 6u45                    No connection
 Java 7u25                    No connection
 Java 8u161                   TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Java 11.0.2 (OpenJDK)
```

```
root@qsccsq: ~/Desktop/testssl.sh
File  Actions  Edit  View  Help
root@qsccsq: ~...top/testssl.sh

 Edge 17 (Win 10)             TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Opera 66 (Win 10)            TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 9 iOS 9               TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 9 OS X 10.11          TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 10 OS X 10.12         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 12.1 (iOS 12.2)       TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Safari 13.0 (macOS 10.14.6)  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Apple ATS 9 iOS 9            TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Java 6u45                    No connection
 Java 7u25                    No connection
 Java 8u161                   TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Java 11.0.2 (OpenJDK)        TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Java 12.0.1 (OpenJDK)        TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 OpenSSL 1.0.2e               TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 OpenSSL 1.1.0l (Debian)      TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 OpenSSL 1.1.1d (Debian)      TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Thunderbird (68.3)           TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)


 Rating (experimental)

 Rating specs (not complete)  SSL Labs's 'SSL Server Rating Guide' (version 2009q from 2020-01-30)
 Specification documentation  https://github.com/ssllabs/research/wiki/SSL-Server-Rating-Guide
 Protocol Support (weighted)  100 (30)
 Key Exchange     (weighted)  80 (24)
 Cipher Strength  (weighted)  90 (36)
 Final Score                  90
 Overall Grade                B

 Done 2020-07-06 21:55:27 [ 212s] -->> 158.135.1.242:443 (shsu.edu) <<--


root@qsccsq:~/Desktop/testssl.sh#
```

Homework: Perform SSL Heartbleed Detection, SSL Encryption Issue Detection and TestSSL Common SSL Issue detection methods on a webpage, which is selected by you. Provide half a page of your observations about yours target and differences of tools.