Lab 5: Tunnelling and Web Security

Objective: In this lab we will investigate the usage of SSL/TLS and VPN tunnels.

📖 **YouTube** Demo: https://youtu.be/ASCDJq4Wy9Y

A Web cryptography assessment

The Ssllabs tool (https://ssllabs.com) can be used to assess the security of the cryptography used on a Web site. Pick three of your favouriate sites to scan. Now perform a test on them, and determine:

Site	Site 1:	Site 2:	Site 3:
What grade does the site get?	B rating	B rating	A rating
The digital certificate key size and type?	EC 256 bits SHA256withECDSA	RSA 2048 bits SHA256withRSA	RSA 2048 bits SHA256withRSA
Does the name of the site match the name on the server?	Yes	Yes	Yes
Who is the signer of the digital certificate?	WE2	WR1	DigiCert Global G2 TLS RSA SHA256 2020 CA1
The expiry date on the digital certificate?	Mon, 19 Jan 2026 08:33:50 UTC	Sat, 24 Jan 2026 10:44:47 UTC	Fri, 24 Jul 2026 23:59:59 UTC
What is the hashing method on the certificate?	SHA256withECDSA	SHA256withRSA	SHA256withRSA
If it uses RSA keys, what is the e value that is used in the encryption (M^e mod N)?	-	256	256
Determine a weak cipher suite used and example why it might be weak?	Protocol Support: TLS 1.0 and TLS 1.1	Protocol Support: TLS 1.0 and TLS 1.1	All good
Is SSL v2 supported?	No	No	No
If SSL v2 was supported, what problems might there be with the site (this will require some research)?	-	-	-
Outline the usage of TLS 1.0/1.1 and 1.2, and identify a problem if one of these TLS versions were not supported?	Supports: TLS 1.0 and TLS 1.1 Older devices or software may fail to connect	Supports: TLS 1.0 and TLS 1.1 Older devices or software may fail to connect	Not Supported: TLS 1.0 and TLS 1.1
Is the site vulnerable to Heartbleed? Is the site vulnerable to DROWN? Is the site vulnerable to BEAST? Is the site vulnerable to POODLE?	No No Not mitigated server-side No	No No Not mitigated server-side No	No No Mitigated server-side No

Research questions:

What does TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 identify?

```
It is the cipher suite used in a TLS connection.
- Key exchange: ECDHE (Elliptic-Curve Diffie-Hellman Ephemeral)
- Authentication: RSA
- Encryption: AES-256 in CBC mode
- Hashing / HMAC: SHA-384
```

If a site gets a 'T' grade, what is the problem?

```
Trust issues (T); If ssllabs don't trust a certificate (and there aren't any other security issues),
it assigns it a T grade (for "trust").
```

If the site was susceptible to Poodle, what is the vulnerability?

```
POODLE is an attack on SSL 3.0's padding (and occasionally TLS padding).
- SSLv3 is enabled, and
- An attacker can decrypt cookies or session data by exploiting CBC padding.
```

Can you find a site which gets an "A+"? What features does a site need to get an "A+" grade?

```
I didn't find any 'A+' grade website
To get an 'A+' grade on SSLLabs is to install a valid SSL certificate with CA bundle and configure HSTS in .htaccess.
```

A.2 We will now create a Python program which calls up the SSLlabs assessment. First create a CSV file (sites.csv) with your sites in it. The format is Name of site, URL:

```
web,site
Cloudflare,www.cloudflare.com
BBC,bbc.co.uk
```

Next enter the following code and run it:

```
# Code from
https://github.com/TrullJ/ssllabs/blob/master/ssllabsscanner.py
import requests
import time
import sys
import logging

API = 'https://api.ssllabs.com/api/v2/'


def requestAPI(path, payload={}):
    '''This is a helper method that takes the path to the relevant
        API call and the user-defined payload and requests the
        data/server test from Qualys SSL Labs.
        Returns JSON formatted data'''

    url = API + path

    try:
        response = requests.get(url, params=payload)
    except requests.exception.RequestException:
        logging.exception('Request failed.')
        sys.exit(1)

    data = response.json()
    return data


def resultsFromCache(host, publish='off', startNew='off', fromCache='on',
all='done'):
    path = 'analyze'
    payload = {
                'host': host,
                'publish': publish,
```

```
                'startNew': startNew,
                'fromCache': fromCache,
                'all': all
            }
    data = requestAPI(path, payload)
    return data


def newScan(host, publish='off', startNew='on', all='done',
ignoreMismatch='on'):
    path = 'analyze'
    payload = {
                'host': host,
                'publish': publish,
                'startNew': startNew,
                'all': all,
                'ignoreMismatch': ignoreMismatch
            }
    results = requestAPI(path, payload)

    payload.pop('startNew')

    while results['status'] != 'READY' and results['status'] != 'ERROR':
        time.sleep(30)
        results = requestAPI(path, payload)

    return results


import csv
with open('sites.csv') as csvfile:
      reader = csv.DictReader(csvfile)
      for row in reader:

            url = row['site'].strip()

            a = newScan(url)
            with open("out3.txt", "a") as myfile:
                  myfile.write(str(row['web'])+"\n"+str(a)+"\n\n\n")
                  print row['web']
```

Note that it will can take a few minutes to perform a single scan. By reading the out3.txt file, outline your findings:

Site name: `www.cloudflare.com` Site rating: `'grade': 'B'`

Other significant details:


```
'cert': {'subject': 'CN=www.cloudflare.com', 'commonNames': ['www.cloudflare.com'],
'altNames': ['www.cloudflare.com', '*.www.cloudflare.com'], 'notBefore': 1762382500000,
'notAfter': 1770161971000, 'issuerSubject': 'CN=WR1, O=Google Trust Services, C=US',
'issuerLabel': 'WR1', 'sigAlg': 'SHA256withRSA', 'revocationInfo': 3, 'crlURIs': ['http://
c.pki.goog/wr1/eOb30fCtrhU.crl'], 'ocspURIs': ['http://o.pki.goog/s/wr1/wUo'],
'revocationStatus': 2, 'crlRevocationStatus': 3, 'ocspRevocationStatus': 2, 'sgc': 0,
'issues': 0, 'sct': True, 'mustStaple': 0, 'sha1Hash':
'ef30e93943ad05c2710ad0bd8b736dda10347bf7', 'pinSha256':
'BU5rjSqVz3GpFjjGOUFSBuVSbrzjLuK04tq3o3eJnjU='}
```


Site name: `bbc.co.uk` Site rating: `'grade': 'B'`

Other significant details:


```
'cert': {'subject': 'CN=www.bbc.com, O=BRITISH BROADCASTING CORPORATION, L=London, ST=London,
C=GB', 'commonNames': ['www.bbc.com'], 'altNames': ['www.bbc.com', 'www.bbc.co.uk',
'www.bbcrussian.com', 'bbc.co.uk', 'bbcrussian.com', 'session.bbc.co.uk', 'search.bbc.co.uk',
'open.live.bbc.co.uk', 'news.bbc.co.uk', 'newsimg.bbc.co.uk', 'wwwnews.live.bbc.co.uk',
'cdnedge.bbc.co.uk', 'newsrss.bbc.co.uk', 'newsvote.bbc.co.uk', 'playlists.bbc.co.uk',
'r.bbci.co.uk', 'node1.bbcimg.co.uk', 'news.bbcimg.co.uk', 'account.bbc.com',
'session.bbc.com', 'bbc.com'], 'notBefore': 1751007241000, 'notAfter': 1785147977000,
'issuerSubject': 'CN=GlobalSign RSA OV SSL CA 2018, O=GlobalSign nv-sa, C=BE', 'issuerLabel':
'GlobalSign RSA OV SSL CA 2018', 'sigAlg': 'SHA256withRSA', 'revocationInfo': 3, 'crlURIs':
['http://crl.globalsign.com/gsrsaovsslca2018.crl'], 'ocspURIs': ['http://ocsp.globalsign.com/
gsrsaovsslca2018'], 'revocationStatus': 2, 'crlRevocationStatus': 3, 'ocspRevocationStatus':
2, 'sgc': 0, 'issues': 0, 'sct': True, 'mustStaple': 0, 'sha1Hash':
'6bea31a9a0fb489820aa4b22d77710bab738ab1a', 'pinSha256':
'DiiZIJmLywdiHumhEESadJ91tgsetSB+QONHXFVxdhM='}
```

B Viewing details

No	Description	Result
B.1	On your VM instance (or your desktop), run Wireshark and capture traffic from your main network connection. Start a Web browser and go to **Google.com**. Stop Wireshark and identify some of your connection details:	Your IP address and TCP port: `IP: 10.0.2.15 | Port: 52670` Google's Web server IP address and TCP port: `IP: 209.85.203.94 | Port: 443` Which SSL/TLS version is used: `TLS 1.2` By examining the Wireshark trace, which encryption method is used for the tunnel (hint: look in the 'Server Hello' response): `TLS_AES_128_GCM_SHA256 (0x1301)` By examining the Wireshark trace, which hashing method is used for the tunnel (hint: look in the 'Server Hello' response): `TLS_AES_128_GCM_SHA256 (0x1301)` By examining the Wireshark trace, what is the length of the encryption key (hint: look in the 'Server Hello' response): `128` Using Firefox, and examining the connection details from the site (click on green padlock), can you verify the TLS version, the symmetric key encryption method, the handshaking method and the hashing method used within the tunnel? A sample is shown below. `Yes` **Technical Details** Connection Encrypted (TLS_AES_128_GCM_SHA256, 128 bit keys, TLS 1.3) The page you are viewing was encrypted before being transmitted over the Internet. Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.
B.2	Run Wireshark and capture traffic from your main network connection. Start a Web browser and go to **https://twitter.com**. Stop Wireshark and identify some of your connection details:	Your IP address and TCP port: `IP: 10.0.2.15 | Port: 60390` Twitter's Web server IP address and TCP port: `IP: 13.107.136.10 | Port: 443` Which SSL/TLS version is used: `TLS 1.3` By examining the Wireshark trace, which encryption method is used for the tunnel: `TLS_AES_256_GCM_SHA384 (0x1301)`

		By examining the Wireshark trace, which hash method is used for the tunnel: `TLS_AES_256_GCM_SHA384 (0x1301)` By examining the Wireshark trace, what is the length of the encryption key: `256` Using Firefox, and examining the connection details from the site (click on green padlock), can you verify the TLS version, the symmetric key encryption method, the handshaking method and the hashing method used within the tunnel? `No`

C OpenSSL

No	Description	Result
C.1	On your VM instance (or your desktop), make a connection to the **www.live.com** Web site: `openssl s_client -connect www.live.com:443`	Which SSL/TLS method has been used: `TLSv1.3` Which method is used on the encryption key on the certificate, and what is the size of the public key? `RSA, 2048-bit` Which is the handshaking method that has been used to create the encryption key? `ECDH (secp521r1), 521 bits` Which TLS version is used for the tunnel? `TLSv1.3` Which symmetric encryption method is used for the tunnel: `AES-256-GCM` Which hashing method is used for the tunnel: `SHA-384` What is the length of the symmetric encryption key: `256-bit` Who has signed the certificate: `DigiCert Cloud Services CA-1`

D Examining traces

No	Description	Result
D.1	Download the following file, and examine the trace with Wireshark: `http://asecuritysite.com/log/ssl.zip`	Client IP address and TCP port: `IP: 192.168.0.20 | Port: 2099` Web server IP address and TCP port: `IP: 66.211.169.66 | Port: 443` Determine one of the symmetric key encryption methods, the key exchange, and the hashing methods that the client wants to use (Hint: look at the 'Client Hello' packet)" `AES_256_CBC SHA` Which SSL/TLS method has been used: `TLS 1.0` Which encryption method is used for the tunnel: `3DES_EDE_CBC` Which hashing method is used for the tunnel: `SHA` What is the length of the encryption key: `168-bit`
D.2	Download the following file, and examine the trace with Wireshark: `http://asecuritysite.com/log/https.zip`	Client IP address and TCP port: `IP: 172.16.121.155 | Port: 3923` Web server IP address and TCP port: `IP: 87.106.189.123 | Port: 443` Which SSL/TLS method has been used: `TLS 1.2` Which encryption method is used for the tunnel: `AES_256_CBC` Which hashing method is used for the tunnel: `SHA` What is the length of the encryption key: `256-bit`
D.3	Download the following file, and examine the trace with Wireshark: `http://asecuritysite.com/log/heart.zip`	Client IP address and TCP port: `IP: 172.16.121.1 | Port: 64666` Web server IP address and TCP port: `IP: 172.16.121.150 | Port: 443` Which SSL/TLS method has been used: `TLS 1.2`

		Which encryption method is used for the tunnel: `AES_256_GCM` Which hashing method is used for the tunnel: `SHA384` What is the length of the encryption key: `256-bit`
D.4	Download the following file, and examine the trace with Wireshark: `http://asecuritysite.com/log/ipsec.zip`	Which is the IP address of the client and of the server: `Client: 192.168.0.20` `Server: 146.176.210.2` Which packet number identifies the start of the VPN connection (Hint: look for UDP Port 500): `2` Determine one of the encryption and the hashing methods that the client wants to use: `AES-CBC` `SHA` Now determine the encryption and hashing methods that are agreed in the ISAKMP: `3DES-CBC` `MD5`
	Download the following file, and examine the trace with Wireshark: `http://asecuritysite.com/log/tor.zip`	Which TCP port does the client use to send to? `9001` What is the IP address of the Tor node that the client connects to? `IP: 172.16.121.169 | Port: 1113` What is strange about the packet size? `Tor traffic often uses fixed-size "cells" (512 bytes) – strange to see identical-length packets.` Is SSL/TLS used for the connection? `No, tor is not using TLS for that hop.` `I wasn't be able to find any tls handshakes (but found some packets)` Can you trace any content in the conversation? `Because Tor encrypts its application data heavily, you typically cannot read the content of the conversation in Wireshark.` Can you determine the Web site that is being connected to? `No, because the Tor node decrypts data at exit and then sends the request to the destination – the original site name is not visible in the TLS handshake.`

What I should have learnt from this lab?

The key things learnt:

- How do perform a cryptography assessment on a Web site (using ssllabs) and in how to spot weaknesses.

- Able to interpret an SSL/TLS session, and identity the important elements of the Client Hello, and the Server Hello.