# SSL Virtual Clean Room Security Architecture

## 1 Problem Statement

A local area network (LAN) is currently isolated from the public internet. Any access into this LAN is physical i.e. a user is to be physically located in the LAN site. It would be more *convenient* to be able to securely access various services on the LAN through trusted clients on the public internet. This document presents technologies that can be used to achieve this goal and outlines an end-to-end solution towards it.

## 2 Technology Overview

### 2.1 VPN

A virtual private network (VPN) extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network [1]. A VPN is created by establishing a virtual point-to-point connection through the use of dedicated connections, virtual tunneling protocols, or traffic encryption. From a user perspective, the resources available within the private network can be accessed remotely [1]. Traditional VPNs are characterized by a point-to-point topology, and they do not tend to support or connect broadcast domains. Designers have developed VPN variants, such as Virtual Private LAN Service (VPLS), and layer-2 tunneling protocols, to overcome this limitation.

**Best Practices**

1. Plan out a comprehensive network security policy. Factor in the following questions:
   a. What are the classes of users?
   b. What level of access is allowed to a class?
   c. Which devices are allowed to connect to the corporate network through a VPN?
   d. Which authentication method will be used and how will it be implemented?
   e. What is the maximum idle VPN connection time allowed before automatic termination?
   f. What level of training will be provided to users before they are allowed to access the VPN?
2. Ensure that **only** organization-issued hardware devices are able to connect to the network, with or without a VPN.
3. To ensure that no unauthorized software is able to install itself, or by a user, and cause a virus, worm, Trojan or malware infection on a device, each device must deny administrator rights to the user of that particular device or all the users in general. This ensures protection against Distributed Denial of Service (DDoS) attacks.
4. Another major security measure that must be adopted is the installation of antivirus and a firewall on all organization provided hardware. Malicious files are kept out by an anti-virus, while more direct hacking attempts are thwarted by a firewall.

We discuss these tools in greater detail in subsequent sections. In deciding to use a VPN based solution, organizations have 2 options [5]. We now discuss these in greater detail.

### 2.1.1 IPsec

IPsec is an IP packet authentication and encryption method. It uses cryptographic keys to protect data flows between hosts and security gateways. IPsec is used to connect a remote user to an *entire network*. This gives the user access to *all* IP based applications. The VPN gateway is located at the perimeter of the network, and the firewall too is setup right at the gateway. However, client software must be installed in order to achieve IPsec VPN access.

**Advantages:**

1. The unique feature of IPsec is that it operates at the Network Layer of the Open Systems Interconnection (OSI) protocol mode, providing a great deal of transparency to applications.

**Disadvantages:**

1. Once a computer is attached to the IPsec-based network, any vulnerabilities that exist at the IP layer in the remote network could be passed to the corporate network across the IPsec tunnel. Avoiding such issues are achievable at higher support costs.
2. It is becoming increasingly difficult to use the home Internet connection for corporate network access if using an IPsec-encrypted VPN tunnel. ISPs consider anything IPsec-encrypted to be a "business-class" transmission, and wish to charge higher rates for IPsec traffic. They might block IPsec traffic if the service type is not business class.

**Implications/Requirements:**

1. The user will only be able to access the network from a single, authorized device.
2. The (capable) networking department has the ability to configure each user’s hardware device individually (installing client software, enforcing security policies etc.).
3. For many use cases, XAUTH and L2TP methods of IPsec authentication are prone to security lapses. An Internet Key Exchange (IKE) [2, 3], or Kerberized Internet Negotiation of Keys (KINK) [4] is typically required as an authentication framework.

### 2.1.2 SSL

A Secure Sockets layer connection operates at the Transport Layer or Application Layer of the OSI Model of protocols. SSL VPN gateways are deployed behind the perimeter firewall, with rules which grant or deny access to specific applications. Thus, SSL provides “granular” access to the corporate network. The remote user is able to access only those applications which are relevant to his or her work, and is not able to access other areas of the network.

**Advantages:**

1. Web SSL VPNs are simpler to setup and troubleshoot.
2. SL VPNs cost less than traditional IPsec VPNs. They do not require proprietary VPN client software to be purchased or licensed (in most cases).
3. SSL makes use of Port 443. This almost guarantees it will work through any firewall that provides standard Internet access, without the need of any special configuration.
4. Web SSL VPNs are compatible with all operating systems and Web browsers. Web SSL VPNs have full IP application support -- replacing IPsec VPN client programs completely.
5. Web SSL VPNs are available on servers, firewalls and even routers. One doesn’t necessarily need a dedicated machine only for VPN users as it is supported even on small devices such as Cisco 870 series routers.

**Disadvantages:**

1. SSL Tunneling is not supported on Linux or non-Windows OS.
2. SSL is processor-intensive leading to poor performance under high loads. However, this can be addressed by clustering, and load-balancing multiple appliances, or through traffic prioritization.
3. Some enterprises need broader application support than SSL provides.

## 2.2 Secure Shell

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. SSH is important in cloud computing to solve connectivity problems, avoiding the security issues of exposing a cloud-based virtual machine directly on the Internet. An SSH tunnel can provide a secure path over the Internet, through a firewall to a virtual machine. SSH operates using the Trust On First Use (TOFU), or Trust Upon First Use (TUFU) security model, where client software needs to establish a trust relationship with an unknown or not-yet-trusted endpoint.

SSH uses public-key cryptography to authenticate the remote computer and allow it to authenticate the user. There are several ways to use SSH:

1. Automatically generated public-private key pairs to simply encrypt a network connection, and then use password authentication to log on.
2. Manually generate public-private key pair to perform the authentication, allowing users or programs to log in without having to specify a password i.e. in such a case, password authentication is disabled.

SSH also supports password-based authentication that is encrypted by automatically generated keys.

In this case the attacker could imitate the legitimate server side, ask for the password, and obtain it (man-in-the-middle attack). However, this is possible only if the two sides have never authenticated before, as SSH remembers the key that the server side previously used. The SSH client raises a warning before accepting the key of a new, previously unknown server.

To avoid the issue stemming from TOFU, administrators can generate the public-private key pair and physically place it in the servers under consideration. Though cumbersome, this process is more secure as SSH only verifies whether the same person offering the public key also owns the matching private key. It is **important** to note that one must verify unknown public keys, i.e. associate the public keys with identities, before accepting them as valid.

SSH is typically used to log into a remote machine and execute commands, but it also supports tunneling, forwarding TCP ports and X11 connections; it can transfer files using the associated SSH file transfer (SFTP) or secure copy (SCP) protocols. The standard TCP port 22 has been assigned for contacting SSH servers. However, Windows is one of the few modern desktop/server OSs that does not include SSH by default.

## Best Practices:

1. Use public key authentication. Passwords are a constant source of security problems, and most intrusions can, at some point, be linked to the cracking or stealing of a password. Passwords are: (i) relatively easy to crack, (ii) sniffable, using cleverly designed MitM attacks, or by installing a backdoor in the SSH server, and (iii) reused, and poorly managed. Keys are, however, (i) hard to crack (assuming the right key size is chosen), (ii) un-sniffable as signatures (and not the keys themselves) are used as part of the authentication protocol, and (iii) can be properly managed, with an authentication agent and passphrase encryption.
2. Do not create DSA keys, this key type is insecure and obsolete. Use elliptic curve keys instead.
3. Encrypt and store the private key. There are well known attacks that are targeted at extracting clear passwords.
4. Ensure that you are indeed communicating with the target server. Unlike SSL/TLS that uses an authority-based model, SSH uses a public key pinning/TOFU at first use model.
5. Use a different key for every computer.
6. Use SSH certificates when needed. Individual keys are difficult to manage on big installations, and it can be used to replace the key-pinning model of SSH with a more classical CA-based model.
7. Use a hardware token as part of multi-factor authentication in SSH.

# 3 Additional Factors

## 3.1 Firewalls

A firewall is a network security system, either hardware- or software-based, that uses rules to control incoming and outgoing network traffic.

**Best practices:**

1. Document all firewall rule changes. This makes troubleshooting and overall policy management much easier and more efficient.
2. Install all access rules with minimal access rights. A rule where the service field is 'ANY' opens up 65,535 TCP ports i.e. 65,535 attack vectors.
3. Verify every firewall change against compliance policies and change requests.
4. Remove unused rules from the firewall rule bases when services are decommissioned.
5. Perform a complete firewall review at least twice per year.
6. Many organizations prefer to deny all traffic and permit only that traffic that is necessary, a security model known as Deny All Permit Exception (DAPE). This is a more secure posture than using a blacklist. However, upkeep on a firewall policy with a **whitelist** is more labor-intensive because you need to keep adding to the firewall whitelist whenever the communication does not fit the existing set of rules.

## 3.2 Authentication Primitives:

1. **Multi-factor authentication**: Multi-factor authentication (MFA) is a method of computer access control in which a user is granted access only after successfully presenting several separate pieces of evidence to an authentication mechanism. As stated earlier, this can be achieved using dedicated hardware tokens, or the more common mechanism of offline verification through text message, or email i.e. 2FA.
2. **Password strength**: Important factors to remember while creating a password are (i) the password must be long and complex, and hard to guess. Typically, the password must include a string of randomness, and must not comprise of coherent sequence of words thereby increasing the entropy of the string, (ii) remembering to think multiple times before sharing the password with other parties, and (iii) avoid reusing passwords across accounts.
3. **Limit sign-in attempts**: In the scenario where passwords are preferred (over public keys), an adversary can attempt to brute force the system by trying all combinations. Thus, it is important to cap the number of login attempts provided to any participant, or include tasks such as pattern or image recognition when the cap is reached. Though these types of recognition problems can be solved using neural networks, image poisoning can be used to steady the defense.
4. **Public Key Authentication:** An alternative to password-based authentication is public key authentication. The server has access to the public key while the client solves a challenge using its private key. Public key authentication has already been discussed for SSH. SSL/TLS also provides for public key authentication by what are known as client

certificates. In case authentication is enabled, there is an additional step in the SSL/TLS exchange whereby the server verifies the identity of the client using the client certificate. For VPNs, it is common that the VPN server itself acts as the certificate authority (CA) and an administrator provides the clients with client certificates generated by this CA.

## 3.3 Logging

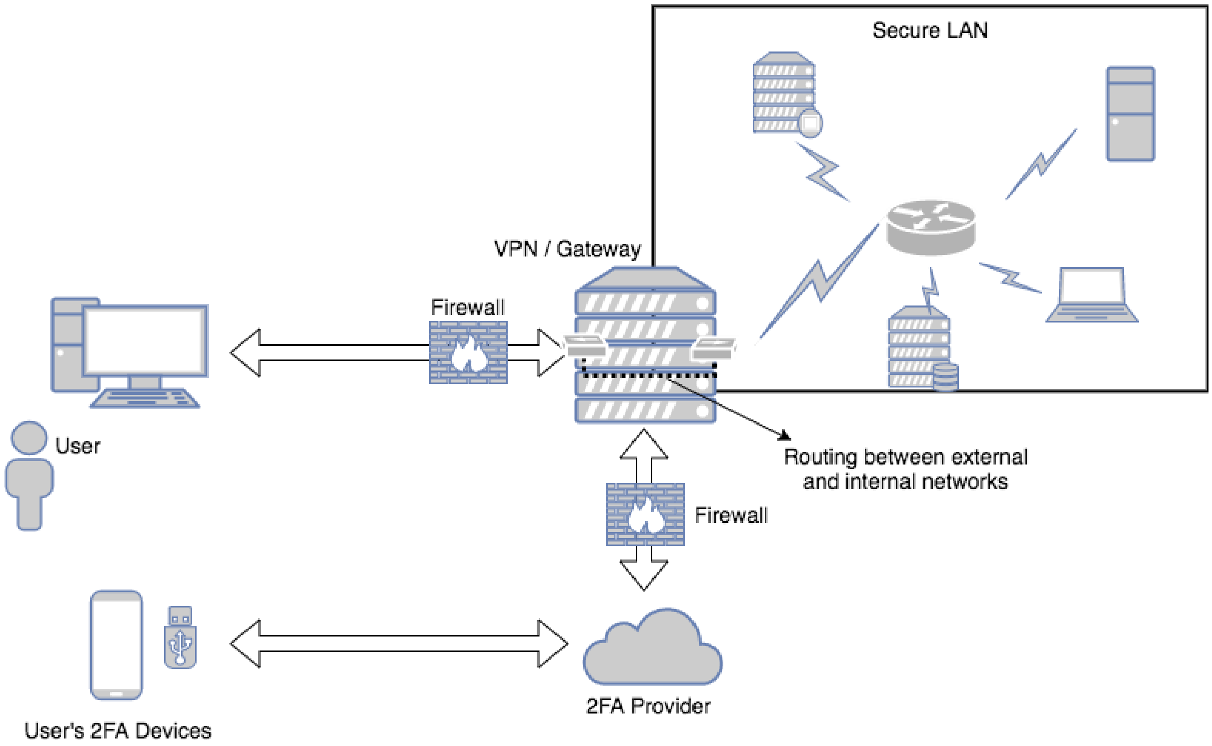
Logging events that transpire at the ingress and egress points of any connection is crucial in maintaining security. Knowledge of the precise sequence of activities that affect a specific operation, procedure, or event is very valuable. Logging creates an "audit trail"—a security-relevant chronological record that documents an organization's digital footsteps. Keeping detailed records of daily activities allows further visibility into users' actions. For example, logs act as a detective control because their trails provide evidence if a hacker or user engages in unauthorized activity. Having a log of user activities helps organizations remain organized and also helps when dealing with unforeseen circumstances, including security violations, performance problems, and system flaws.

Best practices associated with logging include:

1. Create events that humans can read and understand.
2. Use timestamps for **every** event.
3. Use unique identifiers.
4. Log in text format.
5. Log more than just debugging events.
6. Identify the source of the event being logged.
7. Log locally to files.
8. Use rotation policies - logs often occupy a lot of space.
9. Collect logs from:
    a. Applications
    b. Databases
    c. Networks
    d. Configuration files
    e. Performance data

# 4 End-to-end Solution

We now outline a solution based on the above technologies. The architecture is presented in the figure below. In this architecture, we use *OpenVPN* on Linux for providing SSL-based VPN, use certificate authentication for clients (there is no possibility to authenticate with passwords), and use *Duo* for second factor in authentication (other VPN servers and 2FA providers are possible with minor changes). *iptables* is used as a firewall to restrict access to the VPN to specific subnets, to prevent incoming and outgoing connections in the secure network, and to provide forwarding between internal (secure LAN) and external (public internet) networks.



The VPN server is an SSL-type and operates its own certificate authority. It generates a self-signed certificate for itself and certificates for clients. It is recommended that elliptic curve or at least 2048-bit RSA keys be used for certificates. These certificates are shared with clients (user devices) through a secure channel. A client authenticates with the server using its client certificate. For OpenVPN, such a configuration is documented well on the Internet, e.g., [6].

For greater security, we recommend enabling 2-factor authentication. The second factor could be a phone call, SMS message, hardware token supporting one-time password protocols (OTP) or U2F, interaction with a smartphone application, and so forth. We suggest Duo Security that provides easy solutions that support a vast majority of 2-factor authentication mechanisms and easily integrate with OpenVPN [7]. It is however possible to use other solutions such as *Yubikey's* PAM-based authentication [8]. In most solutions, an administrator is needed to add trusted 2FA devices for a user.

Traffic routing between the Internet and the secure LAN takes place by setting up appropriate forwarding rules between the two interfaces in both Iptables and the routing table. Helpful documentation is available for OpenVPN [9]. There are two possible ways making this connection: bridging and routing. Bridging works at layer 2 and thus can support non-IP network protocols. If this is not necessary, routing is an easier and more appropriate option. More details are provided in [9].

iptables also provides a general firewalling of the secure LAN. Except for the VPN/Gateway server, no inbound/outbound traffic is allowed from the LAN. Moreover, the VPN is allowed to accept incoming connections at the VPN's TLS port (e.g., 443) and is allowed to initiate outgoing connections to the 2FA provider.

Finally, it is useful to keep log of all VPN accesses to help post-incident investigation. OpenVPN logs typically go /var/log/syslog and can easily be directed to a remote syslog server.

Through this architecture, observe that:

1. The firewall allows configuration of :
   a. Inbound/Outbound IP addresses.
   b. Port Blocking/Unblocking.
2. The logging framework setup on each node (both in internal and cloud) facilitates logging the required system parameters.

## References:

[1] Virtual Private Networking: An Overview - https://technet.microsoft.com/enus/library/bb742566.aspx
[2] Internet Key Exchange - http://www.ciscopress.com/articles/article.asp?p=25474
[3] IPsec & IKE - https://sc1.checkpoint.com/documents/R77/CP_R77_VPN_AdminGuide/13847.htm
[4] Kerberized Internet Negotiation of Keys - https://tools.ietf.org/html/rfc4430
[5] http://searchsecurity.techtarget.com/feature/Tunnel-vision-Choosing-a-VPN-SSL-VPN-vs-IPsec-VPN
[6] https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-ubuntu-14-04
[7] https://duo.com/docs/openvpn
[8] https://developers.yubico.com/yubico-pam/YubiKey_and_OpenVPN_via_PAM.html
[9] https://community.openvpn.net/openvpn/wiki/BridgingAndRouting
[9] http://www.rsyslog.com/

-