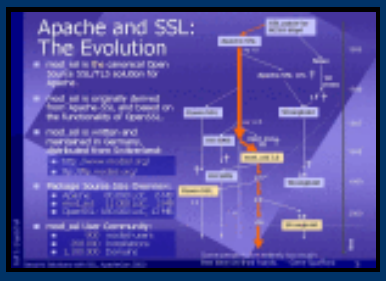
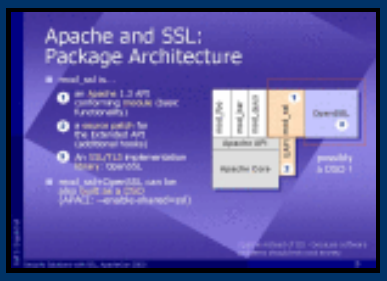
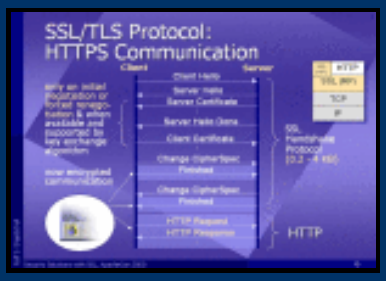
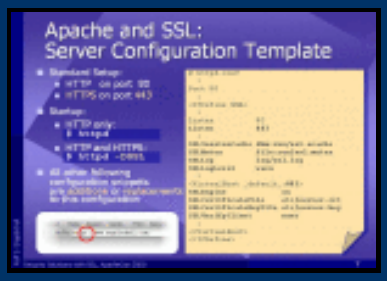
Security Solutions with SSL

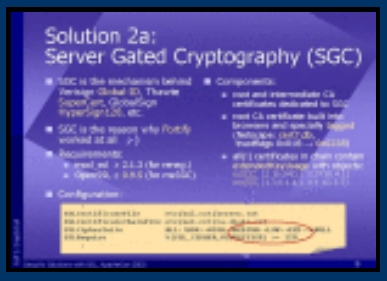
ApacheCon 2000, Ralf S. Engelschall

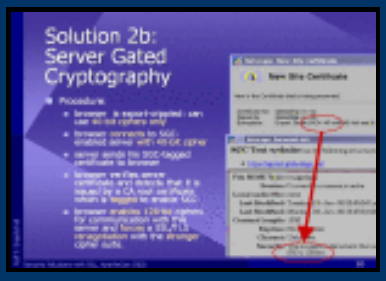


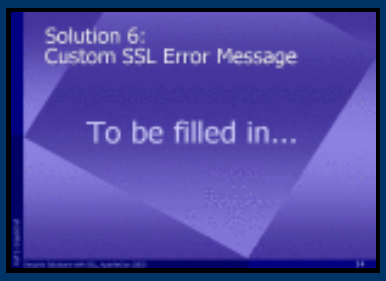


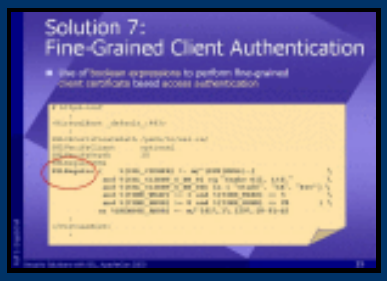


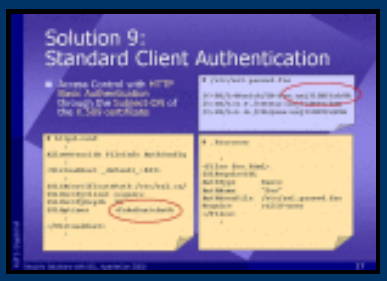


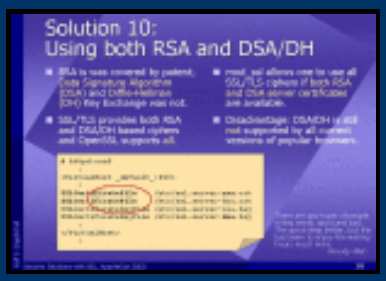


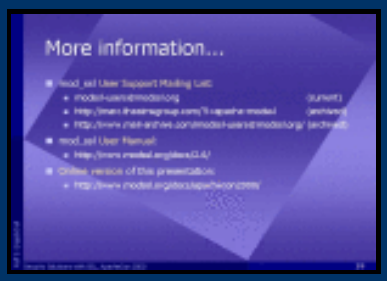












Security Solutions with SSL

Selected security solutions for the Apache webserver, based on the SSL/TLS protocol

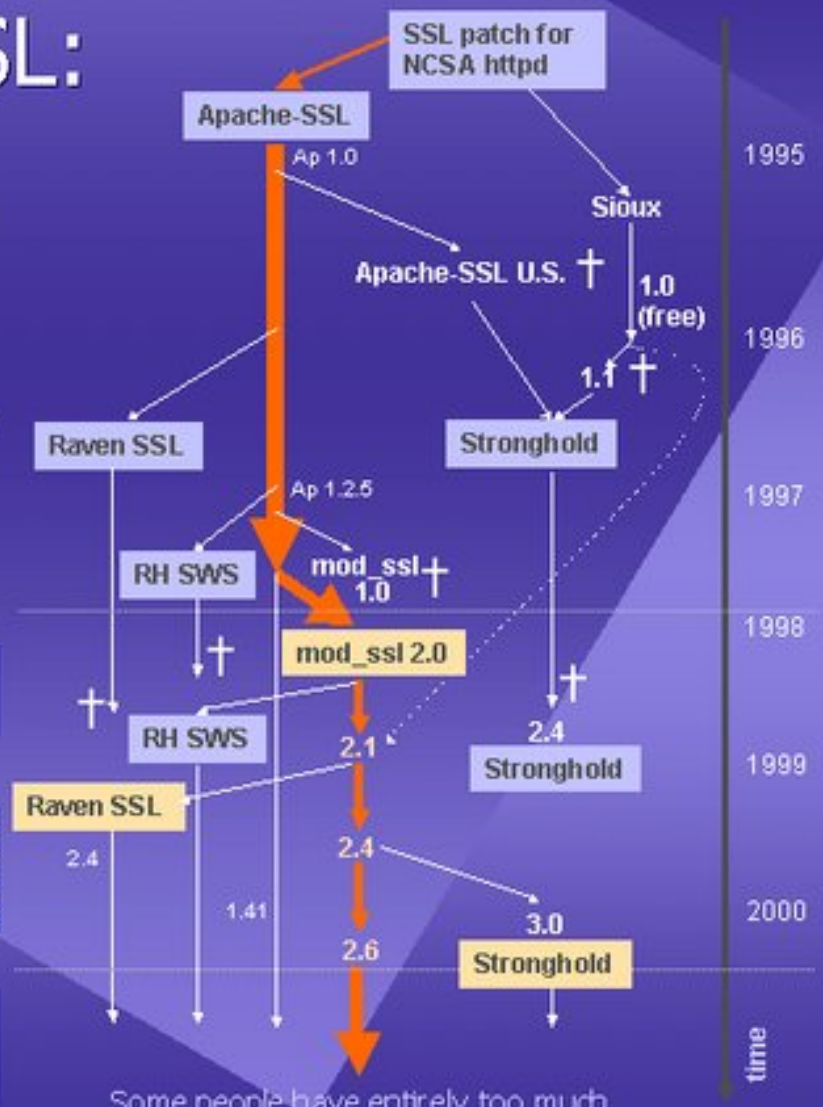
M21 APACHECON 2000

Mon, 23 Oct 2000 18:30-20:00

Ralf S. Engelschall

Apache and SSL: The Evolution

- mod_ssl is the canonical Open Source SSL/TLS solution for Apache.
- mod_ssl is originally derived from Apache-SSL and based on the functionality of OpenSSL.
- mod_ssl is written and maintained in Germany, distributed from Switzerland:
  - http://www.modssl.org/
  - ftp://ftp.modssl.org/
- Package Source Size Overview:
  - Apache: 80.000 LoC, 6 MB
  - mod_ssl: 11.000 LoC, 3 MB
  - OpenSSL: 180.000 LoC, 12 MB
- mod_ssl User Community:
  - 900 modssl-users
  - 200.000 Installations
  - 1,100.000 Domains



Some people have entirely too much free time on their hands. - Gene Spafford

Apache and SSL: Installation Illustration

Seek simplicity,
but distrust it.
- A. N. Whitehead

```
$ gunzip <apache_1.3.12.tar.gz         | tar xf -
$ gunzip <mod_ssl-2.6.6-1.3.12.tar.gz | tar xf -
$ gunzip <openssl-0.9.5a.tar.gz        | tar xf -

$ cd openssl-0.9.5a
$ ./config
$ make

$ cd ../mod_ssl-2.6.6
$ ./configure --with-apache=../apache_1.3.12 \
              --with-ssl=../openssl-0.9.5a \
              --prefix=/usr/local/apache

$ cd ../apache_1.3.12
$ make
$ make certificate TYPE=dummy
$ make install

$ /usr/local/apache/bin/httpd -DSSL
$ netscape https://localhost/
```

1. extract sources
2. build OpenSSL
3. apply mod_ssl to Apache source tree and configure it
4. build and install Apache plus mod_ssl
5. test the server!

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-003-n.html [9/23/2000 4:54:30 PM]

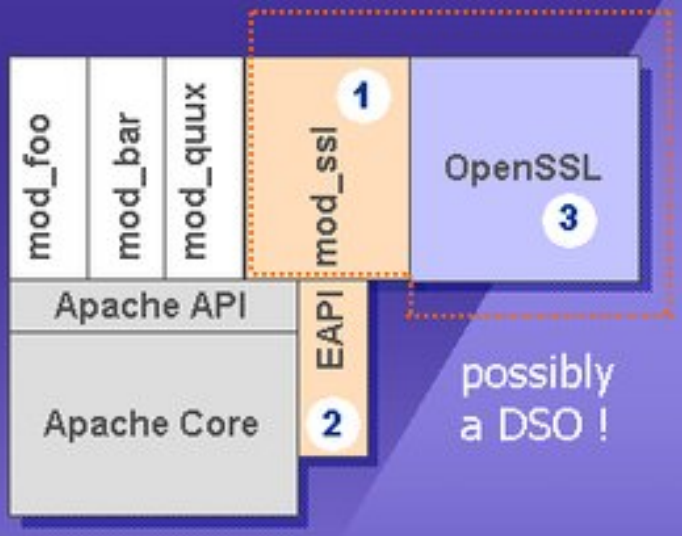
Apache and SSL: Feature Overview

mod_ssl - The best SSL solution for Apache money can't buy.

- Open-Source software
- Useable for commercial and non-commercial use
- Available for both Unix and Win32
- 128-bit strong crypto world-wide
- Supports SSLv2, SSLv3 and TLSv1
- Very clean reviewable ANSI C source code
- Clean Apache module architecture
- Integrates seamlessly into Apache through Extended API (EAPI)
- Full Dynamic Shared Object (DSO) support via mod_so+EAPI
- Advanced pass-phrase handling for private keys
- X.509 based client & server authentication
- X.509 Certificate Revocation List (CRL) support
- per-URL renegotiation of SSL /TLS parameters
- Supports RSA and DSA/DH ciphers
- Support for explicit PRNG seeding
- Boolean-expression based access control facility
- Backward compatibility to other SSL solutions (Apache-SSL, SH, ...)
- Inter-process SSL session cache (DBM or Shared Memory)
- Powerful dedicated SSL engine logging facility
- Simple and robust application to Apache source
- Fully integrated into Apache 1.3 configuration mechanism
- Additionally integration into APACI
- Assistance in X.509v3 certificate generation (RSA and DSA)

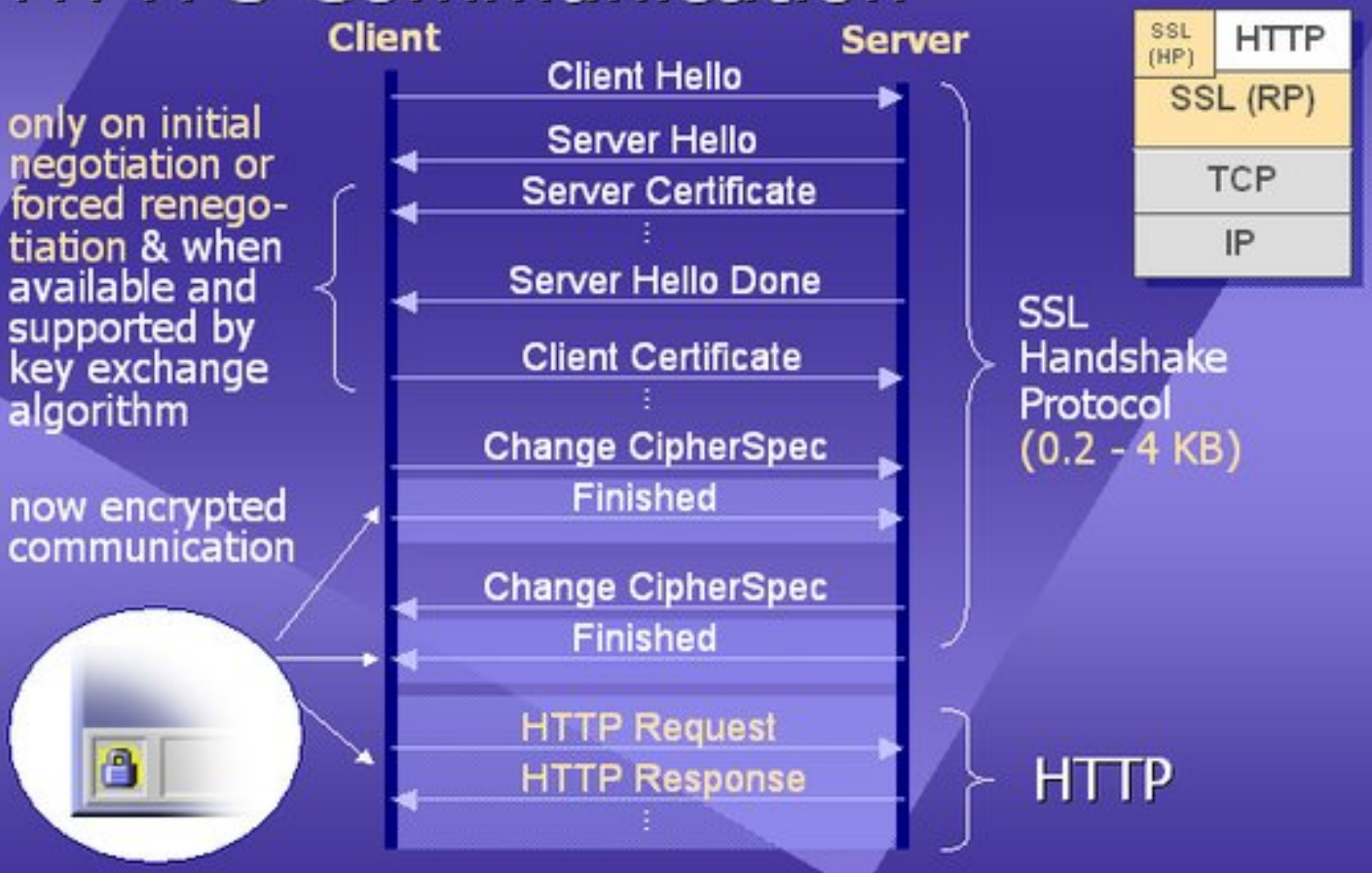
Apache and SSL: Package Architecture

- mod_ssl is...
  1. an Apache 1.3 API conforming module (basic functionality)
  2. a source patch for the Extended API (additional hooks)
  3. An SSL/TLS implementation library: OpenSSL
- mod_ssl+OpenSSL can be also built as a DSO (APACI: --enable-shared=ssl)



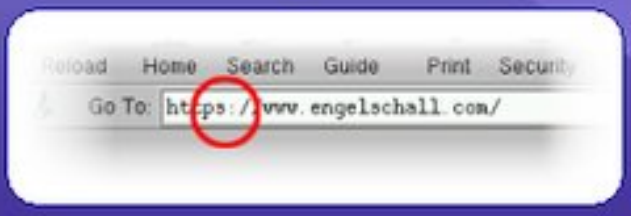
Apache instead of IIS - because software problems should not cost money.

SSL/TLS Protocol: HTTPS Communication



Apache and SSL: Server Configuration Template

- Standard Setup:
  - HTTP on port 80
  - HTTPS on port 443
- Startup:
  - HTTP only:
    ```
    $ httpd
    ```
  - HTTP and HTTPS:
    ```
    $ httpd -DSSL
    ```
- All other following configuration snippets are additions or replacements to this configuration




```
# httpd.conf
  :
Port 80
  :
<IfDefine SSL>
  :
Listen          80
Listen          443
  :
SSLSessionCache dbm:run/ssl.scache
SSLMutex        file:run/ssl.mutex
SSLLog          log/ssl.log
SSLLogLevel     warn
  :
<VirtualHost _default_:443>
SSLEngine               on
SSLCertificateFile      etc/server.crt
SSLCertificateKeyFile   etc/server.key
SSLVerifyClient         none
  :
</VirtualHost>
</IfDefine>
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-007-n.html [9/23/2000 4:54:43 PM]

Solution 1: Speed Up Processing

- Use Shared Memory based SSL/TLS Session Cache:
  MM: Shared Memory Library, http://www.engelschall.com/sw/mm/
  Build-Time: (mod_ssl) `configure --with-mm=../mm-1.1.3`
  Run-Time:
  ```
  SSLSessionCache shm:/var/run/scache(512000)
  ```
- Increase Session Cache Timeout:
  Run-Time:
  ```
  SSLSessionCacheTimeout 1200
  ```
- Use Semaphore based Mutex:
  Run-Time:
  ```
  SSLMutex sem
  ```
- Use minimal logging:
  Run-Time:
  ```
  SSLLog      /dev/null
  SSLLogLevel error
  ```
- Export SSI/CGI Variables only for reasonable contexts:
  Run-Time:
  ```
  <Files ~ "*.(cgi|shtml)$">
  SSLOptions +StdEnvVars
  </Files>
  ```
- Use optimized renegotiations:
  Run-Time:
  ```
  SSLOptions +OptRenegotiate
  ```
- Use HTTP Keepalive facility:
  Run-Time:
  ```
  KeepAlive             on
  MaxKeepAliveRequests  100
  KeepAliveTimeout      15
  ```

Premature optimization is the root of all evil.
- D.E.Knuth

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-008-n.html [9/23/2000 4:54:46 PM]

Solution 2a: Server Gated Cryptography (SGC)

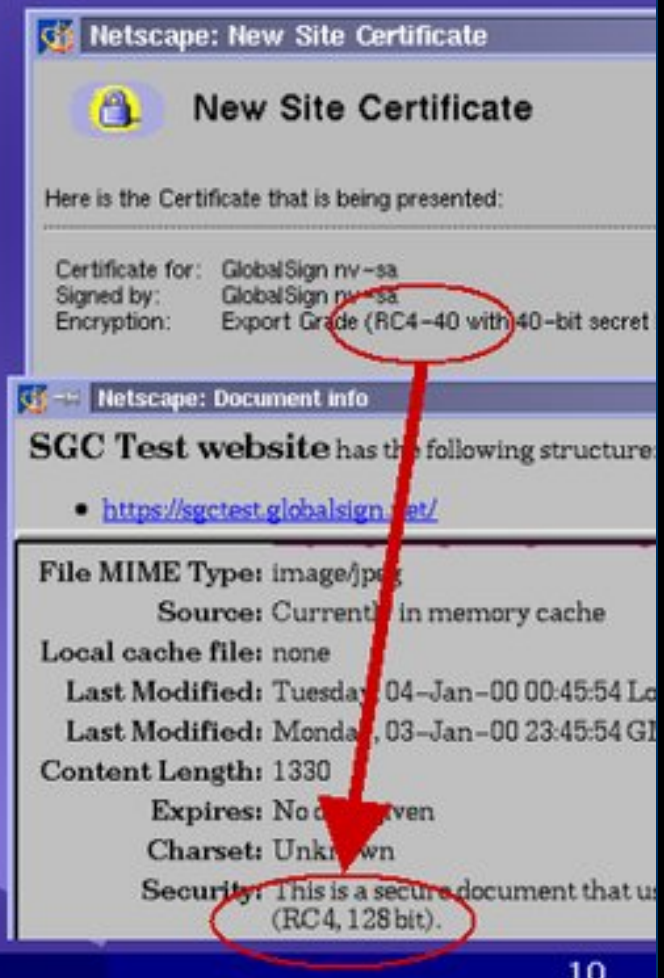
- SGC is the mechanism behind Verisign Global-ID, Thawte SuperCert, GlobalSign HyperSign128, etc.
- SGC is the reason why Fortify worked at all ;-)
- Requirements:
  - mod_ssl ≥ 2.1.3 (for reneg.)
  - OpenSSL ≥ 0.9.5 (for msSGC)
- Components:
  - root and intermediate CA certificates dedicated to SGC
  - root CA certificate built into browsers and specially tagged (Netscape: cert7.db, trustflags 0x018 → 0x0238)
  - all(!) certificates in chain contain extendedKeyUsage with objects:
    nsSGC (2.16.840.1.113730.4.1)
    msSGC (1.3.6.1.4.1.311.10.3.3)
- Configuration:

```
    :
SSLCertificateFile      etc/ssl.crt/server.crt
SSLCertificateChainFile etc/ssl.crt/ca-chain.crt
SSLCipherSuite          ALL:!ADH:+HIGH:+MEDIUM:+LOW:+EXP:+eNULL
SSLRequire              %{SSL_CIPHER_USEKEYSIZE} >= 128
    :
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-009-n.html [9/23/2000 4:54:48 PM]

Solution 2b: Server Gated Cryptography

- Procedure:
  - browser is export-crippled: can use 40-bit ciphers only
  - browser connects to SGC-enabled server with 40-bit cipher
  - server sends his SGC-tagged certificate to browser
  - browser verifies server certificate and detects that it is issued by a CA root certificate which is tagged to enable SGC
  - browser enables 128-bit ciphers for communication with this server and forces a SSL/TLS renegotiation with the stronger cipher suite.



Solution 3: HTTP/HTTPS Gateway

To be filled in...

Solution 4: Per-Directory SSL/TLS Parameters

To be filled in...

Solution 5: Internet/Intranet Access

To be filled in...

Solution 6:
Custom SSL Error Message

To be filled in...

Solution 7: Fine-Grained Client Authentication

- Use of boolean expressions to perform fine-grained client certificate based access authentication

```
# httpd.conf
    :
<VirtualHost _default_:443>
    :
SSLCACertificatePath /path/to/ssl.ca/
SSLVerifyClient      optional
SSLVerifyDepth       10
SSLRequireSSL
SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)-/                  \
           and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd."          \
           and %{SSL_CLIENT_S_DN_OU} in { "Staff", "CA", "Dev" } \
           and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5            \
           and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20        ) \
          or %{REMOTE_ADDR} =~ m/^141\.1\.129\.[0-9]+$/
    :
</VirtualHost>
    :
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-015-n.html [9/23/2000 4:54:59 PM]

Solution 8:
Certificate Revocation Lists (CRL)

To be filled in...

Solution 9: Standard Client Authentication

- Access Control with HTTP Basic Authentication through the Subject-DN of the X.509 certificate

```
# /etc/ssl.passwd.foo

/C=DE/L=Munich/CN=Foo:xxj31ZMTZzkVA
/C=US/L=S.F./CN=Bar:xxj31ZMTZzkVA
/C=US/L=L.A./CN=Quux:xxj31ZMTZzkVA
```

```
# httpd.conf
    :
AllowOverride FileInfo AuthConfig
    :
<VirtualHost _default_:443>
    :
SSLCACertificatePath /etc/ssl.ca/
SSLVerifyClient require
SSLVerifyDepth  10
SSLOptions      +FakeBasicAuth
    :
</VirtualHost>
    :
```

```
# .htaccess

    :
<Files foo.html>
SSLRequireSSL
AuthType        Basic
AuthName        "foo"
AuthUserFile    /etc/ssl.passwd.foo
Require         valid-user
</Files>
    :
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-017-n.html [9/23/2000 4:55:04 PM]

Solution 10: Using both RSA and DSA/DH

- RSA is was covered by patent, Data Signature Algorithm (DSA) and Diffie-Hellman (DH) Key Exchange was not.
- SSL/TLS provides both RSA and DSA/DH based ciphers and OpenSSL supports all.
- mod_ssl allows one to use all SSL/TLS ciphers if both RSA and DSA server certificates are available.
- Disadvantage: DSA/DH is still not supported by all current versions of popular browsers.

```
# httpd.conf
    :
<VirtualHost _default_:443>
    :
SSLCertificateFile     /etc/ssl.server-rsa.crt
SSLCertificateFile     /etc/ssl.server-dsa.crt
SSLCertificateKeyFile /etc/ssl.server-rsa.key
SSLCertificateKeyFile /etc/ssl.server-dsa.key
    :
</VirtualHost>
    :
```

There are two types of people in this world, good and bad. The good sleep better, but the bad seem to enjoy the waking hours much more.
Woody Allen

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-018-n.html [9/23/2000 4:55:06 PM]

More information...

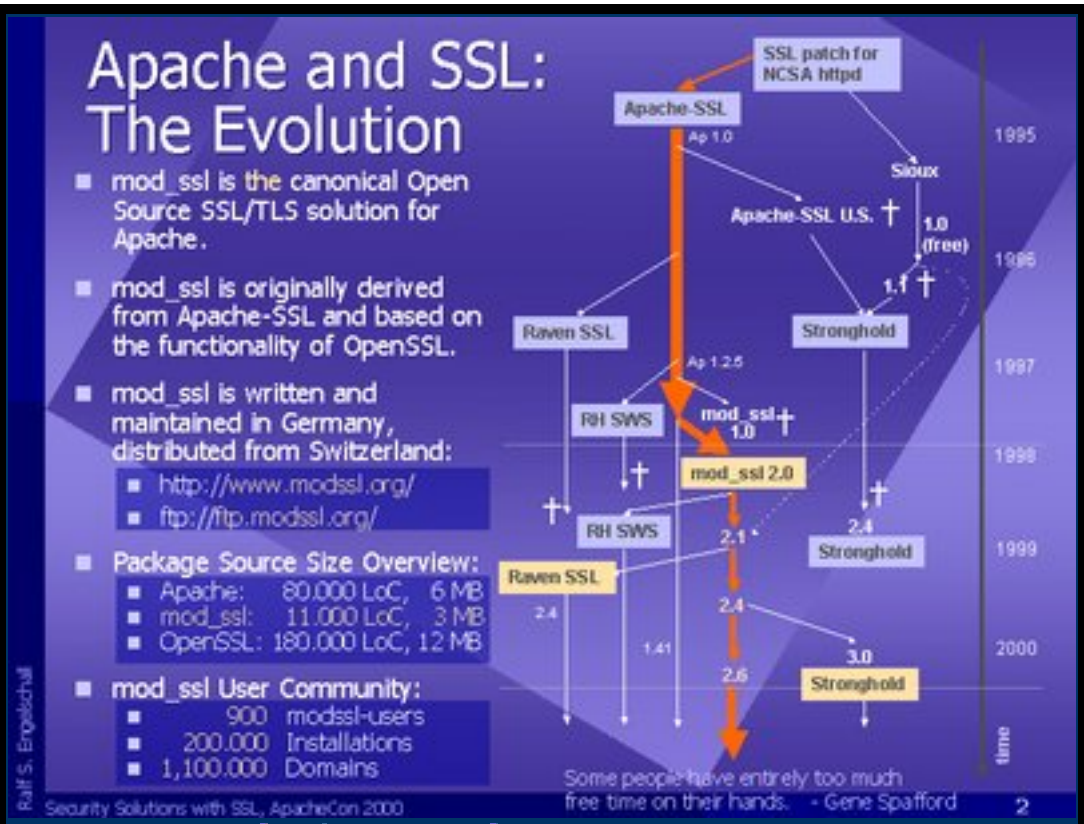
- mod_ssl User Support Mailing List:
  - modssl-users@modssl.org (current)
  - http://marc.theaimsgroup.com/?l=apache-modssl (archived)
  - http://www.mail-archive.com/modssl-users@modssl.org/ (archived)
- mod_ssl User Manual:
  - http://www.modssl.org/docs/2.6/
- Online version of this presentation:
  - http://www.modssl.org/docs/apachecon2000/

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-019-n.html [9/23/2000 4:55:08 PM]

More information...

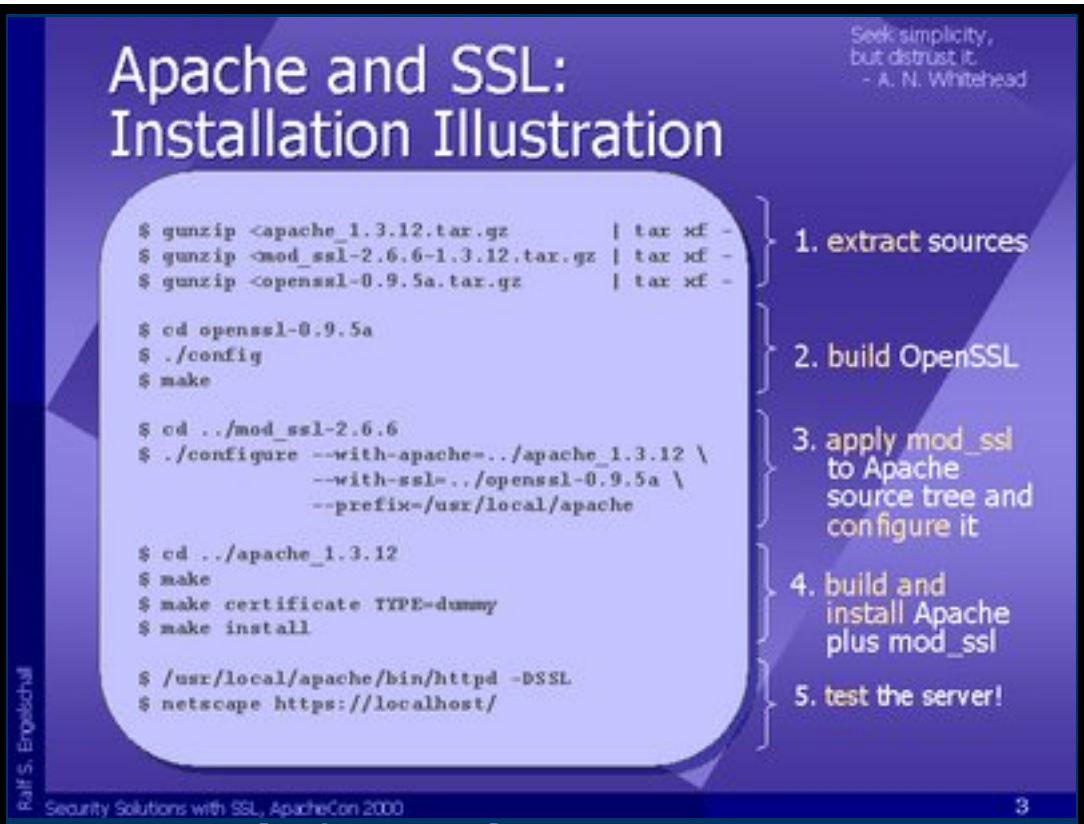
- mod_ssl User Support Mailing List:
  - modssl-users@modssl.org (current)
  - http://marc.theaimsgroup.com/?l=apache-modssl (archived)
  - http://www.mail-archive.com/modssl-users@modssl.org/ (archived)
- mod_ssl User Manual:
  - http://www.modssl.org/docs/2.6/
- Online version of this presentation:
  - http://www.modssl.org/docs/apachecon2000/

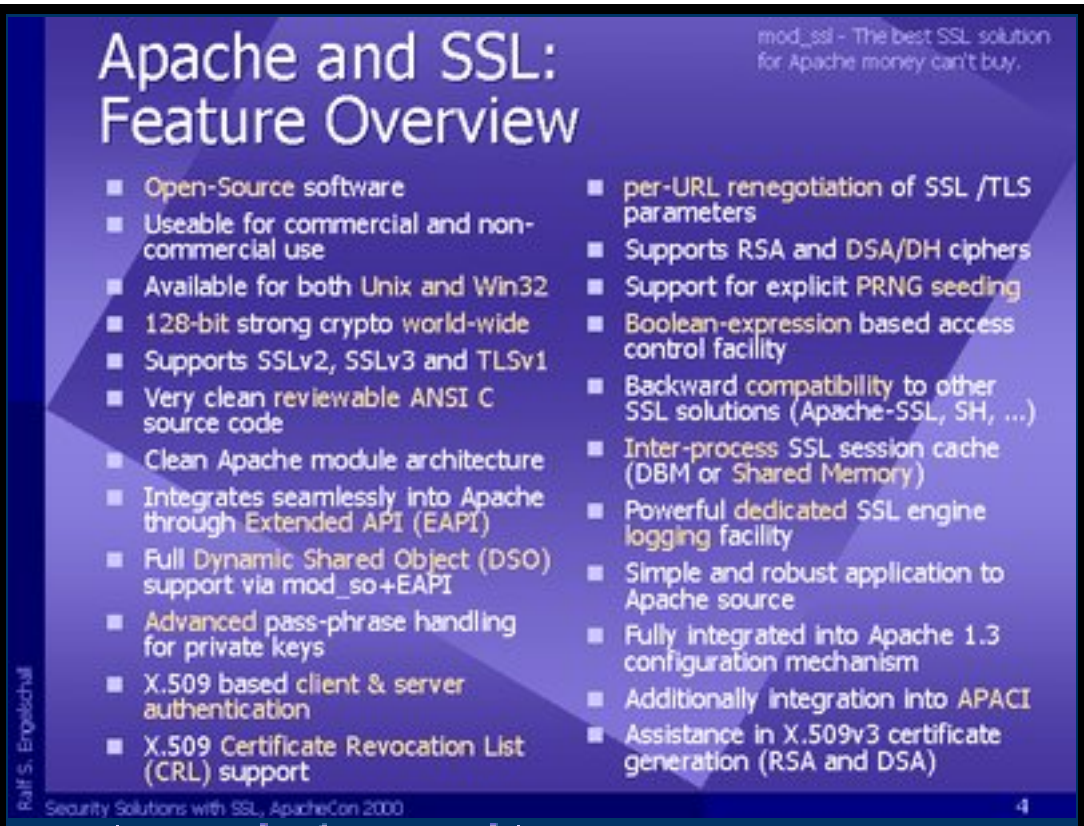

Security Solutions with SSL
Selected security solutions for the Apache webserver, based on the SSL/TLS protocol
M21
APACHECON 2000
Mon, 23 Oct 2000 18:30-20:00
mod_SSL
The Apache Interface To
OpenSSL
Engelschall
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
1



prev index next

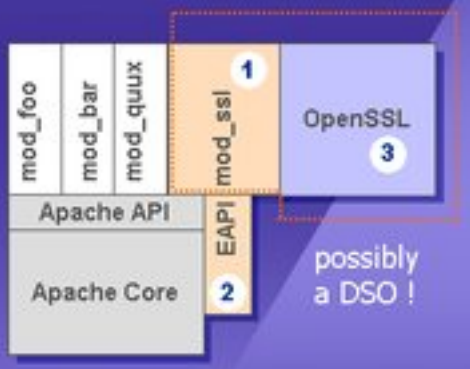
small | normal | large


Apache and SSL:
Installation Illustration
Seek simplicity,
but distrust it.
- A. N. Whitehead
$ gunzip <apache_1.3.12.tar.gz | tar xf -
$ gunzip <mod_ssl-2.6.6-1.3.12.tar.gz | tar xf -
$ gunzip <openssl-0.9.5a.tar.gz | tar xf -
$ cd openssl-0.9.5a
$ ./config
$ make
$ cd ../mod_ssl-2.6.6
$ ./configure --with-apache=../apache_1.3.12 \
--with-ssl=../openssl-0.9.5a \
--prefix=/usr/local/apache
$ cd ../apache_1.3.12
$ make
$ make certificate TYPE=dummy
$ make install
$ /usr/local/apache/bin/httpd -DSSL
$ netscape https://localhost/
1. extract sources
2. build OpenSSL
3. apply mod_ssl to Apache source tree and configure it
4. build and install Apache plus mod_ssl
5. test the server!
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
3


Apache and SSL:
Feature Overview
mod_ssl - The best SSL solution for Apache money can't buy.
Open-Source software
Useable for commercial and non-commercial use
Available for both Unix and Win32
128-bit strong crypto world-wide
Supports SSLv2, SSLv3 and TLSv1
Very clean reviewable ANSI C source code
Clean Apache module architecture
Integrates seamlessly into Apache through Extended API (EAPI)
Full Dynamic Shared Object (DSO) support via mod_so+EAPI
Advanced pass-phrase handling for private keys
X.509 based client & server authentication
X.509 Certificate Revocation List (CRL) support
per-URL renegotiation of SSL /TLS parameters
Supports RSA and DSA/DH ciphers
Support for explicit PRNG seeding
Boolean-expression based access control facility
Backward compatibility to other SSL solutions (Apache-SSL, SH, ...)
Inter-process SSL session cache (DBM or Shared Memory)
Powerful dedicated SSL engine logging facility
Simple and robust application to Apache source
Fully integrated into Apache 1.3 configuration mechanism
Additionally integration into APACI
Assistance in X.509v3 certificate generation (RSA and DSA)
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
4

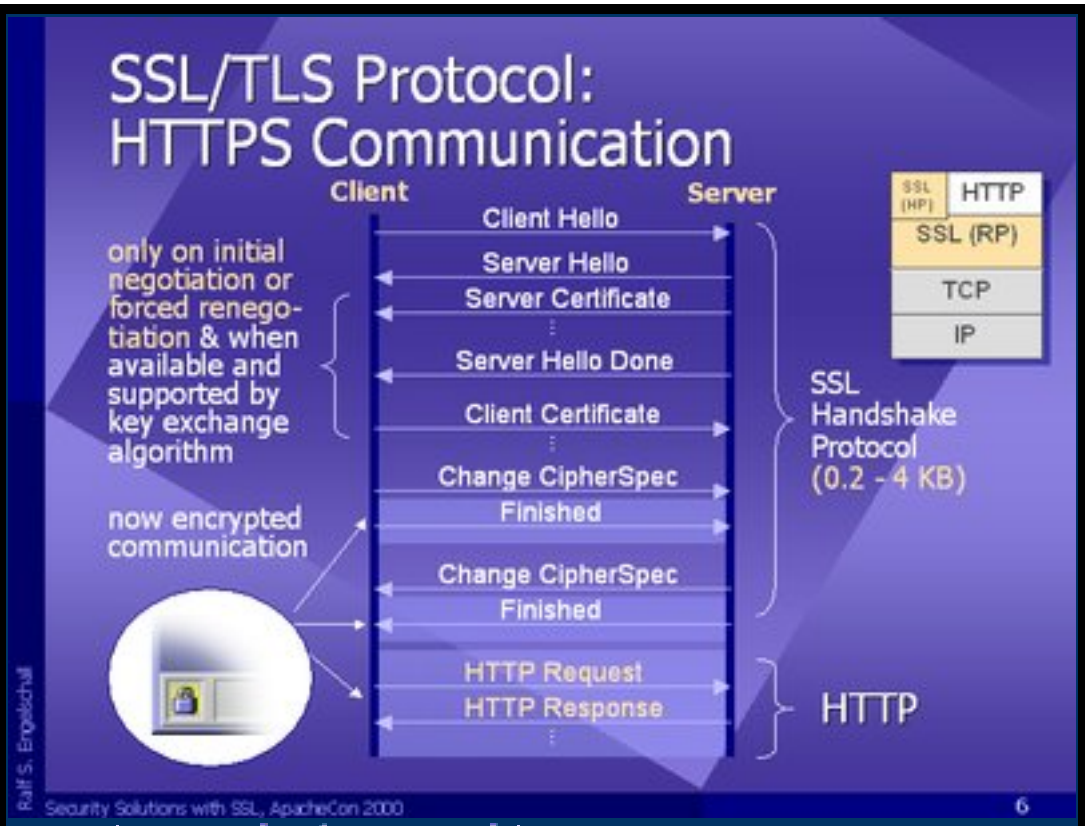
Apache and SSL: Package Architecture

- mod_ssl is...
  1. an Apache 1.3 API conforming module (basic functionality)
  2. a source patch for the Extended API (additional hooks)
  3. An SSL/TLS implementation library: OpenSSL
- mod_ssl+OpenSSL can be also built as a DSO (APACI: --enable-shared=ssl)



Apache instead of IIS - because software problems should not cost money.

prev index next
small | normal | large



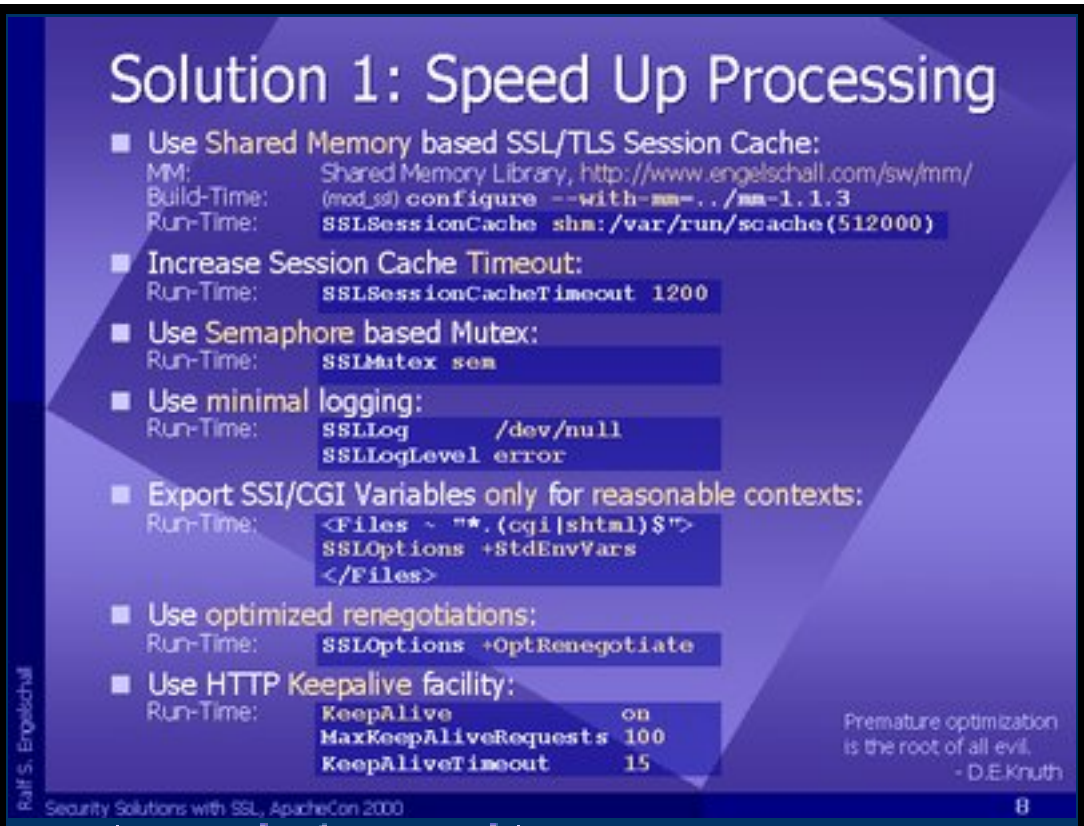
prev index next

small | normal | large

Apache and SSL: Server Configuration Template

- Standard Setup:
  - HTTP on port 80
  - HTTPS on port 443
- Startup:
  - HTTP only:
    `$ httpd`
  - HTTP and HTTPS:
    `$ httpd -DSSL`
- All other following configuration snippets are additions or replacements to this configuration

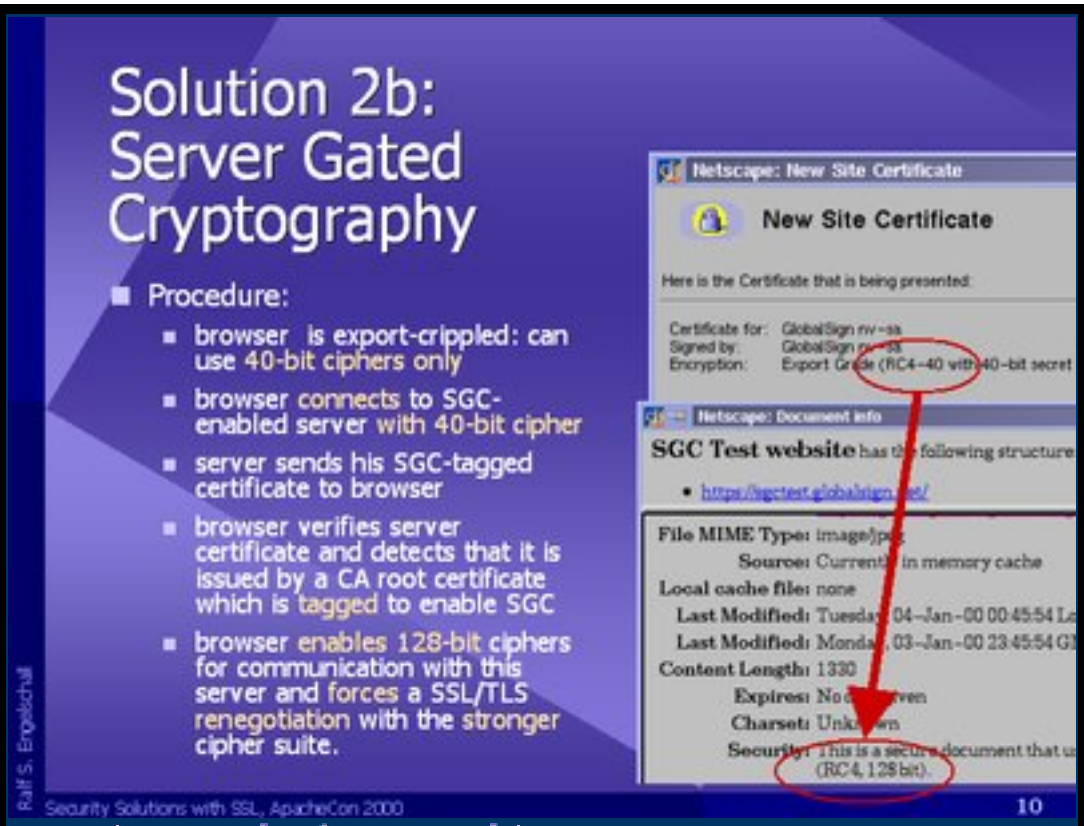

```
# httpd.conf
  :
Port 80
  :
<IfDefine SSL>
  :
Listen          80
Listen          443
  :
SSLSessionCache dbm:run/ssl.scache
SSLMutex        file:run/ssl.mutex
SSLLog          log/ssl.log
SSLLogLevel     warn
  :
<VirtualHost _default_:443>
SSLEngine               on
SSLCertificateFile      etc/server.crt
SSLCertificateKeyFile etc/server.key
SSLVerifyClient         none
  :
</VirtualHost>
</IfDefine>
```

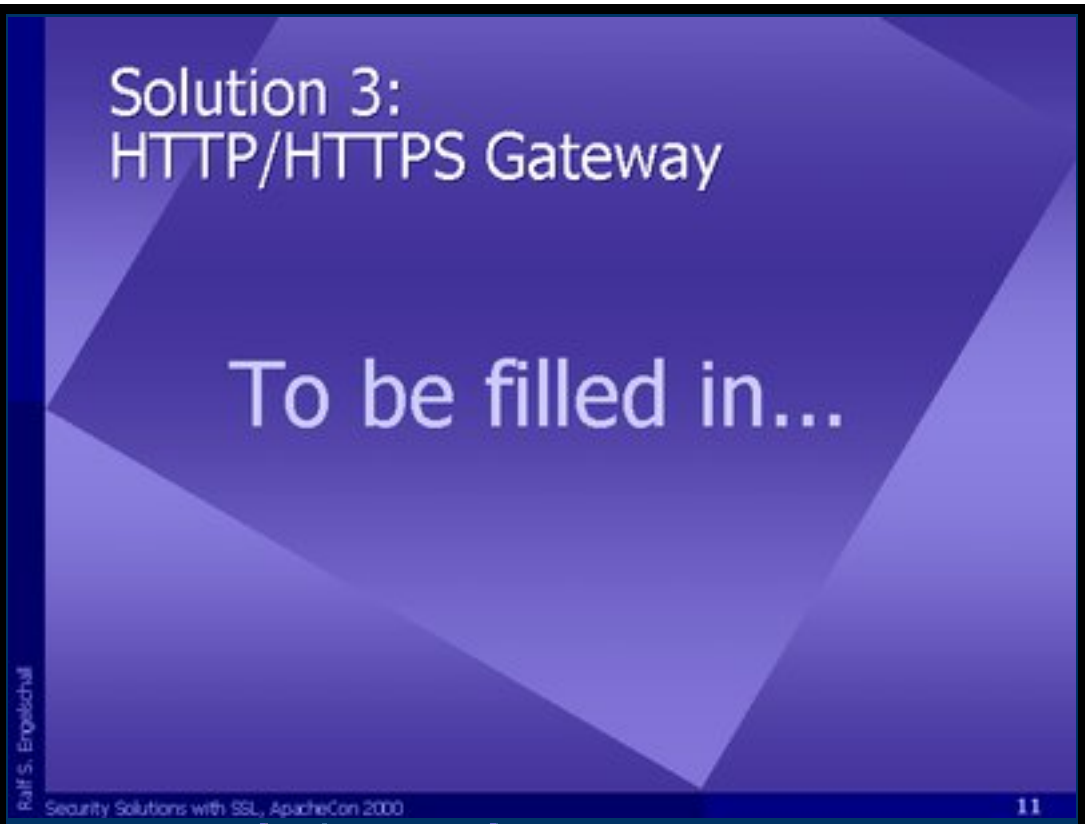

Solution 1: Speed Up Processing
Use Shared Memory based SSL/TLS Session Cache:
MM: Shared Memory Library, http://www.engelschall.com/sw/mm/
Build-Time: (mod_ssl) configure --with-mm=../mm-1.1.3
Run-Time: SSLSessionCache shm:/var/run/scache(512000)
Increase Session Cache Timeout:
Run-Time: SSLSessionCacheTimeout 1200
Use Semaphore based Mutex:
Run-Time: SSLMutex sem
Use minimal logging:
Run-Time: SSLLog /dev/null
SSLLogLevel error
Export SSI/CGI Variables only for reasonable contexts:
Run-Time: <Files ~ "*.(cgi|shtml)$">
SSLOptions +StdEnvVars
</Files>
Use optimized renegotiations:
Run-Time: SSLOptions +OptRenegotiate
Use HTTP Keepalive facility:
Run-Time: KeepAlive on
MaxKeepAliveRequests 100
KeepAliveTimeout 15
Premature optimization is the root of all evil.
- D.E.Knuth
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
8

Solution 2a: Server Gated Cryptography (SGC)

- SGC is the mechanism behind Verisign Global-ID, Thawte SuperCert, GlobalSign HyperSign128, etc.
- SGC is the reason why Fortify worked at all ;-)
- Requirements:
  - mod_ssl ≥ 2.1.3 (for reneg.)
  - OpenSSL ≥ 0.9.5 (for msSGC)
- Components:
  - root and intermediate CA certificates dedicated to SGC
  - root CA certificate built into browsers and specially tagged (Netscape: cert7.db, trustflags 0x018 → 0x0238)
  - all(!) certificates in chain contain extendedKeyUsage with objects:
    nsSGC (2.16.840.1.113730.4.1)
    msSGC (1.3.6.1.4.1.311.10.3.3)
- Configuration:

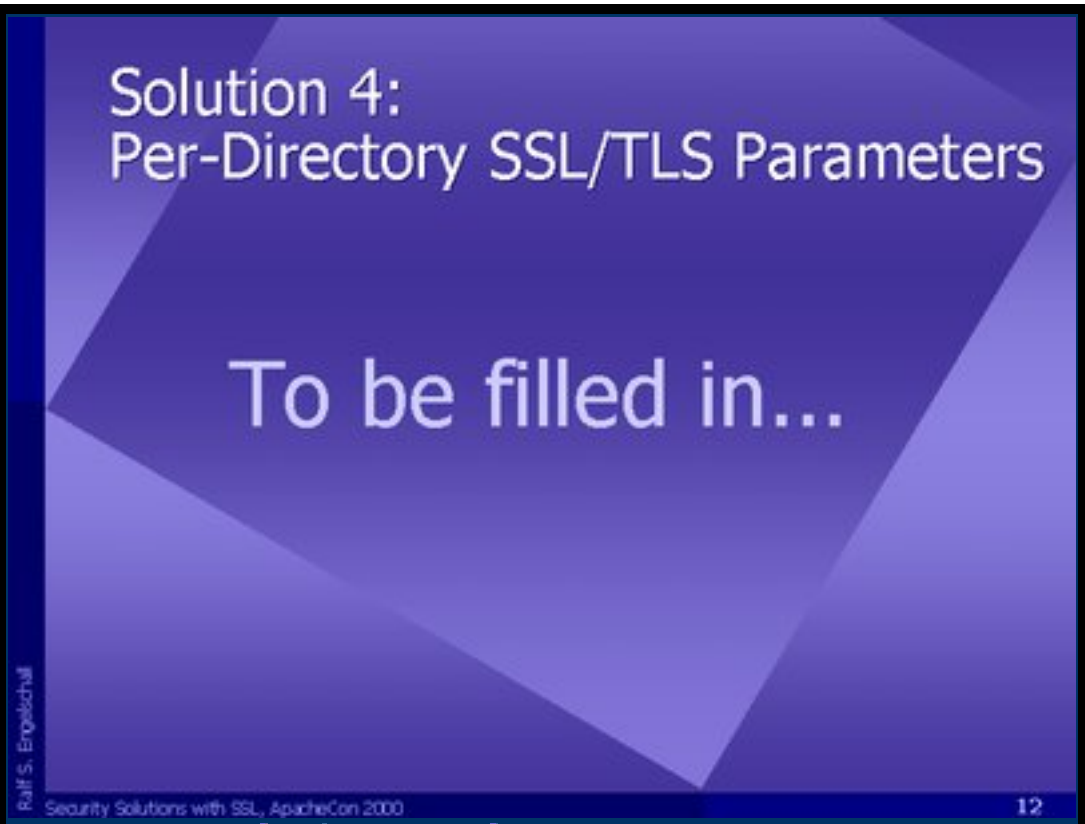
```
    :
SSLCertificateFile      etc/ssl.crt/server.crt
SSLCertificateChainFile etc/ssl.crt/ca-chain.crt
SSLCipherSuite          ALL:!ADH:+HIGH:+MEDIUM:+LOW:+EXP:+eNULL
SSLRequire              %{SSL_CIPHER_USEKEYSIZE} >= 128
    :
```


Solution 2b:
Server Gated
Cryptography
Procedure:
browser is export-crippled: can use 40-bit ciphers only
browser connects to SGC-enabled server with 40-bit cipher
server sends his SGC-tagged certificate to browser
browser verifies server certificate and detects that it is issued by a CA root certificate which is tagged to enable SGC
browser enables 128-bit ciphers for communication with this server and forces a SSL/TLS renegotiation with the stronger cipher suite.
Netscape: New Site Certificate
New Site Certificate
Here is the Certificate that is being presented:
Certificate for:
Signed by:
Encryption:
Netscape: Document Info
SGC Test website
File MIME Type:
Source:
Local cache file: none
Last Modified:
Last Modified:
Content Length: 1330
Expires:
Charset:
Security:
(RC4, 128 bit).
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
10



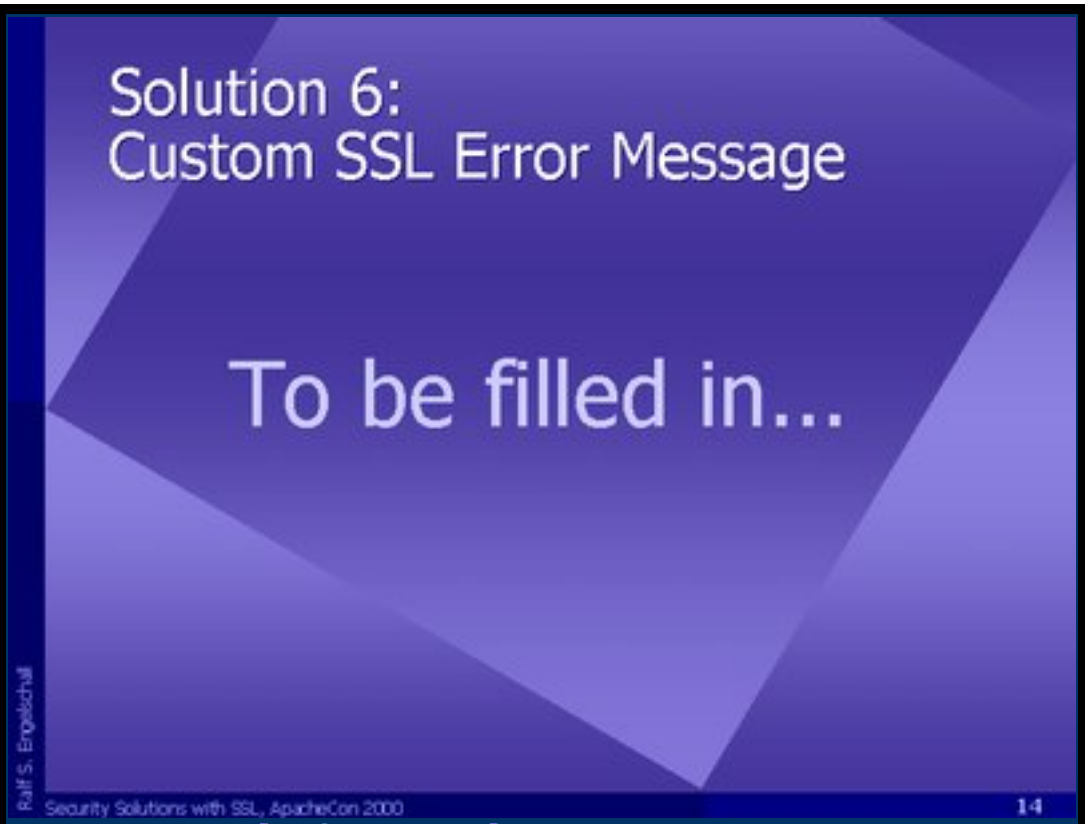
prev index next

small | normal | large


Solution 4:
Per-Directory SSL/TLS Parameters
To be filled in...
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
12

prev index next

small | normal | large



prev index next

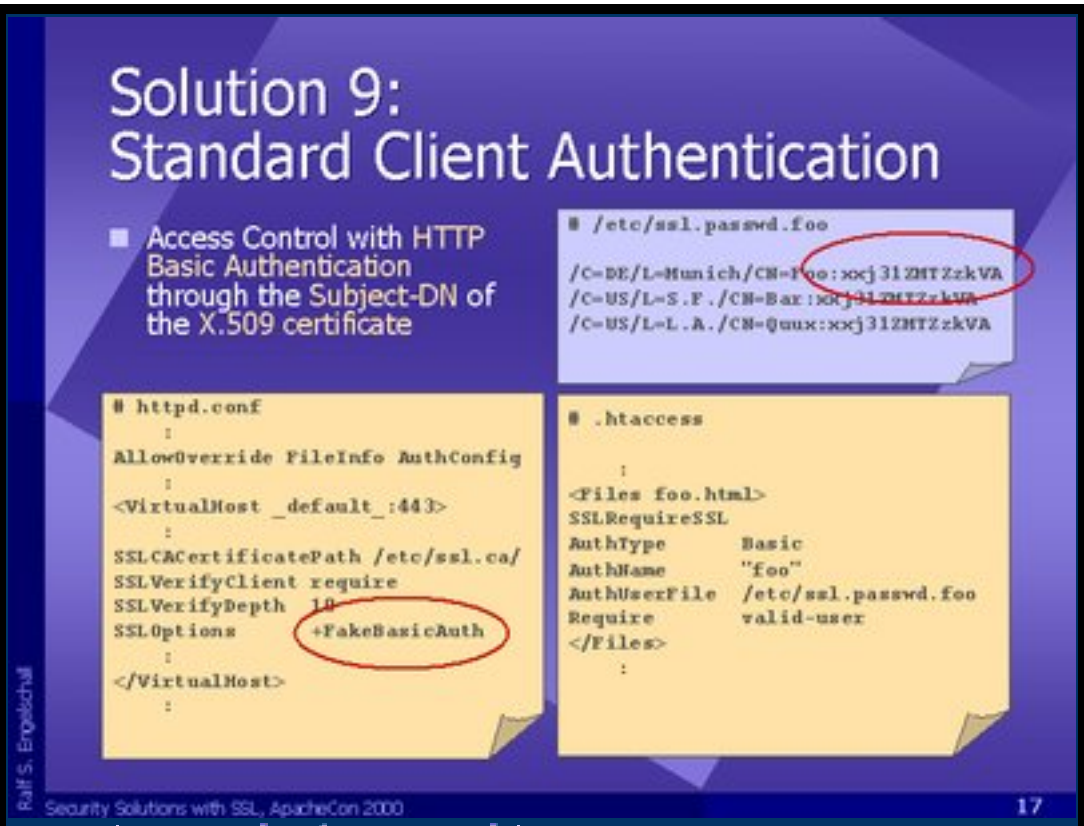
small | normal | large

Solution 7: Fine-Grained Client Authentication

- Use of boolean expressions to perform fine-grained client certificate based access authentication

```
# httpd.conf
    :
<VirtualHost _default_:443>
    :
SSLCACertificatePath /path/to/ssl.ca/
SSLVerifyClient      optional
SSLVerifyDepth       10
SSLRequireSSL
SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)-/                  \
           and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd."          \
           and %{SSL_CLIENT_S_DN_OU} in { "Staff", "CA", "Dev"} \
           and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5            \
           and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20        ) \
          or %{REMOTE_ADDR} =~ m/^141\.1\.129\.[0-9]+$/
    :
</VirtualHost>
    :
```


Solution 8:
Certificate Revocation Lists (CRL)
To be filled in...
Ralf S. Engelschall
Security Solutions with SSL, ApacheCon 2000
16



prev index next

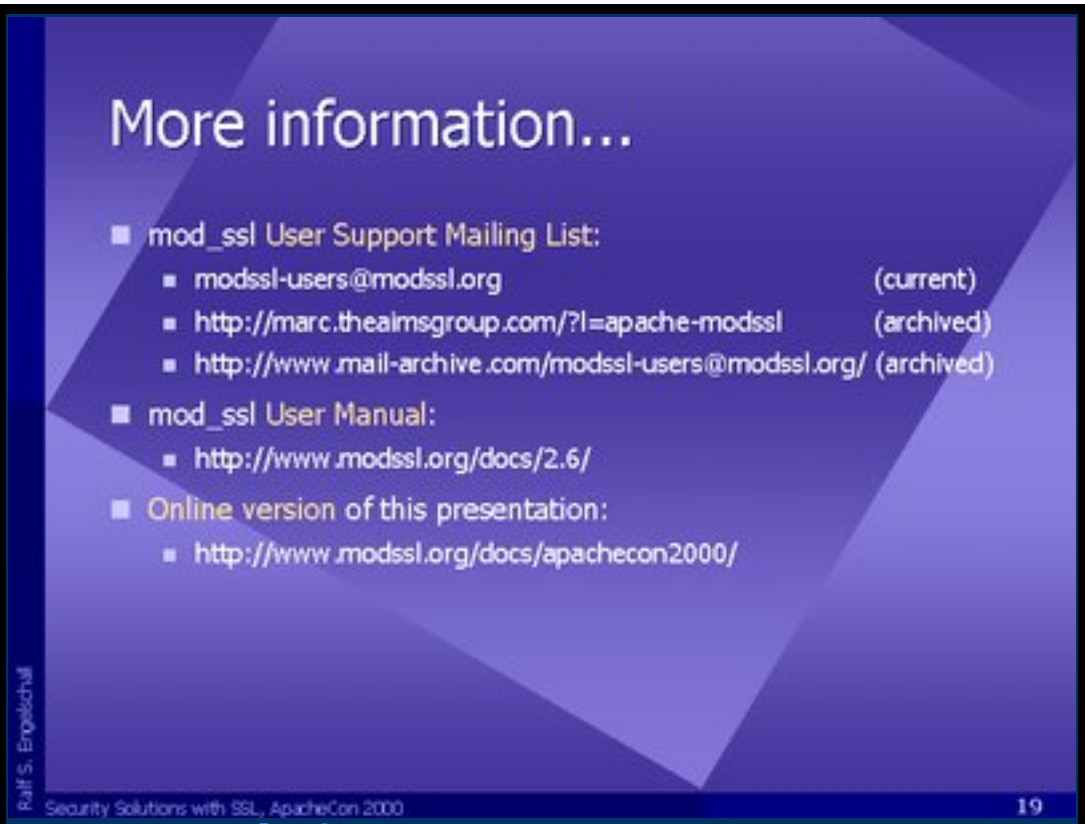
small | normal | large

Solution 10: Using both RSA and DSA/DH

- RSA is was covered by patent, Data Signature Algorithm (DSA) and Diffie-Hellman (DH) Key Exchange was not.
- SSL/TLS provides both RSA and DSA/DH based ciphers and OpenSSL supports all.
- mod_ssl allows one to use all SSL/TLS ciphers if both RSA and DSA server certificates are available.
- Disadvantage: DSA/DH is still not supported by all current versions of popular browsers.

```
# httpd.conf
    :
<VirtualHost _default_:443>
    :
SSLCertificateFile    /etc/ssl.server-rsa.crt
SSLCertificateFile    /etc/ssl.server-dsa.crt
SSLCertificateKeyFile /etc/ssl.server-rsa.key
SSLCertificateKeyFile /etc/ssl.server-dsa.key
    :
</VirtualHost>
    :
```

There are two types of people in this world, good and bad. The good sleep better, but the bad seem to enjoy the waking hours much more.
Woody Allen



prev index small | normal | large

Security Solutions with SSL

Selected security solutions for the Apache webserver, based on the SSL/TLS protocol

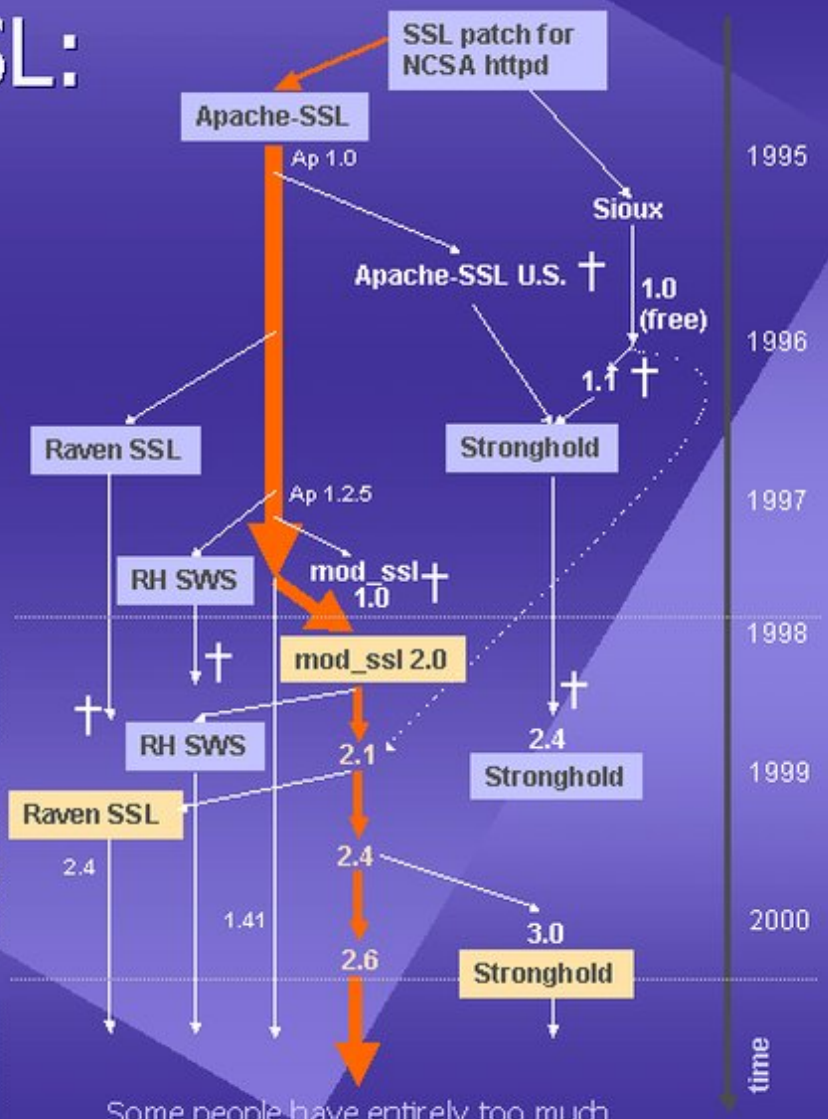
M21 APACHECON 2000

Mon, 23 Oct 2000 18:30-20:00

Ralf S. Engelschall

Apache and SSL: The Evolution

- mod_ssl is the canonical Open Source SSL/TLS solution for Apache.
- mod_ssl is originally derived from Apache-SSL and based on the functionality of OpenSSL.
- mod_ssl is written and maintained in Germany, distributed from Switzerland:
  - http://www.modssl.org/
  - ftp://ftp.modssl.org/
- Package Source Size Overview:
  - Apache: 80.000 LoC, 6 MB
  - mod_ssl: 11.000 LoC, 3 MB
  - OpenSSL: 180.000 LoC, 12 MB
- mod_ssl User Community:
  - 900 modssl-users
  - 200.000 Installations
  - 1,100.000 Domains



Some people have entirely too much free time on their hands. - Gene Spafford

Apache and SSL: Installation Illustration

Seek simplicity,
but distrust it.
- A. N. Whitehead

```
$ gunzip <apache_1.3.12.tar.gz         | tar xf -
$ gunzip <mod_ssl-2.6.6-1.3.12.tar.gz | tar xf -
$ gunzip <openssl-0.9.5a.tar.gz        | tar xf -

$ cd openssl-0.9.5a
$ ./config
$ make

$ cd ../mod_ssl-2.6.6
$ ./configure --with-apache=../apache_1.3.12 \
              --with-ssl=../openssl-0.9.5a \
              --prefix=/usr/local/apache

$ cd ../apache_1.3.12
$ make
$ make certificate TYPE=dummy
$ make install

$ /usr/local/apache/bin/httpd -DSSL
$ netscape https://localhost/
```

1. extract sources
2. build OpenSSL
3. apply mod_ssl to Apache source tree and configure it
4. build and install Apache plus mod_ssl
5. test the server!

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-003-l.html [9/23/2000 4:56:56 PM]

Apache and SSL: Feature Overview

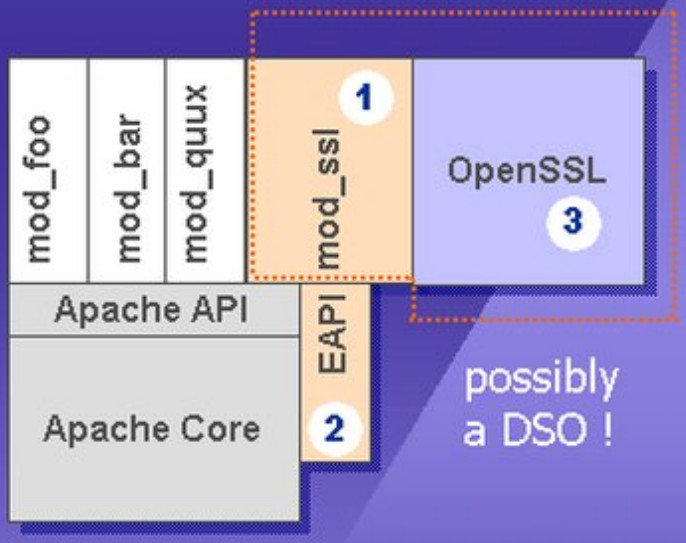
mod_ssl - The best SSL solution for Apache money can't buy.

- Open-Source software
- Useable for commercial and non-commercial use
- Available for both Unix and Win32
- 128-bit strong crypto world-wide
- Supports SSLv2, SSLv3 and TLSv1
- Very clean reviewable ANSI C source code
- Clean Apache module architecture
- Integrates seamlessly into Apache through Extended API (EAPI)
- Full Dynamic Shared Object (DSO) support via mod_so+EAPI
- Advanced pass-phrase handling for private keys
- X.509 based client & server authentication
- X.509 Certificate Revocation List (CRL) support
- per-URL renegotiation of SSL /TLS parameters
- Supports RSA and DSA/DH ciphers
- Support for explicit PRNG seeding
- Boolean-expression based access control facility
- Backward compatibility to other SSL solutions (Apache-SSL, SH, ...)
- Inter-process SSL session cache (DBM or Shared Memory)
- Powerful dedicated SSL engine logging facility
- Simple and robust application to Apache source
- Fully integrated into Apache 1.3 configuration mechanism
- Additionally integration into APACI
- Assistance in X.509v3 certificate generation (RSA and DSA)

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-004-l.html [9/23/2000 4:56:59 PM]

Apache and SSL: Package Architecture

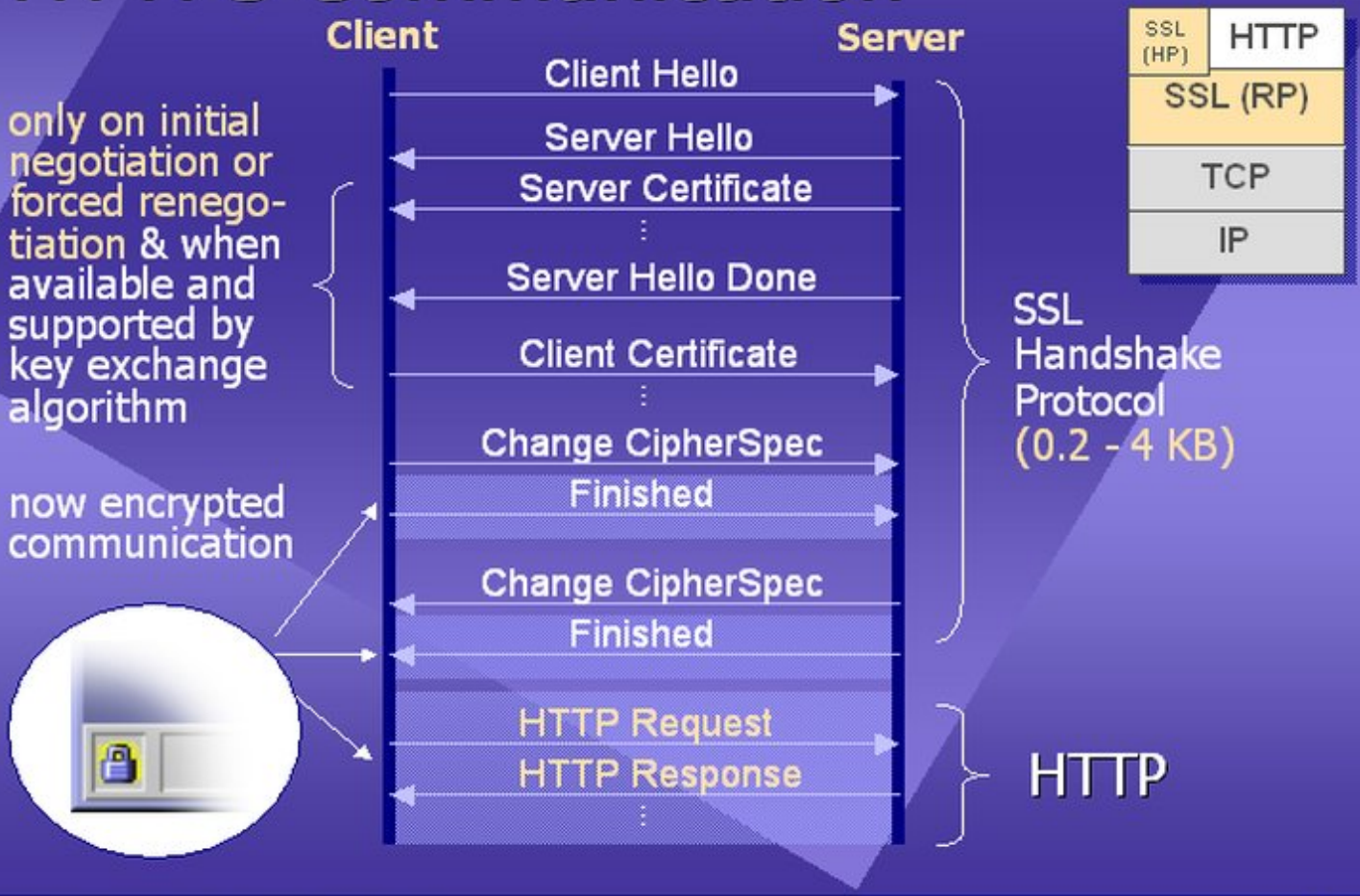
- mod_ssl is...
  1. an Apache 1.3 API conforming module (basic functionality)
  2. a source patch for the Extended API (additional hooks)
  3. An SSL/TLS implementation library: OpenSSL
- mod_ssl+OpenSSL can be also built as a DSO (APACI: --enable-shared=ssl)



Apache instead of IIS - because software problems should not cost money.

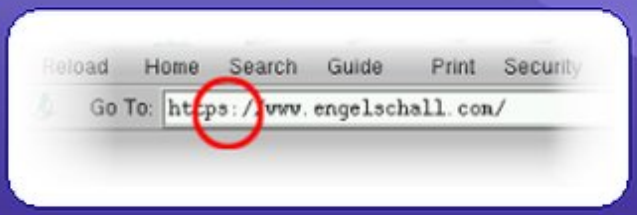
file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-005-l.html [9/23/2000 4:57:03 PM]

SSL/TLS Protocol: HTTPS Communication



Apache and SSL: Server Configuration Template

- Standard Setup:
  - HTTP on port 80
  - HTTPS on port 443
- Startup:
  - HTTP only:
    ```
    $ httpd
    ```
  - HTTP and HTTPS:
    ```
    $ httpd -DSSL
    ```
- All other following configuration snippets are additions or replacements to this configuration




```
# httpd.conf
  :
Port 80
  :
<IfDefine SSL>
  :
Listen          80
Listen          443
  :
SSLSessionCache dbm:run/ssl.scache
SSLMutex        file:run/ssl.mutex
SSLLog          log/ssl.log
SSLLogLevel     warn
  :
<VirtualHost _default_:443>
SSLEngine               on
SSLCertificateFile      etc/server.crt
SSLCertificateKeyFile   etc/server.key
SSLVerifyClient         none
  :
</VirtualHost>
</IfDefine>
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-007-l.html [9/23/2000 4:57:09 PM]

Solution 1: Speed Up Processing

- Use Shared Memory based SSL/TLS Session Cache:
  MM: Shared Memory Library, http://www.engelschall.com/sw/mm/
  Build-Time: (mod_ssl) `configure --with-mm=../mm-1.1.3`
  Run-Time:
  ```
  SSLSessionCache shm:/var/run/scache(512000)
  ```
- Increase Session Cache Timeout:
  Run-Time:
  ```
  SSLSessionCacheTimeout 1200
  ```
- Use Semaphore based Mutex:
  Run-Time:
  ```
  SSLMutex sem
  ```
- Use minimal logging:
  Run-Time:
  ```
  SSLLog      /dev/null
  SSLLogLevel error
  ```
- Export SSI/CGI Variables only for reasonable contexts:
  Run-Time:
  ```
  <Files ~ "*.(cgi|shtml)$">
  SSLOptions +StdEnvVars
  </Files>
  ```
- Use optimized renegotiations:
  Run-Time:
  ```
  SSLOptions +OptRenegotiate
  ```
- Use HTTP Keepalive facility:
  Run-Time:
  ```
  KeepAlive             on
  MaxKeepAliveRequests  100
  KeepAliveTimeout      15
  ```

Premature optimization is the root of all evil.
- D.E.Knuth

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-008-l.html [9/23/2000 4:57:12 PM]

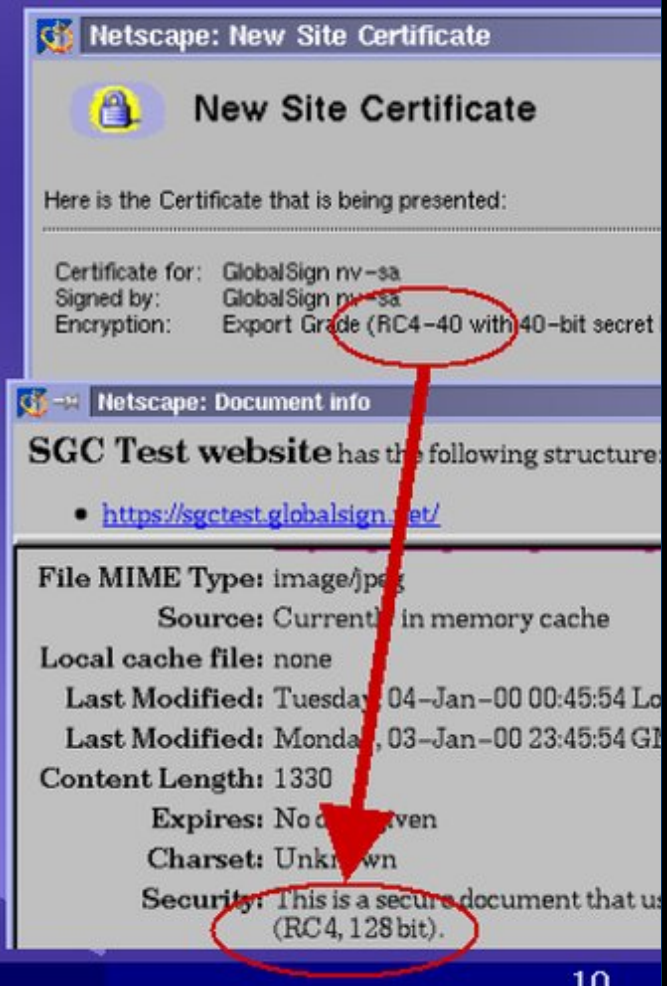
Solution 2a: Server Gated Cryptography (SGC)

- SGC is the mechanism behind Verisign Global-ID, Thawte SuperCert, GlobalSign HyperSign128, etc.
- SGC is the reason why Fortify worked at all ;-)
- Requirements:
  - mod_ssl ≥ 2.1.3 (for reneg.)
  - OpenSSL ≥ 0.9.5 (for msSGC)
- Components:
  - root and intermediate CA certificates dedicated to SGC
  - root CA certificate built into browsers and specially tagged (Netscape: cert7.db, trustflags 0x018 → 0x0238)
  - all(!) certificates in chain contain extendedKeyUsage with objects:
    nsSGC (2.16.840.1.113730.4.1)
    msSGC (1.3.6.1.4.1.311.10.3.3)
- Configuration:

```
    :
SSLCertificateFile      etc/ssl.crt/server.crt
SSLCertificateChainFile etc/ssl.crt/ca-chain.crt
SSLCipherSuite          ALL:!ADH:+HIGH:+MEDIUM:+LOW:+EXP:+eNULL
SSLRequire              %{SSL_CIPHER_USEKEYSIZE} >= 128
    :
```

Solution 2b: Server Gated Cryptography

- Procedure:
  - browser is export-crippled: can use 40-bit ciphers only
  - browser connects to SGC-enabled server with 40-bit cipher
  - server sends his SGC-tagged certificate to browser
  - browser verifies server certificate and detects that it is issued by a CA root certificate which is tagged to enable SGC
  - browser enables 128-bit ciphers for communication with this server and forces a SSL/TLS renegotiation with the stronger cipher suite.



Solution 3: HTTP/HTTPS Gateway

To be filled in...

Solution 4: Per-Directory SSL/TLS Parameters

To be filled in...

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-012-l.html [9/23/2000 4:57:24 PM]

Solution 5: Internet/Intranet Access

To be filled in...

Solution 6: Custom SSL Error Message

To be filled in...

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-014-l.html [9/23/2000 4:57:29 PM]

Solution 7: Fine-Grained Client Authentication

- Use of boolean expressions to perform fine-grained client certificate based access authentication

```
# httpd.conf
    :
<VirtualHost _default_:443>
    :
SSLCACertificatePath /path/to/ssl.ca/
SSLVerifyClient      optional
SSLVerifyDepth       10
SSLRequireSSL
SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)-/                  \
           and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd."          \
           and %{SSL_CLIENT_S_DN_OU} in { "Staff", "CA", "Dev" } \
           and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5            \
           and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20        ) \
          or %{REMOTE_ADDR} =~ m/^141\.1\.129\.[0-9]+$/
    :
</VirtualHost>
    :
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-015-l.html [9/23/2000 4:57:32 PM]

Solution 8: Certificate Revocation Lists (CRL)

To be filled in...

Solution 9: Standard Client Authentication

- Access Control with HTTP Basic Authentication through the Subject-DN of the X.509 certificate

```
# /etc/ssl.passwd.foo

/C=DE/L=Munich/CN=Foo:xxj31ZMTZzkVA
/C=US/L=S.F./CN=Bar:xxj31ZMTZzkVA
/C=US/L=L.A./CN=Quux:xxj31ZMTZzkVA
```

```
# httpd.conf
    :
AllowOverride FileInfo AuthConfig
    :
<VirtualHost _default_:443>
    :
SSLCACertificatePath /etc/ssl.ca/
SSLVerifyClient require
SSLVerifyDepth  10
SSLOptions      +FakeBasicAuth
    :
</VirtualHost>
    :
```

```
# .htaccess

    :
<Files foo.html>
SSLRequireSSL
AuthType        Basic
AuthName        "foo"
AuthUserFile    /etc/ssl.passwd.foo
Require         valid-user
</Files>
    :
```


file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-017-l.html [9/23/2000 4:57:39 PM]

Solution 10: Using both RSA and DSA/DH

- RSA is was covered by patent, Data Signature Algorithm (DSA) and Diffie-Hellman (DH) Key Exchange was not.
- SSL/TLS provides both RSA and DSA/DH based ciphers and OpenSSL supports all.
- mod_ssl allows one to use all SSL/TLS ciphers if both RSA and DSA server certificates are available.
- Disadvantage: DSA/DH is still not supported by all current versions of popular browsers.

```
# httpd.conf
    :
<VirtualHost _default_:443>
    :
SSLCertificateFile      /etc/ssl.server-rsa.crt
SSLCertificateFile      /etc/ssl.server-dsa.crt
SSLCertificateKeyFile /etc/ssl.server-rsa.key
SSLCertificateKeyFile /etc/ssl.server-dsa.key
    :
</VirtualHost>
    :
```

There are two types of people in this world, good and bad. The good sleep better, but the bad seem to enjoy the waking hours much more.

Woody Allen

file:///C|/APACHE~1/PROCEE~1/M21/AC2K-RSE/slide-018-l.html [9/23/2000 4:57:43 PM]