Setting up a secure server with Apache and mod_ssl

Daniel Lopez Ridruejo

About Me

ASF member and long time Apache user

Interested in usability and lowering the learning curve for Apache

Comanche GUI configuration tool

“Teach Yourself Apache” book

What this presentation is about

- Introduction to SSL
- mod_ssl module for Apache
- Building SSL-enabled Apache
- Creating key and certificate with openssl
- Signing the certificate
- Accessing your secure server

Cryptography

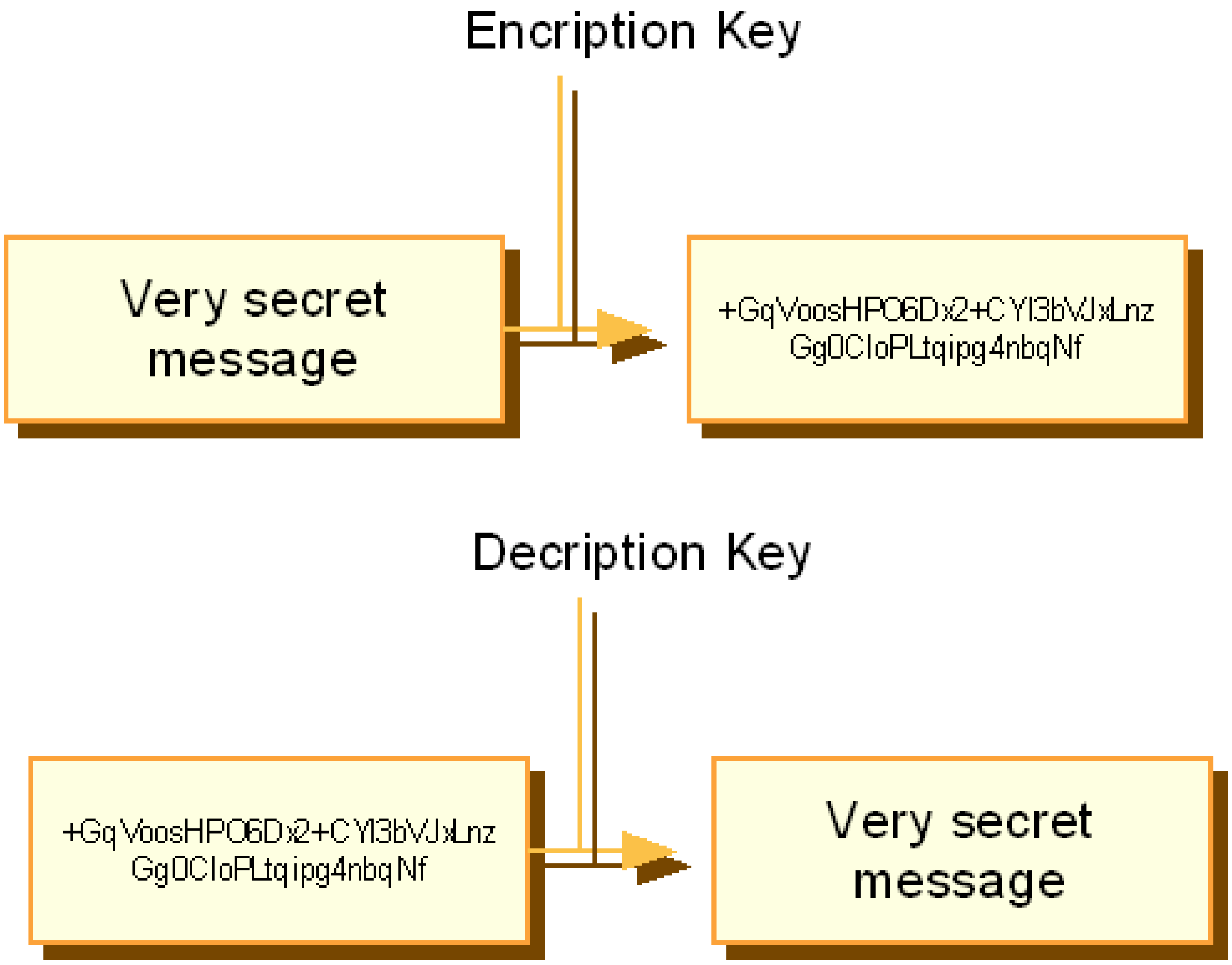
What does security mean?

- Confidentiality
- Integrity
- Authentication

The SSL/TLS protocols

- Secure Socket Layers / Transport Layer Security
- Used to secure HTTP and other protocols
- URLs start with http:// instead of https://
- Encryption: Confidentiality
- Certificates: Authentication
- Digest Algorithms: Integrity

Encryption: Confidentiality



Encription Key

Very secret message

+GqVoosHPO6Dx2+CYl3bVJxLnz
Gg0CloPLtqipg4nbqNf

Decription Key

+GqVoosHPO6Dx2+CYl3bVJxLnz
Gg0CloPLtqipg4nbqNf

Very secret message

Symmetric Key

- Encryption/Decryption keys are the same
- Fast
- Key exchange problems
- Algorithms: DES, Triple-DES, RC4, RC2

Asymmetric Keys

- Encryption and Decryption keys are different
- Generally Slower
- Distribution is easier, mark one of the keys as ‘public’ and distribute widely. Keep the other private
- Messages encrypted with one key can only be decrypted with the other
- Algorithms: RSA

Digests : Integrity

- Fixed-length “fingerprint” of a message
- If the message changes, the fingerprint changes
- Algorithms: MD5, SHA
- Attacker can replace both message and digest. Include additional secret: Message Authentication Codes (MACs)

Certificates : Authentication

- Ties your public key to your identity
- Write your public key and data about you in a message, then ask a trusted third-party to encrypt it with their private key.
- Everybody with the third-party public key can verify the data is true (if you trust that third-party, that is ;)

More on certificates

- Trusted third-party: Certification Authority (Verisign, Thawte…)
- Their public keys are shipped with most browsers
- Certificates can have expiration dates
- Certificates are used by browsers to verify the identity of the server at the other end

How SSL works

- Client (browser) tries to connect to server
- Handshake: client and server exchange keys and certificates
- Verify certificate validity
- Use each other’s public key to securely agree on a symmetric key
- Transmit data using symmetric key

mod_ssl for Apache

- Several SSL alternatives for Apache
- Commercial: Covalent, Red Hat, IBM
- Open Source: mod_ssl, Apache SSL
- mod_ssl is the most popular, originally based on Apache SSL
- Apache 1.3 version distributed separately because of export restrictions. Requires patching the server
- Bundled with Apache 2.0

Building mod_ssl

- Apache 1.3 this requires applying EAPI patches and different build options
- This talk covers Apache 2.0
- Although compilation is different, configuration is similar
- You can also get packaged and binary versions for Linux, BSD, others

OpenSSL

- Underlying cryptography library

```
# gunzip < openssl*.tar.gz | tar xvf -
# cd openssl*
# ./config --prefix=/usr/local/ssl/install
--openssldir=/usr/local/ssl/install/openssl
# make
# make install
```

mod_ssl

- Bundled in Apache 2.0
- Needs to pass the following command line options when building the server

```
--enable-ssl
--with-ssl=/usr/local/ssl/install/openssl
```

- Add the following to config file (if built dynamically)

```
LoadModule ssl_module modules/libmodssl.so
```

Certificates

- Now we need to create a certificate for our server. First step is to create a pair of public/private keys:

```
openssl genrsa -des3 -rand file1:file2:file3 -out
  www.example.com.key 1024
625152 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 a CSR

- We are going to provide data about us and our server and create a Certificate Signing Request
- The CA can then sign the certificate and return it to us

```
# ./usr/local/ssl/install/bin/openssl req -new -key
  www.example.com.key -out www.example.com.csr
```

Certificate Information

```
Using configuration from
  /usr/local/ssl/install/openssl/openssl.cnf
Enter PEM pass phrase:
[…]
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []: San Francisco
Organization Name (eg, company) [Some Company]:.
Organizational Unit Name (eg, section) []:.
Common Name (eg, YOUR name) []:www.example.com
Email Address []:administrator@example.com
Please enter the following 'extra' attributes to be
  sent with your certificate request
A challenge password []:
An optional company name []:
```

Creating a Self-Signed Certificate

- Commercial certificates cost money and take time
- If the server is private or you want to have a temporary solution you can sign your own certificate:

```
# ./usr/local/ssl/install/bin/openssl x509 -req
-days 30 -in www.example.com.csr -signkey
  www.example.com.key -out www.example.com.cert
```

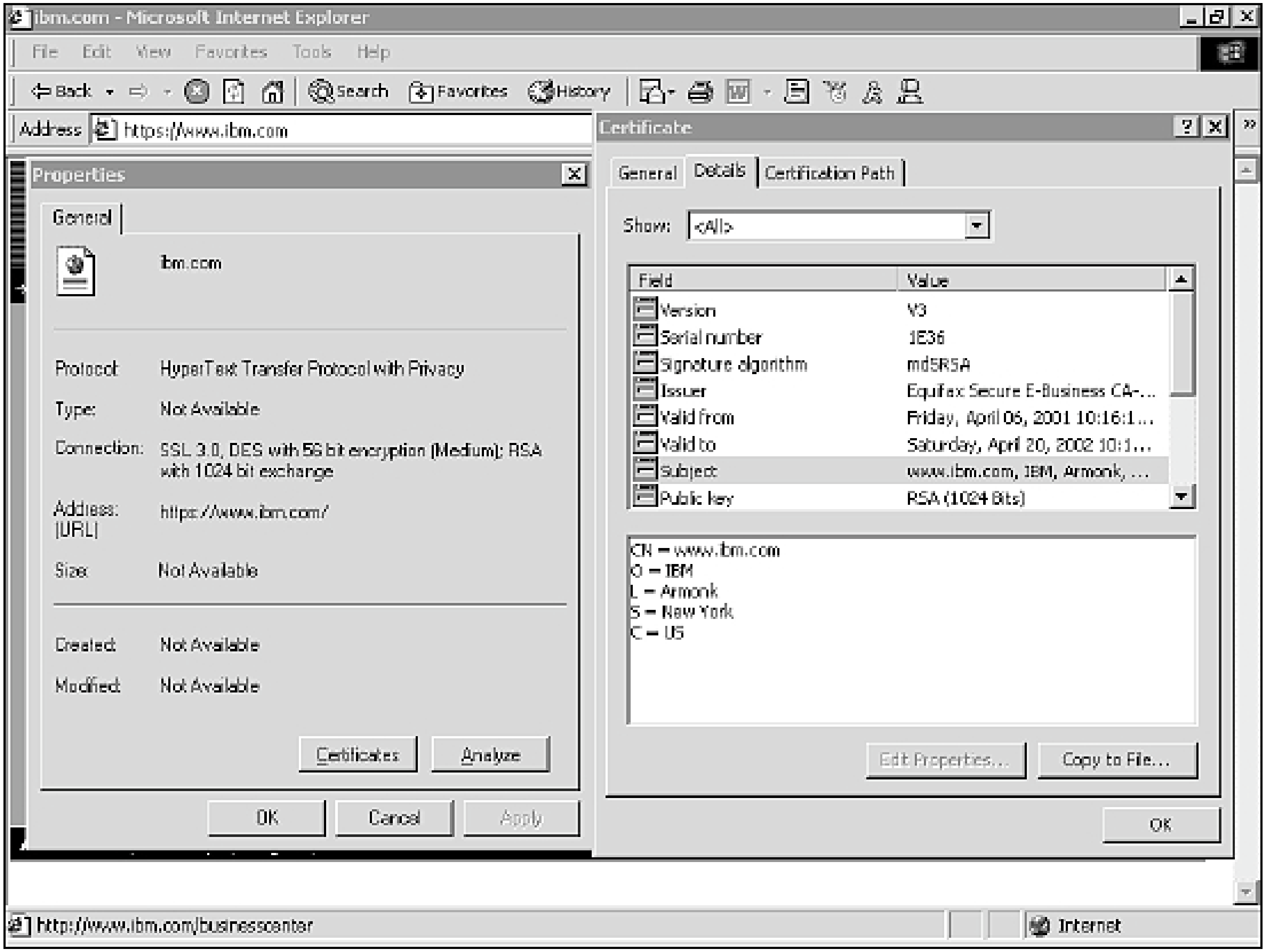
Configuring Apache

```
Listen 80
Listen 443
<VirtualHost _default_:443>
ServerName www.example.com
SSLEngine on
SSLCertificateFile \
  /path/openssl/certs/www.example.com.cert
SSLCertificateKeyFile
  /path/openssl/certs/www.example.com.key
</VirtualHost>
```

Test the installation

- Start Apache
- Prompt for password
- The URL is https://www.example.com
- Or https://www.example.com:8443 if installing as regular user

A look at the server certificate



Address: https://www.ibm.com

ibm.com

Protocol: HyperText Transfer Protocol with Privacy

Type: Not Available

Connection: SSL 3.0, DES with 56 bit encryption (Medium); RSA with 1024 bit exchange

Address: (URL) https://www.ibm.com/

Size: Not Available

Created: Not Available

Modified: Not Available

Show: <All>

Field	Value
Version	V3
Serial number	1E36
Signature algorithm	md5RSA
Issuer	Equifax Secure E-Business CA-...
Valid from	Friday, April 06, 2001 10:16:1...
Valid to	Saturday, April 20, 2002 10:1...
Subject	www.ibm.com, IBM, Armonk, ...
Public key	RSA (1024 Bits)

CN = www.ibm.com
O = IBM
L = Armonk
S = New York
C = US

http://www.ibm.com/businesscenter

Other mod_ssl capabilities

- Control which protocols clients can use
- Client Certificates
- Reverse proxy (offload SSL from App Servers)
- SSL session caching
- Access Control

Questions?

- http://www.modssl.org
- http://httpd.apache.org
- http://www.apacheworld.org
- You can reach me at daniel@rawbyte.com