Technology Transfer Partners

APACHE HTTP SERVER PROJECT

Configuring Apache web server
on OES servers and relatives

Joe Doupnik
jrd@netlab1.net
jdoupnik@microfocus.com
Mindworksuk and Micro Focus
Prof (ret.) Univ of Oxford

UNIVERSITY OF OXFORD

MICRO FOCUS

About this presentation

- The topic is Apache web server, which as we all know is in OES, appliances, and related apparatus.
- The main goal here is perform configuration steps to defend against bad guy penetration attempts (which go on all day every day).
- Nothing needs to be purchased. The cost is our time. The benefit is the health of our site. We also add some useful features.
- Much of this discussion is taken from recent work, in presentations
  Intro to Apache web server and SSL/TLS for system admins
- Primary doc reference is the Apache2 web server manual, which may be installed locally, plus https://httpd.apache.org/ and Apache's security tips. Third party project mod security is not addressed here.
- Careful analysis involves a detailed review of directives.

Typical net.idiot scripted attacks

- Apache access log snippet. Apache logs are in /var/log/apache2
- Note the 301 status code. 301 is a redirect (from our work here).

  200 is success, 400-500 is Failure. (list of codes)

```
185.7.214.104 - - [09/Jul/2022:11:11:55 +0100] "GET / HTTP/1.1" 301 232
185.7.214.104 - - [09/Jul/2022:11:12:00 +0100] "\x16\x03\x01" 301 217 "-" "-"
185.7.214.104 - - [09/Jul/2022:11:12:00 +0100] "\x16\x03\x01" 301 217 "-" "-"
185.7.214.104 - - [09/Jul/2022:11:12:00 +0100] "\x16\x03\x01" 301 217
45.61.186.139 - - [09/Jul/2022:11:14:50 +0100] "GET /config/getuser?index=0 HTTP/1.1" 301 240 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:76.0) Gecko/20100101 Firefox/76.0"
45.61.186.139 - - [09/Jul/2022:11:14:50 +0100] "GET /config/getuser?index=0 HTTP/1.1" 301 240 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:76.0) Gecko/20100101 Firefox/76.0"
45.61.186.139 - - [09/Jul/2022:11:14:50 +0100] "GET /config/getuser?index=0 HTTP/1.1" 301 240
185.180.143.6 - - [09/Jul/2022:11:16:40 +0100] "GET / HTTP/1.1" 200 6936
187.200.196.84 - - [09/Jul/2022:11:18:09 +0100] "POST /cgi-bin/ViewLog.asp HTTP/1.1" 301 226 "-" "r00ts3c-owned-you"
187.200.196.84 - - [09/Jul/2022:11:18:09 +0100] "POST /cgi-bin/ViewLog.asp HTTP/1.1" 301 226 "-" "r00ts3c-owned-you"
187.200.196.84 - - [09/Jul/2022:11:18:09 +0100] "POST /cgi-bin/ViewLog.asp HTTP/1.1" 301 226
187.200.196.84 - - [09/Jul/2022:11:18:09 +0100] "ep.zyxel80;rm+-rf+arm7%3b%23&remoteSubmit=Save" 301 217 "-" "-“
34.75.168.191 - - [09/Jul/2022:11:31:42 +0100] "GET / HTTP/1.0" 500 - "-" "ZoominfoBot (zoominfobot at zoominfo dot com)"
```

Starting from scratch…

- A regular installation of Apache on our SLES based apparatus creates a “default server” for http:// on port 80, and a template for our creating a “virtual host” using SSL/TLS https:// on port 443.
- It turns out that plain http:// port 80 is vulnerable to a multitude of bad things. Changing to https:// removes many of them.
- A practical solution is “redirect” client browsers to use https://. The change is done invisibly by their browser in response to our (server side) Apache “redirect” response. Idiot bots seem unable to comply, and fail.
- Additionally we allow only polite network behaviour, thus try to rebuff the bad guys and script kiddies/idiots.

- First we adjust basic configuration in file */etc/sysconfig/apache2*, then secondly create a https:// port 443 server.
- Third we deal with redirecting bad guy probes. Lastly we embellish Apache with a few useful things.

File /etc/sysconfig/apache2 details

APACHE_MODULES="actions alias ssl auth_basic authn_core authn_dbm authn_file authnz_ldap authz_core authz_groupfile authz_host authz_user autoindex cgi dav dav_fs dav_lock deflate dir env expires fcgid filter h264_streaming headers http2 include ldap log_config log_forensic mime negotiation perl proxy proxy_ajp proxy_http reqtimeout rewrite setenvif socache_shmcb speling status substitute suexec unique_id userdir“

Above, ssl is moved early in the list to prevent dependency issues

APACHE_SERVER_FLAGS="HTTP2 SSL STATUS“

DOC_SERVER="yes“ use as https://example.com/manual

APACHE_MPM="worker“ default is “forked”. “worker” and “event are threaded

APACHE_SERVERADMIN=“postmaster@example.com”

APACHE_SERVERNAME=“example.com“

APACHE_SERVERSIGNATURE="off“

APACHE_LOGLEVEL="warn“

APACHE_USE_CANONICAL_NAME="on“

APACHE_SERVERTOKENS="Minimal“

APACHE_EXTENDED_STATUS="off“

APACHE_DISABLE_SSL_COMPRESSION="on"

Module story: created modules live within /usr/lib64, in directories apache2, apache2-prefork, apache2-worker etc. Apache start up scripts read this apache2 script and from the APACHE_MODULES= line create a list of LoadModule commands to actually load modules. That list is file /etc/apache2/sysconfig.s/load_modules.conf.

Avoid YaST’s httpd selection; it can be a trouble maker. We edit /etc/sysconfig/apache2 to control matters.

File /etc/apache2/vhosts.d/vhost-ssl.conf

- The primary virtual host for https:// is configured by the above file.
- We should avoid YaST's *http* menu offering because it can/will clobber our configurations. Just ignore that clickable.
- Vhost-ssl.conf has entries for a commercial certificate (obviously we avoid self-signed certs when dealing with the public).
- Importantly, it has selections for SSL/TLS crypto details, and we need to improve upon its defaults. Presentation SSL/TLS for system admins discusses this in depth, so please do review it. Following are some key screens. It starts with choosing modern crypto which fits our machine.
- We need expert advice about choosing crypto algorithms, and Mozilla can provide it. Accompanying crypto are many behavioural directives needed to restrict activities to safe ones.

Mozilla advisor for OES2018 SP2 Apache web

(A convenient helper, not a tester, advisory only, use our good judgements)

mozilla

SSL Configuration Generator

Server Software

- (●) Apache
- ( ) AWS ALB
- ( ) AWS ELB
- ( ) Caddy
- ( ) Dovecot
- ( ) Exim
- ( ) Go
- ( ) HAProxy
- ( ) Jetty
- ( ) lighttpd
- ( ) MySQL
- ( ) nginx
- ( ) Oracle HTTP
- ( ) Postfix
- ( ) PostgreSQL
- ( ) ProFTPD
- ( ) Redis
- ( ) Tomcat
- ( ) Traefik

Mozilla Configuration

- ( ) Modern
  Services with clients that support TLS 1.3 and don't need backward compatibility
- (●) Intermediate
  General-purpose servers with a variety of clients, recommended for almost all systems
- ( ) Old
  Compatible with a number of very old clients, and should be used only as a last resort

Environment

Server Version	2.4.23
OpenSSL Version	1.0.2p

Miscellaneous

- [ ] HTTP Strict Transport Security — Avoid!
  This also redirects to HTTPS, if possible
- [x] OCSP Stapling

From https://ssl-config.mozilla.org
Also see its Resources section for discussion

Mozilla advisor for OES2018 SP2 Apache web

```
# this configuration requires mod_ssl and mod_socache_shmcb
<VirtualHost *:443>
    SSLEngine on

    # curl https://ssl-config.mozilla.org/ffdhe2048.txt >> /path/to
/signed_cert_and_intermediate_certs_and_dhparams
    SSLCertificateFile      /path/to/signed_cert_and_intermediate_certs_and_dhparams
    SSLCertificateKeyFile   /path/to/private_key

    # enable HTTP/2, if available
    Protocols h2 http/1.1
</VirtualHost>

# intermediate configuration
SSLProtocol             all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite          ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-
ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-
CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
SSLHonorCipherOrder     off
SSLSessionTickets       off

SSLUseStapling On
SSLStaplingCache "shmcb:logs/ssl_stapling(32768)"
```

Copy

This report is most useful for suggesting SSL Cipher Suite names (as one line)

SSLHonorCipherOrder should be ***on*** to have server choose cipher suites

Better is use app's recommendation

Worth remembering: we should think about details, not just copy & paste.

Items to note:

SSLProtocol
SSLCiperSuite
SSLHonorCipherOrder
SSLSessionTickets
SSLUseStapling
SSLStaplingCache

The next step

- From that report we copy the long line for *SSLServerSuite* and paste it into vhost-ssl.conf . It defines the allowed crypto suites.
- The next slide has an example, plus several needed directives.
- An important directive is *SSLProtocols* which defines which are acceptable. We need to use TLS 1.2 and later, not earlier, and use only strong ciphers.
- Further, the order of choosing must be controlled by the server. Do not let clients play clever games. That is directive *SSLHonorServerOrder on*.
- Directive *SSLCompression off* is needed to avoid a common crypto vulnerability, where the SSL engine itself could, but must not, compress the payload.
- Certificate stapling, OCSP, speeds connection establishment.
- Avoid HSTS. It foolishly invades clients while trying to achieve https usage. My redirect triad alternative is far better.

SSL directives in a vhost-ssl.conf file

# Enable/Disable SSL for this virtual host.	Each vhost can have its own cert
SSLEngine on	and SSL/TLS details
SSLProxyEngine on	if proxypass etc is used within
SSLCompression off	Default. Avoid SSL engine compression
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1	TLSv1 & TLSv1.1 are depreciated
SSLHonorCipherOrder on	Use server's order of ciphers

From Mozilla Advisor and check for weak cyphers using SSL Labs. Must be one long line:

SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384

Yes, that SSLCipherSuite line is a challenge. Consider copy&paste from Mozilla Advisor.
I comment out reference to *ssl-global.conf* in httpd.conf. Use your best judgement.
<u>Test OCSP results</u> via command **openssl s_client -connect my.host:443 -status**
and review section "OCSP Response Data:" particularly line "OCSP Response Status:"

On-line Cert Status Protocol: OCSP Certificate Stapling

As of 2017-10, **No**.

Dovecot does not have any OCSP support whatsoever, as of 2016 was considering the feature for a future release, no work has been done on that since.

Postfix does not have any OCSP support whatsoever, and as of 2017 is not planning to ever to *ever* implement such feature.

Exim can provide clients with an OCSP response, yet acquiring such is yet left as an exercise to the admin.

The main arguments against adding such support are:

1. Security features should be simple so they have more benefit than added risks. OCSP is complex. Short certificate validity is simple and mitigates the same issue.
2. The Chicken-Egg problem of OCSP support in servers being entirely useless until MUAs add such support.

This does not hinder the usage of `must-staple` certificates in web servers. Just have the option enabled on your web server certificate (e.g. `www.example.com`) and disabled on your mail server certificate (e.g. `mail1.example.com`).

Stapling: Periodically query the CA about cert validity, and cache results. Even if the certificate does state an OCSP responder URL our web server can supply the results within the TLS handshake. That is good.
This is a TLS extension in RFC 6066. Also see your Apache2 manual about *mod_ssl* for OCSP details.

Stapling avoids Cert Revocation List complications.

Web servers may support local Stapling, but Postfix, Dovecot, LDAPs, SSHd and so on likely do not (they lack an ability to talk to the CA). Beware *Must Staple*. See https://scotthelme.co.uk/ocsp-must-staple/

from ssllabs.com

OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl.starfieldtech.com/sfig2s1-187.crl OCSP: http://ocsp.starfieldtech.com/
Revocation status	Good (not revoked)

From https://serverfault.com/questions/830434/do-postfix-and-dovecot-support-ocsp-stapling

Also see https://raymii.org/s/articles/OpenSSL_Manually_Verify_a_certificate_against_an_OCSP.html

Some global directives regarding SSL and Certificates

See presentation SSL/TLS for system admins for details and discussion

SSLStrictSNIVHostCheck on	Require host name in vhost arrivals
SSLInsecureRenegotiation off	Default. Don't haggle in public
SSLSessionTickets off	Resumption: use cache, not tickets
SSLSessionCache shmcb:/var/lib/apache2/ssl_scache(512000)	Its cache
SSLUseStapling on	Desirable: offer local cert stapling
SSLStaplingCache shmcb:/var/run/ocsp(12800)	Cache for local cert stapling

 # shmcb: is memory caching using Apache2 module ***socache_shmcb***

SSLStaplingReturnResponderErrors off	
SSLRandomSeed startup "file:/dev/urandom" 1024	Better random number generator
ProtocolsHonorOrder on	Use server's TLS protocol ordering

These items are made global by appearing outside of <…> clauses. See Apache docs. SLES scatters controls within *Include* files in /etc/apache2. Double check them. See also https://www.digitalocean.com/community/tutorials/how-to-configure-ocsp-stapling-on-apache-and-nginx and Apache web server documentation.

Additional global level directives in my OES2018

Globals can be placed in /etc/apache2/globals.conf or /etc/apache2/conf.d, or just outside of <Foo> clauses . Many are in file /etc/sysconfig/apache2.

```
ServerSignature off
UseCanonicalName on
ServerTokens ProductOnly
LogLevel warn
CustomLog /var/log/apache2/access.log combined
KeepAlive on
KeepAliveTimeout 180
MaxKeepAliveRequests 50
ExtendedStatus on
Trace Enable off
```

```
FileETag MTime Size
LimitRequestFieldSize 1512
Header add Cache-Control "private,proxy- revalidate,
max-age=86400"
ProxyPreserveHost on
IndexStyleSheet "/index.css“

<IfModule mod_http2.c>
    Protocols h2 http/1.1
</IfModule>
```

LimitRequestFieldSize is set to block URL buffer overflow attempts.
Header cache control is for remote proxy servers feeding users.

Blocking some bad guy penetration methods

These items can be placed in <Directory> clauses or best placed outside of them and thus apply globally. There are three protections below: fail, fail, redirect.

```
<IfModule mod_rewrite.c>
# Fail OPTIONS, CONNECT etc and HTTP/1.0 requests, and require https://
# ^ is "starts with", $ is "ends with"
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD}  ^(CONNECT|DELETE|HEAD|OPTIONS|PUT|TRACE|TRACK)
    RewriteRule ".*"  "-"  [F,L,END]                    F is report failure, L is last, END is end this rule set
      RewriteEngine On                                    ".*"  "-" is replace incoming command text with -
      RewriteCond %{THE_REQUEST} HTTP/1\.0$             HTTP/1.0 is old and vulnerable to abuse
      RewriteRule ".*"  "-"  [F,L,END]
    RewriteEngine On                                     Please repeat the request using https, not http
    RewriteCond %{HTTPS} off                             If arrival did not use https then do the rule
    RewriteRule (.*)  https://%{HTTP_HOST}/%{REQUEST_URL}  [QSA,L,R=301]     R is redirect, args are repeated
</IfModule>
```

Also see presentations [IPtables & blocking the bad guys]() and [SSL/TLS for system admins]()

Further bad guy defence

Normally applied globally, but can be within a <Directory> clause.

Redirect pests to their localhost. We are being “kind” rather than nasty.

Redirect permanently if *this-URL-phrase* to *this-new-request-line*

```
ReDirectPermanent  /admin  https://127.0.0.1/
ReDirectPermanent  /wp-login.php  https://127.0.0.1/
ReDirectPermanent  /pub/wp-login.php  https://127.0.0.1/
ReDirectPermanent  /cgi-bin/ViewLog.asp  https://127.0.0.1/
ReDirectPermanent  /4VG*  https://127.0.0.1/
ReDirectPermanent  /ViewLog.asp  https://127.0.0.1
ReDirectPermanent  /.env  https://127.0.0.1
ReDirectPermanent  /HNAPI  https://127.0.0.1
ReDirectPermanent  /boaform  https://127.0.0.1
  etc
```

Apache manual about merging sections

How the sections are merged

The configuration sections are applied in a very particular order. Since this can have important effects on how configuration directives are interpreted, it is important to understand how this works.

The order of merging is:

1. `<Directory>` (except regular expressions) and `.htaccess` done simultaneously (with `.htaccess`, if allowed, overriding `<Directory>`)
2. `<DirectoryMatch>` (and `<Directory "~">`)
3. `<Files>` and `<FilesMatch>` done simultaneously
4. `<Location>` and `<LocationMatch>` done simultaneously
5. `<If>`

Some important remarks:

- Apart from `<Directory>`, within each group the sections are processed in the order they appear in the configuration files. For example, a request for */foo/bar* will match `<Location "/foo/bar">` and `<Location "/foo">` (group 4 in this case): both sections will be evaluated but in the order they appear in the configuration files.
- `<Directory>` (group 1 above) is processed in the order shortest directory component to longest. For example, `<Directory "/var/web/dir">` will be processed before `<Directory "/var/web/dir/subdir">`.
- If multiple `<Directory>` sections apply to the same directory they are processed in the configuration file order.
- Configurations included via the `Include` directive will be treated as if they were inside the including file at the location of the `Include` directive.
- Sections inside `<VirtualHost>` sections are applied *after* the corresponding sections outside the virtual host definition. This allows virtual hosts to override the main server configuration.
- When the request is served by `mod_proxy`, the `<Proxy>` container takes the place of the `<Directory>` container in the processing order.

Order is important. Last mention wins.
<Location>s are processed after <Directory>s.

Security: employ "*require*" to isolate items from intruders.

https://localhost/manual/sections

Example, restricted access to area and program


```
Alias /config  /home/search/config.php
<Directory /home/search>
    <requireall>
    require ip 11.22.33.100 10.0.0.1/24 127.0.0.1
    AuthType Basic
    AuthBasicProvider ldap
    AuthName "search"
    AuthLDAPUrl  ldap://example.com/o=oucs?uid?one?(objectClass=user)
    require ldap-user admin jrd
    </requireall>
</Directory>
```

config is URL/program for admin use only

Restrict area access by IP and username

Must satisfy all “require” conditions

admin moves about

The second “require” condition

(There is also <requireany> and <requirenone>)

Require directives, Manual fine print worth reading

Require Directive

Description: Tests whether an authenticated user is authorized by an authorization provider.
Syntax: `Require [not] entity-name [entity-name] ...`
Context: directory, .htaccess
Override: AuthConfig
Status: Base
Module: mod_authz_core

This directive tests whether an authenticated user is authorized according to a particular authorization provider and the specified restrictions. `mod_authz_core` provides the following generic authorization providers:

`Require all granted`
Access is allowed unconditionally.

`Require all denied`
Access is denied unconditionally.

`Require env env-var [env-var] ...`
Access is allowed only if one of the given environment variables is set.

`Require method http-method [http-method] ...`
Access is allowed only for the given HTTP methods.

`Require expr expression`
Access is allowed if *expression* evaluates to true.

<RequireNone> Directive

Description: Enclose a group of authorization directives of which none must succeed for the enclosing directive to not fail.

<RequireAny> Directive

Description: Enclose a group of authorization directives of which one must succeed for the enclosing directive to succeed.

<RequireAll> Directive

Description: Enclose a group of authorization directives of which none must fail and at least one must succeed for the enclosing directive to succeed.

Clippings are from the Apache manual

Require, continued

Using multiple providers

With the introduction of the new provider based authentication and authorization architecture, you are no longer locked into a single authentication or authorization method. In fact any number of the providers can be mixed and matched to provide you with exactly the scheme that meets your needs. In the following example, both the file and LDAP based authentication providers are being used.

```
<Directory "/www/docs/private">
    AuthName "Private"
    AuthType Basic
    AuthBasicProvider file ldap
    AuthUserFile "/usr/local/apache/passwd/passwords"
    AuthLDAPURL ldap://ldaphost/o=yourorg
    Require valid-user
</Directory>
```

```
<Directory "/www/docs/private">
    AuthName "Private"
    AuthType Basic
    AuthBasicProvider file
    AuthUserFile "/usr/local/apache/passwd/passwords"
    AuthLDAPURL ldap://ldaphost/o=yourorg
    AuthGroupFile "/usr/local/apache/passwd/groups"
    Require group GroupName
    Require ldap-group cn=mygroup,o=yourorg
</Directory>
```

```
Require ip address
```

```
Require host domain_name
```

Note that some connections receive a negative invitation, a not. Shoo, go away.

```
<RequireAll>
    Require all granted
    Require not ip 10.252.46.165
</RequireAll>
```

```
<RequireAll>
    Require all granted
    Require not ip 192.168.205
    Require not host phishers.example.com moreidiots.example
    Require not host ke
</RequireAll>
```

```
<Directory "/www/docs">
    <RequireAll>
        Require group alpha beta
        Require not group reject
    </RequireAll>
</Directory>
```

```
AuthType Basic
AuthName "Restricted Resource"
AuthBasicProvider file
AuthUserFile "/web/users"
AuthGroupFile "/web/groups"
Require group admin
```

What Apache does when pointed to a file or directory

- If an incoming URL specifies a filename then it is sought in the URL provided path/filename, and that path is relative to the *DocumentRoot* file system location. If the file is not found then an error is reported to the user.
- If no filename is stated, only a path (relative to *DocumentRoot*), then Apache looks in that directory for a default index file, typically named *index.html*. Directive *DirectoryIndex* defines such file names.
- If a DirectoryIndex file is not present then a directory listing is performed. "*Options Indexes*" and module *mod_autoindex* are required to show a listing, else an empty screen.
- Ah ha! We can show directory listings (if we wish). Note the Options part.
- We must take care to not mistakenly expose directories or specific files. Protection involves using Apache directives to forbid kinds of access, often conditional on the caller's identity. <Directory> and <Location> clauses are standard filtration places, aided by *require* directives. Beware clever guessers

Directory listings, default and with local styling

Index of /pub/bsuk2001

Name	Last modified	Size
bp100.pdf	2016-06-22 12:04	1.4M
bp100.ppt	2001-07-14 00:31	433K
io131.pdf	2016-06-22 12:04	6.0M
io131.ppt	2001-07-19 06:35	13M
jdp400.pdf	2016-06-22 12:04	2.0M
jdp400.ppt	2001-07-17 04:36	716K
tut103.pdf	2016-06-22 12:04	1.7M
tut103.ppt	2001-07-14 01:55	2.0M
tut156.pdf	2016-06-22 12:04	1.2M
tut156.ppt	2001-07-14 02:09	1.0M
tut162.pdf	2016-06-22 12:04	3.1M
tut162.ppt	2001-07-31 03:48	3.1M

Index of /pub/bsuk2001

Name	Last modified	Size
bp100.pdf	2016-06-22 12:04	1.4M
bp100.ppt	2001-07-14 00:31	433K
io131.pdf	2016-06-22 12:04	6.0M
io131.ppt	2001-07-19 06:35	13M
jdp400.pdf	2016-06-22 12:04	2.0M
jdp400.ppt	2001-07-17 04:36	716K
tut103.pdf	2016-06-22 12:04	1.7M
tut103.ppt	2001-07-14 01:55	2.0M
tut156.pdf	2016-06-22 12:04	1.2M
tut156.ppt	2001-07-14 02:09	1.0M
tut162.pdf	2016-06-22 12:04	3.1M
tut162.ppt	2001-07-31 03:48	3.1M

<Directory foo>

Options Indexes FollowSymLinks MultiViews IncludesNoExec State **Indexes** for a listing

IndexStyleSheet “/index.css” Has my embellishments

Visit https://netlab1.net/, do a browser “View Page Source”, and grab a copy of my index.css file

A mixed ingredients example

```
# cat /etc/apache2/conf.d/phpLdapPasswd.conf
Alias  /cpw  /home/user/phpLdapPasswd
<Directory /home/user/phpLdapPasswd>
        Options FollowSymLinks
        DirectoryIndex index.php
        require ip 11.22.33.44/28 127.0.0.1
        RewriteEngine On                          Rewrite triad redirects the caller to try again with https, not  http
        RewriteCond %{HTTPS} off                  Report True (do the rule below) if https was not used
        RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI} [QSA,R]      query string append, redirect
</Directory>
<Location /cpw/hidden>
        require all denied                        Don't go here, I warn you. Apache rebuffs everyone at that door.
</Location>
```

```
<?php
// Include the configuration and functions, and display an error message
// if unable to do so.
if ((!@include './hidden/config.php') || (!@include './hidden/functions.php')) {
?>
<html>
<head>
<title>Fatal Error</title>
</head>
<body>
  … on and on
```

Ordinary file *index.html* is replaced here by running a PHP program.
Directive *DirectoryIndex* does the replacement.

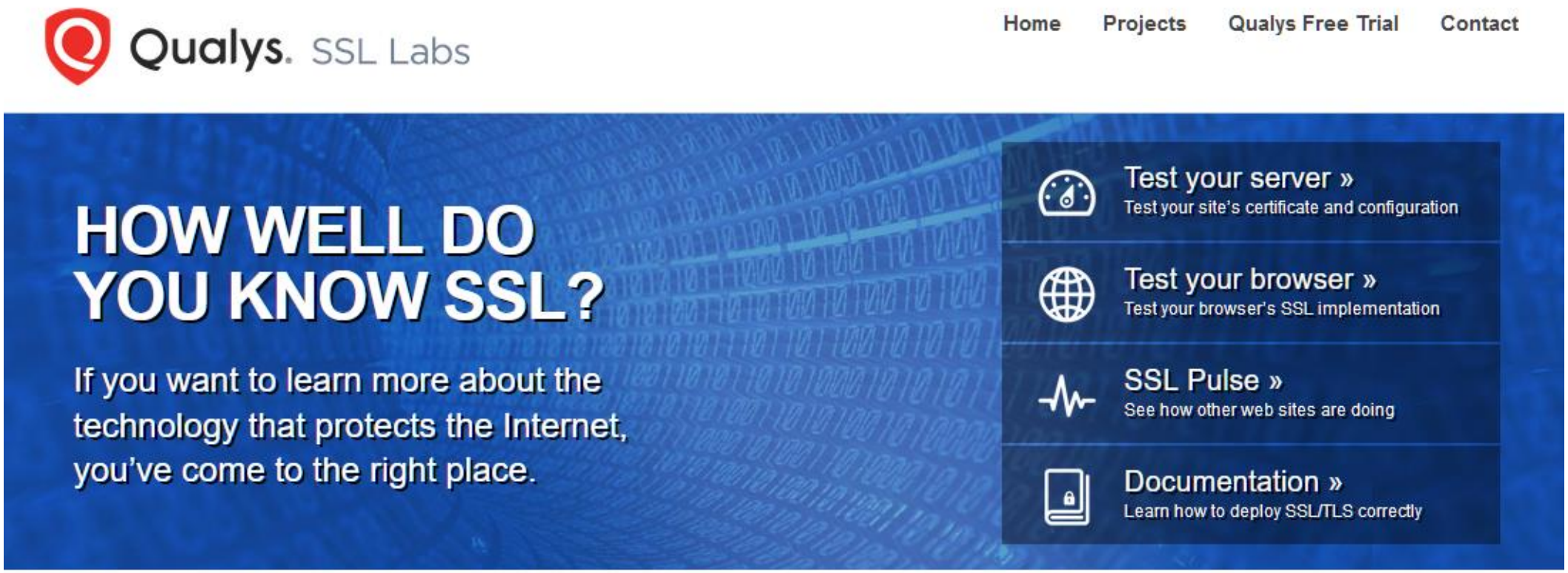
OES components proper

- OES components for Apache are spread around
- At top level is file /etc/apache2/conf.d/OES-httpd.conf which brings in many files via directive *Include /etc/opt/novell/httpd/conf.d/*.conf.*
- Of those we find netstorage, xsrv, nps, ganglia, and welcome-novell.
- The main welcome page web files are located in */srv/www/htdocs/welcome-novell/** (index.html language variations)
- There are many files, with plenty of details, as a complicated set. Patience is required to trace and decode matters if needs be.
- Note use of <Location> clauses holding directive "*require all denied*" to protect areas from web access while permitting local direct access.
- We should place "*require*" directives in many files to allow administration, but not tourists nor local clever guessers.

Comments on exposing OES Java applications

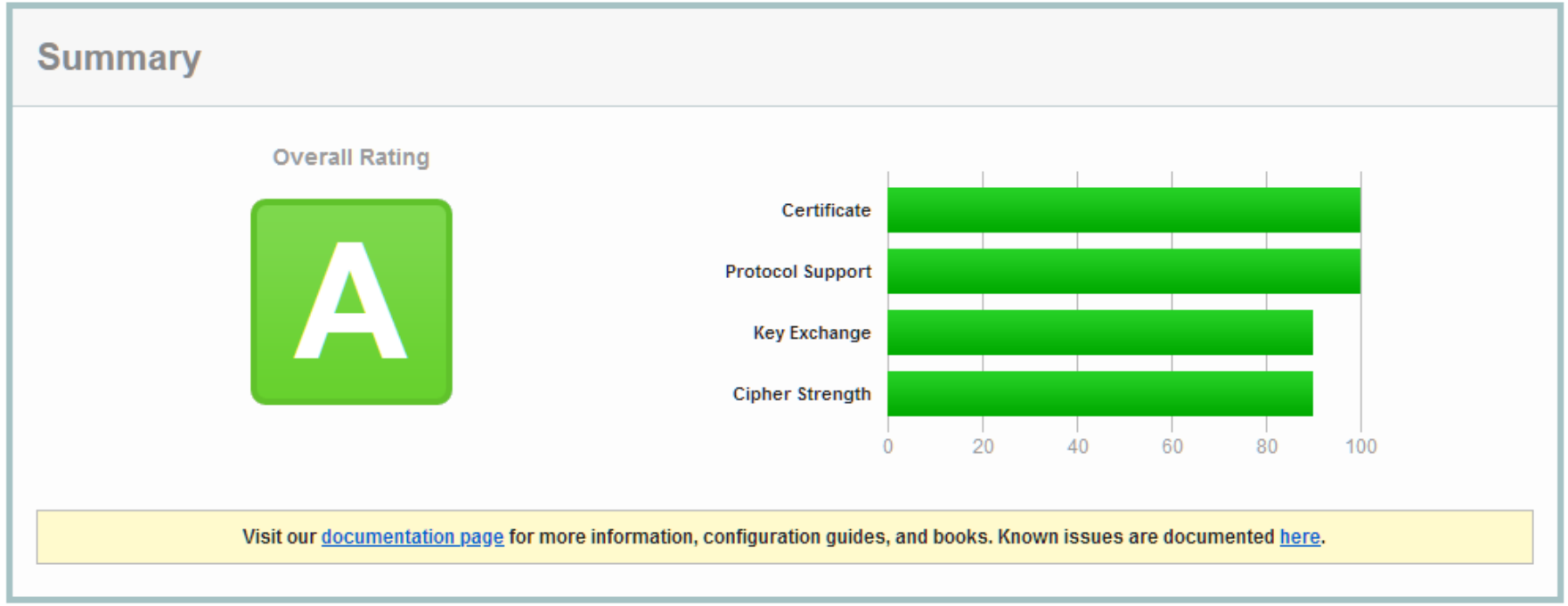
- I choose to avoid exposing Java ports 8xxx and the like. Instead I configure Apache to support nice URLs and then proxy commands to Java. Thus the user connection remains on https:// port 443. See presentation Hiding Tomcat behind Apache.
- The proxy employs directive pair *proxypass* and *proxypassreverse* which should use the same arguments. The pair handles bidirectional traffic. Apache masquerades as doing the work, which is actually done by a background process or machine. Tsk tsk, a fake worker, outsourcing.
- The proxy terminates SSL/TLS connections from/to the outside, thus Apache is the server for certificate and crypto work with the world.
- Proxy work is discussed at length in the Apache documentation, in what has become a small industry of activity. Thus please review those docs.

A primary test tool, Internet based free web testing



From https://www.ssllabs.com
Also review its document collection and implementation census

SSL Labs report about a well configured web site (TLS v1.2)



OES2018 SP2, configured

From https://www.ssllabs.com

Avoid adding HSTS for an ego-boosting-only A+ grade. HSTS is invasive to clients.

SSL Labs report about a well configured site, controls

Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info)
GOLDENDOODLE	No (more info)
OpenSSL 0-Length	No (more info)
Sleeping POODLE	No (more info)
Downgrade attack prevention	Unknown (requires support fo
SSL/TLS compression	No

Secure Renegotiation is good, but do not let clients initiate it

Compression by SSL/TLS, avoid

Forward Secrecy (change keys often) is very desirable, set by choice of crypto algorithm

Forward Secrecy	Yes (with most browsers) ROBUST (more info)
ALPN	Yes h2 http/1.1
NPN	No
Session resumption (caching)	Yes
Session resumption (tickets)	No
OCSP stapling	Yes
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	No (more info)
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No

Session resumption: tickets are less good than caching

OCSP stapling is good

HSTS modifies clients, be wary

Please note carefully:
Controls improve security and robust comms. Usage is via each application which passes settings to the crypto engine. Thus review application docs.

More pages, not shown here

SSL Labs report about a modern web (TLS v1.2, 1.3)

Configuration

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

This machine offers two Protocols, latest and current, each with its own Cipher Suites list.

Cipher Suites

# TLS 1.3 (suites in server-preferred order) ← (server's preferred order)	
TLS_AES_256_GCM_SHA384 (0x1302) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_AES_128_GCM_SHA256 (0x1301) ECDH x25519 (eq. 3072 bits RSA) FS	128
# TLS 1.2 (suites in server-preferred order) ← (server's preferred order)	
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) DH 2048 bits FS	128
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) DH 2048 bits FS	256

TLS v1.3 mandated cipher list (built-in)

TLS v1.2 via ***our*** cipher suite list

Latest TLS v1.3 and fallback v1.2

Two TLS version dependent cipher suites.

"FS" tag is Forward Secrecy, desirable

A SLES 15 machine, thus a look ahead to OES2023

Locally run script testssl.sh as *./testssl.sh hostname*

```
##################################################################
     testssl.sh       3.0 from https://testssl.sh/

       This program is free software. Distribution and
              modification under GPLv2 permitted.
       USAGE w/o ANY WARRANTY. USE IT AT YOUR OWN RISK!

        Please file bugs @ https://testssl.sh/bugs/

##################################################################

 Using "OpenSSL 1.0.2-chacha (1.0.2k-dev)" [~183 ciphers]
 on netlab:./bin/openssl.Linux.x86_64
 (built: "Jan 18 17:12:17 2019", platform: "linux-x86_64")


 Start 2021-05-18 14:12:06

 rDNS
 Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    not offered and downgraded to a weaker protocol
 NPN/SPDY   not offered
 ALPN/HTTP2 h2, http/1.1 (offered)
```

Testing OES2018 SP2 Apache.

This small script is a useful tool, similar to that of SSL Labs, and it can reveal other useful detail.

It can test more than web serving.

testssl.sh -t smtp host:25
testssl.sh -t imap host:143
testssl.sh --mx host
testssl.sh host:636

plus -t ftp, lmtp, xmpp, telnet, ldap, etc.
-t means try StartTLS with the protocol.
See testssl.sh --help for full listing.
To use your local openssl add option
--openssl /usr/bin/openssl

See https://testssl.sh

Only a subset is shown here

Testssl.sh

```
 Testing cipher categories

 NULL ciphers (no encryption)                  not offered (OK)
 Anonymous NULL Ciphers (no authentication)    not offered (OK)
 Export ciphers (w/o ADH+NULL)                 not offered (OK)
 LOW: 64 Bit + DES, RC[2,4] (w/o export)       not offered (OK)
 Triple DES Ciphers / IDEA                     not offered
 Obsolete: SEED + 128+256 Bit CBC cipher       not offered
 Strong encryption (AEAD ciphers)              offered (OK)

 Testing robust (perfect) forward secrecy, (P)FS -- omitting Null Authentication/Encryption, 3DES, RC4

 PFS is offered (OK)          ECDHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-GCM-SHA256
                              DHE-RSA-AES128-GCM-SHA256
 Elliptic curves offered:     secp256k1 prime256v1 secp384r1 secp521r1 brainpoolP256r1 brainpoolP384r1
                              brainpoolP512r1
 DH group offered:            RFC3526/Oakley Group 14 (2048 bits)

 Testing server preferences

 Has server cipher order?     yes (OK)
 Negotiated protocol          TLSv1.2
 Negotiated cipher            ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
 Cipher order
    TLSv1.2:   ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES128-GCM-SHA256
               DHE-RSA-AES256-GCM-SHA384
```

- Noted protocols, ciphers, controls and certificate pc[3]

MindWorks UK

Testssl.sh

Testing vulnerabilities

```
 Heartbleed (CVE-2014-0160)                not vulnerable (OK), timed out
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session ticket extension
 ROBOT                                     Server does not support any cipher suites that use RSA key trans
port
 Secure Renegotiation (RFC 5746)           supported (OK)
 Secure Client-Initiated Renegotiation     not vulnerable (OK)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    no HTTP compression (OK)  - only supplied "/" tested
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK), no SSLv3 support
 TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible (OK), no protocol below TLS 1.2 offered
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate elsewhere with SSLv2 en
abled services
                                           https://censys.io/ipv4?q=91D2D1E0BECDD2BB60A0388BFD6937902ED9EEE
1D0FD9FCA45A81B879B00EE8C could help you to find out
 LOGJAM (CVE-2015-4000), experimental      common prime with 2048 bits detected: RFC3526/Oakley Group 14 (2
048 bits),
                                           but no DH EXPORT ciphers
 BEAST (CVE-2011-3389)                     not vulnerable (OK), no SSL3 or TLS1
 LUCKY13 (CVE-2013-0169), experimental     not vulnerable (OK)
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)
```

Vulnerabilities see: https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/

MindWorks UK

Testssl.sh

We recall “long goodbyes”. Here is a “long hello”.

```
 Testing server defaults (Server Hello)

 TLS extensions (standard)    "server name/#0" "renegotiation info/#65281" "EC point formats/#11"
                              "status request/#5" "heartbeat/#15" "application layer protocol negotiation/#16"
 Session Ticket RFC 5077 hint no -- no lifetime advertised
 SSL Session ID support       yes
 Session Resumption           Tickets no, ID: yes
 TLS clock skew               Random values, no fingerprinting possible
 Signature Algorithm          SHA256 with RSA
 Server key size              RSA 2048 bits
 Server key usage             Digital Signature, Key Encipherment
 Server extended key usage    TLS Web Server Authentication, TLS Web Client Authentication
 Serial / Fingerprints        31610444D74273BA / SHA1 48DA8E8B6F5DEAC46ADC19AFCCED02D1E268A1AD
                              SHA256 91D2D1E0BECDD2BB60A0388BFD6937902ED9EEE1D0FD9FCA45A81B879B00EE8C

 Common Name (CN)             *.netlab1.net
 subjectAltName (SAN)         *.netlab1.net netlab1.net
 Issuer                       Starfield Secure Certificate Authority - G2 (Starfield Technologies, Inc. from US)
 Trust (hostname)             Ok via SAN (same w/o SNI)
 Chain of trust               Ok
 EV cert (experimental)       no
 ETS/"eTLS", visibility info  not present
 Certificate Validity (UTC)   291 >= 60 days (2020-03-24 03:38 --> 2022-04-07 08:00)
 # of certificates provided   3
 Certificate Revocation List  http://crl.starfieldtech.com/sfig2s1-187.crl
 OCSP URI                     http://ocsp.starfieldtech.com/
 OCSP stapling                offered, not revoked
 OCSP must staple extension   --
 DNS CAA RR (experimental)    not offered
 Certificate Transparency     yes (certificate extension)
```

Details are in the URL below, if interested

Server name is same as in the cert

Intermediary chain is valid

Cert offers OCSP responder URL, Apache also offers OCSP results

Extensions: https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml

Streaming video support, mod_h264

1/2

- Enhance browser's control of playing a video file (skip about quickly).
- Obtain the code from h264.code-shop.com, download option.
- Ensure Apache2's APXS support is present: in YaST, Software Management, search on apache2, enable apache2-devel (a repo addition might be needed).

- In the unpacked mod_h264_streaming-2.27 directory, do

  ./configure --with-apxs=/usr/sbin/apxs2
  Run *make* followed by *make install*

  The result is file */usr/lib64/apache2/mod_h264_streaming.so*
  which is in the Apache2 core (MPM independent) module area

- Add term *h264_streaming* to list APACHE_MODULES= in file /etc/sysconfig/apache2

Streaming video support, mod_h264

2/2

In /etc/apache2/conf.d create a small file, say mp4.conf, holding:

```
<IfModule mod_h264_streaming.c>
AddHandler h264-streaming.extensions .mp4 .webm .ogv
AddHandler smooth-streaming.extensions .ism
AddType video/webm .webm
AddType video/mp4 .mp4
AddType video/x-m4v .m4v
AddType video/ogg .ogv
AddType video/quicktime .mov
</IfModule>
```

Restart Apache (rcapache2 restart), check logs, double check via *apachctl –t –D DUMP_MODULES* to see our addition appear. Test with a large video file.

On the wire zip compression, mod_deflate

Apache Module mod_deflate

Description:	Compress content before it is delivered to the client
Status:	Extension
Module Identifier:	deflate_module
Source File:	mod_deflate.c

My file */etc/apache2/conf.d/deflate.conf* reads:

```
<IfModule mod_deflate.c>
 AddOutputFilterByType  DEFLATE  text/css video/mp4 video/webm image/png image/jpeg application/pdf  text/plain
BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
</IfModule>
```

Add *deflate* to list APACHE_MODULES= in file /etc/sysconfig/apache2.

See regular expression note about the optional BrowserMatch line.

See presentation SSL/TLS for system admins about avoiding crypto cracking via user interactive input carried in compressed files. Stay with static files for mod_deflate.

Mod_speling manual snippet

Apache Module mod_speling

Available Languages: en | fr | ja | ko

Description:	Attempts to correct mistaken URLs by ignoring capitalization, or attempting to correct various minor misspellings.
Status:	Extension
Module Identifier:	speling_module
Source File:	mod_speling.c

Summary

Requests to documents sometimes cannot be served by the core apache server because the request was misspelled or miscapitalized. This module addresses this problem by trying to find a matching document, even after all other modules gave up. It does its work by comparing each document name in the requested directory against the requested document name **without regard to case**, and allowing **up to one misspelling** (character insertion / omission / transposition or wrong character). A list is built with all document names which were matched using this strategy. **Erroneous extension** can also be fixed by this module.

If, after scanning the directory,

- no matching document was found, Apache will proceed as usual and return an error (`404 - document not found`).
- only one document is found that "almost" matches the request, then it is returned in the form of a redirection response (`301 - Moved Permanently`).
- more than one document with a close match was found, then the list of the matches is returned to the client, and the client can select the correct candidate (`300 - Multiple Choices`).

Forgiving finger fumbles

```
<IfModule mod_speling.c>
## CheckSpelling alone does case
## insensitivity and one-character
## typo  corrections
      CheckSpelling on
      CheckCaseOnly on
</IfModule>
```

Module's default on/off settings:

Checkbasenamematch	on
CheckCaseOnly	off
CheckSpelling	off

Add *speling* to list APACHE_MODULES= in file /etc/sysconfig/apache2

In conclusion

- Add modern crypto and certificate handling to Apache.
- Include directives about crypto controls and proper net.behaviour.
- Investigate use of *require* to protect selected areas.
- Consider directory index local styling.
- Consider streaming video support and on-the-wire compression of selected file kinds.
- Review the two companion presentations, Intro to Apache and SSL/TLS for system admins, for further information.
- This business is “relatively” easy to accomplish, yet careful reading of the Apache documentation plus testing are recommended.
- Enjoy the results of your good deeds.

MindWorks UK

MindWorks Inc. Ltd
210 Burnley Road
Weir
Bacup
OL13 8QE UK

Telephone: +44 (0) 170 687 1900
Fax: +44 (0) 170 687 8203
Web: www.mindworksuk.com
Email: training@mindworksuk.com