MODULE TITLE: Implementing Secure Systems

MODULE CODE: WM242-24 (cw2)

STUDENT ID NUMBER: U1834961 - U1824952

GPG FINGERPRINT:

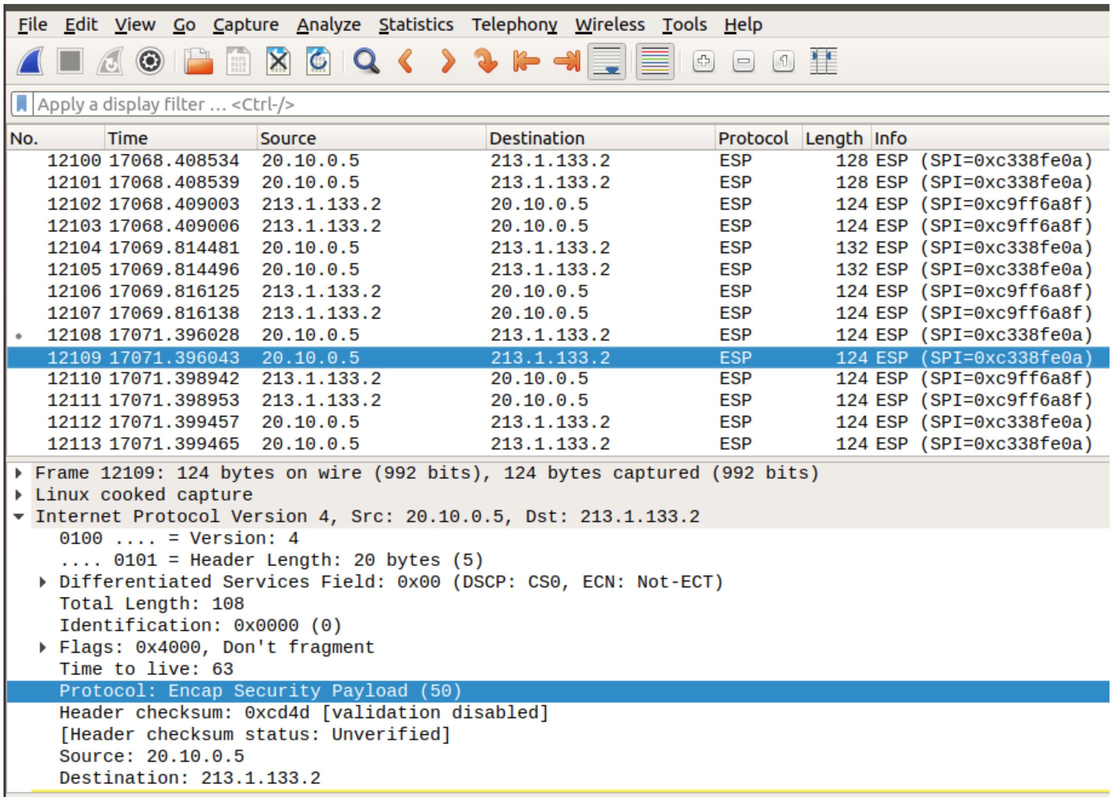
F157202342CC4C1BAD90C09463167BBAE805C8CC

01D30C70CBEB3B45273ED0E3C280BCD3EB9771C1

PHASE 1:

Implementation	Reasoning	Evidence
Implemented an x509 structure across the organisations network. This included one root CA and one intermediate CA.	The organisation has a requirement that an x509 certificate authority hierarchy be implemented. Creating an Intermediate CA that generates a certificate signing request to the root CA is an example of the x509 hierarchy; the intermediate CA can go on to be used to sign users requesting remote access to the organisations network.	Build_x509.sh Network Model Diagram
Generated keys and certificates for the root CA, Intermediate CA, all mobusrs, gw-u1834961-u1824952.cyber.test and www. u1834961-u1824952.cyber.test	The root CA generates a key and a self-signed certificate. The I_CA then generates its own key and generates a certificate signing request to the root CA to initiate a chain of trust.	From build_x509.sh; key-gen cmd for root CA: ipsec pki --gen --type rsa --size 4096 --outform pem > CA.key.pem Intermediate CA certificate signing request cmd: ipsec pki --issue --ca --in ica.csr.pem --lifetime 365 --cacert CA.crt.pem --cakey CA.key.pem --dn "C=UK, O=University of Warwick - Cyber Security Centre, OU=ISS CW2, CN=i_ca.cyber.test" --san 192.168.65.5 --san @192.168.65.5 --flag serverAuth --flag ikeIntermediate --outform pem > ica.crt.pem
Ipsec VPN implemented using StrongSwan for one remote user wishing to access the network. Included creating a user that sits on the 192.168.65.0/24 network (Warwick internal) having some	Setting up a secure VPN connection for authenticated users requesting remote access to the organisations network.	Within internet.startup is the tcpdump cmd: tcpdump -i any -w /*hostlab*/internetCapture.pcap which will collect traffic

communication with an external user (mobusr1); configured such that communication between the two will be encrypted through the authentication of trusted certificates.		destined for the organisations network. Using a netcat listening on host ‘a’ with the command: nc -lvnp 8000 creates a listener on port 8000 on mobusr 1which has the ip address 20.10.0.5 connects to hist ‘a’ using the cmd: nc 192.168.65.20 8000 and generated some traffic for proof that the packets being sent across the internet are encrypted.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

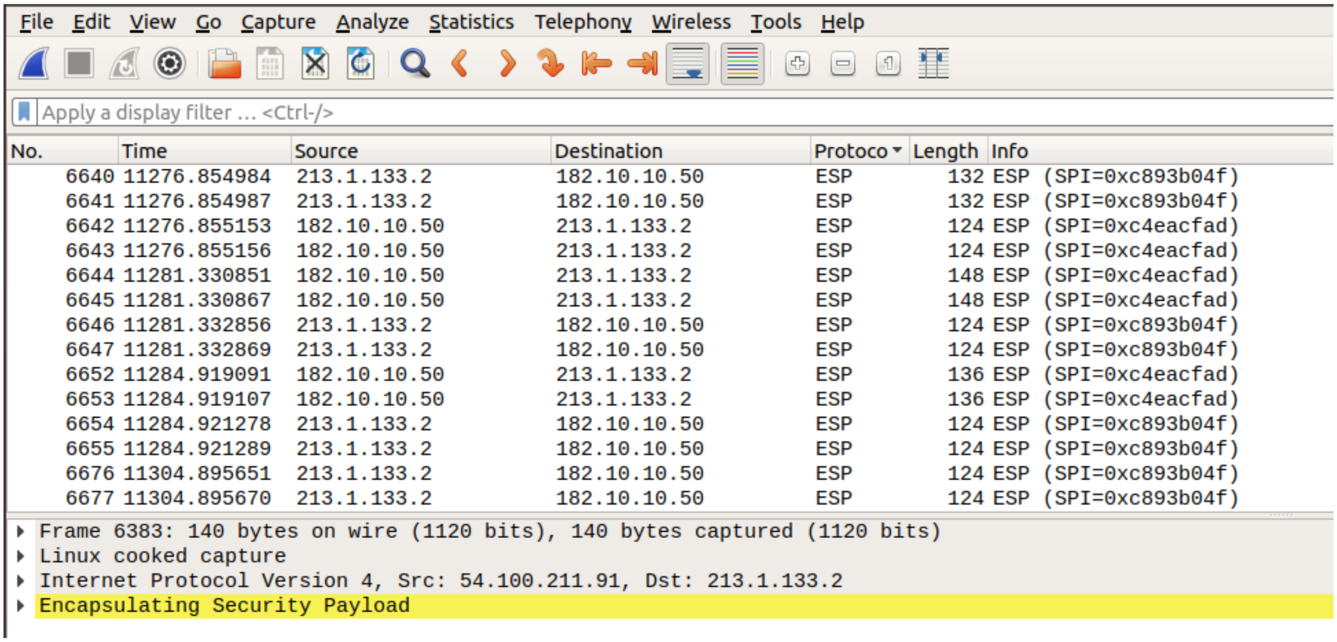
```
No.   Time            Source       Destination   Protocol Length Info
12100 17068.408534    20.10.0.5    213.1.133.2   ESP      128 ESP (SPI=0xc338fe0a)
12101 17068.408539    20.10.0.5    213.1.133.2   ESP      128 ESP (SPI=0xc338fe0a)
12102 17068.409003    213.1.133.2  20.10.0.5     ESP      124 ESP (SPI=0xc9ff6a8f)
12103 17068.409006    213.1.133.2  20.10.0.5     ESP      124 ESP (SPI=0xc9ff6a8f)
12104 17069.814481    20.10.0.5    213.1.133.2   ESP      132 ESP (SPI=0xc338fe0a)
12105 17069.814496    20.10.0.5    213.1.133.2   ESP      132 ESP (SPI=0xc338fe0a)
12106 17069.816125    213.1.133.2  20.10.0.5     ESP      124 ESP (SPI=0xc9ff6a8f)
12107 17069.816138    213.1.133.2  20.10.0.5     ESP      124 ESP (SPI=0xc9ff6a8f)
12108 17071.396028    20.10.0.5    213.1.133.2   ESP      124 ESP (SPI=0xc338fe0a)
12109 17071.396043    20.10.0.5    213.1.133.2   ESP      124 ESP (SPI=0xc338fe0a)
12110 17071.398942    213.1.133.2  20.10.0.5     ESP      124 ESP (SPI=0xc9ff6a8f)
12111 17071.398953    213.1.133.2  20.10.0.5     ESP      124 ESP (SPI=0xc9ff6a8f)
12112 17071.399457    20.10.0.5    213.1.133.2   ESP      124 ESP (SPI=0xc338fe0a)
12113 17071.399465    20.10.0.5    213.1.133.2   ESP      124 ESP (SPI=0xc338fe0a)

▸ Frame 12109: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
▸ Linux cooked capture
▾ Internet Protocol Version 4, Src: 20.10.0.5, Dst: 213.1.133.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 108
    Identification: 0x0000 (0)
  ▸ Flags: 0x4000, Don't fragment
    Time to live: 63
    Protocol: Encap Security Payload (50)
    Header checksum: 0xcd4d [validation disabled]
    [Header checksum status: Unverified]
    Source: 20.10.0.5
    Destination: 213.1.133.2
```

Figure 1: An Encapsulated Security Payload from mobusr1 that uses the internet to access the internal network.

PHASE 2:

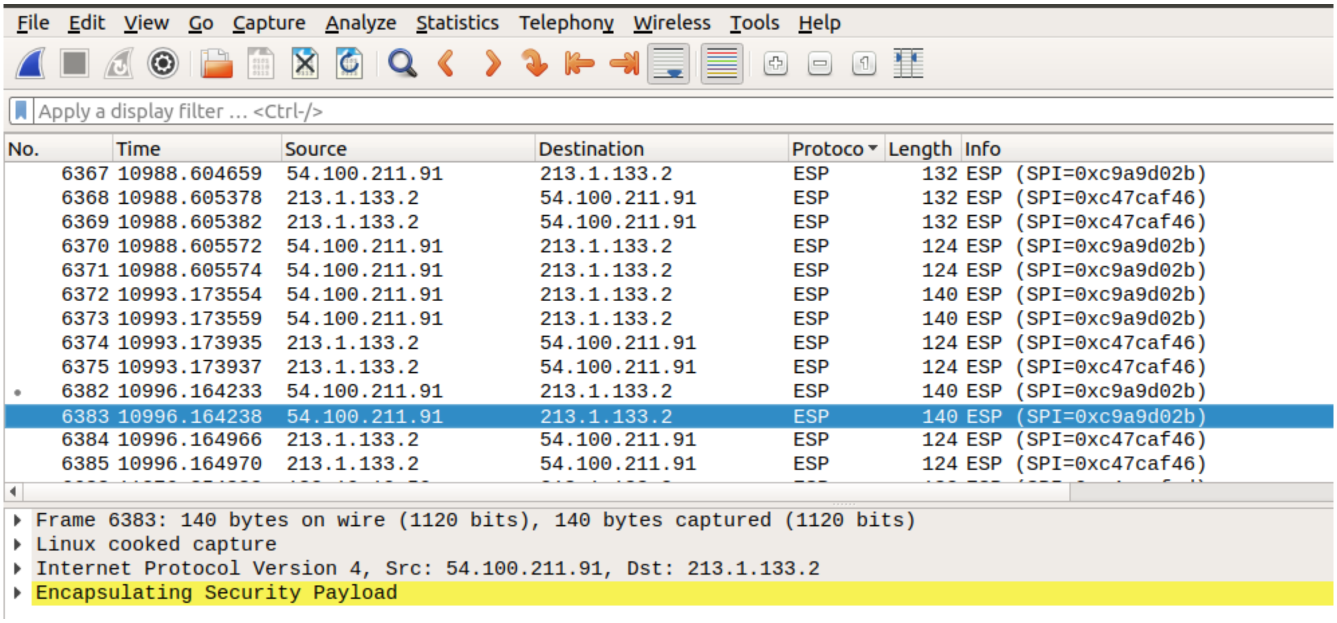
Implementation	Reasoning	Evidence
Implement IPsec VPN for two or more users. Achieved by creating two more external users that each perform achieve communication with a host sat on the Warwick internal network (192.168.65.0/24). Again, the authentication used is the x509 certificates.	Setting up further secure VPN connections for 2 more users both requesting remote access to Warwick's internal network.	Using the same test as before, setting up a netcat listener on host 'a': nc -lvnp 8000 and connection to it one mobusr 2 and mobusr 3. Figure 2: Is mobusr2 sending encrypted packets across the internet from the IP address 182.10.10.50 Figure 3: Is mobus3 sending the encrypted packets across the internet from the IP address 54.100.211.91
Correctly allocate the IP addresses and domain names on the network.	Ensuring that the IP addresses for the internal network are correct, in the 192.168.65.0/24 rand. Changes to the name of the VPN gateway (gw.u 1834961-u1824952.cyber.test) and the Apache webserver (www.u1834961-u1824952.cyber.test)	gw.u 1834961-u1824952.cyber.test.startup includes: #for the internal network ifconfig eth0 192.168.65.1/24 up #designated public ip for the organisation ifconfig eth1 213.1.133.2/27 up # 1:1 routing for the Apache web server. ifconfig eth1:1 213.1.133.3/27 up www.u1834961-u1824952.cyber.test.startup ifconfig eth0 192.168.65.10 a.startup ifconfig eth0 192.168.65.20


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.       Time            Source            Destination        Protoco  Length  Info
     6640 11276.854984    213.1.133.2       182.10.10.50       ESP         132  ESP (SPI=0xc893b04f)
     6641 11276.854987    213.1.133.2       182.10.10.50       ESP         132  ESP (SPI=0xc893b04f)
     6642 11276.855153    182.10.10.50      213.1.133.2        ESP         124  ESP (SPI=0xc4eacfad)
     6643 11276.855156    182.10.10.50      213.1.133.2        ESP         124  ESP (SPI=0xc4eacfad)
     6644 11281.330851    182.10.10.50      213.1.133.2        ESP         148  ESP (SPI=0xc4eacfad)
     6645 11281.330867    182.10.10.50      213.1.133.2        ESP         148  ESP (SPI=0xc4eacfad)
     6646 11281.332856    213.1.133.2       182.10.10.50       ESP         124  ESP (SPI=0xc893b04f)
     6647 11281.332869    213.1.133.2       182.10.10.50       ESP         124  ESP (SPI=0xc893b04f)
     6652 11284.919091    182.10.10.50      213.1.133.2        ESP         136  ESP (SPI=0xc4eacfad)
     6653 11284.919107    182.10.10.50      213.1.133.2        ESP         136  ESP (SPI=0xc4eacfad)
     6654 11284.921278    213.1.133.2       182.10.10.50       ESP         124  ESP (SPI=0xc893b04f)
     6655 11284.921289    213.1.133.2       182.10.10.50       ESP         124  ESP (SPI=0xc893b04f)
     6676 11304.895651    213.1.133.2       182.10.10.50       ESP         124  ESP (SPI=0xc893b04f)
     6677 11304.895670    213.1.133.2       182.10.10.50       ESP         124  ESP (SPI=0xc893b04f)

▸ Frame 6383: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits)
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 54.100.211.91, Dst: 213.1.133.2
▸ Encapsulating Security Payload
```

Figure 2: Shows an Encapsulated Security Payload that uses the internet to access the internal network for mobusr2.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.       Time            Source            Destination        Protoco  Length  Info
     6367 10988.604659    54.100.211.91     213.1.133.2        ESP         132  ESP (SPI=0xc9a9d02b)
     6368 10988.605378    213.1.133.2       54.100.211.91      ESP         132  ESP (SPI=0xc47caf46)
     6369 10988.605382    213.1.133.2       54.100.211.91      ESP         132  ESP (SPI=0xc47caf46)
     6370 10988.605572    54.100.211.91     213.1.133.2        ESP         124  ESP (SPI=0xc9a9d02b)
     6371 10988.605574    54.100.211.91     213.1.133.2        ESP         124  ESP (SPI=0xc9a9d02b)
     6372 10993.173554    54.100.211.91     213.1.133.2        ESP         140  ESP (SPI=0xc9a9d02b)
     6373 10993.173559    54.100.211.91     213.1.133.2        ESP         140  ESP (SPI=0xc9a9d02b)
     6374 10993.173935    213.1.133.2       54.100.211.91      ESP         124  ESP (SPI=0xc47caf46)
     6375 10993.173937    213.1.133.2       54.100.211.91      ESP         124  ESP (SPI=0xc47caf46)
  •  6382 10996.164233    54.100.211.91     213.1.133.2        ESP         140  ESP (SPI=0xc9a9d02b)
     6383 10996.164238    54.100.211.91     213.1.133.2        ESP         140  ESP (SPI=0xc9a9d02b)
     6384 10996.164966    213.1.133.2       54.100.211.91      ESP         124  ESP (SPI=0xc47caf46)
     6385 10996.164970    213.1.133.2       54.100.211.91      ESP         124  ESP (SPI=0xc47caf46)

▸ Frame 6383: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits)
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 54.100.211.91, Dst: 213.1.133.2
▸ Encapsulating Security Payload
```

Figure 3: Shows an Encapsulated Security Payload that uses the internet to access the internal network for mobusr3

Implementation	Reasoning	Evidence
Creating keys of specific size and validity period, depending on the machines functionality.	All of the keys generated are 4096 bits in length. The root CA has a certificate that is valid for 10 years where as all others have a validity period of 1 year.	build_x509.sh: # generate root CA key ipsec pki --gen --type rsa --size 4096 --outform pem > CA.key.pem # generate root certificate ipsec pki --self --ca --lifetime 3650 --in CA.key.pem --type rsa --dn "C=UK, O=University of Warwick - Cyber Security Centre, OU=ISS CW2, CN=CA" --outform pem > CA.crt.pem # gw certificate creation # generate key ipsec pki --gen --type rsa --size 4096 --outform pem > gw.key.pem # generate csr ipsec pki --pub --in gw.key.pem --type rsa --outform pem > gw.csr.pem # sign certificate with root key ipsec pki --issue --in gw.csr.pem --lifetime 365 --cacert CA.crt.pem --cakey CA.key.pem --dn "C=UK, O=University of Warwick - Cyber Security Centre, OU=ISS CW2, CN=gw.u1834961-u1824952.cyber.test" --san 213.1.133.2 --san @213.1.133.2 --san 192.168.65.1 --san @192.168.65.1 --flag serverAuth --flag ikeIntermediate --outform pem > gw.crt.pem

Creating a certificate revocation list (CRL) for users that are no longer permitted access to the internal network externally. For the purpose of demonstration, a rogue user has been added that has had their certificate adding to the CRL.	Within organisations such privileges are remotely accessing networks may need to be revoked (eg. change in occupation). A revocation list ensures the networks integrity by restricting access to those who were once granted entry. The list is added to the I_CA which will perform a check against to determine entry.	Within the build_x509.sh: ipsec pki --signcrl --carcert ica.crt.pem --cakey ica.key.pem --reason superseeded --cert rogueOne.crt.pem > ica.crl.pem
Creating a robust Apache web server.	Creating an Apache web server that has thought out features to ensure the organisations security.	www.u1834961-u1824952.cyber.test.startup a2enmod headers initiated to set some rules. In this case the ssl-params.conf file includes: Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains" which enables strict transport security so that browsers always use this site via https. Another thing added within the ssl-params.conf file is the decision to use only specific cypher suites. SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH Cypher suites chosen use a combination of Diffie Hellman key exchange and/or elliptical curve encryption. Also within ssl-params.conf is a configuration of what version of TLS and SSL to include only allow TLSv1.1 and

		TLSv1.2 (and TLSv1.3 on apache 2.4 e.g. not the netkit version) SSLProtocol All -SSLv2 -SSLv3 -TLSv1
Adding 1:1 NAT for the internal Apache server.	Mapping the internal address of the Apache webserver 192.168.65.10 to one of the organisations public IP addresses (213.1.133.3)	gw.u1834961-u1824952.cyber.test.startup iptables -t nat -A PREROUTING -i eth1 -d 213.1.133.3 -j DNAT --to-destination 192.168.65.10 iptables -t nat -A POSTROUTING -s 192.168.65.10 -j SNAT --to-source 213.1.133.3 iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to-source 213.1.133.2
Adding the root CA to *etc/ssl/certs*	Added our root CA to the *etc/ssl/certs* folder. This allows ssl connection without any trust issues. For example without this, curl throw up an error.	From the build_x509.sh: cp CA.crt.pem ../lab/shared/usr/local/share/ca-certificates/CA.crt Within shared.startup: update-ca-certificates