Report 2: Virtual Private Network Implementations

Shai Levin

Foreword

In this report we will discuss the common types of VPNs, go in depth into two specific types, and their suitable applications.

If a business requires remote access to secure networks, servers or clients across the internet, a Virtual Private Network may need to be utilised. Communication across the internet presents a vulnerability in which attackers can exploit. When traffic passes through the web, attackers can eavesdrop, stage man-in-the-middle attacks, and more. This risks data breach, or entry into secure networks. VPNs allow us to securely communicate between two trusted locations across the internet, by creating a 'bridge' or 'tunnel' of encrypted traffic from host to host, host to network, or network to network. In this report we will discuss the features of different VPNs in Section I, and implement OpenVPN and IKEv2 in Section II.

Section I: Overview

SSL vs IPsec

Most VPNs are based on either SSL or IPSec protocols:

A **SSL** based VPN operates at the Transport layer, using SSL or the more recent TLS encryption protocol. The protocol forms an encrypted tunnel between two ports on two hosts. The two hosts can interact as if they are directly attached. The source/destination ports and IP addresses are unencrypted and may be used by attackers to derive some information about the use case. It is typically simpler to implement, but provides more limited capability, and thus useful for smaller scale usage. In essence, it is more 'granular' than the alternatives. It is thus recommended for specific use cases where regular yet small interactions take place, or where speed and convenience is preferable.

IPSec provides a bridge connection from a remote host (or network) to a secure network. When the bridge is configured, the remote host behaves as if it is on the same local network. IPSec offers a range of cryptographic suites and operates at the internet layer (IP). As such it obscures data like ports and session numbers. Deployment requires specialized software whereas SSL does not. It provides the highest level of security but as a consequence requires more effort to configure and manage.

OpenVPN and IKEv2 & Others

Today, we typically implement one of the following VPNs.

OpenVPN is built on SSL, but adds multiple features, such as adding UDP support for increased speed, providing an internet layer bridge or TAP mode, and is open source. OpenVPN still uses SSL for authentication and set-up, but in operation it can vary substantially from traditional SSL based VPNs. It is considered slower and harder to configure than IKEv2, but much simpler. The biggest benefit of OpenVPN is that it is open source, allowing any vulnerabilities to be vetted by external security professionals.

IKEv2 is essentially an extension of IPSec with a faster session set up and more flexibility in cryptographic suites. It is more stable, and is well integrated with cross-platform support. It is much easier to configure than traditional IPSec VPNs, resolving it's biggest flaw. IKEv2 is proprietary, and it is rumoured that the NSA has compromised it. True or not, it highlights the primary reason why people might choose OpenVPN over IKEv2.

Other proprietary implementations of SSL VPNs exist, which are integrated into software packages such as mobile banking apps. This implementation is preferable when dealing with large distributed systems, a situation where an IPsec solution may prove unwieldy.

Authentication Suites

OpenVPN uses TLS for their authentication handshake. By default, suites used are RSA for public key pairs, and Blowfish for their symmetric key encryption, but can be configured to use any OpenSSL supported ciphers.

IKEv2 protocol handshakes are used in IKEv2 instead of the standard TLS handshake as in OpenVPN. The protocol is considered better and IKEv2 is faster when connections drop and need to be re-established. IKEv2 supports a manner of similar standard cipher suites as with TLS.

The more complex and larger the cipher we use, the longer it will take to generate keys and negotiate the VPN connection. Typically RSA 2048 bit for public keys, and AES 128 bit for symmetric key is recommended. This option is widely supported and offers good security. Note that some modern cipher suites will not work with older or simpler

hardware, so care is needed when selecting which cipher suites are allowed in the VPN configuration.

Section II: VPN Implementations

Network Configuration

In both implementations, we use a network configuration similar to the one described in Report 1, except that the firewall is configured to block all non-essential traffic between trusted and untrusted/DMZ zones. This facilitates the need for a VPN in order to pass through the firewall into the trusted network. The mobile client is an android phone which has already connected to the wireless network with a user/password set up similarly to that in Report 1.

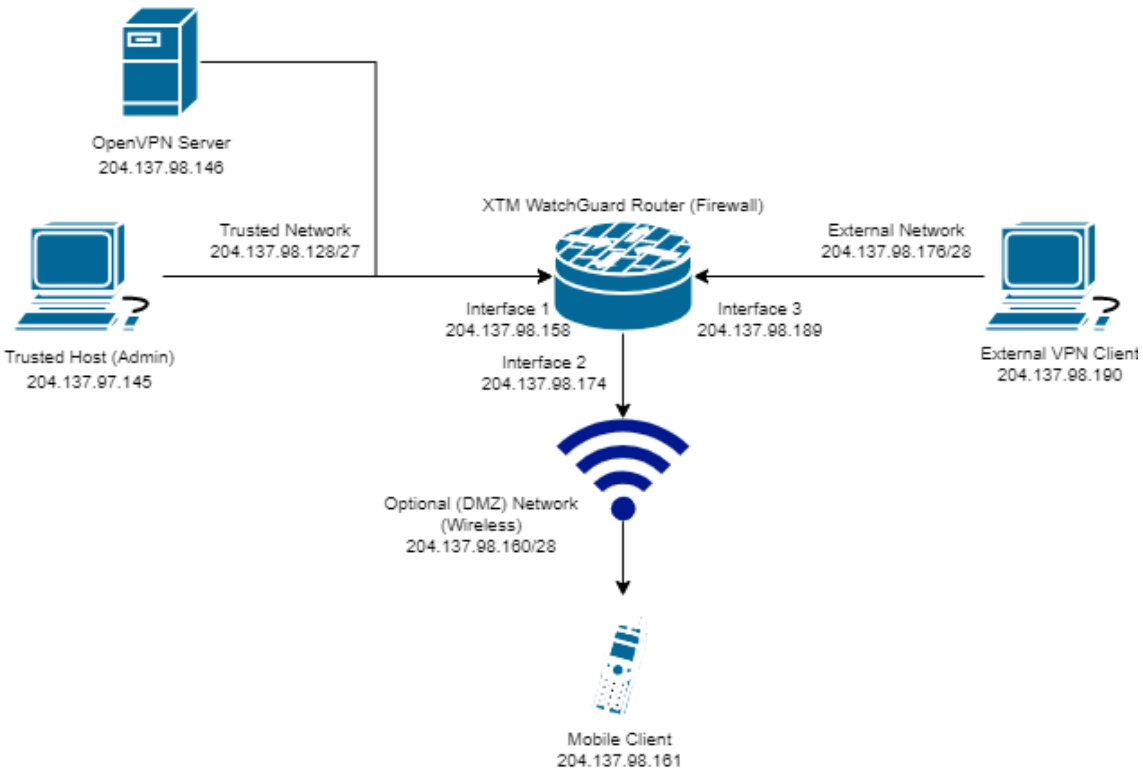
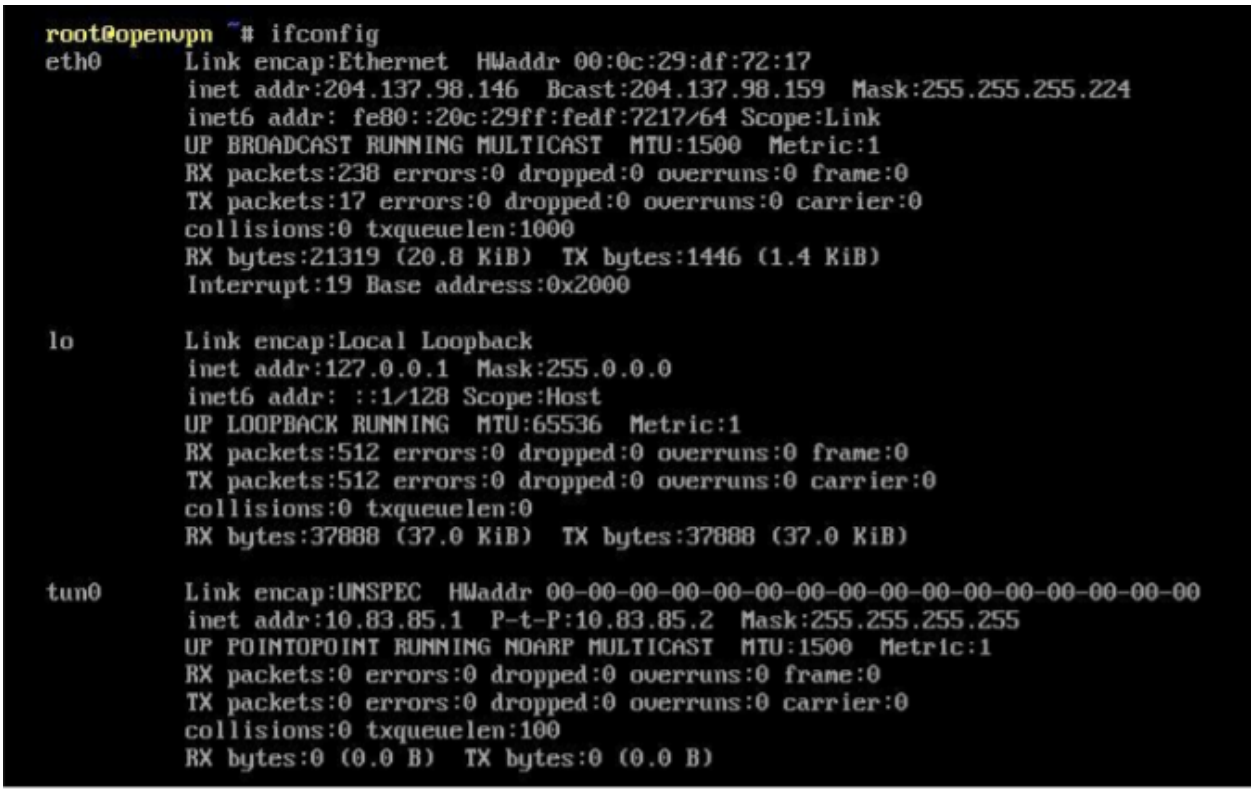


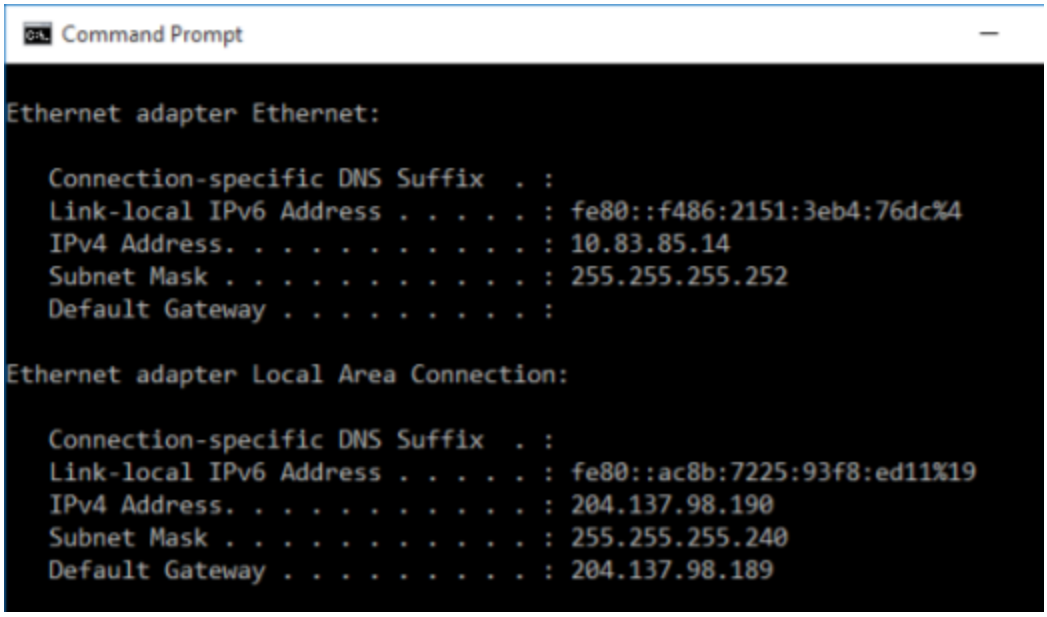
Figure 1

OpenVPN Configuration

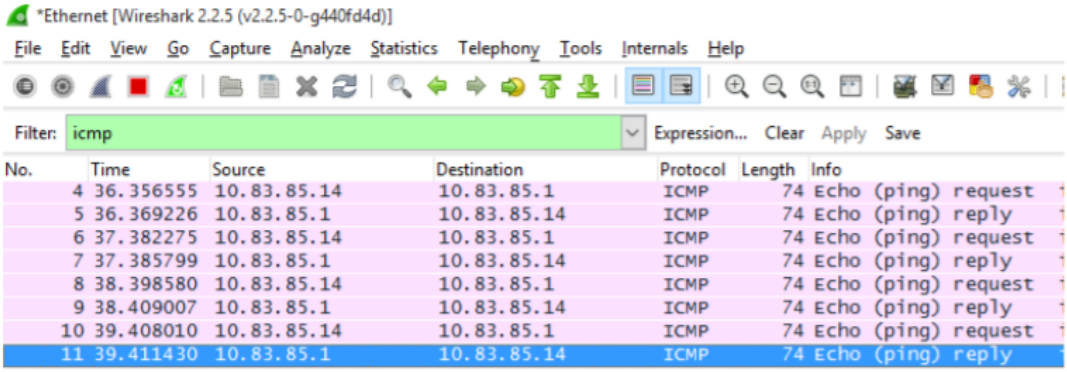
We have chosen to use OpenVPN connected to the OpenVPN server on the trusted net with UDP TAP mode (bridge mode). This will bridge either a remote client on the wireless network, or on the external network, to the trusted network. We use UDP mode because it is faster, but at a cost of less security. There are two modes for authentication - static key, and SSL/TLS certificate. Static key is the most convenient but is not as secure. We will choose the digital certificate option. We will use the default OpenVPN encryption suites.

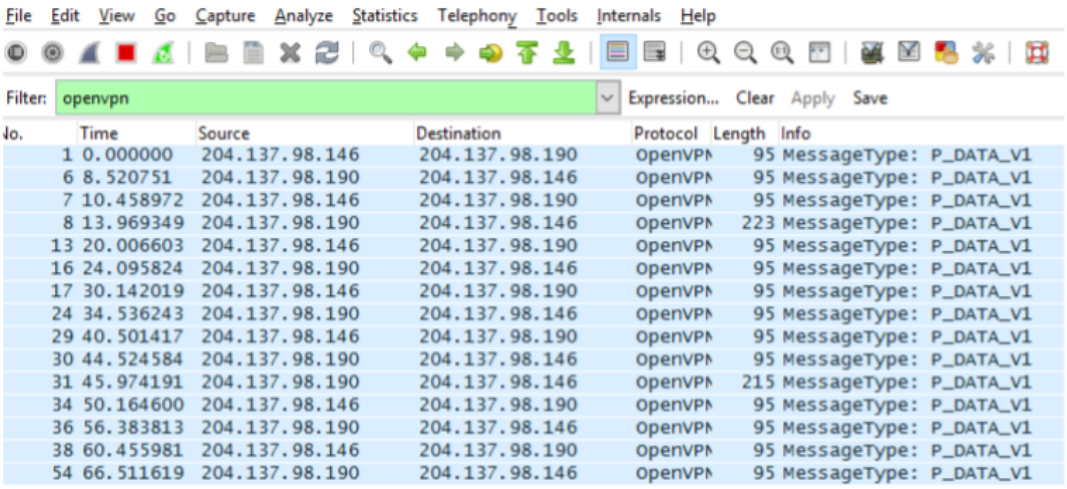


We load the OpenVPN.xml config onto the firewall to allow OpenVPN UDP traffic to pass through. The OpenVPN server will show a tunnel connection once started.
Now we will configure the client side. We are using certificate authentication. As such, we add a OpenVPN client which automatically generates a .ovpn signed digital certificate file. This is what the client will use to authenticate their VPN connection. We FTP the file onto the clients machine, but in practice a more secure method of transfer should be used. We import the .ovpn file onto our OpenVPN software on the client machine, and can then connect. Checking ipconfig on the machine, we can see a connection to the 10.83.85.XX subnet, suggesting that the tunnel has been established.

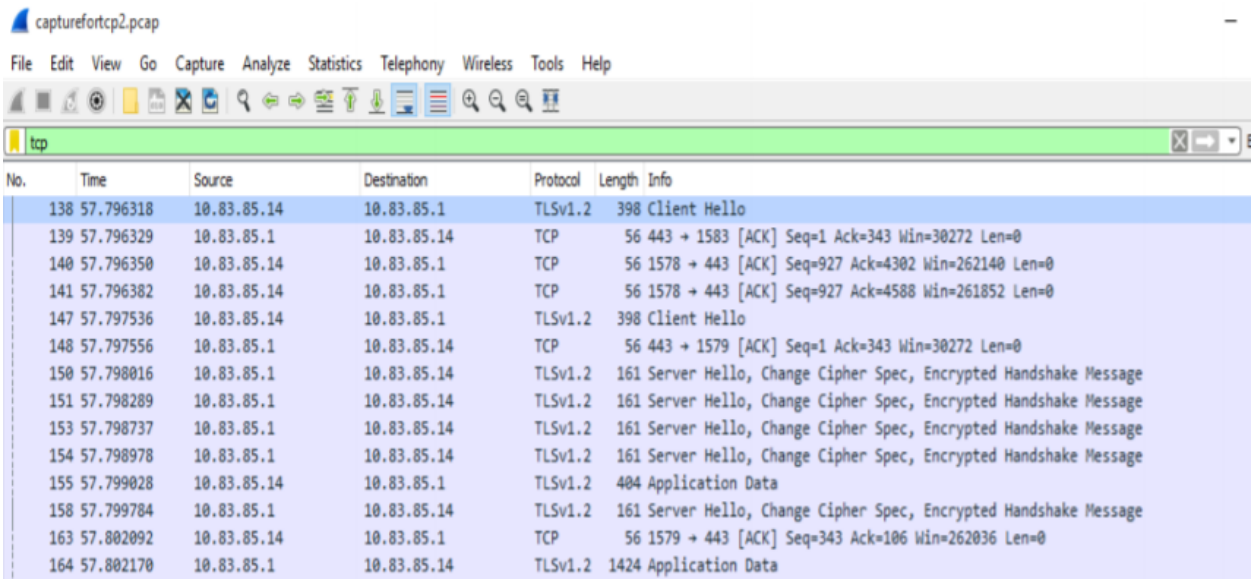


To test the tunnel has been established, we can now ping the servers tunnel address 10.83.85.1. We capture the traffic on wireshark both at the tunnel layer, and at the ethernet layer.





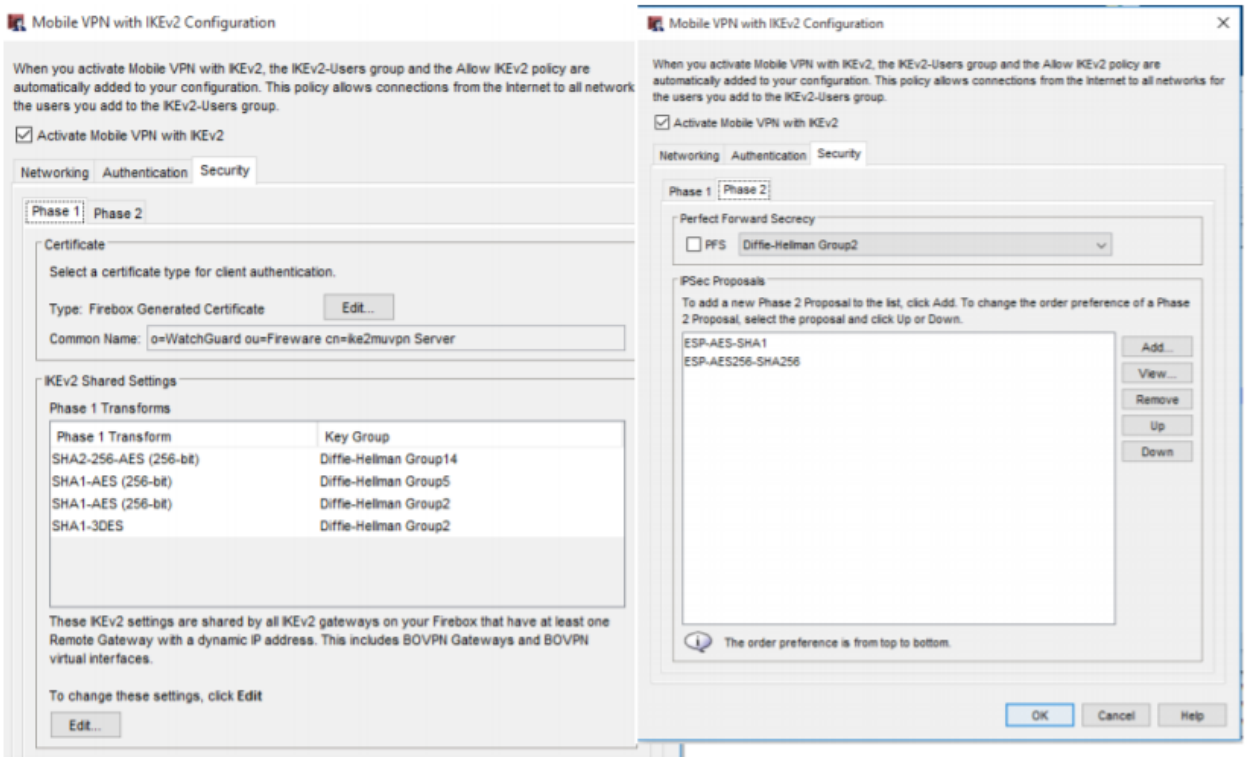
Note how the ping traffic is transmitted over the wire as OpenVPN protocol packets. This is our raw encrypted traffic sent as UDP packets, whereas the packets in the first case are unencrypted ICMP packets before encryption. Now, looking at the server side, here is a trace of the client/server negotiation stage. This follows a typical TLS handshake process, which is an extremely secure key exchange protocol in regular use over web traffic.



The tunnel has been established and we can now use the VPN for its intended purpose. Note connecting an android client over the optional wireless network follows a **very** similar process, with the .ovpn file and the OpenVPN app. Once we transfer the file, we simply import the file into the app using the import profile option.

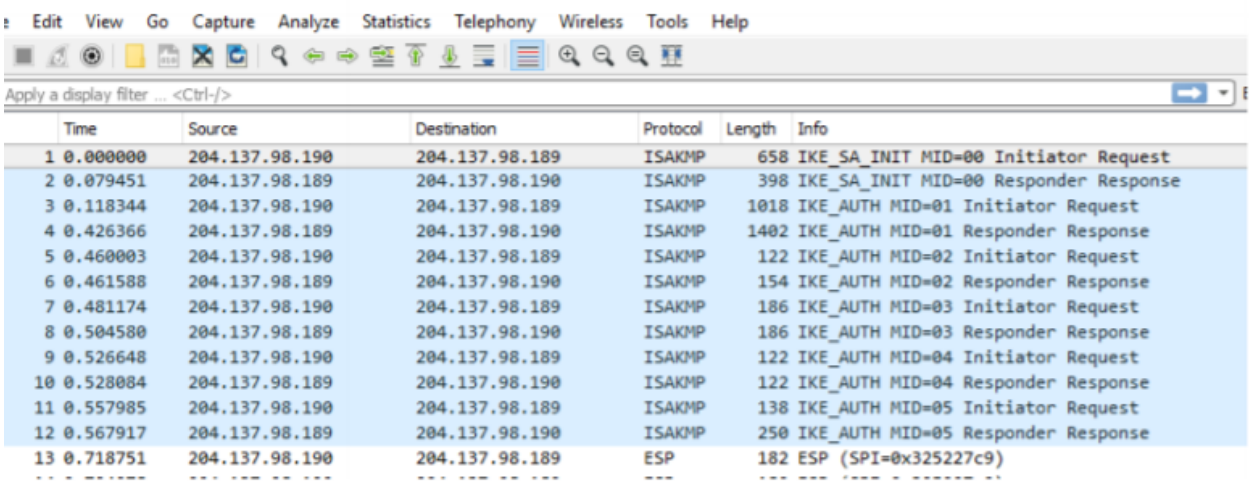
IKEv2 Configuration

Since IKEv2 is so widely supported natively on many platforms, it is in fact extremely convenient to set up in our network. In fact, our router natively supports Mobile IKEv2 usage so we do not need to use a back-end VPN server. It can be implemented similarly to in Report 1, either through the Firebox-DB or with a back-end RADIUS server. But we will be using certificate based authentication through the Firebox-DB in this case. We run through the Watchguard Mobile VPN with IKEv2 setup wizard, and choose an IP pool for the VPN tunnel. A separate IP pool is needed for wireless clients connecting in the optional network, and those connecting via the external network. We then configure the Mobile VPN. The security config is shown below, and like SSL, offers a wide range of cipher suites.



We are satisfied with the default config in this case. Adding a user for the remote client to the Firebox-DB is necessary to establish a connection. We then ensure the firewall is configured to allow IPSec traffic from external & optional networks to the Firebox router, and to allow IKEv2 users to connect to pass through the firewall to any locations through the router. We then download a client instruction file from the Policy Manager. The file is a zipped file required for any clients who wish to connect to the VPN. It contains all necessary config, install files and the digital certificate in one package. We transfer it to the client the same way as in the SSL implementation.

On our remote windows machine, we now unzip the .tgz file, and install the certificate to the OS which will authenticate our connection. We enter the windows settings and add a VPN connection. We enter the IP address of the server (the Firebox router) with relevant options and credentials. After connecting to the VPN, we may need to configure our routing to force traffic to prioritise transmitting traffic over the VPN tunnel rather than directly to the router. We add a default route to the VPN tunnel end-point (192.168.114.2) with a low metric to prioritize traffic over the default router gateway (204.137.98.189). Below is a wireshark trace of the handshake negotiation phase of the VPN connection.



	Time	Source	Destination	Protocol	Length	Info
1	0.000000	204.137.98.190	204.137.98.189	ISAKMP	658	IKE_SA_INIT MID=00 Initiator Request
2	0.079451	204.137.98.189	204.137.98.190	ISAKMP	398	IKE_SA_INIT MID=00 Responder Response
3	0.118344	204.137.98.190	204.137.98.189	ISAKMP	1018	IKE_AUTH MID=01 Initiator Request
4	0.426366	204.137.98.189	204.137.98.190	ISAKMP	1402	IKE_AUTH MID=01 Responder Response
5	0.460003	204.137.98.190	204.137.98.189	ISAKMP	122	IKE_AUTH MID=02 Initiator Request
6	0.461588	204.137.98.189	204.137.98.190	ISAKMP	154	IKE_AUTH MID=02 Responder Response
7	0.481174	204.137.98.190	204.137.98.189	ISAKMP	186	IKE_AUTH MID=03 Initiator Request
8	0.504580	204.137.98.189	204.137.98.190	ISAKMP	186	IKE_AUTH MID=03 Responder Response
9	0.526648	204.137.98.190	204.137.98.189	ISAKMP	122	IKE_AUTH MID=04 Initiator Request
10	0.528084	204.137.98.189	204.137.98.190	ISAKMP	122	IKE_AUTH MID=04 Responder Response
11	0.557985	204.137.98.190	204.137.98.189	ISAKMP	138	IKE_AUTH MID=05 Initiator Request
12	0.567917	204.137.98.189	204.137.98.190	ISAKMP	250	IKE_AUTH MID=05 Responder Response
13	0.718751	204.137.98.190	204.137.98.189	ESP	182	ESP (SPI=0x325227c9)

Below is a trace of some traffic between the tunnel. The IPSec tunnel transmits traffic in encrypted, authenticated packets.

No.	Time	Source	Destination	Protocol	Length	Info
22	3.727675	204.137.98.190	204.137.98.189	ESP	182	ESP (SPI=0x325227c9)
23	3.733202	204.137.98.190	204.137.98.189	ESP	182	ESP (SPI=0x325227c9)
24	3.734011	204.137.98.190	204.137.98.189	ESP	182	ESP (SPI=0x325227c9)
25	3.788262	204.137.98.190	204.137.98.189	ESP	406	ESP (SPI=0x325227c9)
29	3.859754	204.137.98.190	204.137.98.189	ESP	118	ESP (SPI=0x325227c9)
32	3.862544	204.137.98.190	204.137.98.189	ESP	118	ESP (SPI=0x325227c9)
34	3.864098	204.137.98.190	204.137.98.189	ESP	150	ESP (SPI=0x325227c9)
40	4.005044	204.137.98.190	204.137.98.189	ESP	150	ESP (SPI=0x325227c9)

Now android does not natively support IKEv2 VPNs, but we can use the StrongSwan app. We use the same .tgz file which contains a .sswan file, and the certificate. The certificate is installed by tapping on it, and the .sswan file is imported into the app. It follows a similar process to the windows client from this point.

Conclusion

While both VPNs investigated are very secure, the choice for which one you might use is up to the use case. OpenVPN is very easy to set up with a basic hardware configuration, whereas IKEv2 might prove easier in an enterprise grade network with for example, a Firebox Router. OpenVPN is open source but slightly slower in general compared to IKEv2 and is thus recommended in smaller networks. In a situation where you have huge networks and many remote clients connecting, an IKEv2 VPN setup may be preferable.

In both cases, digital certificates authentication is recommended when security is a priority, as they are very hard to forge and provide the client both confidentiality and authenticity.

Two examples and recommendations:

Case 1: A large company wishes to bridge two branches networks together with a VPN.
Recommendation: IKEv2

Case 2: A small startup who has workers that need to remotely access their machines from home.
Recommendation: OpenVPN (TUN mode)

References:

- IPSec vs SSL
  https://searchsecurity.techtarget.com/tip/IPSec-VPN-vs-SSL-VPN-Comparing-respective-VPN-security-risks
- NSA Leaked slides suggesting IKE is compromised
  https://edwardsnowden.com/2015/01/07/fielded-capability-end-to-end-vpn-spin-9-design-review/