26. DECUS Symposium
Bonn

OpenSSL in OpenVMS und STunnel

Helmut Ammer
OpenVMS Support
CSSC München

2F06

Presentation Overview

- Product information
  - What is the secure sockets layer (SSL)?
  - Overview of SSL/OpenSSL/SSL on OpenVMS
  - VMS changes & uses
- Technical information
  - SSL in an application
  - Crypto library
  - OpenSSL command line utility examples
- STunnel
- Questions?

What is SSL?

- Secure Sockets Layer
- Secures data communication between a client and server at the transport layer
- Authenticates the Server (by default) and the client (optionally)
- Provides data confidentiality
- Ensures data integrity

SSL & OpenSSL

- Netscape developed SSL V2 & V3
- Transport layer security (TLS) is RFC 2246
- OpenSSL is a toolkit that provides:
  - Sslv2 & v3 protocols
  - TLS v1 protocol
  - Cryptographic algorithms
- OpenSSL is packaged as
  - An SSL library
  - A cryptographic library
  - A command line utility

VMS Changes to 0.9.6b

- Added 64-bit API support.
- Added a menu-driven certificate tool.
- Enabled SSL to run on any TCP/IP product.
- Added VMS PRNG support.
- Added some better documentation.
- And many more … all of which are being sent back to the OpenSSL group

  ftp://ftp.openssl.org/snapshot/

  openssl-VMS_64bit-snap-yyyymmdd.Tar.gz

SSL for OpenVMS Alpha V1.0-B

- V1.0 port of OpenSSL 0.9.6B
  - ~~V1.0 : based on OpenSSL 0.9.6B & distributed on V7.3-1 LP CD~~
- Buffer Overflow Security vulnerabilities fixed
  - Based on 0.9.6B but includes security patches, use this!
- Download V1.0-B from the OpenVMS security website

  www.openvms.compaq.com/openvms/products/ssl/ssl.html

- Layered Product kit (.PCSI)
- Installation steps:

$ product install ssl[/dest=dev:[dir]]

$ @sys$startup:ssl$startup

$ @ssl$com:ssl$utils

SSL for OpenVMS Alpha – The source kit

- Source available on the web
  http://www.openvms.compaq.com/openvms/products/ssl/ssl_source.html
  Same sources that were used to create the .PCSI kit
- Instructions are on the website:
  - Downloading
  - Expanding the image
  - Unpacking the save set
  - Building the sources

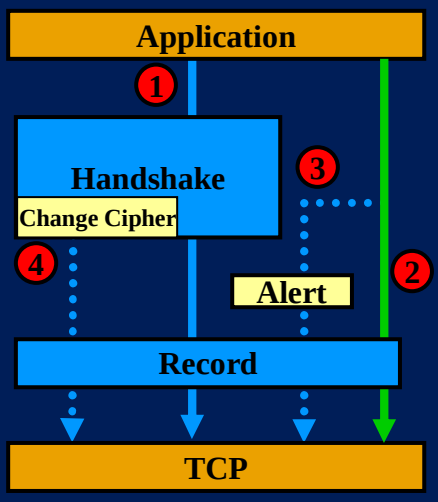
SSL for OpenVMS in Use Today

- Currently being used in:
  - Common data security architecture
  - Compaq secure web server (apache)
  - PHP
  - Galaxy configuration manager
  - Lightweight directory access protocol (LDAP) API
- Next release
  - 0.9.6g
    - Bug fixes since 0.9.6b
    - Improve documentation
    - Alpha/Itanium
  - CRL support

OpenSSL Development Issues

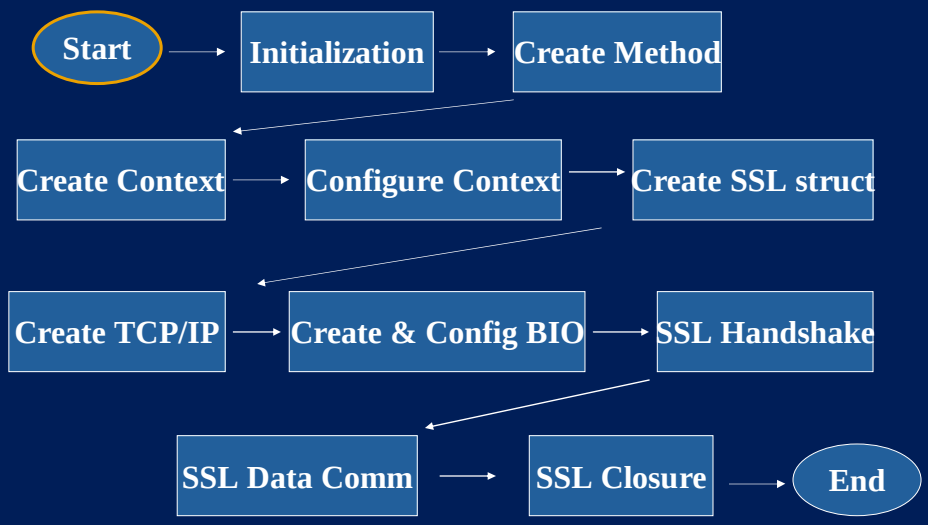
- Backward Compatibility
- Crypto Documentation
- Certificate Management
- Architecture Differences

SSL/TLS Protocol Overview



- 1. Handshake
  - Establish shared secret for encryption
- 2. Application Data
  - Encryption & data integrity for SSL
- 3. Alert
  - Signaling errors & SSL closure
- 4. Change cipher spec
  - Notify that crypto algorithms & keys are being changed

Overview of an SSL application



Initialization

```
/* load encryption & hash algorithms. */
SSL_library_init();

/* load error strings for better reporting. */
SSL_load_error_strings();
```

Method Creation

Protocol	Combined Method	Server Method	Client Method
SSLv2	SSLv2_method	SSLv2_server_method	SSLv2_client_method
SSLv3	SSLv3_method	SSLv3_server_method	SSLv3_client_method
TLSv1	TLSv1_method	TLSv1_server_method	TLSv1_client_method
SSLv23	SSLv23_method	SSLv23_server_method	SSLv23_client_method

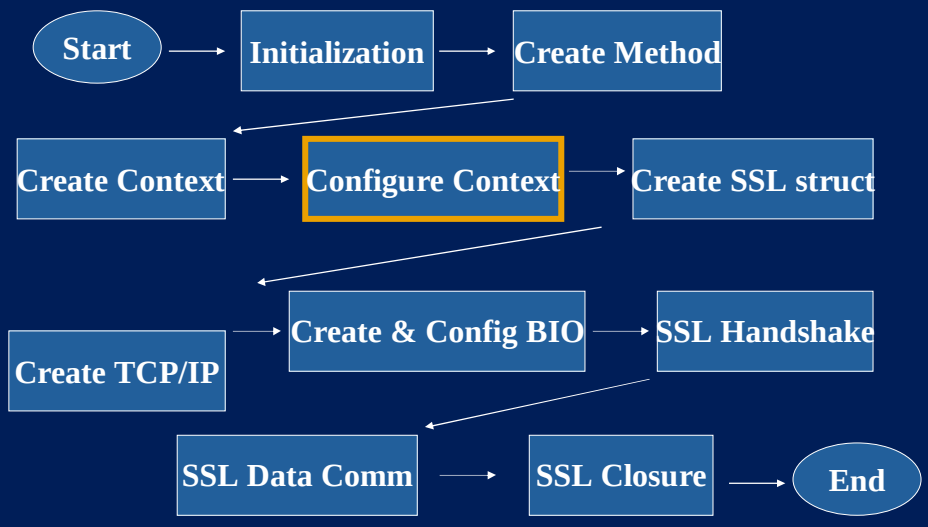
Method Creation (cont'd)

```
SSL_METHOD *meth;
…
meth = SSLv23_method();
```

Context Creation

```
SSL_CTX *ctx;
…
ctx = SSL_CTX_new(meth);
```

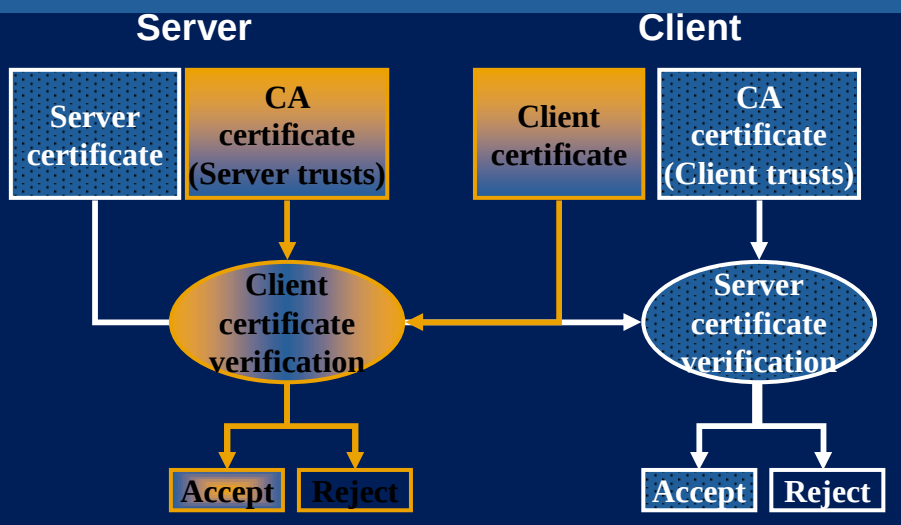
Overview of an SSL application



Context Configuration

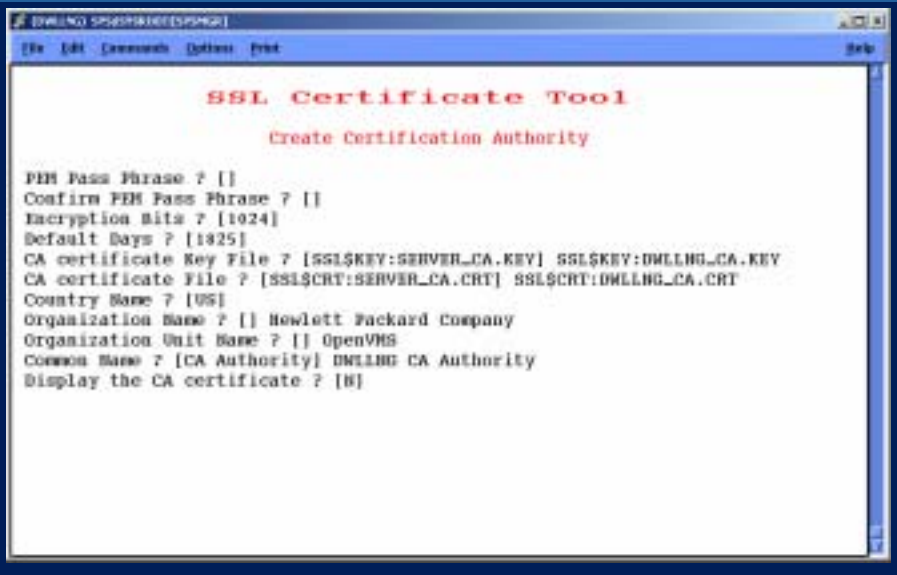
- Certificates & Keys
  - Client, Server & Certificate Authority
  - Certificates aka Public Keys
  - Created with OPENSSL.EXE or SSL$COM:SSL$CERT_TOOL
- Verification
  - Client
  - Server

Server Authentication and Client Authentication

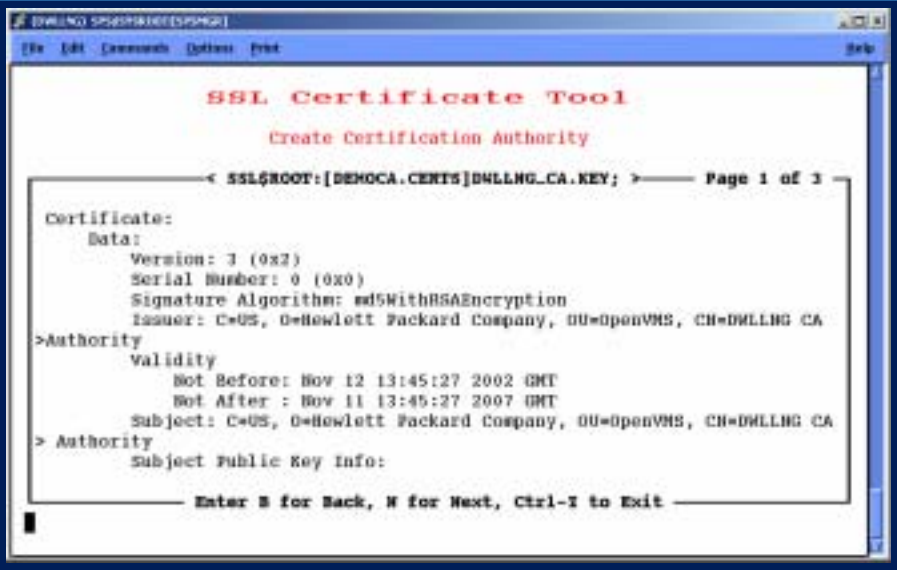


Certificate Tool –
$ @SSL$COM:SSL$CERT_TOOL

Create Certificate Authority Certificate



Display Certificate Authority certificate



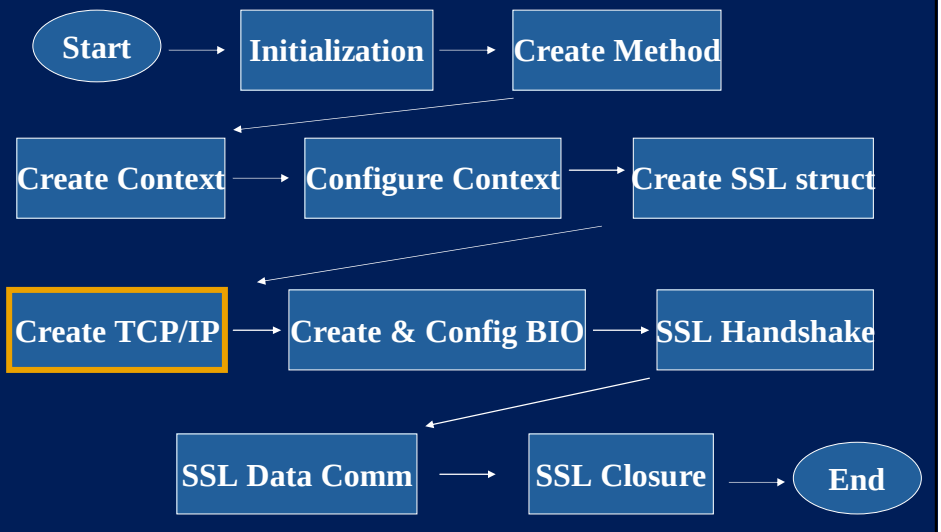
Context Configuration (cont'd)

SSL_CTX_use_certificate_file
(ctx, server_cert, SSL_FILETYPE_PEM);
SSL_CTX_use_PrivateKey
(ctx, server_key, SSL_FILETYPE_PEM);
SSL_CTX_load_verify_locations
(ctx, CAfile, CApath);

SSL Creation

```
SSL *ssl;
…
ssl = SSL_new(ctx);
```

Overview of an SSL application



TCP/IP Socket Creation - Server

```
listen_sock = socket
      (PF_INET, SOCK_STREAM, IPPROTO_TCP);
sa_serv.sin_family = AF_INET;
sa_serv.sin_addr.s_addr = INADDR_ANY;
sa_serv.sin_port = htons(s_port);
err = bind(listen_sock, &sa_serv, sizeof(sa_serv));
sock = accept (listen_sock, &sa_cli, &client_len);
```

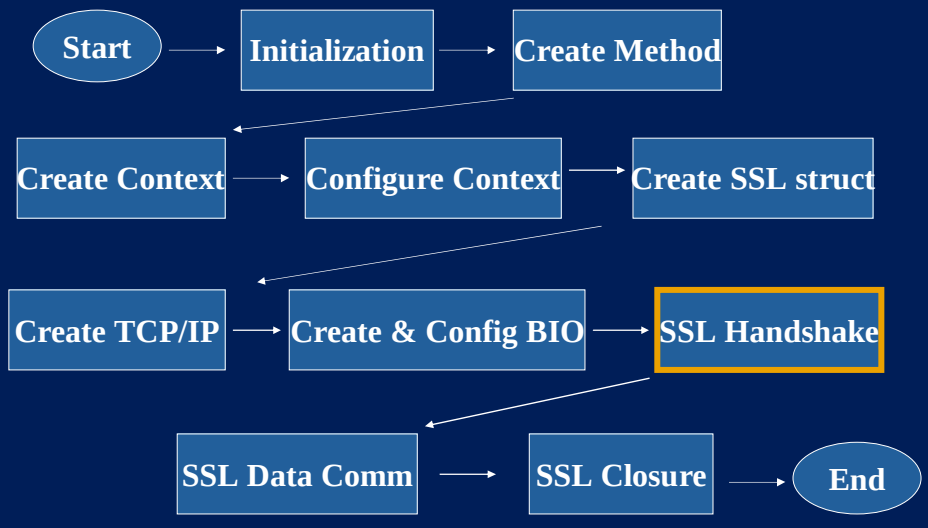
TCP/IP Socket Creation - Client

```
sock = socket
      (AF_INET, SOCK_STREAM,IPPROTO_TCP);
serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(s_port);
serv_addr.sin_addr.s_addr = inet_addr(s_ipaddr);
err = connect (sock, &serv_addr, sizeof(serv_addr);
```

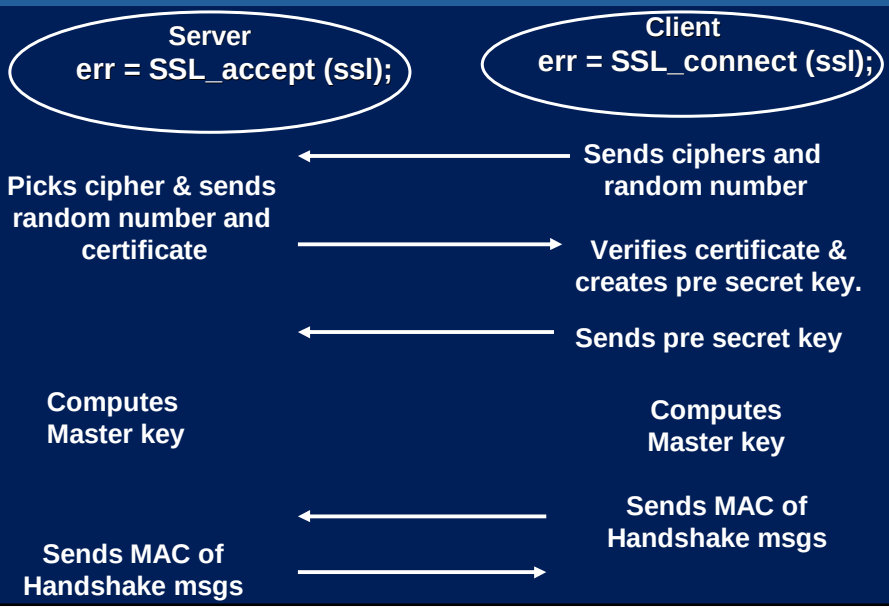
BIO Creation & Configuration

```
SSL_set_fd (ssl, sock);
Or
sbio = BIO_new (BIO_s_socket() );
BIO_set_fd (sbio, sock, BIO_NOCLOSE);
SSL_set_bio (ssl, sbio, sbio);
```

Overview of an SSL application



Handshake



SSL Data Communication

- Sending data –
  err = SSL_write (ssl, buffer, sizeof(buffer));

- Receiving data –
  err = SSL_read (ssl, buffer, sizeof(buffer));

SSL Closure

err = SSL_shutdown (ssl);
err = close (sock);
SSL_free (ssl);
SSL_free (ctx);

Link against
SSL$LIBSSL_SHR.EXE
SSL$LIBCRYPTO_SHR.EXE

Crypto Library

- Symmetric Ciphers
  - Blowfish, Cast, DES, *Idea*,* RC2, RC4, *RC5**
- Public Key Cryptography & Key Agreement
  - DSA, Diffie-Helman(DH), RSA
- Certificates
  - x509 & x509v3

* - Note: Idea & RC5 are not supported in SSL for OpenVMS

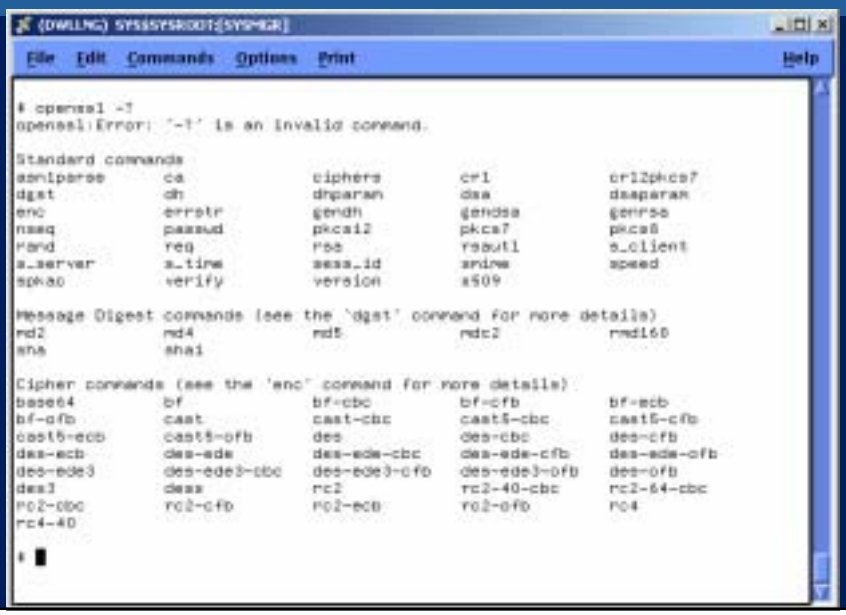
Crypto Library (Cont'd)

- Authentication Codes & Hash Functions
  - hmac, md2, md4, md5, mdc2, ripemd, sha
- Auxiliary Functions
  - threads, rand
- I/O & Data Encoding
  - asn1, pem, pkcs7, pkcs12

Crypto APIs

- Nearly 2,000 crypto APIs
  - symmetric cryptography
  - Hashes and MACs
  - Public Key Algorithms
- Link against:
  - SSL$LIBCRYPTO_SHR.EXE

Command Line Utility – $@SSL$COM:SSL$UTILS



Configuration File

SSL$ROOT:[000000]OPENSSL-VMS.CNF
SSL$ROOT:[000000]OPENSSL-VMS.CNF_TEMPLATE
Environmental variables:

$foo

${foo} – SSL on OpenVMS will only accept this format.

```
##################################################################
[ CA_default ]

dir             = ssl$root:[demoCA      # Where everything is kept
certs           = ${dir}.certs]         # Where the issued certs are kept
crl_dir         = ${dir}.crl]           # Where the issued crl are kept
database        = ${dir}]index.txt      # database index file.
new_certs_dir   = ${dir}.certs]         # default place for new certs.

certificate     = ${dir}]cacert.pem             # The CA certificate
serial          = ${dir}]serial.txt             # The current serial number
crl             = ${dir}]crl.pem                # The current CRL
private_key     = ${dir}.private]cakey.pem      # The private key

x509_extensions = usr_cert              # The extentions to add to the cert
```

S_Server

```
Server> @ssl$com:ssl$utils
Server> s_server -cert ssl$certs:server.crt -key ssl$key:server.key -state
Using default temp DH parameters
ACCEPT
SSL_accept:before/accept initialization
SSL_accept:SSLv3 read client hello A
SSL_accept:SSLv3 write server hello A
SSL_accept:SSLv3 write certificate A
SSL_accept:SSLv3 write key exchange A
SSL_accept:SSLv3 write server done A
SSL_accept:SSLv3 flush data
SSL_accept:SSLv3 read client key exchange A
SSL_accept:SSLv3 read finished A
SSL_accept:SSLv3 write change cipher spec A
SSL_accept:SSLv3 write finished A
SSL_accept:SSLv3 flush data
```

S_server (Cont'd)

```
-----BEGIN SSL SESSION PARAMETERS-----
MHUCAQECAgMBBAIAFgQg1KFEzJfmJFmdcm2idGaM4OhxL8RZr/ktB/Pv/F99KdwEMH/tormk
    acVAlpCLNhzgOrjkwANo+zvfVDgkfBkP87Q75B6/4G8FXexHqbx2Ds42UaEGAgQ9j25+ogQC
    AgEspAYEBAEAAAA=
-----END SSL SESSION PARAMETERS-----
Shared ciphers:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DES-CBC3-SHA:DHE-
    DSS-RC4-SHA:RC4-SHA:RC4-MD5:EXP1024-DHE-DSS-RC4-SHA:EXP1024-RC4-
    SHA:EXP1024-DHE-DSS-DES-CBC-SHA:EXP1024-DES-CBC-SHA:EXP1024-RC2-CBC-
    MD5:EXP1024-RC4-MD5:EDH-RSA-DES-CBC-SHA:EDH-DSS-DES-CBC-SHA:DES-CBC-
    SHA:EXP-EDH-RSA-DES-CBC-SHA:EXP-EDH-DSS-DES-CBC-SHA:EXP-DES-CBC-
    SHA:EXP-RC2-CBC-MD5:EXP-RC4-MD5
CIPHER is EDH-RSA-DES-CBC3-SHA

This is a test.

SSL3 alert read:warning:close notify
DONE
shutting down SSL
CONNECTION CLOSED
ACCEPT
```

S_Client (1 of 3)

```
Client> @ssl$com:ssl$utils
Client> s_client "-CAfile" ssl$certs:dwllng_ca.crt -state
CONNECTED(00000005)
SSL_connect:before/connect initialization
SSL_connect:SSLv2/v3 write client hello A
SSL_connect:SSLv3 read server hello A
depth=1 /C=US/O=Compaq Computer Corp/OU=OpenVMS/CN=DWLLNG CA
   Authority
verify return:1
depth=0 /C=US/ST=New Hampshire/L=Nashua/O=Hewlett Packard
/OU=OpenVMS/CN=dwllng.compaq.com/Email=webmaster@dwllng.compaq.com
verify return:1
SSL_connect:SSLv3 read server certificate A
SSL_connect:SSLv3 read server key exchange A
SSL_connect:SSLv3 read server done A
SSL_connect:SSLv3 write client key exchange A
SSL_connect:SSLv3 write change cipher spec A
SSL_connect:SSLv3 write finished A
SSL_connect:SSLv3 flush data
SSL_connect:SSLv3 read finished A
---
```

S_Client (2 of 3)

```
Certificate chain
 0 s:/C=US/ST=New Hampshire/L=Nashua/O=Hewlett Packard
/OU=OpenVMS/CN=dwllng.compaq.com/Email=webmaster@dwllng.compaq.com
i:/C=US/O=Compaq Computer Corp/OU=OpenVMS/CN=DWLLNG CA Authority
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDTzCCArigAwIBAgI
…
/bsxw7IvIJ4=
-----END CERTIFICATE-----
subject=/C=US/ST=New Hampshire/L=Nashua/O=Hewlett Packard
/OU=OpenVMS/CN=dwllng.compaq.com/Email=webmaster@dwllng.compaq.com
issuer=/C=US/O=Compaq Computer Corp/OU=OpenVMS/CN=DWLLNG CA
   Authority
---No client certificate CA names sent---
SSL handshake has read 1279 bytes and written 250 bytes
---
```

S_Client (3 of 3)

```
New, TLSv1/SSLv3, Cipher is EDH-RSA-DES-CBC3-SHA

Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : EDH-RSA-DES-CBC3-SHA
    Session-ID: E914688EA19D97E593775A2EDB9E8887891305265C95A0105033758A927BB9BC
    Session-ID-ctx:
    Master-Key: 2DEA3A1FDE90226F736F652031EB5B6F3F32D3421F6303D6664B5487421B57…
    Key-Arg   : None
    Start Time: 1036438609
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
This is a test.
Q
DONE
SSL3 alert write:warning:close notify
```

Command line utility – ENCrypting & decrypting

```
$ @ssl$com:ssl$utils
$ openssl enc -des3 -salt -in sys$login:login.com -out sys$login:login.enc
enter des-ede3-cbc encryption password:
Verifying password - enter des-ede3-cbc encryption password:
$
$ openssl enc -d -des3 -in sys$login:login.enc -out sys$login:login.dec
enter des-ede3-cbc decryption password:
$
$ diff sys$login:login.dec sys$login:login.com
Number of difference sections found: 0
Number of difference records found: 0

DIFFERENCES /IGNORE=()/MERGED=1-
    SYS$SYSROOT:[SYSMGR]LOGIN.DEC;1-
    SYS$SYSROOT:[SYSMGR]LOGIN.COM;12
$
```

Command line utility - RSA public & private keys

```
$ @ssl$com:ssl$utils
$ openssl genrsa -out privatekey.pem -des3 1024
Generating RSA private key, 1024 bit long modulus
.....................................................++++++
.....................++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
$
$ openssl rsa -in privatekey.pem -pubout -out publickey.pem
read RSA key
Enter PEM pass phrase:
writing RSA key
$
```

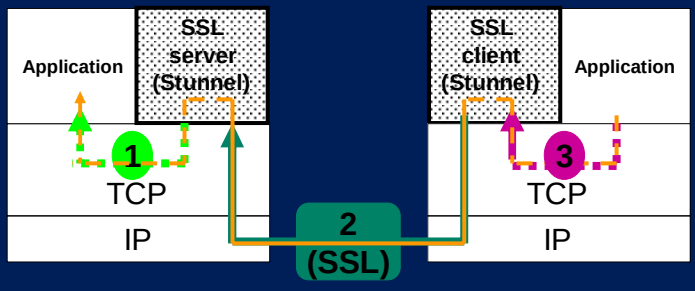
Command line utility – sign & verify using SHA1

```
$ openssl sha1 -sign privatekey.pem -out loginsign.bin login.com
Enter PEM pass phrase
$
$ openssl sha1 -verify publickey.pem -signature loginsign.bin login.com
Verified OK
$
```

Stunnel (Secure Tunnel)

- Stunnel is a program that allows you to encrypt arbitrary TCP connections inside an SSL (secure sockets layer) connection from your OpenVMS system to any other Stunnel capable machine
- Stunnel allows you to secure non-SSL aware applications (like telnet, ftp, RCP, IMAP, etc) by having Stunnel provide the encryption and not requiring changes to the original application
- Alpha only
- Tested on OpenVMS version 7.2-2 and up
- Requires “Compaq SSL for OpenVMS alpha V1.0”
- Needs “Compaq/DEC C for OpenVMS V6.0” or higher to build from source
- http://www.openvms.compaq.com/opensource/

Using Stunnel (telnet example)



1. SSL server: (stunnel -d 992 -r localhost:23 -p stunnel.pem)

2. SSL client: (stunnel -c -d 992 -r remote:992)

3. Application: (telnet localhost 992)

Using Stunnel (ftp example)

1.) Start Stunnel server
($ stunnel -d 990 -r 192.168.0.1:21 –p stunnel.pem)
2.) Start Stunnel client
($ stunnel -c -d 990 -r 192.168.0.1:990)
3.) Start FTP (client) at the host running Stunnel client
($ ftp 192.168.0.2 990)

Reference

- *SSL and TLS: Designing and Building Secure Systems* by Eric Rescorla
- *Network Security with OpenSSL:Cryptography for Secure Communications* by John Viega, Matt Messier & Pravir Chandra
- *Open Source Security for OpenVMS Alpha Vol 2: Compaq SSL (Secure Sockets Layer) for OpenVMS Alpha*

Reference (Cont'd)

- www.openvms.compaq.com/openvms/products/ssl/ssl.html
- www.openvms.compaq.com/openvms/products/ssl/ssl_source.html
- www.openssl.org
- wp.netscape.com/eng/ssl3/
- www.ietf.org/rfc/rfc2246.txt
- www.tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html

- www.openvms.compaq.com/openvms/security.html

Questions?

Questions ? ? ?


invent