Journal of Theoretical and Applied Information Technology

www.jatit.org

ON THE SECURITY OF FIREWALL POLICY DEPLOYMENT

[1]A. KARTIT, [2]M. EL MARRAKI, [3]A. RADI, [4]B. REGRAGUI

[1, 2]Département Informatique, Faculté des Sciences Rabat, FSR, Rabat, Morocco
[3, 4]Département Physique, FSR, Rabat, Morocco

ABSTRACT

Due to the sensitive nature of information transmitted during a policy deployment, the communication between management tool and firewall should be confidential.
Confidentiality can be achieved by using encrypted communication protocols such as SSH, SSL and IPSec. Much research has already addressed the specification of policies, conflict detection and optimization, but very little research is devoted to the security of policy deployment. In the previous article [1], we proposed an exact algorithm for the deployment of security policies but in this paper, we will propose an effective solution that will allow us to secure the deployment process of a political target. This solution has the objective to create secure tunnels between the different entities (firewalls).

Keywords: *Security Protocol (SP), Securing Exchanges (SE), Security of Policy Deployment (SPD), Authentication Protocol (AP).*

1. INTRODUCTION

Computer security is the set of measures implemented to reduce vulnerability of a system against accidental or intentional threats. The security covers so many aspects:

- Integrity of information (no modification or destruction).
- Confidentiality (no disclosure to unauthorized parties).
- Authentication of parties (signature).

A firewall is a system or system group that manages and represents an access control policy defined by network administrators. A firewall is always on the single link (or at least a link in traffic routes) that connects a private network (intranet) to the rest of the world (Internet) to be able to filter all traffic leaving and returning. Generally, firewalls are configured to protect against unauthenticated access the external network. They ensure, among other things, a filtering function at different levels of the OSI layer and prevent intruders to log on machines of the internal network. Packet filter, you can look at higher-level protocols, or watch the transported data or verify that they do not contain viruses or does not contain a particular word for example. Here is a diagram illustrating a conventional architecture of security using a firewall:

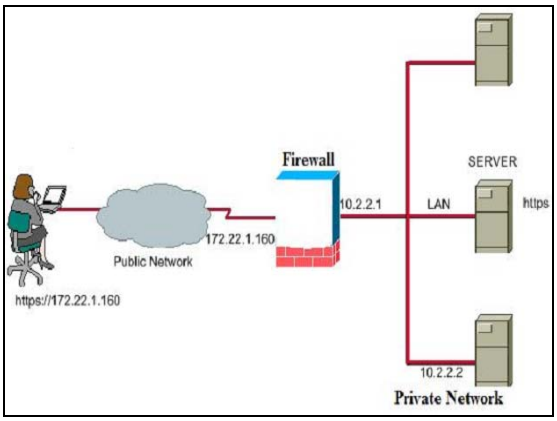

FIGURE 1: FIREWALL-ARCHITECTURE

However, this system firewall is insufficient if not accompanied by other protections. Indeed, it does not provide security services outlined above (Authentication of the source data, integrity and confidentiality) [2]. Several types of passive and active attacks are not protected (traffic analysis, IP spoofing, IP Flooding …) Vulnerabilities or configuration tools for automatic analysis of system vulnerabilities. With this in mind we thought to implement other security protocols within the firewall (SSL, SSH) to ensure the security of data exchange with other firewalls and especially the Security of Firewall Policy Deployment.

2. SECURITY MECHANISMS

There are four generally accepted security properties that one may require when establishing a secure channel between the user and the server:

- ✓ User/server authentication: before sending sensitive information over the Internet, the user should be assured that they are communicating with the right server; the server should also be able to verify the identity of the user before processing the requested transactions.
- ✓ Confidentiality: only the authorized entities (i.e. the user and the server) should have access to the content of the messages being exchanged.
- ✓ Data integrity: the user and the server should be able to detect any manipulation (including insertion, deletion and substitution) or replay of data by unauthorized parties.
- ✓ Non-repudiation: neither the user nor the server should be able to deny previous commitments or actions; for instance, in case of disputes, the administrator should be able to prove to a third party that the user has performed certain transactions.

Nowadays, different protocols are implemented in the field of security exchanges. There are SSL / TLS (Secure Socket Layer / Transport Layer Security) {[3],[4]} that provides the same services to all Internet applications. However, it does not provide the services of non-repudiation and access control. In the same area, we can cite the SSH (Secure Shell) [5] which provides a secure, interactive session between a client (user or machine) and a remote (server). This protocol does not currently support key distribution by a trusted third party. IPSec can make confidential the contents of packets transmitted by the protocol. IPSec provides confidentiality services, and authentication level data transferred by the IP protocol, via the establishment of the header authentication extension.

3. ANALYSIS OF EXISTING SECURITY SOLUTIONS

3.1 Protocol SSH (Secure Shell)

3.1.1 Introduction

SSH Secure Shell is developed in 1995 by Tatun Ylönen a Finnish professor. The first version of the protocol, SSH-1 [6], was designed to secure communications to remote Unix server, especially remote controllers (rsh, rlogin ...).

Because of several security vulnerabilities in the first version of this protocol [7], the IETF established a working group called SECSH (Secure Shell) protocol to standardize and guide its development in the public interest. The group submitted a set of Drafts detailing a decisive new version of SSH 2.0 (or SSH-2) {[5], [8], [9]}. This version contains new algorithms and services such as file transfer (SFTP) and the "tunneling" or "port forwarding protocols. In the following, we only describe the version 2 of the protocol.

3.1.2 Architecture

SSH uses client/server architecture to provide authentication, encryption and integrity of data transmitted in a network. Version 2 of the protocol specifies an architecture divided into three sub-protocols working together. **(See Figure 2)**

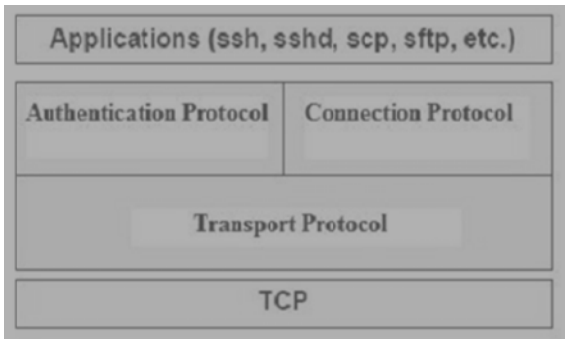

FIGURE 2: SSH-2 ARCHITECTURE

In the initialization phase of the SSH protocol, many important points are negotiated. In the case of using the TCP / IP network, the following procedure takes place between the client and the server machine. **(See Figure 3)**

The client and server negotiate automatically configure each channel, depending on the type of service requested by the client and the connection mode the user to the network. This allows to easily managing different types of remote connections without changing the basic infrastructure of the protocol. (**See Figure 4)**

3.1.3 Advantages

SSH is a powerful and practical approach to secure communications on a network of computers. Through its mechanism of authentication, SSH allows for a tunnel secure remote logins, file transfers, transfer of TCP / IP ports and other important features.

3.1.4 Disadvantages

SSH is able to bypass many security threats related to the system.

However, it is vulnerable to denial of service attacks, inheriting the weaknesses of TCP / IP on which it rests. In addition, and depending on the environment, SSH is susceptible to certain methods of attack, such as analysis and the diversion of traffic.

3.2 Protocol SSL/TLS

3.2.1 Introduction

The SSL (Secure Socket Layer) is originally developed by Netscape in 1996 [3]. In 1999, the first version of the protocol standard has emerged in the IETF under the name TLS (Transport Layer Security). TLS3 corresponds to version 3.1 [4] SSL.
SSL/TLS is a modular protocol whose goal is to secure Internet transactions between the client and the server independently of any type of application. Indeed, SSL/TLS acts as an extra layer above TCP, thus ensuring the services of authentication, confidentiality and integrity.

3.2.2 Architecture

SSL/TLS is designed to use TCP to provide a secure end to end reliable. SSL/TLS is not a simple protocol, but rather consists of two layers of protocol, as shown in **Figure 5**.

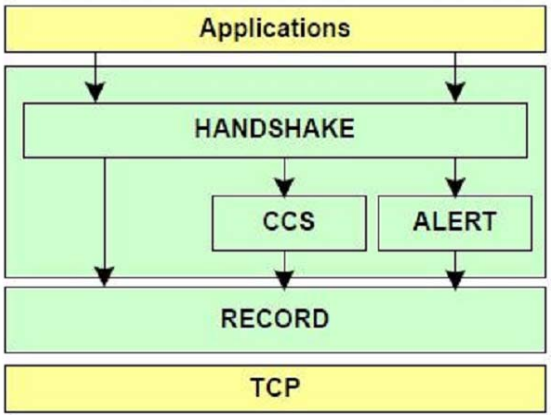

FIGURE 5: SSL/TLS ARCHITECTURE

3.2.3 Handshake Protocol

The handshake protocol is the most complex part of SSL/TLS. It allows the exchange of security parameters (random numbers, list of encryption algorithms, hash, etc...) Between the client and server before the application data is transmitted. It also allows the authentication of both communicators. However, in most cases, only the server is authenticated.
This protocol can operate in two ways: either it provides a complete exchange with negotiation of security settings (the handshake complete), or it tries to use an old session SSL/TLS already negotiated (abbreviated handshake).
(See Figure 6)

3.1.3 Advantages

SSL/TLS is a security protocol that enables secure exchange between a client and a server SSL/TLS in a transparent manner, placing himself between the application layer and transport. Main advantages of this protocol are:

- ✓ Strong authentication and integrity of messages.
- ✓ Interoperability and ease of deployment
- ✓ Usability

3.2.4 Disadvantages

SSL/TLS, like any other innovative technology in the security domain has its share of disadvantages:

- ✓ Cryptographic problem (related to the implementation)
- ✓ Attacks linked to the network
- ✓ Use of weak encryption keys

3.3 Protocol IPSec

3.3.1 Introduction

IPSec is a set of protocols designed by IETF to secure IP traffic.
Originally conceived in the philosophy of IPv6, IPSec is an encapsulation system offering security services required at the IP level. However, since the needs of security on the Internet and intranets can not wait until all (or at least a majority) of the computer world has migrated to IPv6, it is necessary that IPSec is also used IPv4. A convenient way to proceed, which has the advantage of ensuring compatibility with all existing IP implementations without having to modify, is regarded as an IPSec protocol independent, implementable as an add form of a software or equipment dedicated.

3.3.2 Architecture

[10] describes the IPSec protocol to the highest level. In particular, it states that an implementation is supposed to allow configuration in terms of security policy (i.e. what exchanges IP must be protected by IPSec and if so, what (s) protocol (s) use). On each system capable of using IPSec, this must be an SPD (Security Policy Database), whose precise form is at the discretion of the implementation, which allows to specify the security policy applied to the system. Each entry in this database is identified by an SPI (Security Parameter Index) single (32 bit).
A secure communication using IPSec is called an SA (Security Association). A SA based on a single

application of AH or ESP a single application. This does not preclude the simultaneous use of AH and ESP between two systems, or for example the encapsulation of datagrams in other datagrams HA HA, but several SA will then be activated between the two systems. In addition, an SA is unidirectional. The protection of a communication taking place in both directions, therefore, requires the activation of two SA. Each SA is uniquely identified by an SPI, a destination IP address (ip address of the multicast or broadcast) and a protocol (AH or ESP). Active SAs are grouped in a SAD (Security Association Database). The SPD is consulted during the processing of any IP datagram, incoming or outgoing, including the non-IPSec packets. For each datagram, three behaviors are possible: reject, accept without IPSec processing, or apply IPSec. In the third case, the SPD also specifies what treatments applied IPSec (ESP, AH, tunnel mode or transport, what (s) algorithm (s) signing and / or use encryption).

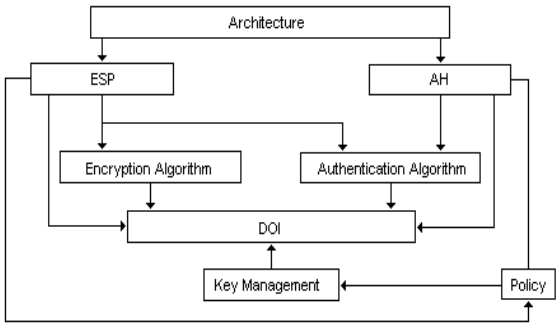

FIGURE 7: IPSEC ARCHITECTURE

3.3.3 The services offered by IPSec

A communication between two hosts, protected by IPSec, is likely to operate in two modes: transport mode and tunnel mode.
The first mode provides protection primarily for upper layer protocols, the second allows for him to encapsulate IP datagrams within IP datagrams other content which is protected. The major advantage of this second mode is that it treats all part of an IPSec communication and transmits datagrams purged their party IPSec their true destination feasible. It is also possible to encapsulate communication IPSec tunnel mode, which is treated by a security gateway, which transmits packets after striking their first envelope to a host processing to turn the remaining protections or a second bridge Security.

3.3.4 Key management in IPSec

The presence of asymmetric encryption mechanisms requires taking into account issues of key management and distribution to all systems to be the source and / or destination of an IPSec communication.
In the case of small infrastructure, it is possible, even preferable to opt for management and manual distribution of keys. This simple technique should however be coupled with a good organization and a certain rigor to first, to ensure proper operation and secondly to meet the criteria rekey. However, this method is not applicable on many systems including infrastructure, for obvious reasons, this volume of data to process and timelines for implementation. IKE (Internet Key Exchange) [11] has therefore been set as the default protocol for the management and key distribution. Key management is provided by the ISAKMP protocol [12], the principle of exchange provided by a mechanism derived from Oakley [13].

3.3.5 Advantages

The fact that IPSec is a security protocol at the network level, it allows a very high level of security compared to other solutions application layers. [14] and [15] list the advantages of IPSec and the present as follows:

- ✓ Protection against circumvention and traffic analysis
- ✓ Protocol transparent to applications
- ✓ Implementation of a VPN

3.3.6 Disadvantages

The problems with IPSec are related to the ambiguity documentary, implementation and redundancy features [16].

- ✓ Interoperability between implementations
- ✓ Redundancy features
- ✓ Broadcast and multicast

4. COMPARISON BETWEEN DIFFERENT SOLUTIONS

In this section, we compare the three security protocols already studied in relation to the requirements defined in the preceding chapter.

4.1 The choice of cryptographic mechanisms

Three security protocols are similar in their initialization phase and mechanism for protecting data.
For the initialization phase of IPSec and SSH are based on the negotiation of a DH group for the generation of shared keys. SSL/TLS, even if it supports a DH exchange, most of its negotiations are based on asymmetric encryption with the public key server.
For the protection of data, the three protocols use cryptographic function symmetric and hashing to

secure the services of confidentiality and integrity of data. For many reasons (political, former specification, etc...), three protocols begin negotiating with low-cost cryptographic algorithms such as DES encryption (56 bit) and MD5 hash.

4.2 Authentication

Authentication is a service that differs depending on the level of protection that we wish to establish. With IPSec, authentication of users is mainly related to network equipment such that machinery users and routers.

Both SSH and SSL/TLS are protocols for exchange. They authenticate only the two ends of the communication (client, server). For the protocol SSL/TLS, it provides a single method of authentication based on certificatesX.509 identity. However, SSH gives customers a choice between several methods. Authentication negotiated through a secure tunnel. To do this, even the authentication methods that appear for the first time non-secure (password), can be used securely with SSH.

4.3 Integrity and confidentiality

These two services are the foundation of any security protocol. Unlike the two other protocols (SSL/TLS and SSH), integrity and confidentiality in IPSec negotiated. They can also be used jointly or separately according protection mechanism negotiated (AH or ESP).

With SSL/TLS and SSH services of integrity and confidentiality are explicit. The messages from the initialization phase of these two protocols are protected by integrity and data applications are encrypted and integrity protected.

4.4 Protection against active and passive attacks

The three protocols studied in the previous chapter provide protection mechanisms for initialization phases and the phases of data protection.

If the integrity service is activated (if SSH, SSL, IPSec AH), the traffic is protected against any changes made by malicious third.

If the confidentiality service is activated (if SSH, SSL, IPSec ESP), the Traffic is protected against attack by intelligence information and analysis traffic.

If the service is not triggered replay (if SSH, SSL, IPSec-ESP, AH, ISAKMP), the data sent can not be replayed after a time.

If the service of protection against denial of service is activated (if IPSec-AHESP), traffic is protected against the "IP, TCP and UDP flooding ".

4.5 The protection of exchange or network protection

Among the three protocols studied in the previous chapter, IPSec is the only protocol that provides security at the IP traffic. It can thus ensure the security of all IP-based applications. The SSH and SSL/TLS provide security for all types of traffic flowing over the TCP transport protocol.

4.6 The trust between communicators

The trust between communicators is obtained with the authentication methods negotiated between the players. Both IPSec and SSL/TLS using the trusted authority (PKI) to provide a strong authentication service based on X.509 identity certificates. For the SSH protocol, several studies have integrated its authentication mechanism based on public/private key RSA in a trust infrastructure [17].

5. THE SOLUTION ADOPTED

After having compared the three security protocols, it was found that the solution based on VPN over SSH is best suited for the deployment of policies because it is easy to setup, need non-administrative access and work reliably.

To most users SSH appears to be terminal emulator similar to Telnet. The users do not see the encryption and therefore the security is transparent for the user. For system administrators SSH is a popular remote administration platform.

6. IMPLEMENTING THE CHOSEN SOLUTION (VPN over SSH)

6.1 Creating a VPN tunnel

Here is the sequence of commands that we have entered in our command line:

```
pppd debug updetach noauth \
pty "ssh -l login -t -t @distante \
pppd noauth 200.100.254.254:200.100.253.253"
```

In the first command line, “noauth” request that pppd does not care about the authentication part. This return to SSH.

In the second command line, “pty” option allows here to pass the following commands to the remote shell that we have just opened. The -t option to ssh, in turn, forces the allocation of pseudo-tty on the remote machine.

The last line assigns a private address at each end virtual network. We can connect our machines from

one to another without any worries. We just create our VPN.

6.2 Establish the IP_FORWARDING

The IP_FORWARDING is a flag that tells the Linux kernel if the packets must pass through the machine or, on the contrary, it must be stopped. By default, the current distributions, it is initialized to 0 at startup. The following script allows you to initialize it to 1 if you want one of two machines, or both, serve as a gateway for other machines on your network, allowing not only communication from one station to another but also from one network to another.

```
#!/bin/sh
echo " Setting up IP forwarding rules "
echo 1 > /proc/sys/net/ipv4/ip_forward
echo -n "/proc/sys/net/ipv4/ip_forward: "
cat /proc/sys/net/ipv4/ip_forward
for forwarding in
/proc/sys/net/ipv4/conf/*/forwarding
do
echo -n "$forwarding: ";
interface=‘dirname $forwarding‘
interface=‘basename $interface‘
case "$interface" inppp*|eth1)
# interface list when the transfer must be
#enabled
echo 1 > $forwarding
;;
*) # it deactivates interfaces that do not require
the transfer.
echo 0 > $forwarding
;;
esac
cat $forwarding
done
```

6.3 SSH server configuration

The SSH server is shipped with default configuration file named sshd_config. By default, it listens on port 22; we will modify it to listen on port 9870. This results in two things immediately:

- Robots that scan port 22 to find a fallible will not bore your ssh server.
- Logs authentication normally concerns only access attempts to your vpn.

Here is the entire configuration file used for our example:

```
# vpn/etc
# Specific configuration of the port Port 9870
PidFile /var/run/sshd_vpn.pid
HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
# Configuration of log levels
SyslogFacility AUTH
LogLevel INFO
RSAAuthentication yes
AllowUsers sshvpn
# Restrictions
#
IgnoreRhosts yes
IgnoreUserKnownHosts yes
PermitRootLogin no
StrictModes yes
PasswordAuthentication no
PermitEmptyPasswords no
ChallengeResponseAuthentication no
RhostsAuthentication no
RhostsRSAAuthentication no
X11Forwarding no
PrintMotd no
KeepAlive yes
```

6.4 Configuring the VPN

Since our initial tests have been successful, we certainly will desire to automate our VPN. To do this, it is useful and proper to create a configuration file that contains the variables necessary for the creation of our VPN tunnel:

```
# VPN1 Configuration File
# /opt/ssh-vpn/etc/vpn1
# The networks are connected to a side,
# following the route command:
client_network=192.168.2.0/24
server_network=192.168.1.0/24
# Do you want information to debug?
client_debug="no"
server_debug="yes"
# Take different IPs for each VPN required.
server_ppp_ip=192.168.254.254
client_ppp_ip=192.168.254.253
# is there a PPP authentication required?
client_require_pap="yes"
server_require_pap="yes"
client_require_chap="no"
server_require_chap="no"
# Need non-standard pppd arguments? Put
them here.
#client_pppd_args="usepeerdns"
#server_pppd_args="proxyarp"
# Need additional arguments ssh? Put them
here
```

```
#client_ssh_args="-C"
#server_ssh_args=""
```

7. CONCLUSION

In this article, we discussed the three options the most common security currently i.e. SSH, SSL/TLS and IPSec, while explaining the advantages and disadvantages of each of these solutions. Indeed, we have chosen and implemented VPN over SSH to ensure the security of policy deployment.

REFRENCES:

[1] M. El Marraki and A. Kartit, "On the Correctness of Firewall Policy Deployment", *Journal of Theoretical and Applied Information Technology*, ISSN: 1817-3195, Volume 19, n°1, pages 22 – 27, 2010.

[2] J. Allen, A. Christie, W. Fithen, J. Mackhugh, and J. Pickel, "State of Practice of Intrusion Detection Technologies", *Stoner Carnegie Mellon University, Networked Systems Survivability Program, Technical Report*, CMU/SEI-99-TR-028, 2000.

[3] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol", November 18, 1996.

[4] T. Dierks, C.Aallen, "The TLS Protocol Version 1.0", *IETF RFC* No. 2246, January 1999.

[5] T.Ylonen, T. Kivinen, M. Saarinen, T. Rinne, and S. Lehtinen, "SSH protocol architecture", *IETF Internet Draft, draft-ietf-secsh-architecture-13.txt*, September 2002.

[6] T. Ylonen, "The SSH (Secure Shell) Remote Login Protocol", *Helsinki University of Technology*, November 1995.

[7] D. Barrett and R. Silverman, "SSH, le Shell sécurisé", *la référence*. N° ISBN: 2-84177-147-4, O'Reilly, Paris, 2002

[8] T.Ylonen, T. Kivinen, M. Saarinen, T. Rinne, and S. Lehtinen, "SSH transport layer protocol", *IETF Internet Draft, draft-ietf-secsh-transport-17.txt*, October 2003

[9] T.Ylonen, T. Kivinen, M. Saarinen, T. Rinne, and S. Lehtinen, "SSH connection protocol", *IETF Internet Draft, draft-ietf-secsh-connect-18.txt*, October 2003.

[10] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", *IETF RFC No*. 2401, November 1998

[11] D. Harkins et. Al, "The Internet Key Exchange (IKE)", *IETF RFC No*. 2409, November 1997.

[12] D. maughan, "Internet Security Association and Key Management Protocol (ISAKMP)", *IETF RFC No*. 2408, November 1998.

[13] H. Orman, "The OAKLEY Key Determination Protocol", *IETF RFC No*. 2412, November 1998.

[14] T. Markham, "Internet Security Protocol", *Dr. Bobb's Journal*, juin 1997.

[15] W. Stallings, "Sécurité des réseaux, Applications et standard ", Editions Vuibert Informatique, Paris, 2002. ISBN 2-7117-8653-6

[16] N. Ferguson and B. Schneier, "A cryptographic evaluation of IPSec", *Technical report, Counterpane Internet Security*, 95128, USA, 2000.

[17] T. Verdet, "Interfaçage de OpenSSH avec une PKI", *Master's thesis, Département Informatique et réseaux*, ENST Paris, Octobre 2003

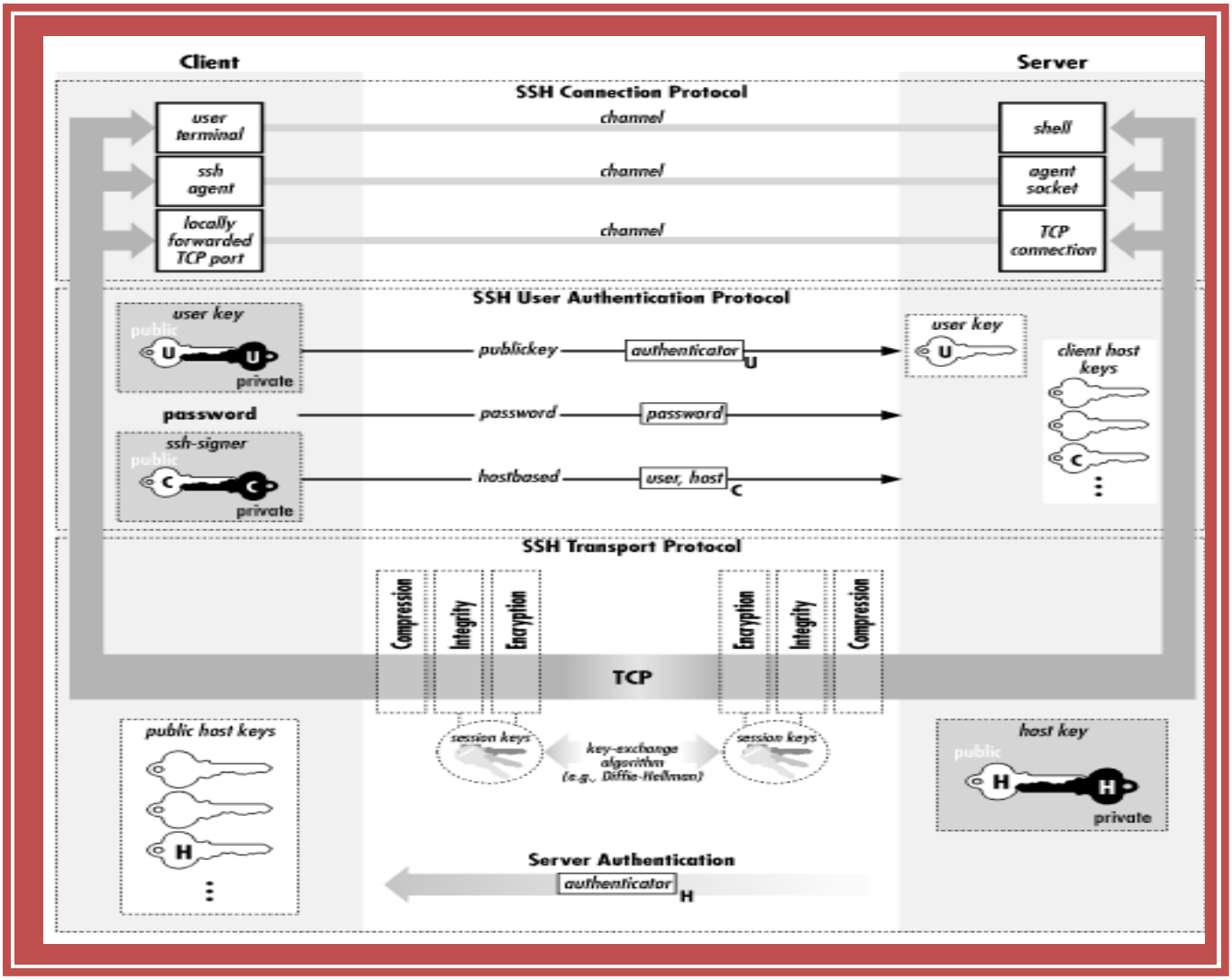

FIGURE 3: SSH-2 HANDSHAKE

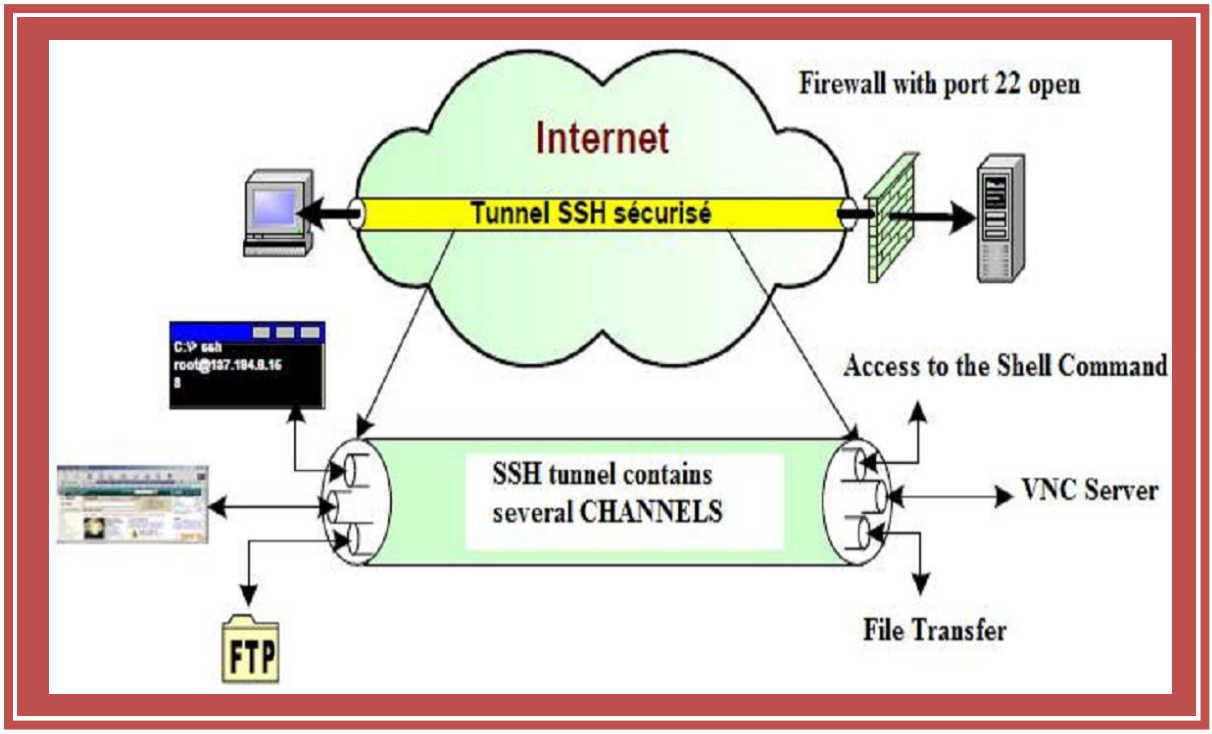

FIGURE 4: SSH TUNNELS AND CHANNELS

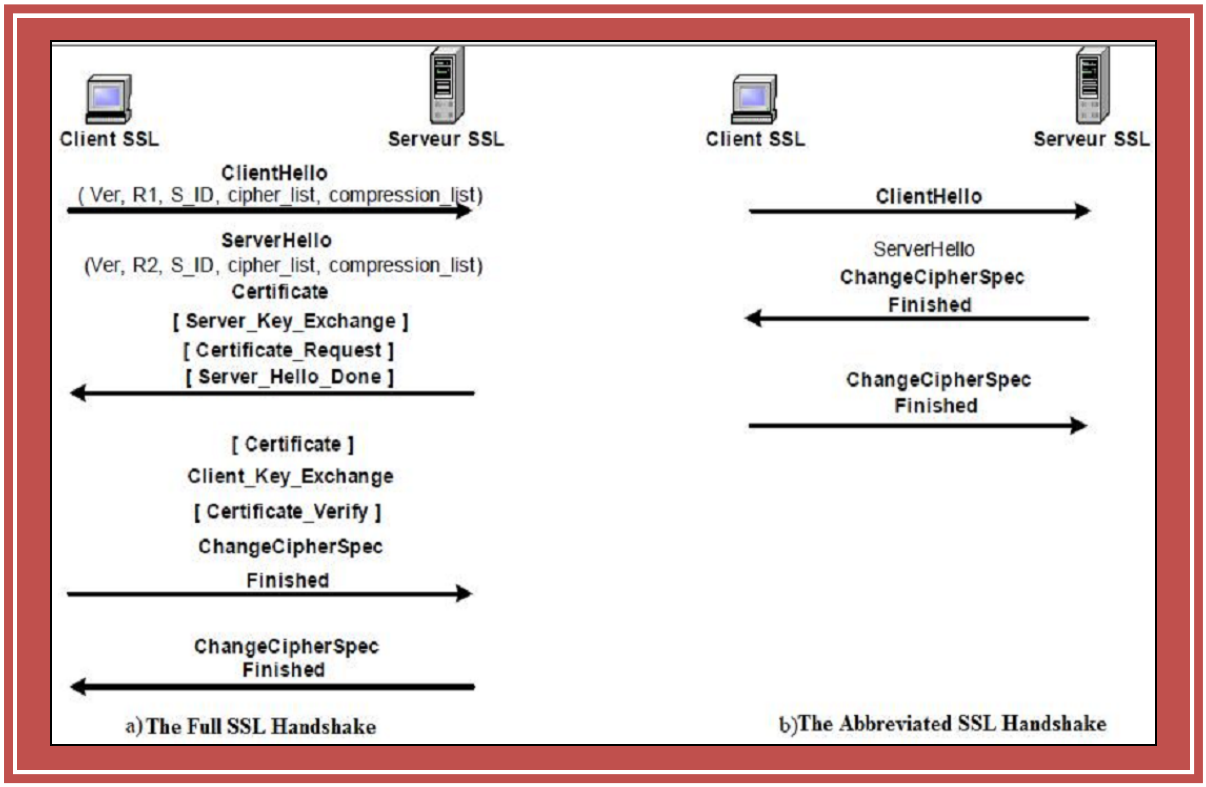

FIGURE 6: SSL/TLS HANDSHAKE