

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 favourite sites to scan. Now perform a test on them, and determine:

Site	Site 1:	Site 2:	Site 3:
What grade does the site get?			
The digital certificate key size and type?			
Does the name of the site match the name on the server?			
Who is the signer of the digital certificate?			
The expiry date on the digital certificate?			
What is the hashing method on the certificate?			
If it uses RSA keys, what is the e value that is used in the encryption ($M^e \bmod N$)?			
Determine a weak cipher suite used and example why it might be weak?			
Is SSL v2 supported?			
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?			
Is the site vulnerable to Heartbleed? Is the site vulnerable to DROWN? Is the site vulnerable to BEAST? Is the site vulnerable to POODLE?			

Research questions:

What does TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 identify?

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

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

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

A.2 We will now create a Python program which calls up the SSLabs 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/TrulliJ/sslabs/blob/master/sslabscanner.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: Site rating:

Other significant details:

Site name: Site rating:

Other significant details:

B Viewing details

No	Description	Result
B.1	<p>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.</p> <p>Stop Wireshark and identify some of your connection details:</p>	<p>Your IP address and TCP port: Google's Web server IP address and TCP port: Which SSL/TLS version is used: By examining the Wireshark trace, which encryption method is used for the tunnel (hint: look in the 'Server Hello' response): By examining the Wireshark trace, which hashing method is used for the tunnel (hint: look in the 'Server Hello' response): By examining the Wireshark trace, what is the length of the encryption key (hint: look in the 'Server Hello' response): 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.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Technical Details</p> <p>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.</p> </div>
B.2	<p>Run Wireshark and capture traffic from your main network connection. Start a Web browser and go to https://twitter.com.</p> <p>Stop Wireshark and identify some of your connection details:</p>	<p>Your IP address and TCP port: Twitter's Web server IP address and TCP port: Which SSL/TLS version is used: By examining the Wireshark trace, which encryption method is used for the tunnel:</p>

	<p>By examining the Wireshark trace, which hash method is used for the tunnel:</p> <p>By examining the Wireshark trace, what is the length of the encryption key:</p> <p>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?</p>
--	---

C OpenSSL

No	Description	Result
C.1	<p>On your VM instance (or your desktop), make a connection to the www.live.com Web site:</p> <pre>openssl s_client -connect www.live.com:443</pre>	<p>Which SSL/TLS method has been used:</p> <p>Which method is used on the encryption key on the certificate, and what is the size of the public key?</p> <p>Which is the handshaking method that has been used to create the encryption key?</p> <p>Which TLS version is used for the tunnel?</p> <p>Which symmetric encryption method is used for the tunnel:</p> <p>Which hashing method is used for the tunnel:</p> <p>What is the length of the symmetric encryption key:</p> <p>Who has signed the certificate:</p>

--	--	--

D Examining traces

No	Description	Result
D.1	<p>Download the following file, and examine the trace with Wireshark:</p> <p>http://asecuritysite.com/log/ssl.zip</p>	<p>Client IP address and TCP port:</p> <p>Web server IP address and TCP port:</p> <p>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”)</p> <p>Which SSL/TLS method has been used:</p> <p>Which encryption method is used for the tunnel:</p> <p>Which hashing method is used for the tunnel:</p> <p>What is the length of the encryption key:</p>
D.2	<p>Download the following file, and examine the trace with Wireshark:</p> <p>http://asecuritysite.com/log/https.zip</p>	<p>Client IP address and TCP port:</p> <p>Web server IP address and TCP port:</p> <p>Which SSL/TLS method has been used:</p> <p>Which encryption method is used for the tunnel:</p> <p>Which hashing method is used for the tunnel:</p> <p>What is the length of the encryption key:</p>
D.3	<p>Download the following file, and examine the trace with Wireshark:</p> <p>http://asecuritysite.com/log/heart.zip</p>	<p>Client IP address and TCP port:</p> <p>Web server IP address and TCP port:</p> <p>Which SSL/TLS method has been used:</p>

		<p>Which encryption method is used for the tunnel:</p> <p>Which hashing method is used for the tunnel:</p> <p>What is the length of the encryption key:</p>
D.4	<p>Download the following file, and examine the trace with Wireshark:</p> <p>http://asecuritysite.com/log/ipsec.zip</p>	<p>Which is the IP address of the client and of the server:</p> <p>Which packet number identifies the start of the VPN connection (Hint: look for UDP Port 500):</p> <p>Determine one of the encryption and the hashing methods that the client wants to use:</p> <p>Now determine the encryption and hashing methods that are agreed in the ISAKMP:</p>
	<p>Download the following file, and examine the trace with Wireshark:</p> <p>http://asecuritysite.com/log/tor.zip</p>	<p>Which TCP port does the client use to send to?</p> <p>What is the IP address of the Tor node that the client connects to?</p> <p>What is strange about the packet size?</p> <p>Is SSL/TLS used for the connection?</p> <p>Can you trace any content in the conversation?</p> <p>Can you determine the Web site that is being connected to?</p>

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.