TUNNELLING DATABASE CONNECTIONS WITH FREES/WAN'S OPPORTUNISTIC ENCRYPTION (OE)

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Juan Pedro Flor Pereira
April 08, 2003

ABSTRACT

This document will explain in a brief way how to solve the problem of Database connections in insecure networks through the VIRTUAL PRIVATE NETWORK, a characteristic of FreeS/WAN, that allows any Gateway that disposes this capacity, OPPORTINISCTIC ENCRYPTION (OE), to establish secure connections without complex configurations or the agreement or previous co-ordination of the administrators of the different Gateways.

The basic idea of securing in deepness the Database connections consists in encrypting the flow of data that travels through the insecure networks.

This document will explain the problem that many organisations have to confront: databases connected to Internet and the problem of the difficult administration and maintenance.

A solution will be given using the new protocol, OPPORTUNISTIC ENCRYPTION, developed by FreeS/WAN. It will be followed with a description of DNS Secure (DNSSec).

The solution explains how to accomplish a more secure network administration, easy administration and maintenance without the necessity of advanced knowledge referring to cryptography, VIRTUAL PRIVATE NETWORK and OE.

An implementation is proposed on how to resolve the problem mentioned before. The document analyses the basic configurations that should be taken into consideration to achieve secured connections . All this through an example for a fictitious business that adjust to the necessity of many real companies today.

This research gives additional advice on how to increase the security systems. It will finally end with a conclusion of this document outlining the benefits de Frees/WAN's OE.

INTRODUCTION

Why securing in deepness the database connections?

The networks expansion, globalisation, technological development, etc. demand that the organisations have to share, facilitate and dispose vital information to the public and among the same organisations through the world network, the Internet.

The financial entities, for example, need to make available their databases to the branches in different parts of the world to be competitive. Organisations of other type need to accomplish as well certain requirements that need connectivity. Because of this and the growing wave of hacking in world level, "securing the database connections" is a vital task because a valuable resource is in jeopardy: the information.

Therefore, securing in depth the database connections is no simple task. The configuration, maintenance, and cost involved can in many cases result in gigantic task. This is why many organisations opt to decrease their level of security hoping that, in the best of the cases, nothing bad happens.

On the other side, the financial entities are victims of hacker attacks that in one way or another try to expose their systems or rob information resulting in great lost of money, trust, and credibility.

For this reasons is that securing in profundity the databases is and must be a primordial task.

In the following problems of a fictitious company named "Storage Systems" it will be analysed how to adjust to the necessity of the companies of today.

THE PROBLEM

A clear description of the problem (which is divided in two sub- problems) will be considered in this sections.

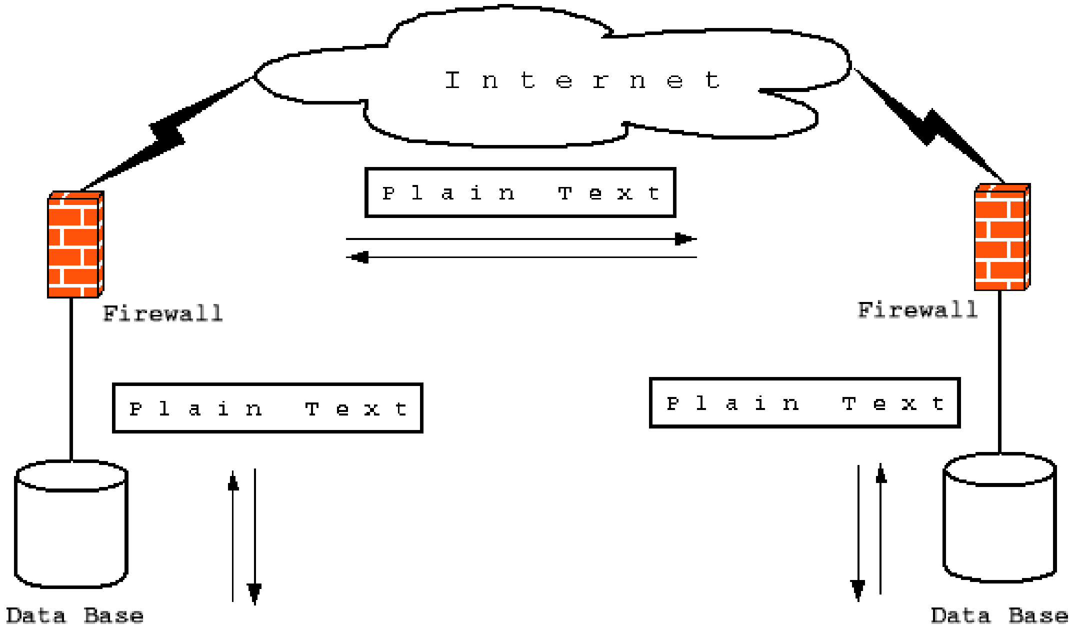
Problem 1 - Information

The information has become a valuable resource that the organisations need to handle. In many occasions it has to travel through “insecure networks” risking falling in wrong hands resulting in prejudice for many. The following graphic shows the current situations of the business “**Storage System**” (StSys) and their database connections:

Fig. 1 [General scheme of insecure database connections]

a) Distributed Databases

Storage Systems (StSys) handles databases distributed, collections of data that belong logically to the same network but are disperse physically among the sites of network LAN and WAN. Therefore all the traffic of data flows from one network to another through Internet without encrypting, “plain text”, which means that with techniques like “sniffing” it is possible to capture all the traffic resulting in a great security problem.



b) Replicated Databases

Because of their policy and business regulations StSys opted to replicate their databases in different physical locations. This also results in a problem because the data flow through insecure networks without any type of cryptography.

Problem 2 - Complex Administration and maintenance

StSys confronts another problem: StSys has more than 125 branches in all the world therefore the configuration, administration and maintenance results complex. Besides the fact the Information Technology (IT) personal must be highly specialised, the administration, configuration and maintenance of security over all the branch imply a great overload to all the team.

Two fundamental problems of Storage Systems

- Valuable information in insecure networks
- Complex administration and maintenance

APPROPRIATE SOLUTION FOR STORAGE SYSTEMS

How to secure in deepness the connections of data base?

Using Frees/WAN

The basic idea of securing in deepness the database connections consist in encrypting the flow of data that travels through insecure networks as illustrates the following graphic:

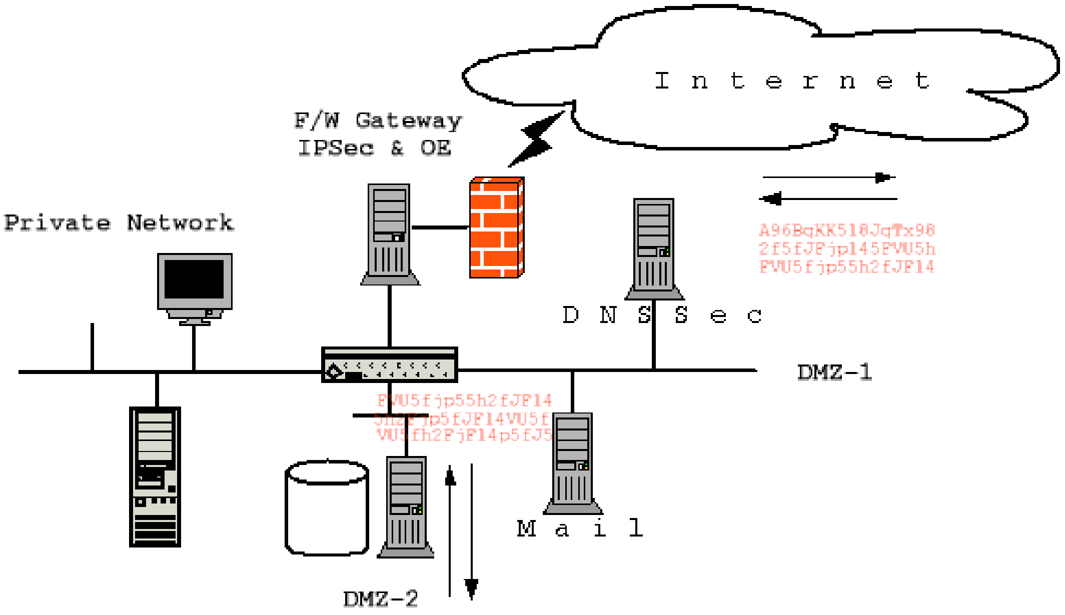


Fig. 2 [Tunnelling database connections with FreeS/WAN's OE]

The idea of “securing in deepness the database connections” consist in transmitting data through secure tunnels (encryption) using VPNs. This way we can accomplish security in deepness.

There are many products of easy configuration, administration and maintenance but none of them proportion the facility inherent to a product (software) in particular: FreeS/WAN.

FreeS/WAN is a free Linux implementation of IPSec that provides encryption and authentication.

The primordial characteristic of FreeS/WAN is the implementation of a new protocol “OE” that pretends to be part of the the IPSec standards.

The primordial characteristic of OE is: allowing any FreeS/WAN Gateway to encrypt the traffic before sending it even if it does not have previous co-ordination of the administrators of the gateways of both parts and any manual configuration for each tunnel.

Each and every one of the tools (software) used to solve problems are Open Source, Open Licences and No Cost.

BRIEF THEORY

FreeS/WAN

FreeS/WAN is an open source implementation (Linux implementation) of the IPsec (IP security) protocols that provides services of encryption and authentication at a network level. The main objective of FreeS/WAN is making of Internet safer and providing simple ongoing administration; OE allows any FreeS/WAN Gateway to encrypt the traffic before sending it even without previous co-ordination of the administrators of the gateways (at the extreme point) of both sides and without any manual configuration for each secure encrypted tunnel.

“Opportunistic encryption permits secure (encrypted, authenticated) communication via Ipsec without connection-by-connection prearrangement, either explicitly between hosts (when the hosts are capable of it) or transparently via packet-intercepting security gateways. It uses DNS records (authenticated with DNSSEC) to provide the necessary information for gateway discovery and gateway authentication, and constrains negotiation enough to guarantee success.“ [1]

“A major goal of the FreeS/WAN project is opportunistic encryption: a (security) gateway intercepts an outgoing packet aimed at a remote host, and quickly attempts to negotiate an IPsec tunnel to that host's security gateway. If the attempt succeeds, traffic can then be secure, transparently (without changes to the host software). If the attempt fails, the packet (or a retry thereof) passes through in clear or is dropped, depending on local policy. Prearranged tunnels bypass the packet interception etc., so static VPNs can coexist with opportunistic encryption.”[2]

> *“Opportunistic encryption [allows] secure (encrypted, authenticated) communication via Ipsec without connection-by-connection pre-arrangement[.]“*

FreeS/WAN is generally executed in a computer that is the gateway to Internet. This gateway implements safe tunnels (encryption) to other safe gateways (FreeS/WAN + OE). When OE is activated, the safe gateway intercepts the first outgoing package with a specific destiny. Then it tries to negotiate a safe tunnel for the encrypted traffic among both extremes (secure gateways).

So that a secure gateway distributes the information required by another and for the authentication to take place between both gateways, DNSSec is needed. The following graph makes a summary of the motives for which DNSSec is necessary:

. DNSSec guarantees that the data obtained by other VPN host are reliable.
. DNSSec allows identifying in a safe way the remote host with which communication is desired
. DNSSec allows obtaining the key of authenticity for the secure gateway.

With respect to the cryptography, FreeS/WAN uses three protocols:

- * AH (Authentication Header)
- * ESP (Encapsulating Security Payload)
- * IKE (Internet Key Exchange)

And divides in three main parts

- * KLIPS (kernel Ipsec)
- * Pluto (an IKE daemon)

And Scripts

How it works

The following graphic shows how FreeS/WAN's OPPORTINISCTIC ENCRYPTION works:

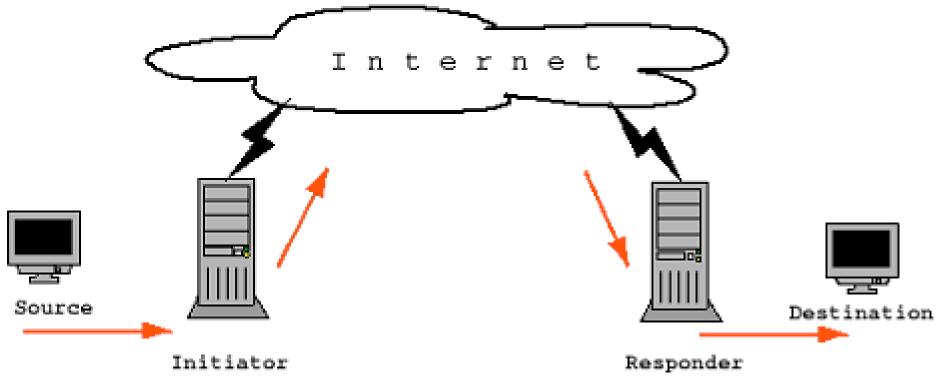


Fig-3 [***Basic steps in a basic negotiation***]

The “source” and “destination” are endpoints of the data-flow that must be protected. The source sends a set of packets with a specific destination. The “initiator” intercepts the first outgoing package, then it tries to negotiate with the “responder” gateway to establish a safe tunnel for encrypting all the data-flow.

When the “initiator” is going to initiate an Opportunistic negotiation with the “responder”, it needs to provide a way for the Responding to the extreme secure gateway to find a public key for the “initiator” to allow authentication. This is made by putting the public key in a KEY record in the reverse-map of the “initiator”.
The “source” and “destination” that it is behind protected Opportunistic Encryption must include a special TXT record in its reverse-map.

Frees/WAN flow

1. Initiator does a DNS reverse lookup on the Destination address, asking for TXT records.
2. Initiator picks one TXT record.
3. The Initiator uses the existing keying channel.
4. Responder gets the first IKE message, and responds.
5. Initiator gets Responder's reply, and sends first message of IKE's D-H exchange.
6. Responder gets Initiator's D-H message, and responds with a matching one.
7. Initiator gets Responder's D-H message; encryption is now established.
8. Responder gets Initiator's authentication message.

10. Initiator initiates IKE Phase 2 negotiation to establish tunnel.
11. Responder gets first Phase 2 message.
12. Initiator gets second Phase 2 message, establishes tunnel.
13. Communication proceeds. [3]

That is the way how FreeS/WAN works.

Open SSL

Open SSL is needed by DNSSec

SSL (Secure Socket Layer) was originally developed by Netscape to transmit private documents via the Internet. Open SSL is a set of Open Source tools that implement the “Secure Socket Layer (SSL v2/v3)” and “Transport Layer Security (TSL v1)“. Basically SSL works by using a private key to encrypt data that it is transferred over the SSL connection. The fundamental goal of SSL protocol is to provide privacy and reliability between two communicating applications. One advantage of SSL is that it is an independent application protocol; it is a higher-layer security protocol. The SSL protocol provides secure connections that has three basic properties:

- Private connections (using strong cryptography)
- Authenticity (with public key schema)
- Reliability (whit has functions that provides integrity)

SSL also implements four cryptographic operations:

- digital signing (digitally-signed)
- stream cipher encryption (stream-ciphered)
- block cipher encryption (block-ciphered)
- public key encryption (public-key-encrypted)

And SSL can be used for:

Create RSA, DH and DSA keys
Create X.509, CSRs and CRLs certificates
Calculate Digest messages
Encrypt and decrypt

BIND (DNSSec)

DNSSec in needed by FreeS/WAN

BIND is the ISC[4] implementation of "Berkeley Internet Name Domain" to provide DNS services. This Open Source software is extensively used in all the world to promote and provide DNS services.

DNSSEC (DNS Security Extensions) is a technique for securing the Domain Name System. It is a set of extensions to DNS, which provide end-to-end authenticity and integrity and was designed to protect the Internet from certain attacks, for example DNS spoofing. DNS spoofing happens when a DNS server accepts or makes use of incorrect information from a host that has no authority for that information. When a DNS server accepts or makes use of false information from a host that has not authority given that information Dnsspoofing occurs. Dnsspoofing attacks can cause serious security problems for DNS servers vulnerable to such attacks, for example causing users to be directed to false or malicious Internet sites. [5]

DNSSec is a method that provides cryptographic verification information along with DNS messages. Since the 8.2 BIND version security mechanism has been introduced, TSIG (Transaction Signatures), that fundamentally allow administrators to secure communications between two name servers.
TSIG uses the 'MD5' hash function with its Hmac-md5 variant. A DNS (query, response, dynamic update) message is passed through to Hmac-md5. In addition, a public key is used like entrance (input), HMAC-MD5(dns_query, shared_key), producing a unique result 'meta-record' in a new 'resource record' called 'TSIG record'. Now this record is added to DNS message automatically and sent to its destiny, which it is stripped off and verified. Everything is made by the BIND software, reason why the DNS administrator does not have to add any TSIG record to the zone files. The DNS server of the ends must be synchronous with the time; for these reason it is necessary to use the protocol NTP (Network Time Protocol). TSIG does not make anything to assure the privacy of a DNS message, reason why the message travels or is sent in clear text.

To provide cryptography "Public keys" and "digital signatures" are used. The "digital signatures" provided cryptographic authentication in DNSSec; proves that certain message originated from a specific source (place) and the message itself has not been changed.
Within DNSSec there are two record types for authentication: the KEY record that stores the public key for a host or administrative zone and the SIG record that stores a digital signature associated with each set of records.

Finally, DNSSec prevents that any client to trust false information because all data must be authenticated before it can be trusted using digitally signatures. Additionally, TSIG guarantee that the client and server performing a DNS transaction are the entities they claim to be.

DNSSec

- Digital Signatures provides height level of trust
- TSIG guarantee that the client and server performing a DNS transaction are the entities they claim to be.

IMPLEMENTATION

Software Requirements:

Bind Version 9.2
ftp://ftp.isc.org/isc/bind9/9.2.2/bind-9.2.2.tar.gz

OpenSSL Version 0.9
http://www.openssl.org/source/openssl-0.9.7b.tar.gz

FreeS/WAN Version 2
ftp://ftp.xs4all.nl/pub/crypto/freeswan/freeswan-2.02.tar.gz

Open SSL

Open SSL needs no special configuration. Proceed just with the installation:

```
$ ./config
$ make
$ make test
# make install
```

Bind (DNSSec)

After accomplishing a satisfactory installation, it should be proceed with the configuration of the DNS server.

DNS server Configuration

The public keys should define each DNS server adding the following entrance to in the file "named conf"

```
key "rndc-key" {
        algorithm hmac-md5;
        secret "ZxwU....5EKTERe3B7t";
};

controls {
        inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

FreeS/WAN

To use the FreeS/WAN's OE characteristic the following is necessary

Linux, this should not be behind the NAT
FreeS/WAN version greater than 2
A static IP address

Once unpacked proceed the installation (see the file INSTALL in the source code). After the proper installation of the FreeS/WAN software initiate the service

Initiate FreeS/WAN (as root)

```
 # service ipsec start
```

Verify that FreeS/WAN works correctly

```
# ipsec verify
```

in this moment the following output should be seen

```
Checking your system to see if IPsec got installed and started correctly
Version check and ipsec on-path                              [OK]
Checking for KLIPS support in kernel                         [OK]
Checking for RSA private key (/etc/ipsec.secrets             [OK]
Checking that pluto is running                               [OK]
```

To make possible FULL OPPORTUNISTIC ENCRIPTATION with inbound and outbound connections (sending and receiving) the following steps should be followed

Create and publish a forward Record DNS
Select a domain name and host (ID) that will be used

Create a TXT record that contains the public key created with ipsec:

```
# ipsec showhostkey --txt @xy.stsys.com
```

The result would be something like this:

```
 ; RSA 2048 bits   xy.stsys.com   Mon Jun  2 12:00:00 2003
    IN      TXT     "X-IPsec-Server(10)=@[Public-IP]"     "5F4S66....654SJFG4/"
```

DNS Record reverse TXT for each machine protected (Data base server)
The objective of this is:

- Discovering the addresses of the gateway with only one IP address of the packages of exit
- Verifying that the gateway has authorisation to encrypt all the traffic for a certain final point.

The before generated keys should be the same, on the contrary it will not work. Now, some changes in the DNS server should be made:

With the generated key

```
IN      TXT     "X-IPsec-Server(10)=@[Public-IP]"     "5F4S66....654SJFG4/"
```

Create an entrance to each computer that is wished to be protected.

```
[SUBNET-IP].in-addr.arpa. IN PTR arthur.stsys.com.
```

The DNS file should contain:

- The address of the sub network in the reverse format. Example:
  192.168.1.0 => 0.1.168.192-in.addr.arpa

- The name of the host should have a period in the end. Example:
  hostx.stsys.com => hostx.stsys.com.

Verify that the TXT register with the published

```
# ipsec verify --host xy.stsys.com
# ipsec verify --host hostx.stsys.com
```

Then, the following output should be displayed:

```
Looking for TXT in forward map: xy.stsys.com          [OK]
```

If everything is fine until here, we can proceed to initiate the service

```
# service ipsec restart
```

Policy Group

Policy is a document (guideline) that states how plans to protect the company's physical and technological information assets. The policy group allows to establish locally lists of Ipsec policy for hosts and remote network by building lists of hosts or networks that desire to have a special treatment.
FreeS/WAN allows the following group policy in the group base:

Private *Only communication privately with the listed CIDR blocks.*
Private-or-clear *Preferentement private communication with the listed CIDR blocks.*
Clear-or-private *Permit communication in plain text with the listed CIDR blocks,*
but also accepts inbound OE connection.
Clear Only communication in plain text with the listed CIDR blocks.
Block Blocks traffic to and from and the listed CIDR blocks.

Policy Groups Location

```
./etc/ipsec.d/
      |--------
              `-- policies
                 |-- block
                 |-- clear
                 |-- clear-or-private
                 |-- private
                 `-- private-or-clear
```

Example of policy defined by CIDIR, IP Adress or FQDN for any policy group type:

```
192.168.1.105                          # Sales
192.168.1.0/16                         # MS Windows Network
admin.domain.tld                       # Subdomain neigh
```

ADVICES

To achieve the maximum security it is not enough to secure just the database connections. It is necessary to take complementary actions. Following, a brief summary of certain things to taking into consideration to build a reasonable security infrastructure:

- Physical Security
- Safe communications
- Security policies
- Access Control
- Intrusion detection systems
- Encryption
- Avoid unknown software
- Backups
- Regular audits
- Minimising unnecessary services
- Software actualisation
- Robust Passwords
- Secure programming

CONCLUSION

In many cases, subtracting the complexity of the configuration, administration and maintenance of security systems can accomplish an increment security.
The use of FreeS/WAN, Bind, OpenSSL and other Open Source tools not only adds to the security of work environment but decreases the load of work with respect to the administration.
Although other solutions exist to secure the connections in databases, as analysed in this document, not all of them present the facility of OPPORTUNISTIC ENCRYPTION dealing with administration.
Furthermore, FreeS/WAN goes more than securing the connections of databases, with the principal characteristic, OPPORTUNISTIC ENCRYPTION, it is possible to encrypt almost all the traffic that travels through Internet.

It is not only possible to create an environment as described in this research for the company Storage Systems. With these technology it is even possible to create a secure telephony infrastructure: Secure Voice Over IP.

REFERENCES

[1] Spencer Henry & Redelmeier D. Hugh. “IKE Implementation Issues“. 26 Feb 2002
URL: http://www.freeswan.org/freeswan_trees/freeswan-2.02/doc/draft-spencer-ipsec-ike-
implementation.txt (02/07/2003)

[2] Spencer Henry & Redelmeier D. Hugh. “Opportunistic Encryption.“ 3 May 2001
URL: http://www.freeswan.org/freeswan_trees/freeswan-2.02/doc/opportunism-spec.txt (03/07/2003)

[3] Spencer Henry & Redelmeier D. Hugh. “Opportunistic Encryption.“ 3 May 2001
URL: http://www.freeswan.org/freeswan_trees/freeswan-2.02/doc/opportunism-spec.txt (03/07/2003)

[4] “Internet Software Consortium (ISC)”
URL: http://www.isc.org/

[5] Doug Sax. “DNS Spoofing (Malicious Cache Poisoning).”
URL: http://www.giac.org/practical/gsec/Doug_Sax_GSEC.pdf (15/08/2003)

Navathe, Elmasri. Sistemas de bases de datos – Conceptos Fundamentales. Mexico: Addison-Wesley Iberoamericana, 1996. 704 – 729.

Brown, Steven. Implementación de redes privadas virtuales (RPV). Mexico: McGraw-Hill. 2000.

Linux Magazine. “Transactional Security in BIND 9.“ November 2001.
URL: http://www.linux-mag.com/2001-11/bind9_01.html (07/06/2003)

Nominum.com team. “DNSSEC FAQs.” 2003
URL: http://www.nominum.com/getOpenSourceResource.php?id=8 (02/07/2003)

Internet RFC/STD/FYI/BCP Archives. “RFC 3226 - DNSSEC and IPv6 A6 aware server/resolver message size requirements.“
URL: http://www.faqs.org/rfcs/rfc3226.html (18/08/2003)

OpenSSL Team. “HOWTO certificates”.
URL: http://www.openssl.org/docs/HOWTO/certificates.txt (03/06/2003)

OpenSSL Team. “HOWTO keys”.
URL: http://www.openssl.org/docs/HOWTO/keys.txt (03/06/2003)

Freier Alan O., Karlton Philip, Kocher Paul C. “The SSL Protocol Version 3.0”. 1998
URL: http://wp.netscape.com/eng/ssl3/draft302.txt (03/06/2003)

Frees/WAN Team. “Linux FreeS/WAN.“ 2003/04/15.
URL: http://www.freeswan.org/freeswan_trees/freeswan-2.01/doc/ (02/07/2003)