Lab 8: Tunnelling

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

1 Web cryptography assessment

The ssllabs tool (https://ssllabs.com) can be used to assess the security of the cryptography used on a Web site. You will be given a range of Web sites to scan in the lab, and you should pick three sites from the list. 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 mod N)?			
Determine a weak cipher suite used and example why it might be weak?			
What does TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 identify?			
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:

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

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

2 Viewing details

No	Description	Result
1	Go to your Kali Linux instance. Run Wireshark and capture traffic from your main network connection. Start a Web browser, and go to **www.napier.ac.uk**.	Your IP address and TCP port: Napier’s Web server IP address and TCP port:

	Stop Wireshark and identify some of your connection details:	Right-click on the GET HTTP request from the client, and follow the stream: What does the red and blue text identify? Can you read the HTTP requests that go from the client to the server? [Yes][No]
2	Go to your Kali Linux instance. 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: 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: By examining the Wireshark trace, which hash method is used for the tunnel: By examining the Wireshark trace, what is the length of the encryption key: By examining the certificate from the browser which encryption method is used for the tunnel: By examining the certificate from the browser, which hash method is used for the tunnel:

		By examining the certificate from the browser is the length of the encryption key:
3	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: 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: By examining the Wireshark trace, which hash method is used for the tunnel: By examining the Wireshark trace, what is the length of the encryption key: By examining the certificate from the browser which encryption method is used for the tunnel: By examining the certificate from the browser, which hash method is used for the tunnel: By examining the certificate from the browser is the length of the encryption key:

3 OpenSSL

No	Description	Result
1	Go to your Kali Linux instance, and 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: Which method is used on the encryption key on the certificate, and what is the size of the public key? Which is the handshaking method that has been used to create the encryption key? Which TLS version is used for the tunnel? Which encryption method is used for the tunnel: Which hash method is used for the tunnel: What is the length of the encryption key: What is the serial number of the certificate: Who has signed the certificate:

4 Examining traces

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

		Which SSL/TLS method has been used: Which encryption method is used for the tunnel: Which hash method is used for the tunnel: What is the length of the encryption key: Can you spot the packet which identifies the Heartbleed vulnerability?
3	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: Which packet number identifies the start of the VPN connection (Hint: look for UDP Port 500): Determine one of the encryption and hashing methods that the client wants to use: Now determine the encryption and hashing methods that are agreed in the ISAKMP:

5 SSL Labs Python

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
```

Using the following code:

```
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
print ("Scanning")
with open('sites.csv') as csvfile:
  reader = csv.DictReader(csvfile)
  for row in reader:
    url = row['site'].strip()
    print ("Scanning...",url)
    a = newScan(url)
    with open("out3.txt", "a") as myfile:
      myfile.write(str(row['web'])+"\n"+str(a)+"\n\n\n")
```

```
        print (row['web'])
```

The repl.it site is here.

Now pick to domains to scan. Note that it 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:

6 IDS

Setup a Microsoft Windows server (from AWS) or use your Windows server on vSoC, and then install Snort in the folder c:\snort. Next locate the snort.exe program and download the coursework as a pcap file here. Next, create a rules file to detect the connection between the bot and the controller, and save it as cw.rules:

```
# Connection detection
alert tcp any any -> any 5000 ( msg:"Port 5000";sid:10000)
alert tcp any any -> any 5001 (msg:"Port 5001";sid:10001)
alert tcp any any -> any 5002 (msg:"Port 5002";sid:10002)
alert tcp any any -> any 5003 (msg:"Port 5003";sid:10003)
alert tcp any any -> any 5004 (msg:"Port 5004";sid:10004)
alert tcp any any -> any 5005 (msg:"Port 5005";sid:10005)
```

```
# Content detection (e.g. "bye")
alert tcp any any -> any 5000 (msg:"Port 5000 command bye"; content:"bye"; sid:11000)
alert tcp any any -> any 5001 (msg:"Port 5001 command bye"; content:"bye"; sid:11001)
alert tcp any any -> any 5002 (msg:"Port 5002 command bye"; content:"bye"; sid:11002)
alert tcp any any -> any 5003 (msg:"Port 5003 command bye"; content:"bye"; sid:11003)
alert tcp any any -> any 5004 (msg:"Port 5004 command bye"; content:"bye"; sid:11004)
alert tcp any any -> any 5005 (msg:"Port 5005 command bye"; content:"bye"; sid:11005)

# Some additional pre-processor things
preprocessor stream5_global: track_tcp yes, \
track_udp yes, \
track_icmp no, \
max_tcp 262144, \
max_udp 131072, \
max_active_responses 2, \
min_response_seconds 5
preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, \
overlap_limit 10, small_segments 3 bytes 150, timeout 180, \
ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137 139 143 \
161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667 6668 6669 \
7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779, \
ports both 80 81 82 83 84 85 86 87 88 89 90 110 311 383 443 465 563 591 593 631 636 901 989 992 993 994 995 1220 1414 1830 2301 2381
2809 3037 3057 3128 3443 3702 4343 4848 5250 6080 6988 7907 7000 7001 7144 7145 7510 7802 7777 7779 \
7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912 7913 7914 7915 7916 \
7917 7918 7919 7920 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180 8222 8243 8280 8300 8500 8800 8888 8899 9000 9060
9080 9090 9091 9443 9999 10000 11371 34443 34444 41080 50000 50002 55555
preprocessor stream5_udp: timeout 180
```

Now create a subfolder named log:

```
mkdir log
```

Now run with:

```
snort.exe -c cw.rules -r cw.pcal -k none
```

What do you observe from the output?
Your alerts should be in the log\alert.ids folder. What do you examine from the contents of this file?
Now try this approach on the pcap file that you have captured for the coursework.