hp e3000

webwise secure web server

hp webwise mpe/ix secure web server

Presented by Mark Bixby

mark_bixby@hp.com

Solution Symposium 2002

prerequisite knowledge

- General Apache knowledge
- POSIX shell basics
- Hierarchical File System basics

hp e3000

webwise secure web server

webwise A.03.00 product overview

- A.01.00 released as a separately purchasable product for 6.5
- A.03.00 now bundled into 7.5 FOS as a drop-in replacement for Apache A.02.00
- adds SSL encryption and X.509 authentication to Apache

hp e3000

webwise secure web server

webwise A.03.00 is built from…

- Apache 1.3.22
- Mod_ssl 2.8.5 SSL/TLS encryption module
  - MM 1.1.3 shared memory library
- OpenSSL 0.9.6b general purpose SSL/TLS and crypto toolkit
- RSA BSAFE Crypto-C 5.2 crypto toolkit
  - RC2, RC4, and RSA algorithms

new apache functionality since 1.3.14

- mostly bug fixes & portability enhancements
- LogFormat %c for logging connection status at request completion
- mod_auth file-owner and file-group authentication enforcement
- rotatelogs utility supports date/timestamp references in logfile names
- Apache manual pages moved outside of the htdocs DocumentRoot; i.e. /APACHE/PUB/htdocs/manual moved to /APACHE/CURRENT/htmanual

webwise changes since A.01.00

- Apache 1.3.9 updated to 1.3.22
- child processes run as WWW.APACHE instead of SECURE.APACHE; may have file ownership and permissions implications!
- uses the same V.UU.FF-based file layout scheme as Apache A.02.00 (the old SECURE.APACHE group is not modified or referenced by A.03.00)

hp e3000

webwise secure web server

migrating from previous versions of apache or webwise

- Create new JHTTPD from JHTTPD.sample
- Create new config files from corresponding *.sample files
- Copy existing WebWise A.01.00 server key and certificate to new A.03.00 locations
- Copy existing WebWise A.01.00 htdocs content and cgi-bin scripts from /APACHE/SECURE to the new A.03.00 /APACHE/PUB locations, or modify the new A.03.00 config files to refer to the old A.01.00 locations

hp e3000

webwise secure web server

migrating to hpux

- WebWise on MPE shares the same core architecture as the Apache bundle on HPUX
- 100% upward compatible
- a few additional standard Apache modules on HPUX
- extra HP modules on HPUX for integration with other HPUX products

hp e3000

webwise secure web server

mod_ssl is...

- The heart of WebWise
  - encrypted TCP connections
  - client and server X.509 authentication
- Consists of:
  - Patches to extend the Apache API (EAPI)
  - the mod_ssl module
  - bin/sign.sh script
- bin/openssl command line utility included for key/certificate management

hp e3000

webwise secure web server

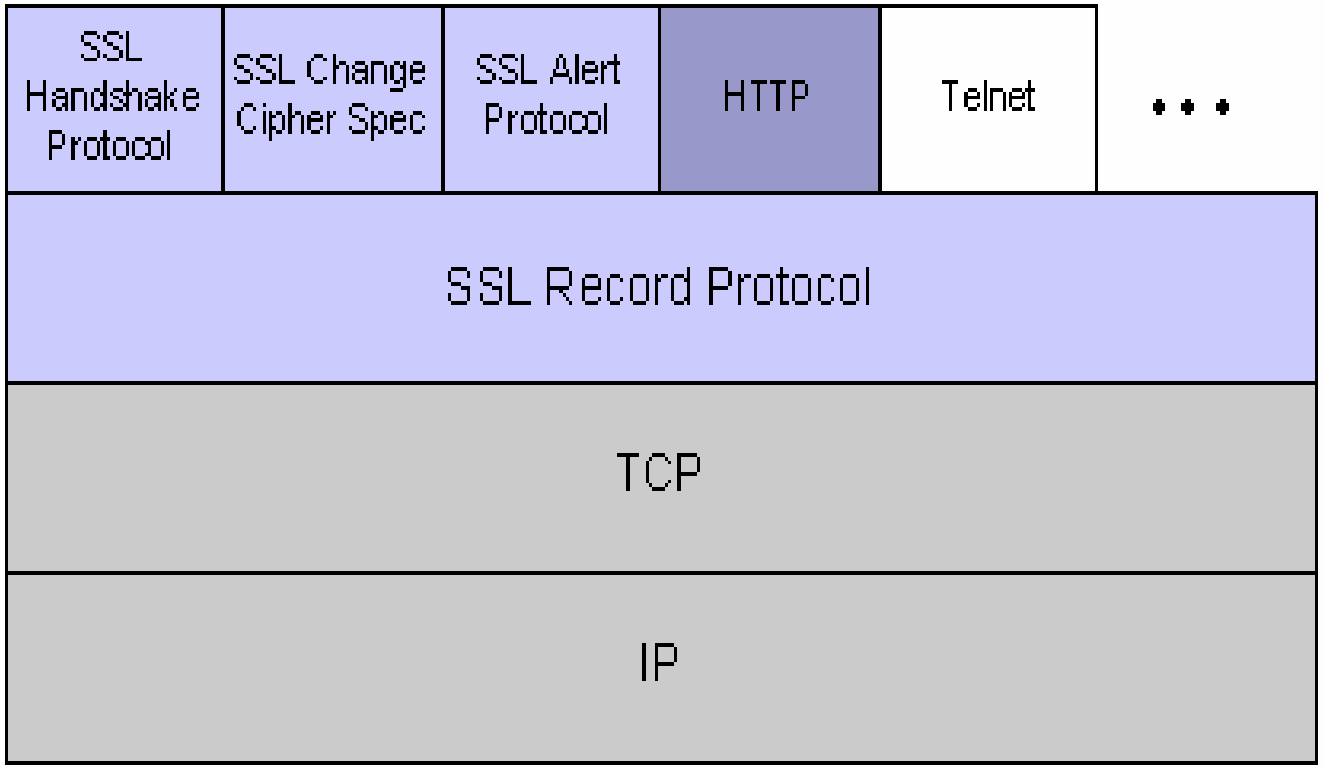
mod_ssl is NOT…

- a substitute for a firewall
- a substitute for good host security practices
- a substitute for good application security practices
- a substitute for good human security practices

definitions: secure sockets layer (ssl)

- A protocol layer between any application stream protocol (such as HTTP) and TCP that allows secure communications via encryption, digests, signatures, and authentication
- SSLv2.0 - vendor standard from Netscape
- SSLv3.0 - expired Internet Draft from Netscape
- Supported by all browsers

definitions: secure sockets layer (cont.)



hp e3000

webwise secure web server

definitions: transport layer security (tls)

- An evolution of SSLv3.0
- Defined in RFC2246
- Supported by all modern browsers

definitions: key

- A really big random number (1024 bits)
  - 40 bits? 56 bits? 128 bits? 1024 bits? SAY WHAT???
- Split into two mathematically related components:
  - private key
  - public key
- A key establishes your identity -- protect it! (chmod 400 and pass phrase)
- Both servers and clients have keys
- RSA keys / algorithm defined by RFC 2437

definitions: private key

- Uniquely identifies you
- Protect it with your life!
- You use it to:
  - create digital signatures
  - create digital certificates
  - decrypt data sent to you that was encrypted with your public key

definitions: public key

- Allows the public to send you encrypted data which only you can decrypt with your private key
- Your public key is also included in your certificate

definitions: message digest

- Short, fixed-length representation of longer, variable-length messages (hash)
- Can’t determine original msg from digest
- No two messages have the same digest
- Digest algorithms:
  - MD5 (128-bit hash)
  - SHA1 (160-bit hash)

definitions: digital signature

- Message digest (plus sequence number) encrypted with sender's private key
- Alter the message and the digest won't match
- Alter the digest and the public key decryption won't work

definitions: certificate

- Validates your identity to others
- Format defined by X.509 standard
- Created by a Certificate Authority
- Contains:
  - your identity (name, company, locality, etc)
  - your public key
  - validity dates
  - the identity and signature of a trusted agency called a Certificate Authority

definitions: certificate authority (ca)

- A trusted agency that issues certificates
- Validates the identity of a person requesting a certificate
- The CA signs the certificate request with their own CA certificate, thus creating a certificate for the requestor
- CA certificate may be self-signed (root-level), or signed by a higher CA
- You can be your own CA!

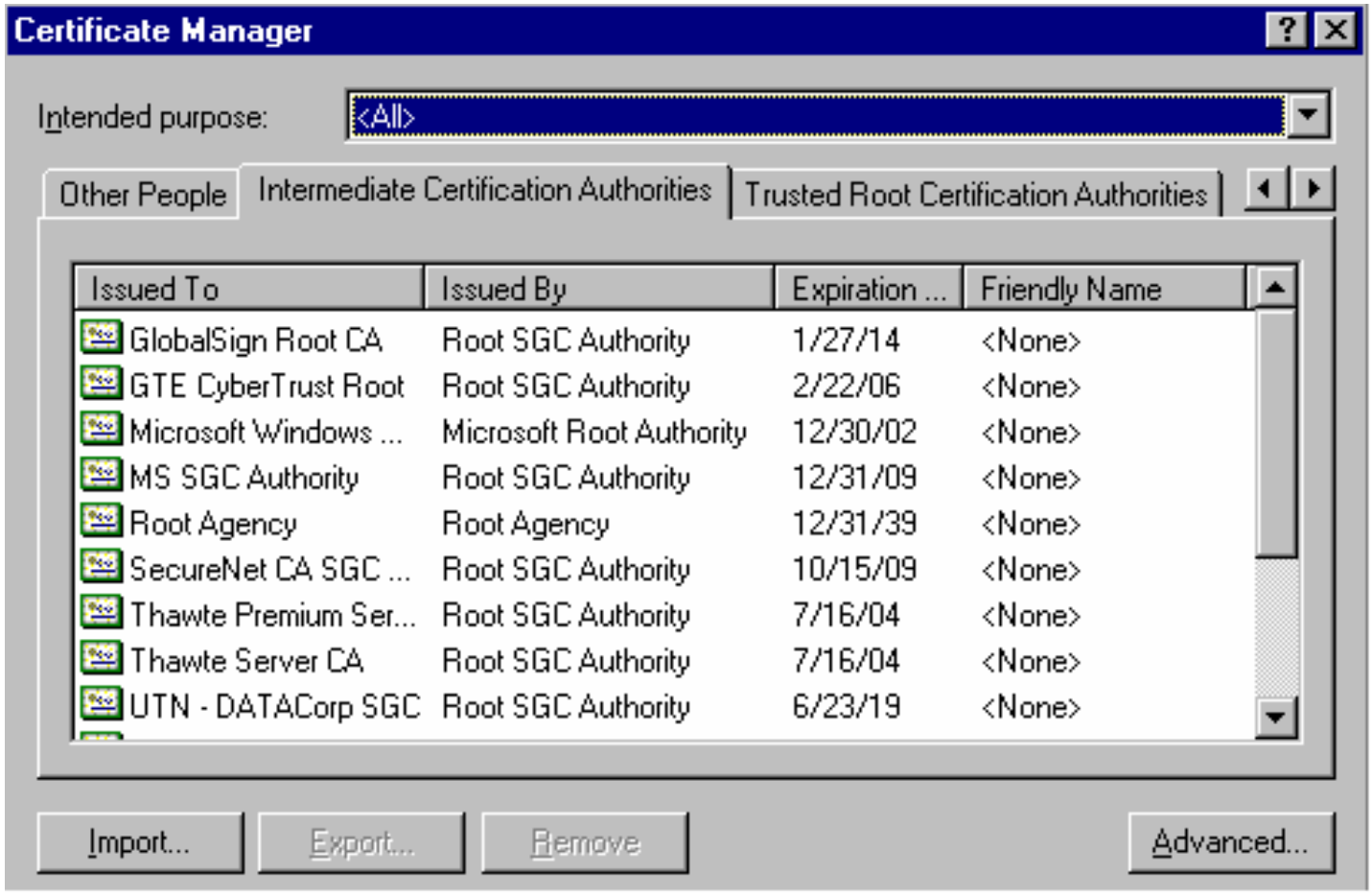
definitions: certificate authority (cont.)

- Browsers are pre-configured to trust certain CAs
  - Netscape: Edit, Preferences, Privacy & Security, Certificates, Manage Certificates, Authorities
  - MSIE: Tools, Internet Options, Content, Certificates, Intermediate Certification Authorities, Trusted Root Certification Authorities
- You can add new trusted CAs
- Server certificates signed by trusted CAs are automatically accepted!

hp e3000

webwise secure web server

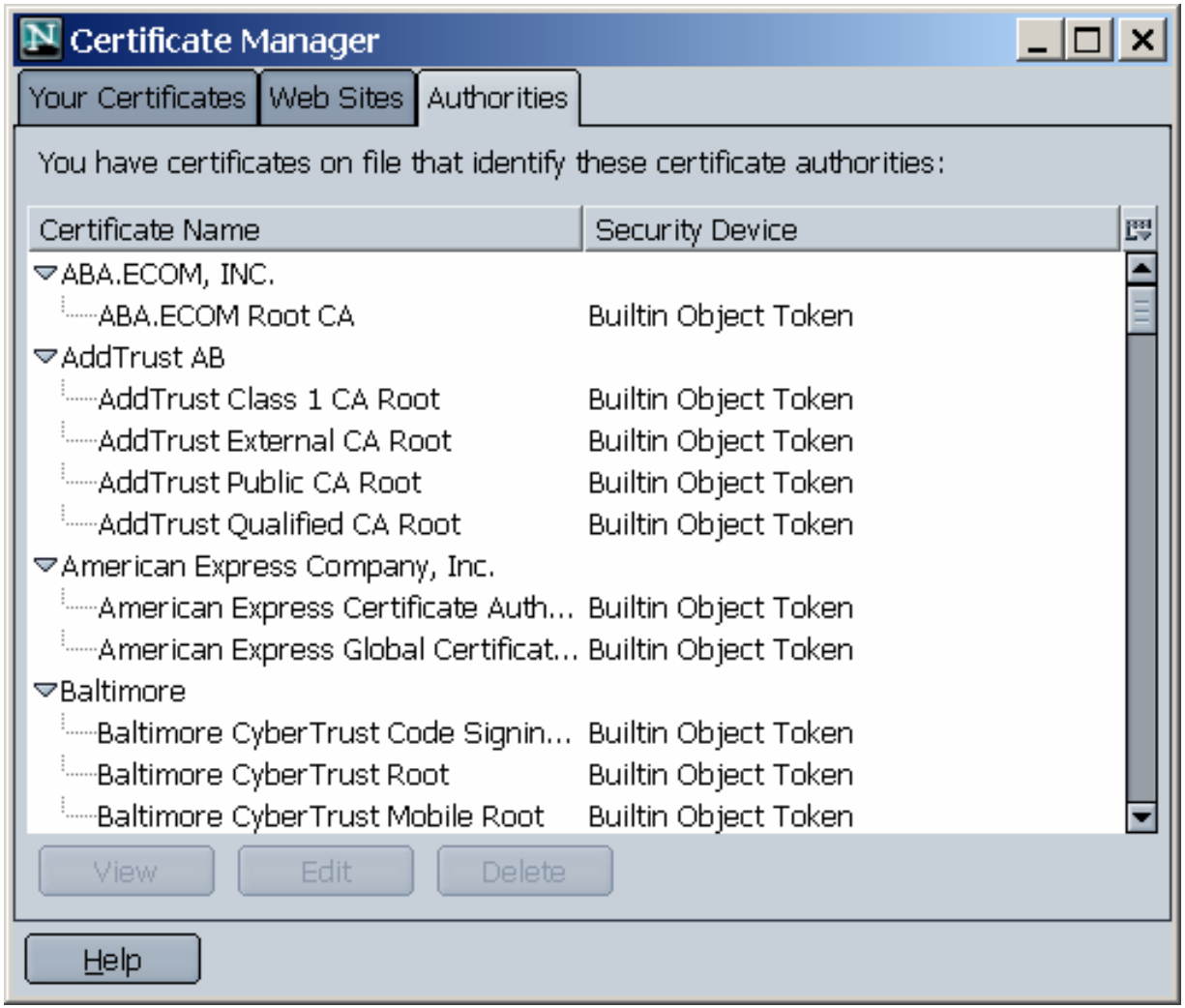
msie5.5 ca window



hp e3000

webwise secure web server

netscape 6.2.1 ca window



hp e3000

webwise secure web server

definitions: certificate signing request (csr)

- What you send to a CA in order to request a certificate
- Contains:
  - your identity (name, company, locality, etc)
  - your public key
- The CA signs your CSR with the CA certificate, resulting in your certificate

definitions: certificate chain

- Every certificate is signed by a CA
- CA certificates are signed by other CAs
- A chain of valid CA signatures (assumes trust is inherited)

hp e3000

webwise secure web server

definitions: certificate revocation list (crl)

- A list of certificates that a CA has revoked (I.e. invalidated)
- An employer CA would revoke the certificate of a terminated employee and list that certificate in a CRL
- Must be obtained manually from the CA

hp e3000

webwise secure web server

mod_ssl configuration directives - sslengine (required)

- Specifies whether SSL/TLS is enabled; typically used inside <VirtualHost>
- on: SSL/TLS is enabled
- off: SSL/TLS is disabled

sslmutex (required)

- Specifies the method of synchronization used between WebWise children
- none - use at your own risk!
- File:/path/to/mutex - uses fcntl() locking on the specified filename with the parent PID appended for uniqueness
- sem - not implemented for MPE!

hp e3000

webwise secure web server

sslrandomseed (required)

- SSLRandomSeed context source [bytes]
- Seeds the Pseudo Random Number Generator (PRNG)
- Context is either “startup” or “connect”
- Sources:
  - builtin - current time, process id, and 1KB of random scoreboard data
  - file:/path/to/source - reads from a file
  - exec:/path/to/program - reads from program stdout

sslsessioncache (recommended)

- Specifies the SSL session cache method used to avoid repeated (slow) SSL handshaking
- none - no cache; terrible performance
- dbm:/path/to/datafile - disk file cache
- shmht:/path/to/datafile(size) - shared memory cache hash table (file not created on MPE)
- shmcb:/path/to/datafile(size) - shared memory cache cyclic buffers (file not created on MPE); best performance!

hp e3000

webwise secure web server

sslsessioncachetimeout (optional)

- Specifies the session cache timeout in seconds
- Default is 300

hp e3000

webwise secure web server

sslprotocol (optional)

- Specifies accepted SSL protocols
- + or - syntax like Options
- Default is all
- SSLv2
- SSLv3
- TLSv1
- All
- SSLProtocol All -SSLv2

sslciphersuite (optional)

- Specifies the ordered list of ciphers to be negotiated during the SSL handshake
- Default: ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
- 128-bit RC4 will be chosen first
- **`/APACHE/CURRENT/bin/openssl ciphers -v`** will list all available ciphers

sslcertificatekeyfile (required)

- Specifies the server key file
- /APACHE/PUB/conf/ssl.key/server.key
- Protect the key file with your life!
- Well, maybe just with chmod 400 permissions and a pass phrase
- Whoever has the key can impersonate you!

sslpassphrasedialog (recommended)

- How to obtain the pass phrase for encrypted private keys
- builtin - read the pass phrase from $STDIN after !RUN HTTPD
- exec:/path/to/program - program prints pass phrase to $STDLIST; two parms:
  - servername:portname
  - RSA or DSA
- Protect the pass phrase!
  - Whoever knows the pass phrase can get your key!

sslcertificatefile (required)

- Specifies the web server certificate file
- /APACHE/PUB/conf/ssl.crt/server.crt
- May also contain a private key in the same file, but this isn't recommended
- Protect this file with chmod 400 permissions

sslcertificatechainfile (optional)

- Specifies the all-in-one file containing the concatenated CA certificates of all CA signers between the server certificate and the CA root
- Makes it easier for browsers to validate your server certificate

sslcacertificatefile (optional)

- Specifies the all-in-one file containing the concatenated CA certificates that might have been used to sign the certificates of your clients
- This directive and/or SSLCACertificatePath is required for client authentication

sslcacertificatepath (optional)

- Specifies the directory containing all of the individual CA certificates that might have been used to sign the certificates of your clients
- Hash symlinks must be present in this directory
- /APACHE/PUB/conf/ssl.crt/Makefile will create the hash symlinks
- This directive or SSLCACertificateFile is required for client authentication

hp e3000

webwise secure web server

sslcarevocationfile (optional)

- Specifies the all-in-one file containing the concatenated CRLs of all of the CAs that might have signed the certificates of your clients
- This directive or SSLCARevocationPath is recommended for client authentication

sslcarevocationpath (optional)

- Specifies the directory containing all of the individual CRLs of all of the CAs that might have signed the certificates of your clients
- Hash symlinks must be present in this directory
- /APACHE/PUB/conf/ssl.crl/Makefile will create the hash symlinks
- This directive or SSLCARevocationFile is recommended for client authentication

sslverifyclient (optional)

- Specifies the type of client certificate authentication desired
  - none: no client certificate is required
  - optional: the client may present a valid certificate
  - require: the client must present a valid certificate
  - optional_no_ca: the client may present a certificate, but it doesn't have to be valid
- "optional" doesn't work with all browsers, and "optional_no_ca" is really for testing

sslverifydepth (optional)

- Specifies the maximum number of CA certificates to be used when validating the client certificate
- 0 means that self-signed client certificates are accepted only
- 1 (default) means the client certificate can be self-signed or has to be signed by a CA which is directly known to the server, etc, etc

hp e3000

webwise secure web server

ssllog (required)

- Specifies the mod_ssl log file
- Serious errors are duplicated to the ErrorLog
- |/path/to/program or /path/to/file

sslloglevel (optional)

- Specifies the logfile verbosity fence
- none - no dedicated logging, but “error” messages still written to ErrorLog
- error - fatal messages
- warn - non-fatal messages
- info - major processing steps
- trace - minor processing steps
- debug - very VERY verbose!

sslrequiressl (optional)

- Forbids access unless SSL is being used for this connection
- Useful for protecting against exposing sensitive data over non-SSL connections

hp e3000

webwise secure web server

sslrequire (optional)

- Allow access only if an arbitrarily complex boolean expression is true
- **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/^192\.76\.162\.[0-9]+$/**

ssloptions (optional)

- Specifies various SSL-related runtime options
- Similar to Options directive
- StdEnvVars - creates SSL-related environment variables for CGI/SSI applications; expensive!
- CompatEnvVars - creates extra environment variables for compatibility with other Apache-based SSL servers

ssloptions (cont.)

- ExportCertData - creates environment variables containing applicable X.509 certificates in PEM format
- FakeBasicAuth - client certificate Subject is used as userid lookup into Basic Authentication password file; user not prompted for password (assumed to be “password”)
- StrictRequire - access denial due to SSLRequire or SSLRequireSSL overrides all other access checking

ssloptions (cont.)

- OptRenegotiate - by default, every per-directory SSL parameter reconfiguration causes a full SSL renegotiation handshake (slow!). This option tries to be more granular, but may cause unexpected results.

hp e3000

webwise secure web server

custom log formats

- Extra format function for use by the mod_log_custom module
- %{varname}x - inserts the value of the varname env variable into the message
- **`CustomLog logs/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"`**

hp e3000

webwise secure web server

accounting structure

- Same scheme as Apache 1.3.14 A.02.00:
  - APACHE account (PM)
  - PUB group (PM)
  - V.UU.FF-based A0300 group (PM)
  - MGR user (PM)
  - WWW user (non-PM)
  - /APACHE/CURRENT symbolic link points to /APACHE/A0300

hp e3000

webwise secure web server

directory & file structure

- Same scheme as Apache 1.3.14 A.02.00
- All files owned & managed by MGR.APACHE
- Sensitive files MUST be protected with owner-only security!

hp e3000

webwise secure web server

new files and directories compared to apache

- bin/openssl - general crypto utility
  - supported for key/cert management only
  - add /APACHE/CURRENT/bin to PATH
- bin/sign.sh - cert-signing shell script
  - supported for self-signed CA cert only

hp e3000

webwise secure web server

new files and directories compared to apache (cont.)

- conf/ssl.crl/ - CRL directory
- conf/ssl.crt/ - certificate directory
  - protect directory with chmod 700
  - server.crt - server certificate (chmod 400)
  - Sensitive data! Protect it!
- conf/ssl.csr/ - CSR directory
- conf/ssl.key/ - key directory
  - protect directory with chmod 700
  - server.key - server private key (chmod 400)
  - Sensitive data! Protect it!

hp e3000

webwise secure web server

new files and directories compared to apache (cont.)

- logs/ssl_engine_log - the SSL error_log
- logs/ssl_request_log - the SSL access_log
  - includes protocol and cipher used
- logs/ssl_mutex.nnn - semaphore file

hp e3000

webwise secure web server

version information

- **`HTTPD -v`** (same as Apache)

```
Server version: Apache/1.3.22 (HP MPE/iX
WebWise A.03.00)
Server built:   Jan 15 2002 15:47:50
```

- **`bin/openssl version`**

```
OpenSSL 0.9.6b 9 Jul 2001
```

hp e3000

webwise secure web server

server configuration

- Copy sample files to normal names
- /APACHE/PUB/JHTTPD.sample
- conf/access.conf.sample, httpd.conf.sample, magic.sample, mime.types.sample, srm.conf.sample
- conf/ssl.crt/server.crt.sample (test only!)
- conf/ssl.key/server.key.sample (test only!)

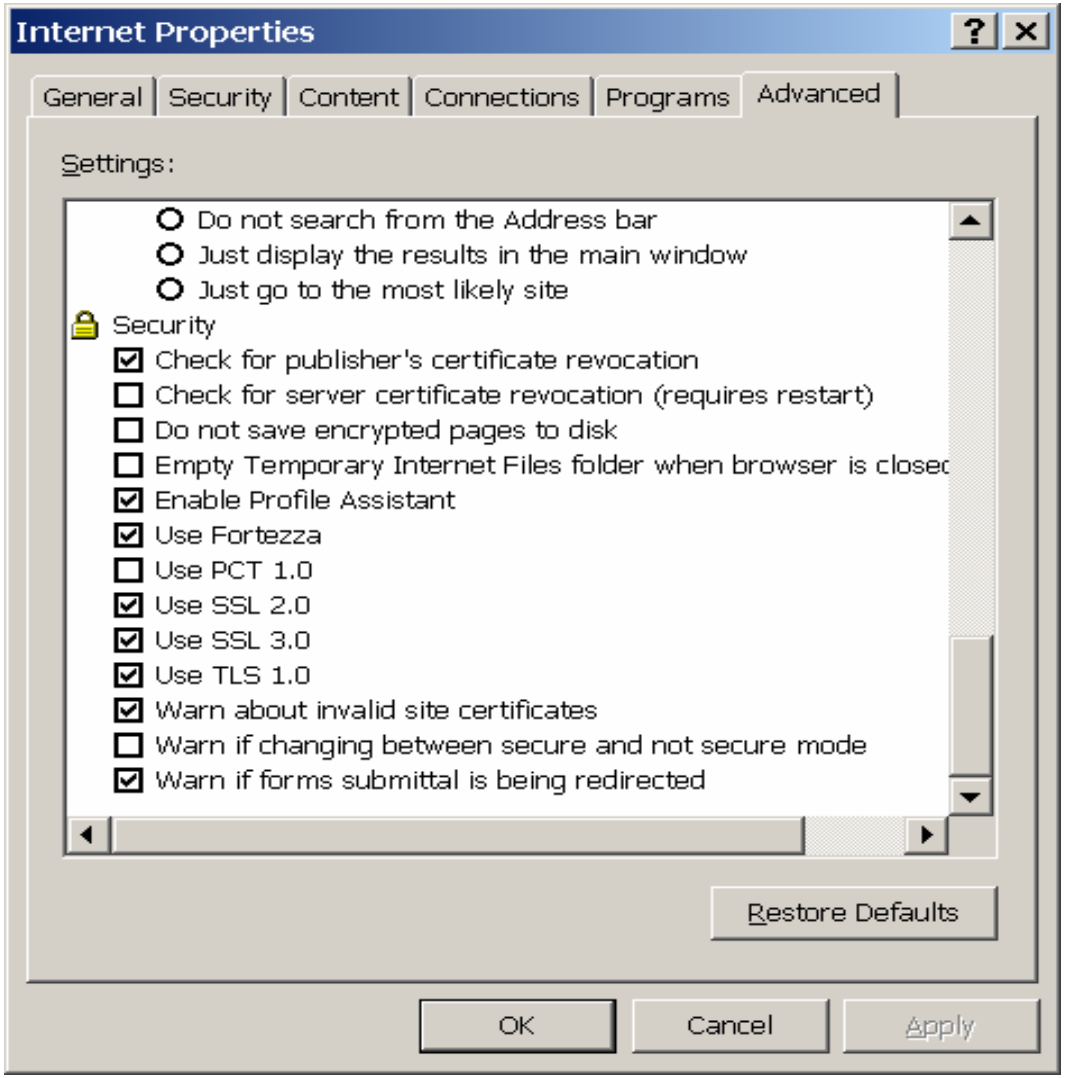
browser configuration

- MSIE allows you to enable/disable SSLv2.0, SSLv3.0, and TLSv1.0; no cipher choice
- Netscape allows you to enable/disable SSLv2.0, SSLv3.0, TLSv1.0, and to choose the ciphers for each one
- Both browsers allow you to manage personal and CA certificates

hp e3000

webwise secure web server

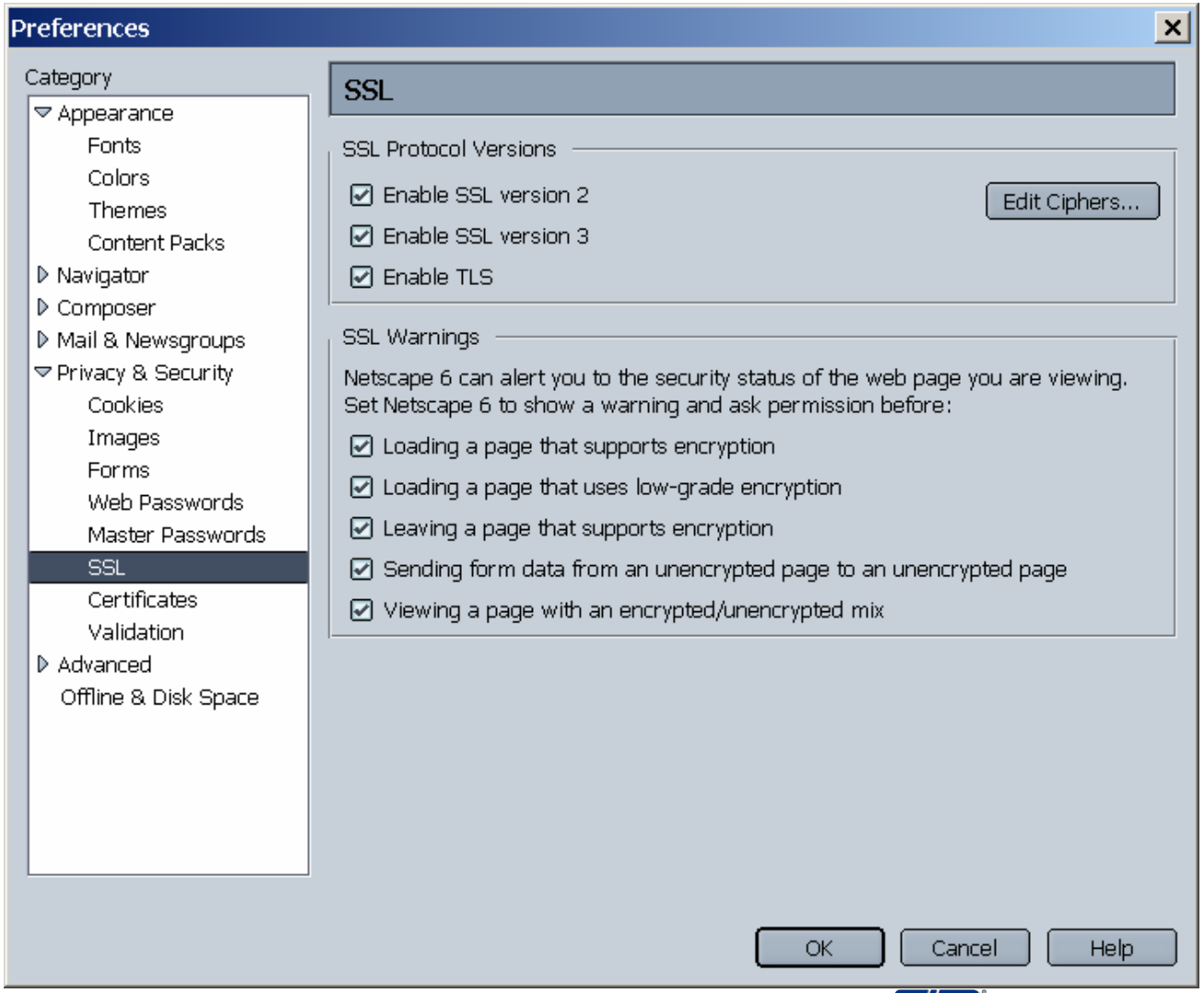
browser configuration - msie5.5



hp e3000

webwise secure web server

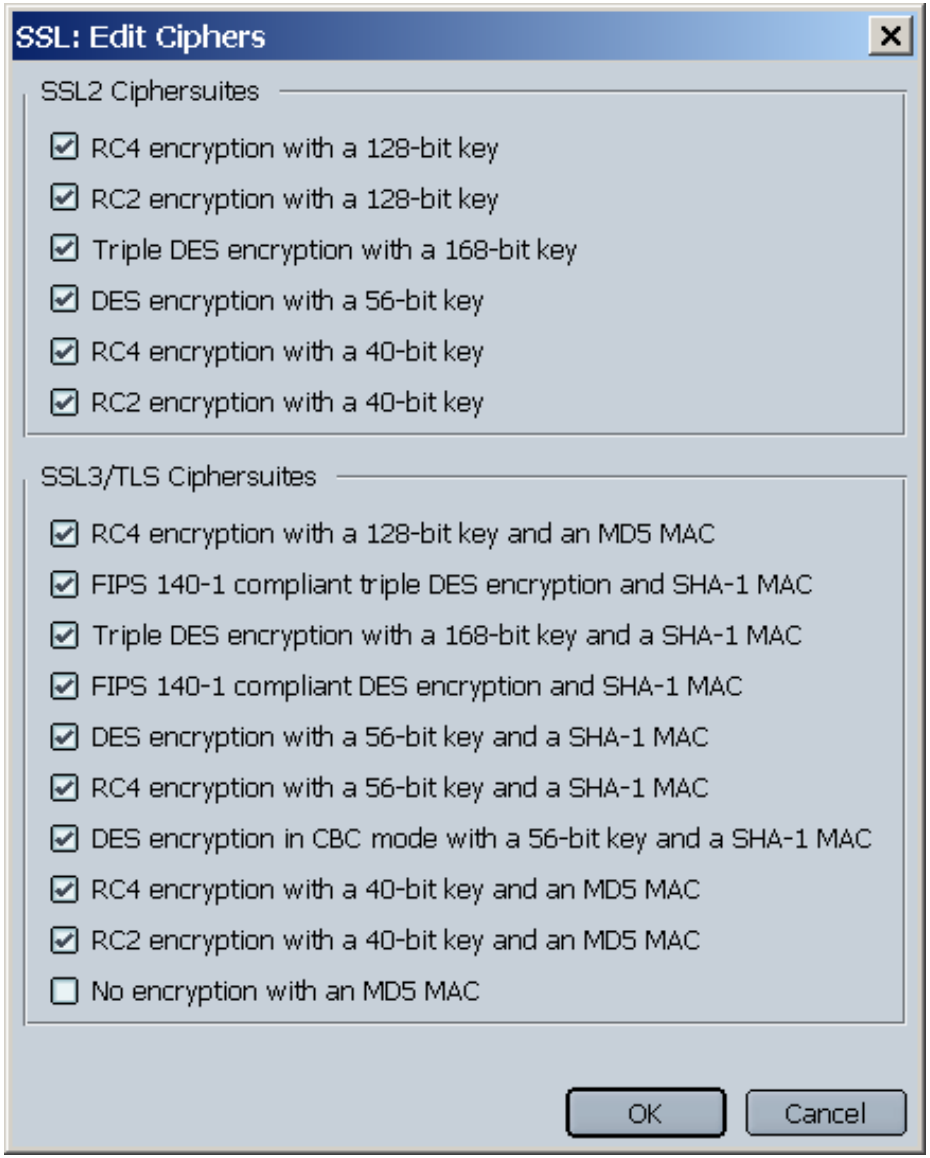
browser configuration – netscape 6.2.1



hp e3000

webwise secure web server

browser configuration – netscape 6.2.1 (cont.)



creating the server key

- conf/ssl.key/server.key.sample (test only)
- key generated as a random number - use openssl -rand parameter to specify random data file for better seed
- pass phrase strongly recommended!
  - Encrypts the key file with DES3 via openssl -des3 option
  - See SSLPassPhraseDialog directive
  - Protect the pass phrase!
- Protect the key file!

creating the server key (cont.)

- **$ cd conf/ssl.key**
- **$ openssl genrsa -rand /SYS/PUB/HPSWINFO **
  -des3 -out server.key 1024

```
unable to load 'random state'
 28199 semi-random bytes loaded
Generating RSA private key, 1024 bit long
modulus
 ..................+++++
 ..................+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

creating the server key (cont.)

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,1EF909EDE2B056B0

cctYqA7Rm5LS6G8vcqlhVRRzg78epZ+SRMs7jF8TuCHJB
ds0ScXxjOd2TRORqNVC/IAsmbc5nc2kB9GJswJ6HhcqcT
m0oI0NXBKiXWnhM2raHHlzBI161+4dBMTpgPjqYj4w4ei
VlveDqqm8W38D/YKm3w+tocUMSwbj8KFFnYDHuvq6TI8u
pRUD79ukSYhIDRs18Od2yuhepEAe9P3P/wAuZDPjRtmjt
4b1UgO5aSt+zflq6Zchikv5GsPQPWaBu3a6eykZwc47zx
a86X1eQLeuLoeV1QlEPAvi4Ade3tQ0n3h1bAfaHDSkgoU
S6toA3oAVrPkeUOP3Y8qF6UEuyP2LCK5vo6Ccp9XgHBDd
-----END RSA PRIVATE KEY-----
```

creating the server key (cont.)

- **`$ openssl rsa -noout -text -in server.key`**
  - displays details about the newly created key
- **`$ chmod 400 server.key`**
- Protect the key file!

server key pass phrase

- SSLPassPhraseDialog builtin
  - HTTPD reads pass phrase from stdin (I.e. JHTTPD)
  - Protect JHTTPD from unauthorized readers!
- SSLPassPhraseDialog exec:/path/to/pgm
  - Program/script prints pass phrase to stdout
  - Protect the program from unauthorized readers or executors!
  - Have program perform security checking before writing to stdout

hp e3000

webwise secure web server

creating the server csr

- Identifies the company and the server
- Attributes chosen here are visible to browser users, so choose carefully

creating the server csr (cont.)

- **$ cd conf/ssl.csr**
- **$ openssl req -new -key ../ssl.key/server.key **
  -out server.csr
  Country Name (2 letter code) [AU]:US
  State or Prov Name (full name) []:My State
  Locality Name (eg, city) []:My City
  Organization Name (eg, company) []:My Company
  Organizational Unit Name []:My Org
  Common Name []:www.mycompany.com
  Email Address []:webmaster@www.mycompany.com
- Leave the “extra attributes” blank

creating the server csr (cont.)

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB4TCCAUoCAQAwgaAxCzAJBgNVBAYTAlVTMREwDwYDV
MA4GA1UEBxMHTXkgQ2l0eTETMBEGA1UEChMKTXkgQ29tc
TXkgT3JnMRowGAYDVQQDExF3d3cubXljb21wYW55LmNvb
ARYbd2VibWFzdGVyQHd3dy5teWNvbXBhbnkuY29tMIGfM
A4GNADCBiQKBgQDS1iRItFKSDzOhDShFeoiWkfnc0yPGp
H/Umn2uM/tSNOiguAPBYce8prLYjNqyXY4QBCzWQNGv/N
+TyPMF/dYdH+1oOaaTZ0ZE0WP0l6CimzzXjvwCupOpcQ8
oAAwDQYJKoZIhvcNAQEEBQADgYEAj1vTRa5SamY2IwkLu
grIsPyS74PBHGQKdPp8y0L6aVD28wO1jZ82j62ihLXoPl
+6erc4gXI5CzSVh/1QJV8YWB+OpI2UC8Kd747eMEnLmxw
-----END CERTIFICATE REQUEST-----
```

creating the server csr (cont.)

- **$ openssl req -noout -text -in server.csr**

```
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=US, ST=My State, L=My
City, O=My Company, OU=My Org,
CN=www.mycompany.com/Email=webmaster@www.myco
mpany.com
        Subject Public Key Info:
            Public Key Algorithm:
rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
```

- **$ chmod 400 server.csr**

get signed by a trusted ca...

- Browsers configured with trusted CAs
  - I.e. www.verisign.com and many others
  - can add additional trusted CAs
- Paste your CSR into a CA web form
- Receive certificate by e-mail, save as conf/ssl.crt/server.crt
- Most CAs offer temporary testing certificates

...or become your own ca

- **`$ cd conf/ssl.key`**
- **`$ openssl genrsa -des3 -out ca.key 1024`**
- **`$ chmod 400 ca.key`**
- Protect the key file!

...or become your own ca (cont.)

- **$ openssl req -new -x509 -days 365 **
  -key ca.key -out ca.crt
  Country Name (2 letter code) [AU]:US
  State or Province Name [Some-State]:My State
  Locality Name (eg, city) []:My City
  Organization Name (eg, company) []:My Company
  Organizational Unit Name []:My Company CA
  Common Name []:Certificate Authority
  Email Address []:ca@mycompany.com

hp e3000

webwise secure web server

...or become your own ca (cont.)

```
-----BEGIN CERTIFICATE-----
eS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMTu
+s3Y2eodSY5GTQIc6vmzeWNS8iMq3OMrXEOXU01i7UPZnU/L
czBYVfMZk+IBXMQbYxgbkWXd5wgo8aLxgIEa3BcIs794KWeN
8kHWgoJcB8z28EL9JsS7irYFAgMBAAGjggEAMIH9MB0GA1Ud
Mz9xW15QUriuZRe0QTCBzQYDVR0jBIHFMIHCgBSZAey+GvYO
QaGBpqSBozCBoDELMAkGA1UEBhMCVVMxETAPBgNVBAgTCE15
VQQHEwdNeSBDaXR5MRMwEQYDVQQKEwpNeSBDb21wYW55MRYw
b21wYW55IENBMR4wHAYDVQQDExVDZXJ0aWZpY2F0ZSBBdXRo
hkiG9w0BCQEWEGNhQG15Y29tcGFueS5jb22CAQAwDAYDVR0T
hkiG9w0BAQQFAAOBgQB2brOu05pOu1JjnyQltijVkQjxqy15
SjvtyOL++IxL7IbrLSYp5ASpGSsjjyRBaNWIYFxIOhnM3Cho
4K8ZH8eVP/TY6W+KsgQJeXMLObagv9HcoZFpQX40A6KJAcFT
-----END CERTIFICATE-----
```

...or become your own ca (cont.)

- **$ openssl x509 -noout -text -in ca.crt**

```
Certificate:
  Data:
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=US, ST=My State, L=My City, O=My
Company, OU=My Company CA, CN=Certificate
Authority/Email=ca@mycompany.com
    Validity
      Not Before: Apr  7 23:19:40 2000 GMT
      Not After : Apr  7 23:19:40 2001 GMT
    Subject: C=US, ST=My State, L=My City, O=My
Company, OU=My Company CA, CN=Certificate
Authority/Email=ca@mycompany.com
```

- **$ chmod 400 ca.crt**

...or become your own ca (cont.)

```
$ sign.sh ../ssl.csr/server.csr
CA signing: ../ssl.csr/server.csr ->
  ../ssl.csr/server.crt:
The Subjects Distinguished Name is as follows
countryName             :PRINTABLE:'US'
stateOrProvinceName     :PRINTABLE:'My State'
localityName            :PRINTABLE:'My City'
organizationName        :PRINTABLE:'My Company'
organizationalUnitName:PRINTABLE:'My Org'
commonName              :PRINTABLE:'www.mycompany.com'
emailAddress
  :IA5STRING:'webmaster@www.mycompany.com'
Certificate is to be certified until Apr  7 23:54:01
  2001 GMT (365 days)
```

...or become your own ca (cont.)

```
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit?
  [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: ../ssl.csr/server.crt <-> CA cert
../ssl.csr/server.crt: OK
```

...or become your own ca (cont.)

- **`$ rm -fR ca.db.*`**
  - remove temporary files from conf/ssl.key
- **`$ cd ..`**
- **`$ mv ssl.csr/server.crt ssl.crt/server.crt`**
  - move newly created server certificate into the correct location
- **`$ mv ssl.key/ca.crt ssl.crt/ca.crt`**
  - move newly created CA certificate into the correct location

installing the server certificate

- **$ openssl x509 -noout -text -in ssl.crt/server.crt**

```
Certificate:
  Data:
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=US, ST=My State, L=My City, O=My Company,
  OU=My Company CA, CN=Certificate
  Authority/Email=ca@mycompany.com
    Validity
      Not Before: Apr 7 23:54:01 2000 GMT
      Not After : Apr 7 23:54:01 2001 GMT
    Subject: C=US, ST=My State, L=My City, O=My Company,
  OU=My Org, CN=www.mycompany.com/
  Email=webmaster@www.mycompany.com
```

installing the server certificate (cont.)

- Rebuild the symlink hash
- **$ cd conf/ssl.crt**
- **$ make**

```
ca-bundle.crt    ... Skipped
ca.crt           ... dc91dd8e.0
server.crt       ... 2f66b362.0
snakeoil-ca-dsa.crt ... 0cf14d7d.0
snakeoil-ca-rsa.crt ... e52d41d0.0
snakeoil-dsa.crt ... 5d8360e1.0
snakeoil-rsa.crt ... 82ab5372.0
zzyzx-ca-rsa.crt ... f28a2a0f.0
```

- **$ chmod 400 server.crt**

starting the web server

- **:STREAM JHTTPD.PUB.APACHE**
- Will spend as much as the first few minutes in a tight CPU loop generating temporary crypto keys before ready to accept requests
- No records written to log files during this time

hp e3000

webwise secure web server

using the web server

- conf/httpd.conf.sample uses ports 80 and 443
- Default browser ports are 80 and 443
  - **`http://your3000.host.name`** (port 80)
  - **`https://your3000.host.name`** (port 443)
- Non-default port numbers can also be used:
  - **`http://your3000.host.name:nnn`** (http port nnn)
  - **`https://your3000.host.name:nnn`** (https port nnn)

restarting the web server

- Why? To reread config files.
- Log on as SM user or MGR.APACHE
- Normal restart
  - **`$ kill -HUP $(cat /APACHE/PUB/logs/httpd.pid)`**
- Graceful restart
  - **`$ kill -USR1 $(cat /APACHE/PUB/logs/httpd.pid)`**

stopping the web server

- Log on as SM user or MGR.APACHE
- **`$ kill $(cat /APACHE/PUB/logs/httpd.pid)`**
- Only use :ABORTJOB as a last resort!
  - Will leak SVIPC semaphores
  - Use IPCS.HPBIN.SYS to display
  - Use IPCRM.HPBIN.SYS to manually remove

hp e3000

webwise secure web server

performance

- First few minutes in tight CPU loop
- Brief CPU burst for new SSL sessions
- Use bytestream instead of MPE record format for content
  - Content-length: header problem
  - Symptom: browser hangs at end of content
- Make sure RESLVCNF.NET.SYS is valid
  - Non-responding DNS servers can add a minute to every browser request
  - Symptom: browser hangs for about a minute before any content is returned
  - /SENDMAIL/CURRENT/bin/dnscheck script

hp e3000

webwise secure web server

security tips

- WebWise only protects the TCP/IP connection between browser and server!
- Protect the key and certificate files!
- Protect the key pass phrase!

security tips (cont.)

- Most security problems BY FAR are the result of sloppy CGI programming
  - Explicitly validate every byte of data sent by browser
  - A CGI hole can give the whole world the same access as a :HELLO WWW.APACHE session
- Restrict CGI/SSI authorship to trusted users
  - Don't allow CGI/SSI to be used outside of the APACHE account

security tips (cont.)

- Minimize the use of world-readable and world-writable permissions
- Make sure all accounts and/or users have passwords
- Change all default vendor passwords
- Disable all services not explicitly being used
- Use a firewall
- Stay current on releases & patches

hp e3000

webwise secure web server

troubleshooting server problems

- All Apache troubleshooting methods apply
- Check the log files first!
- If JHTTPD terminates at startup, investigate Pass Phrase
- Is SSLEngine On?
- Does SSLProtocol match the browser?
- Does SSLCipherSuite match the browser?

troubleshooting server problems (cont.)

- **`echo “HEAD / HTTP/1.0\n” | \`**
  `bin/openssl s_client -connect host:port | \`
  `/bin/cat -`
  - Pipes used due to select() being unimplemented for terminals
  - -state - displays SSL protocol states
  - -debug - displays packet data
  - -ssl2 | -ssl3 | -tls1 - protocol selection
  - -ciper - cipher selection
  - bin/openssl s_client help

troubleshooting server problems (cont.)

- Browser displays pages after long delay while HTTPD chews on the CPU
  - Potential SSL Session Cache malfunction
  - Server doing full SSL handshake with each request
  - Use SSLLogLevel directive to increase verbosity
  - If using a disk-based cache, are the permissions correct?
- Browser displays pages after long delay while HTTPD seems idle
  - HostNameLookups On causing inverse DNS lookups to hang due to misconfigured RESLVCNF or DNS
  - /SENDMAIL/CURRENT/bin/dnscheck script

troubleshooting server problems (cont.)

- Are the configuration file permissions correct?
  - Parent process running as the JHTTPD !JOB user (MGR.APACHE) must be able to read everything
  - Child processes running as the conf/httpd.conf User user (WWW.APACHE) must be able to read CA & CRL files if doing X.509 client authentication
  - Child permissions problems manifest as weird browser errors
- Does the problem also occur in HPUX Apache?

troubleshooting server problems (cont.)

- Check the mod_ssl bug database
  - **`http://www.modssl.org/support/bugdb/`**
- No OpenSSL bug database :-(
  - Search the mailing list archives at **`http://www.openssl.org/support/`**
- Check the Apache bug database
  - **`http://bugs.apache.org/`**

troubleshooting browser problems

- No response to browser
  - Check httpd.conf or SOCKINFO.NET.SYS to verify the correct ports (80, 443) are being listened to
- “The page cannot be displayed” (MSIE)
  - Speaking https to the http server port
  - Speaking the wrong security protocol (I.e. SSLv2 when the server requires SSLv3)
- “A network error occurred while Netscape was receiving data”
  - Speaking https to the http server port
  - Speaking the wrong security protocol (I.e. SSLv2 when the server requires SSLv3)

troubleshooting browser problems (cont.)

- A server certificate dialog box always appears
  - Server certificate not signed by a trusted CA
  - Server certificate expired
  - Server certificate hostname doesn't match URL hostname
- Verifying protocol & cipher
  - Look in logs/ssl_request_log
  - MSIE: right-click, Properties
  - Netscape: right-click, View Page Info

further documentation

- Complete product documentation
  - **`http://your.host.name/manual/`**
- Mod_ssl documentation
  - **`http://www.modssl.org/docs/2.8/`**
- OpenSSL documentation
  - **`http://www.openssl.org/docs/apps/openssl.html`**
- Apache documentation
  - **`http://www.apache.org/docs/`**
- 7.5 Communicator
- 7.5 Configuring and Managing MPE/iX Internet Services Manual

hp e3000

webwise secure web server

join the hp3000-l community

- Available as a mailing list and as the Usenet newsgroup comp.sys.hp.mpe
- In-depth discussions of all things HP e3000
- Talk with other WebWise & Apache users
  - seek advice, exchange tips & techniques
- Keep up with the latest HP e3000 news
- Interact with CSY
- **`http://jazz.external.hp.com/papers/hp3000-info.html`**