Deciphering mod_ssl: Using SSL with the Apache HTTP Server

Joe Orton, Red Hat

jorton@redhat.com

ApacheCon

Leading the Wave of Open Source

Contents

- Introduction
- Basic setup
- Advanced configuration
- Future features
- Conclusion
- Q&A

Introduction

(Not a) History lesson

- Pre-history: Apache-SSL etc
- mod_ssl 2.8.x for httpd 1.3
  - www.modssl.org, Ralf Engelschall
- mod_ssl in the httpd 2.x tree
  - “mod_ssl/2.0.x” > “mod_ssl/2.8.x” !?

mod_ssl Complexity

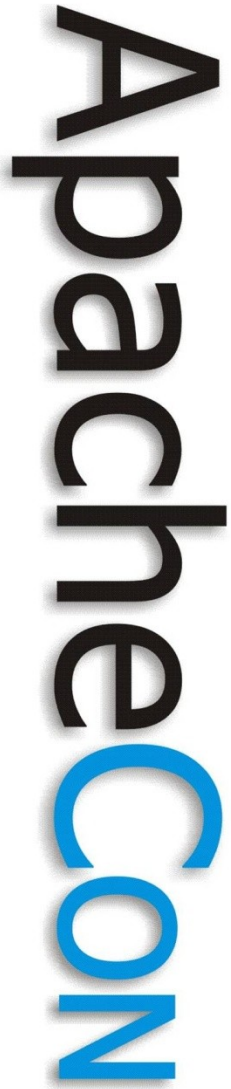
- 42 configuration directives

mod_ssl Complexity

- 42 configuration directives
- 1 expression language

mod_ssl Complexity

- 42 configuration directives
- 1 expression language
- Hooks into/from 5 modules



mod_ssl Complexity

- 42 configuration directives
- 1 expression language
- Hooks into/from 5 modules
- 14K lines of code

mod_ssl Complexity

- 42 configuration directives
- 1 expression language
- Hooks into/from 5 modules
- 14K lines of code
- 30+ exported CGI variables

SSL Server, Step Zero

- You need an SSL certificate!

SSL Server, Step Zero

- ~~You need an SSL certificate!~~
- You need an SSL certificate signed by a CA

SSL Server, Step Zero

- ~~You need an SSL certificate!~~
- ~~You need an SSL certificate signed by a CA~~
- You need an SSL certificate signed by a CA which is trusted by all the web browsers

SSL Server, Step Zero

- ~~You need an SSL certificate!~~
- ~~You need an SSL certificate signed by a CA~~
- You need an SSL certificate signed by a CA which is trusted by all the web browsers
- ... all the web browsers **which will use your SSL site**

Basic Configuration

- Minimal configuration:

```
Listen 443

SSLSessionCache shmcb:run/sslcache(512000)
SSLMutex default

<VirtualHost *:443>
  SSLEngine on
  SSLCertificateFile /path/to/cert.crt
</VirtualHost>
```

Session caching

- Reduces server load
- Reduces per-connection round trips

Tuning the Session Cache

- Enable mod_status

SSL/TLS Session Cache Status:
cache type: **SHMCB**, shared memory: **512000** bytes, current sessions: **2752**
subcaches: **32**, indexes per subcache: **133**
time left on oldest entries' SSL sessions: avg: **157** seconds, (range: 149...166)
index usage: **64%**, cache usage: **99%**
total sessions stored since starting: **5425**
total sessions expired since starting: **0**
total (pre-expiry) sessions scrolled out of the cache: **2673**
total retrieves since starting: **14** hit, **1** miss
total removes since starting: **0** hit, **0** miss

```
SSLSessionCache shmcb:run/sslcache(512000)
SSLSessionCacheTimeout 300
```

Certificate chains

- Increasing depth of CA certificate chains
- Intermediate certs not known/trusted by browsers
- MSIE knows how to fetch them anyway – Firefox does not!
- Configure the server to send them:

```
SSLCertificateChainFile /path/to/ca.crt
```

Exporting SSL state

- Large set of SSL variables
  - Exported to the CGI environment
  - Available to other modules
- Enable per-Location or Directory:

```
<Directory /all/my/php/code>
    SSLOptions +StdEnvVars
</Directory>
```

- Most commonly used:

```
$HTTPS = "on" or "off"
```

Custom SSL logging

- Can use any of the SSL env vars
- Inside the VirtualHost:

```
CustomLog logs/ssl_request_log \
  “%t %h %{SSL_PROTOCOL}x \”%r\””
 ...
[11/Mar/2009:09:58:13 +0000] 127.0.0.1
   TLSv1 "GET /info.php HTTP/1.1"
```

Browsers are broken

- SSL requires exchange of messages to cleanly close connection
- MSIE has... issues (historically)
- Standard workaround:

```
BrowserMatch ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
```

Advanced configuration

Client Certificates

- Secure user authentication
- Widely disliked, deployment issues
- Hardware tokens easier
- ... but (relatively) expensive
- Government adoption increasing
  - National ID schemes
  - Internal ID schemes, e.g. US DoD

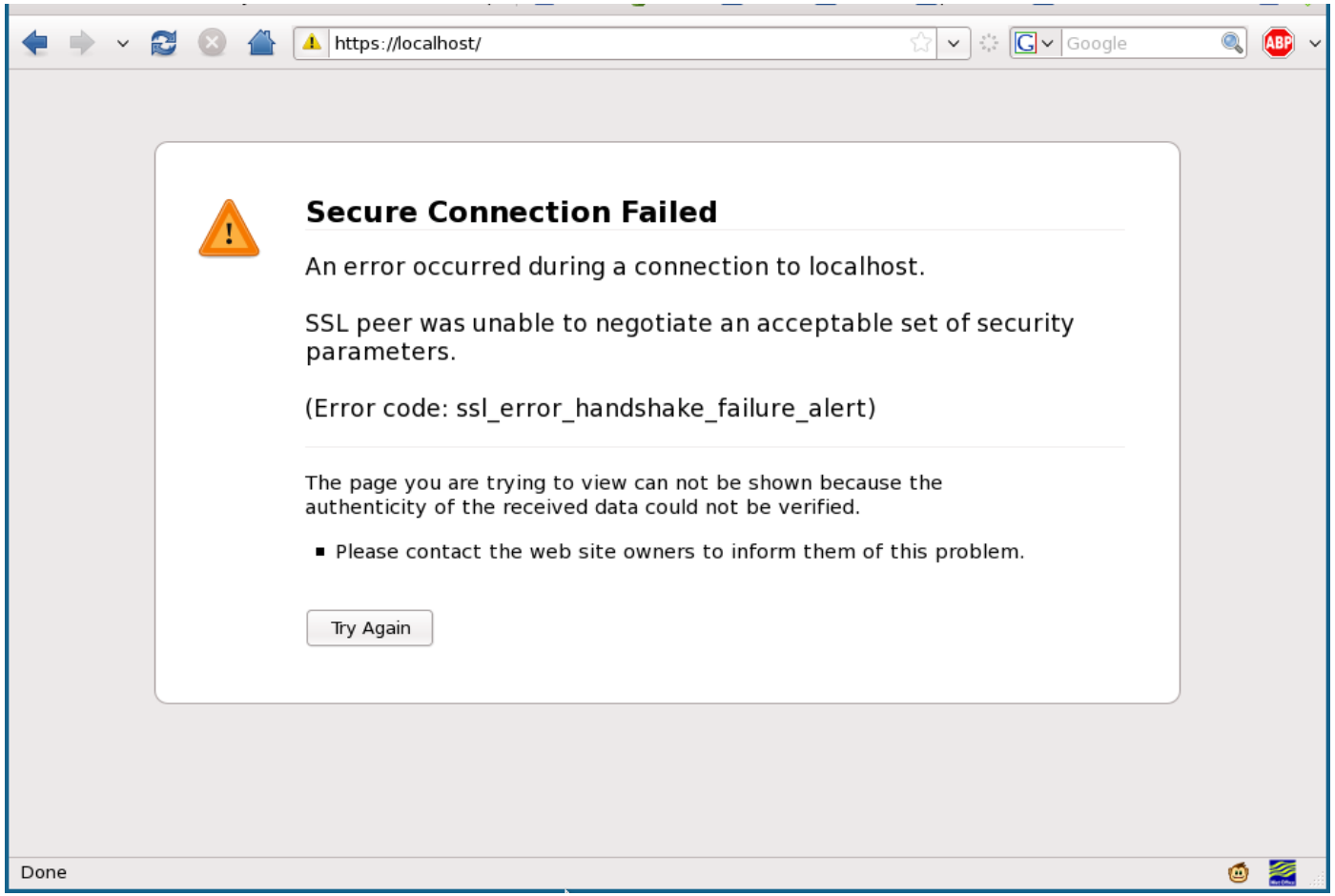
Basic configuration

- In <VirtualHost>:

```
SSLVerifyClient require
SSLCACertificateFile /path/to/myca.crt
```

- Or in <Directory> or <Location>
- ... but use with care

Fail



Fail, politely

- Don't fail the SSL handshake
- Use SSLRequire for access control

```
<Location /secret>
  SSLVerifyClient optional
  SSLRequire “%{SSL_VERIFY_CLIENT}” \
                eq “SUCCESS”
  ErrorDocument 403 /403.html
</Location>
```

Access control

- Structured “Subject” field in client cert
- Can reflect organizational structure:

```
/C=US/O=Red Hat, Inc/OU=Engineering/
  /CN=Joe Orton/
```

- Fine-grained access control based on subject fields:

```
SSLRequire “%{SSL_CLIENT_S_DN_OU}” \
    in {“Engineering”, “Support”}
```

Per-Directory Renegotiation

- Here be dragons!

```
<form action="/secret/foo.cgi"
      method="POST">
  Submit your document:
  <input type="file" name="thedoc">
```

- If /secret/ requires renegotiation
- ... i.e. SSLVerifyClient in <Directory>
- This is a hard problem

Per-Directory Renegotiation

- Client:

```
<SSL handshake>
<HTTP Request Headers + Body>
```

- Server:

```
<SSL handshake>
<HTTP Request Headers>
<SSL handshake>
<HTTP Request body>
```

Per-Directory Renegotiation

- mod_ssl will buffer the request body, then renegotiate – up to 128K of data
- Unlimited buffering == DoS
- New in 2.2.12, SSLRenegBufferSize
- Better solution:
  - Per-dir renegotiation is fine for GET
  - So design the site to avoid per-dir renegotiation on POST

Revocation

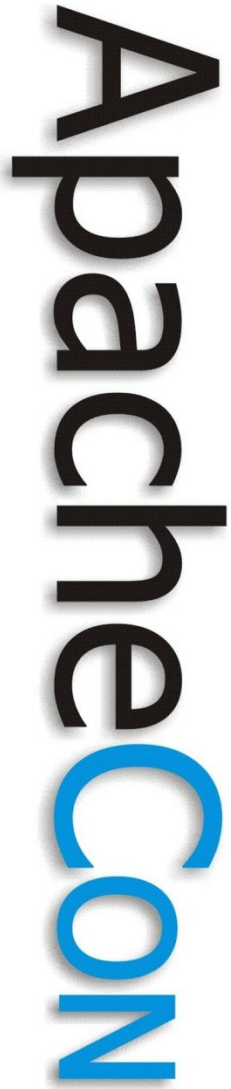
- Revoke certs for ex-employee, citizens you don't like, etc
- Current solution: static CRL files

```
SSLVerifyClient require
SSLCACertificateFile /path/to/myca.crt
SSLCARevocationFile /path/to/myca.crl
```

- Restart the server to reload CRLs (graceful or not)

Future features

httpd 2.3 and beyond



SNI

- Name-based virtual hosts don't work for SSL
- “Server Name Indication” TLS extension fixes this
  - Supported in (relatively) modern browsers: Firefox 2, MSIE7
  - Now supported in httpd trunk

OCSP

- “Online Certificate Status Protocol”
- Because CRLs suck:
  - Static files. How/when to reload?
  - How to update?
- Check client certificate revocation status in real time

OCSP protocol

- OCSP server is an HTTP resource
- Send it a POST request
  - Request body includes details of (client) cert to verify
- Response gives revocation status of given certificate
  - In a signed message
  - Hence, trusted if you trust the signer

OCSP in mod_ssl

- Zawinski's law, compressed version:
  - “Every program attempts to expand until it can read mail.”

OCSP in mod_ssl

- Zawinski's law, compressed version:
  - “Every program attempts to expand until it can read mail.”
- Applies for HTTP clients too:
  - “Every program attempts to expand until it contains an HTTP client.”
- mod_ssl contains an HTTP client

OCSP in mod_ssl

- Zawinski's law, compressed version:
  - “Every program attempts to expand until it can read mail.”
- Applies for HTTP clients too:
  - “Every program attempts to expand until it contains an HTTP client.”
- mod_ssl contains an HTTP client
  - As does OpenSSL.

OCSP in mod_ssl

- Zawinski's law, compressed version:
  - "Every program attempts to expand until it can read mail."
- Applies for HTTP clients too:
  - "Every program attempts to expand until it contains an HTTP client."
- mod_ssl contains an HTTP client
  - As does OpenSSL. PHP brings four.

OCSP Stapling

- Verifying every SSL server cert against the issuing CA's OCSP server(s):
  - Good for security
  - Bad for performance
- OCSP “stapling” solves this

OCSP Stapling

- SSL server obtains OCSP response for its own cert
- Response is “stapled” to the SSL handshake
  - Uses a TLS/1.0 extension
  - Includes timestamp
  - Is signed by CA (or intermediate)
  - Cached by server

Conclusion

- Basic configuration:
  - Server certs, session cache, logging, browser hacks, cert chains, and SSL variables
- Advanced configuration:
  - Client certs, fine-grained access control, per-dir reneg “issues”, failing politely
- Future
  - SNI, OCSP, OCSP stapling

Q & A