

# Apache 2 mod\_ssl by example

ApacheCon 2004

Mads Toftum  
mads@apache.org

# Agenda

- Getting started
- Certificates
- Access control
- Proxy solutions
- Performance

# Building mod\_ssl

- The Apache 1.3 + mod\_ssl way
  - Download mod\_ssl and apache from different sites
  - Patch apache:

```
$ ./configure --with-apache = ../apache-1.3.x/ \  
    --with-ssl=../openssl-0.9.x  
    ...      #extra apache options  
$ cd ../apache-1.3.x  
$ make  
$ make install
```

# Building apache with mod\_ssl

- The Apache httpd 2.x way

- Get the source from apache.org

```
$ cd httpd-2.x/
```

```
$ ./configure --prefix=/usr/local/apache2 \
```

```
    --enable-ssl
```

```
$ make
```

```
$ make install
```

# Practical example



# More build options

- httpd options
  - enable-ssl=shared
  - with-ssl=DIR
- apr options
  - with-egd[=DIR]
  - with-devrandom[=DEV]

# Configuring Apache

- Default config in ssl.conf
- Wrapped in `<IfDefine SSL>`
  - Start with `-DSSL`
  - `apachectl startssl`

```
<IfDefine SSL>
```

```
    LoadModule ssl_module modules/mod_ssl.so
```

```
</IfDefine>
```

```
<IfModule mod_ssl.c>
```

```
    Include conf/ssl.conf
```

```
</IfModule>
```

# Configuring – common part

```
<IfDefine SSL>
```

```
Listen 1.2.3.4:443
```

```
SSLPassPhraseDialog builtin
```

```
SSLSessionCache shm:logs/ssl_scache(512000)
```

```
SSLSessionCacheTimeout 300
```

```
SSLMutex file:logs/ssl_mutex
```

```
SSLRandomSeed startup builtin
```

```
SSLRandomSeed connect builtin
```

# Configuring - VirtualHost

```
<VirtualHost 1.2.3.4:443>
```

**SSLEngine on**

```
ServerName example.com:443
```

```
DocumentRoot "/serverroot/htdocs/"
```

```
SSLCertificateFile conf/ssl.crt/server.crt
```

```
SSLCertificateKeyFile conf/ssl.key/server.key
```

```
</VirtualHost>
```

```
</IfDefine>
```

# Generating certificates with openssl

- Preparations

```
openssl.cnf (/usr/local/ssl/openssl.cnf)
```

```
$ echo '01' > serial
```

```
$ touch index.txt
```

```
$ mkdir certs crl newcerts private
```

# openssl – generating CA

- Generate private key
  - `openssl genrsa -des3 2048`
- Generate CA certificate
  - `openssl req -new -x509 -days 3650`
- Check the certificate
  - `openssl x509 -in cacert.pem -noout -text`

# openssl – server cert

- Generating server keypair
  - `openssl genrsa -des3 -out server.key 1024`
- Generating the request
  - `openssl req -new -key server.key -out server.csr`
- Signing the server certificate with your CA
  - `openssl ca -out server.crt -infiles server.csr`
- Verify the generated certificate
  - `openssl verify -CAfile cacert.pem server.crt`

# Generating certificates - tinyca

Create a new CA

Name for local storage: ExampleCA

Data for CA Certificate

Serial Number (00000000): 00000000

Country Name (2 letter code): UK

Password (needed for signing): password

Password (confirmation): password

State or Province Name: Example

Locality Name (eg. city): Trondheim

Organization Name (eg. Company): Example.org

Organizational Unit Name (eg. section): CA

eMail Address:

Validity (Days): 365

Key length: C: 1024 @: 2048 O: 4096

OK Cancel

Create a new Certificate Request

Common Name (eg. your Name): example.com

your eMail Address (in the form of Name (eg. your Name)):

eMail Address: webmaster@example.com

Password (protect your private Key): password

Password (confirmation): password

Country Name (2 letter code): UK

State or Province Name:

Locality Name (eg. city): Trondheim

Organization Name (eg. Company): ExampleCo

Organizational Unit Name (eg. section): CA

Key length: C: 1024 @: 2048 O: 4096

OK Cancel

<http://tinyca.sm-zone.net/>

# Removing the passphrase

- startup

```
$ umask 077
```

```
$ openssl rsa -in server.key -out unsafe.key
```

- SSLPassPhraseDialog
  - exec:/path/to/program
  - /path/to/program servername:port RSA

# Using Client Certs - 1

- `SSLVerifyClient`
  - none (default)
  - require
  - optional / `optional_no_ca`

## **Ex:**

`SSLCACertificateFile conf/ca.crt`

`SSLVerifyClient require`

`SSLVerifyDepth 1`

# Client cert – error messages

- Failed client cert validation errors are difficult to decipher in the browser

```
SSLVerifyClient optional
```

```
RewriteEngine on
```

```
RewriteCond %{SSL_CLIENT_VERIFY} !="SUCCESS"
```

```
RewriteRule .* /path/client-cert-error.html [L]
```

Note: many other env vars

# Client Cert – tracking users

Environment variables can be used to match client certs to requests:

Combined Log Format:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

With SSL\_CLIENT\_S\_DN

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%{SSL_CLIENT_S_DN}x\"" ssl
```

# Client certs – per directory

- Directives can be applied in a directory context

```
SSLCACertificateFile conf/ca.crt
```

```
SSLVerifyClient none
```

```
<Location /admin>
```

```
    SSLVerifyClient require
```

```
    SSLVerifyDepth 1
```

```
</Location>
```

# Client certs – mapping to users

- **SSLOptions +FakeBasicAuth**
  - `SSL_CLIENT_S_DN`
  - `openssl x509 -noout -subject -in certificate.crt`
  - `C=DK/L=CPH/CN=Mads:xxj31ZMTZzkVA`

**<Directory />**

**SSLOptions +FakeBasicAuth**

**AuthType Basic**

**AuthName Cert**

**AuthUserFile conf/htpasswd**

**require valid-user**

**</Directory>**

# Client certs – group based access

- SSLRequire *ComplicatedExpression*

```
SSLRequire (\
```

```
    %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
```

```
    and \
```

```
    %{SSL_CLIENT_S_DN_OU} in ("Staff ", "Boss") \
```

```
)
```

# Proxy – wrapping legacy services

- Add SSL support to http services
- Offload SSL processing

```
<VirtualHost 1.2.3.4:443>
```

```
    SSLEngine on
```

```
    ProxyPass / http://10.0.0.2/
```

```
    ProxyPassReverse / http://10.0.0.2/
```

```
</VirtualHost>
```

# Proxy - “unwrapping” SSL

- Opposite of previous slide

```
<VirtualHost 1.2.3.4:80>
```

```
SSLProxyEngine on
```

```
ProxyPass / https://www.example.com/
```

```
ProxyPassReverse / https://www.example.com/
```

```
...
```

```
SSLProxyCACertificateFile conf/certs/ca.crt
```

```
SSLProxyVerify require
```

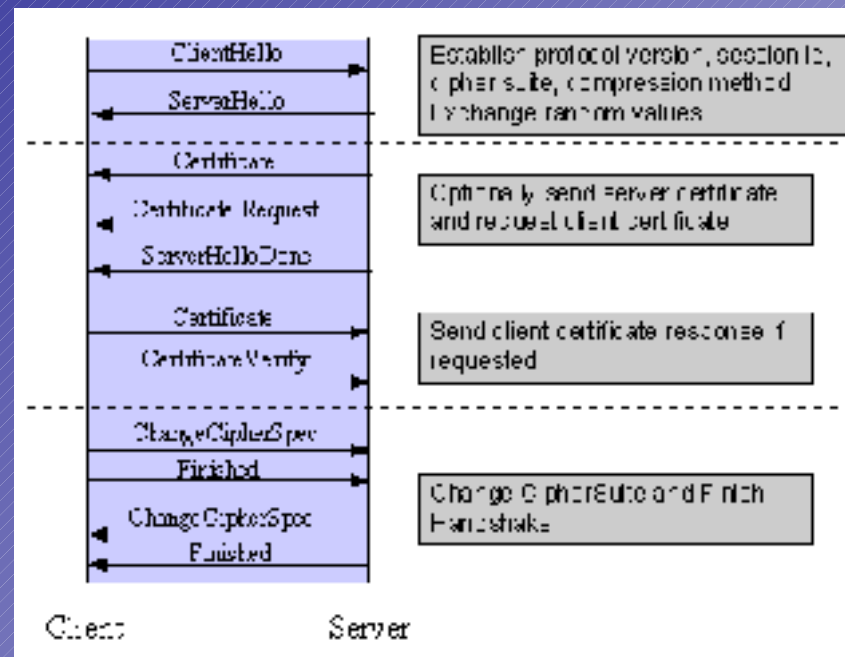
```
</VirtualHost>
```

# speed - keysize

- Size does matter!

		sign	verify	sign/s	verify/s
rsa	512 bits	0.0019s	0.0002s	528.8	5903.0
rsa	1024 bits	0.0090s	0.0005s	110.6	2100.7
rsa	2048 bits	0.0532s	0.0016s	18.8	644.0
rsa	4096 bits	0.3534s	0.0054s	2.8	185.8

# speed - keysize



# Speed – session cache

- SSLSessionCache
  - none
  - dbm:file
  - shm:file(size)
- SSLSessionCacheTimeout
  - Clients may time out sessions
  - %`{SSL_SESSION_ID}`
- distributed - [www.distcache.org](http://www.distcache.org)

# Speed – misc

- [/manual/mod/mod\\_ssl.html#envvars](#)
- [/manual/ssl/ssl\\_compat.html#variables](#)
- SSLOptions
  - StdEnvVars / CompatEnvVars / ExportCertData
  - significantly grows the size of the environment
  - `<Files ~ "\.(pl|cgi)$">`
- OptRenegotiate
  - tries to renegotiate when SSL settings change in directory context to avoid overhead of full handshake

# Questions ?

<http://cvs.apache.org/~mads/ac2004/>

# Intra/extranet

```
<Directory /usr/local/apache/htdocs>
#   Outside the subarea only Intranet access is granted
Order          deny,allow
Deny           from all
Allow         from 192.168.1.0/24
</Directory>
<Directory /usr/local/apache/htdocs/subarea>
#   Force clients from the Internet to use HTTPS
RewriteEngine on
RewriteCond    %{REMOTE_ADDR} !^192\.168\.1\.[0-9]+$
RewriteCond    %{HTTPS} !=on
RewriteRule    ^/(.*)$ https://*{SERVER_NAME}/$1 [R,L]
#   Ask for client certificate and require strong cipher
SSLVerifyClient optional
SSLVerifyDepth 1
SSLCACertificateFile conf/ssl.crt/company-ca.crt
SSLOptions     +FakeBasicAuth +StrictRequire
SSLRequire     %{SSL_CIPHER_USEKEYSIZE} >= 128
#   Allow Network Access and/or Basic Auth
Satisfy        any
#   Network Access Control
Order          deny,allow
Deny           from all
Allow         from 192.168.1.0/24
#   HTTP Basic Authentication
AuthType       basic
AuthName       "Protected Intranet Area"
AuthUserFile   conf/protected.passwd
Require        valid-user
</Directory>
```