Securing Web Access with a Private Certificate Authority

Presented by Paul Weinstein,
Kepler Solutions,
<pdw@keplersol.com>

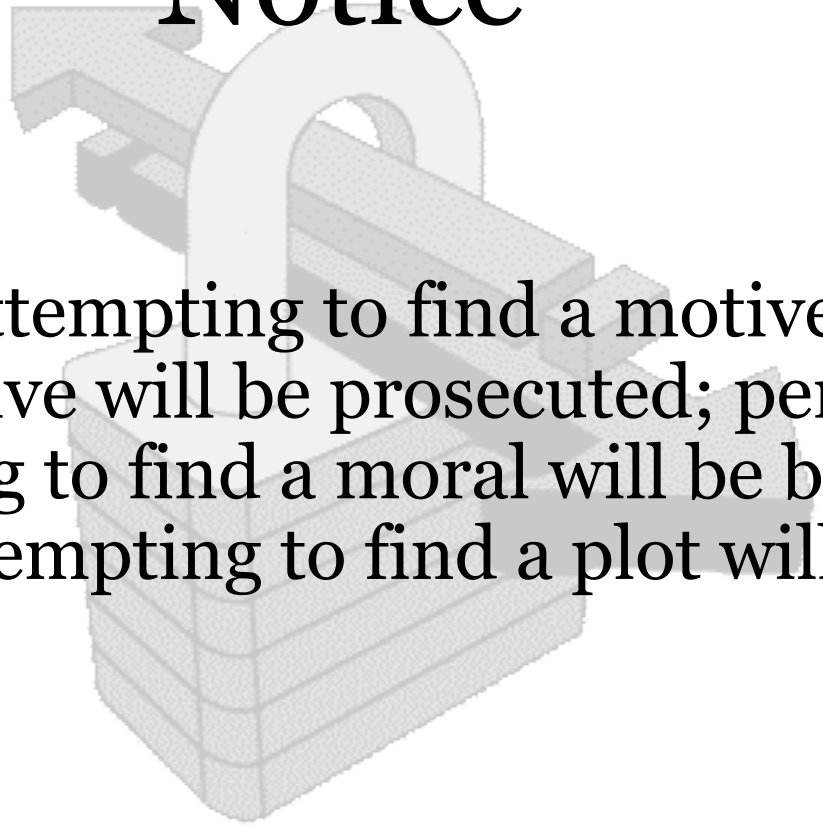
ApacheCon EU 2006

June 28, 2006

Hello World

- Introduction
- The Basics:
  - Review of SSL Protocol
  - Review of Digital Certificates
  - A Private Certificate Authority in Action
- The Nit and Gritty
  - Creating a Private Certificate Authority
  - Publishing the Private Certificate Authority
  - Using Our Private Certificate Authority

Notice



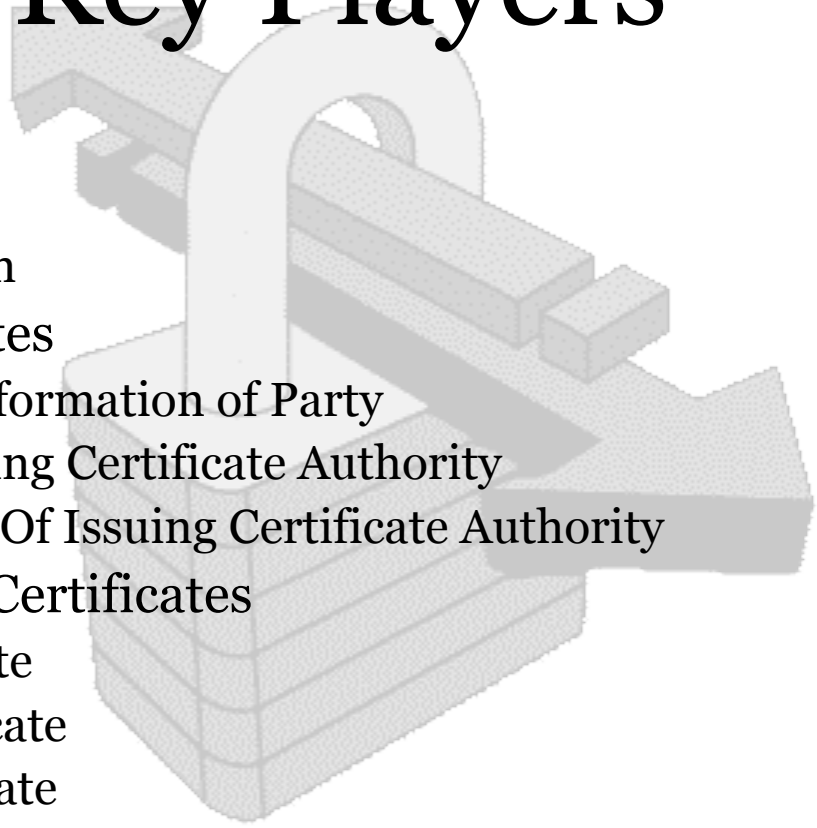
"Persons attempting to find a motive in this narrative will be prosecuted; persons attempting to find a moral will be banished; persons attempting to find a plot will be shot."

- Preface for The Adventures of Huck Finn By Mark Twain

The Basics

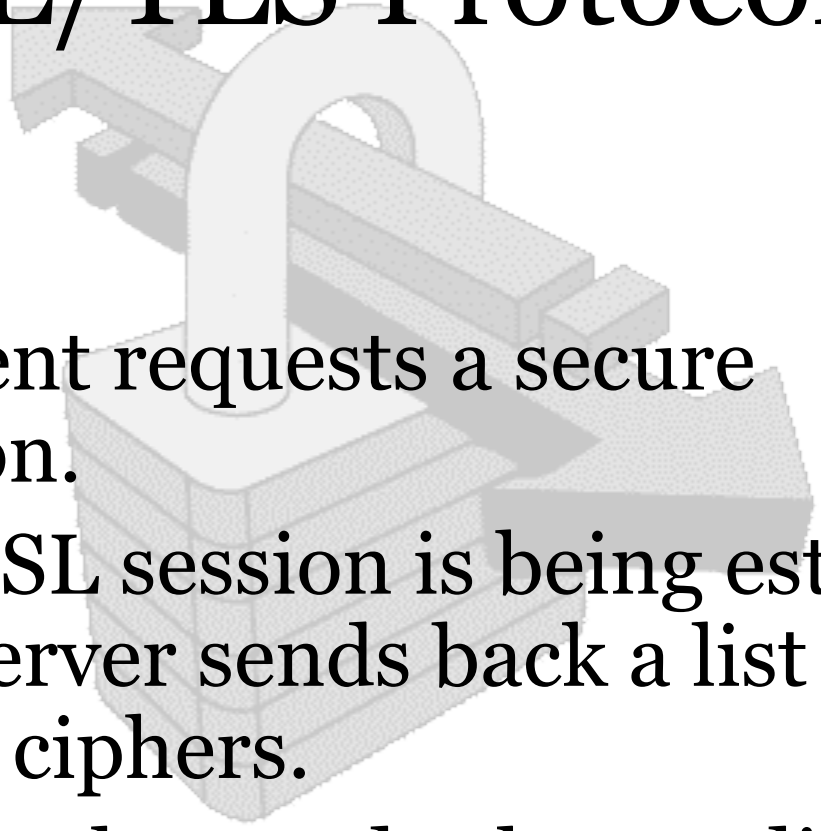
SSL, Digital Certificates and Certificate Authorities

Key Players



- SSL Protocol
  - Encryption
  - Authentication
- Digital Certificates
  - Identifying Information of Party
  - Name Of Issuing Certificate Authority
  - A “Signature” Of Issuing Certificate Authority
- Type Of Digital Certificates
  - Root Certificate
  - Server Certificate
  - Client Certificate

SSL/TLS Protocol



- A web client requests a secure transaction.
- If a new SSL session is being established the web server sends back a list of agreeable ciphers.
- The server also sends along a digital certificate.

SSL/TLS Protocol

- The client authenticates the server.
- The client generates a symmetric key using an agreeable cipher and key size and then encodes the symmetric key.
- If the server has requested a digital certificate to authenticate the client, the client sends it along with the encoded symmetric key.

SSL/TLS Protocol

- Both the client and the server use the symmetric key to generate another symmetric key, know as the session key.
- The client sends a message to the server stating that all future messages from the client will be encrypted with the session key.
- The server sends a message to the client stating that all future messages from the server will be encrypted with the session key.

Digital Certificates

- Digital Certificates
  - A Serial Number
  - Identifying Information
    - Individual and/or Group Name
    - Location/Contact Information
  - Subject's Public Key
  - Name Of Issuing Certificate Authority
  - A "Signature" Of Issuing Certificate Authority
- Type Of Digital Certificates
  - Root Certificate
  - Server Certificate
  - Client Certificate

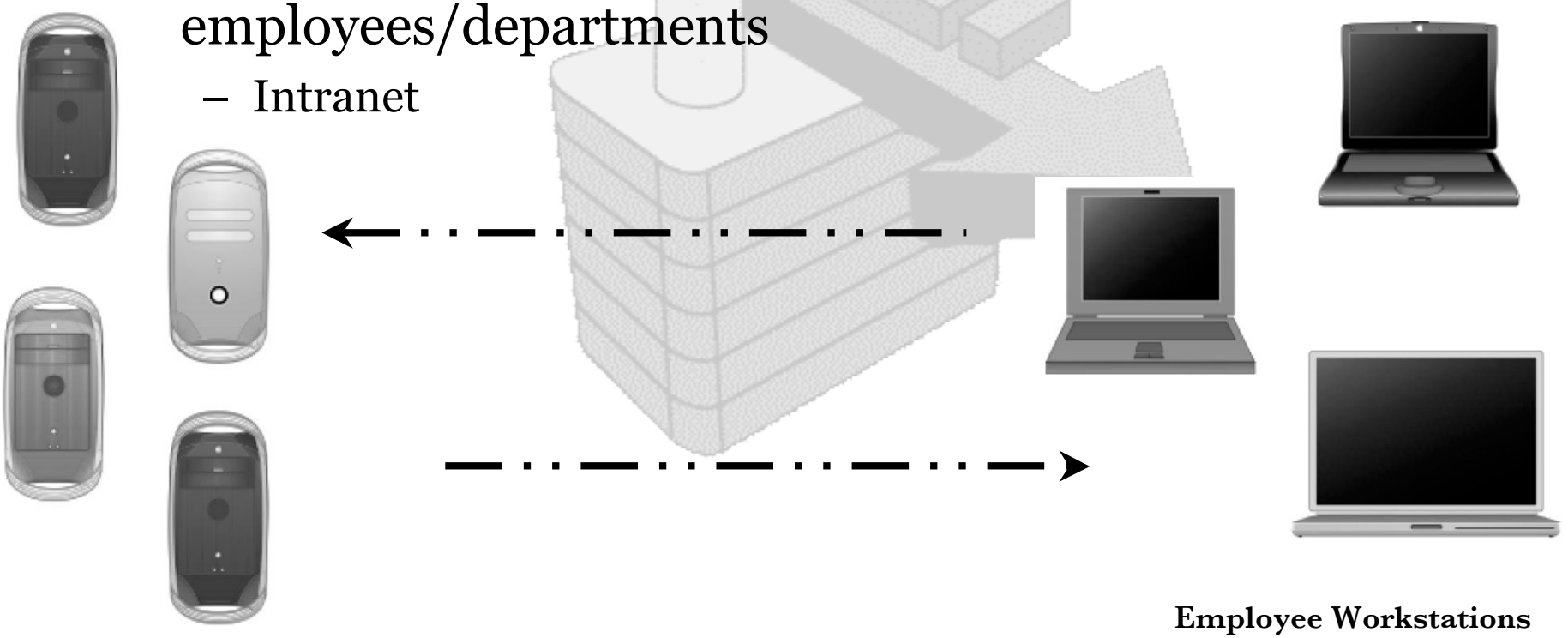
Certificate Authorities

- Public Certificate Authority; Verisign, Thawte, GeoTrust; recognized by default by most web browsers and web servers; used when no other relation exists between two parties.

- Private Certificate Authority; by default not recognized; used when a relationship already exists between two parties.

A PCA in Action

- Secure valuable data in transit between employees/departments
  - Intranet

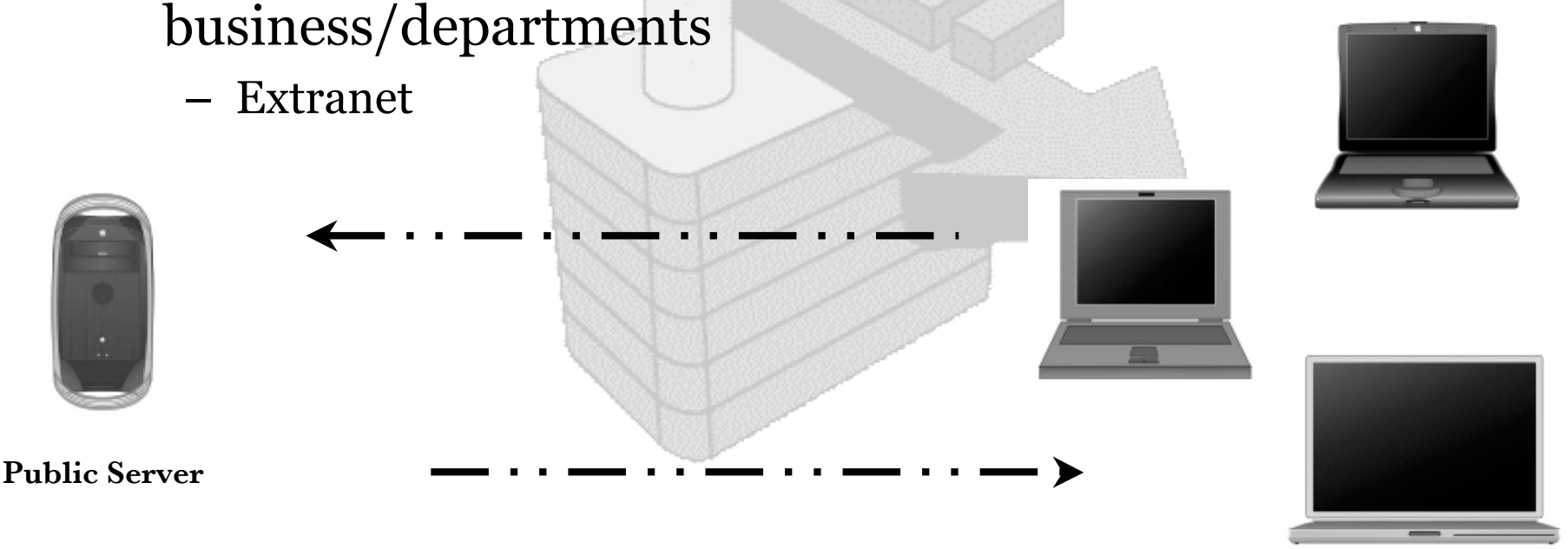


Department Servers

Employee Workstations

A PCA in Action

- Secure valuable data in transit between business/departments
  - Extranet



Public Server

Business Partner's Workstations

The Nit and Gritty

Creating, Publishing and Using a Private Certificate Authority

Creating a Private Certificate Authority

- A self-signed Root Certificate:

```
[pca]$ openssl req -new -x509 -keyout private/ca.key -out certs/ca.cert -config
openssl.cnf
Using configuration from openssl.cnf
Generating a 1024 bit RSA private key
.....++++++
...++++++
writing new private key to 'private/ca.key'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Richmond
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Waubonsie Consulting
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:Paul Weinstein
Email Address []:pdw@waubonsie.com
```

Creating a Private Certificate Authority

- Configuring OpenSSL:

```
[ CA_default ]

dir             = /usr/local/pca        # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
new_certs_dir   = $dir/certs            # default place for new certs.

certificate     = $dir/certs/ca.cert    # The CA certificate
serial          = $dir/serial           # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/ca.key   # The private key
RANDFILE        = $dir/private/.rand    # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert
```

Creating a Private Certificate Authority

- Configuring OpenSSL:

```
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

Publishing the Private Certificate Authority

- Setting MIME-type in Apache:

```
#
#   Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
```

Using Our Private Certificate Authority: Server Certificate

- Creating a Certificate Signing Request:

```
[pca]$ openssl req -new -keyout private/server.key -out certs/server.csr -days 365 -config openssl.cnf
Using configuration from openssl.cnf
Generating a 1024 bit RSA private key
..++++++
.....................++++++
writing new private key to 'private/server.key'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Richmond
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Waubonsie Consulting
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:www.waubonsie.com
Email Address []:webmaster@waubonsie.com
```

Using Our Private Certificate Authority: Server Certificate

- Signing the Certificate Signing Request:

```
[pca]$ openssl ca -out certs/server.cert -config openssl.cnf -infiles certs/server.csr
Using configuration from openssl.cnf
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'California'
localityName          :PRINTABLE:'Richmond'
organizationName      :PRINTABLE:'Waubonsie Consulting'
commonName            :PRINTABLE:'www.waubonsie.com'
emailAddress          :IA5STRING:'webmaster@waubonsie.com'
Certificate is to be certified until Oct  7 04:16:40 2003 GMT (365 days)
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
```

Using Our Private Certificate Authority: Server Certificate

- Configuring Apache:

```
#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again. A test
#   certificate can be generated with `make certificate' under
#   built time. Keep in mind that if you've both a RSA and a DSA
#   certificate you can configure both in parallel (to also allow
#   the use of DSA ciphers, etc.)
SSLCertificateFile /usr/local/apache/conf/ssl.crt/server.cert

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /usr/local/apache/conf/ssl.key/server.key
```

Using Our Private Certificate Authority: Client Certificate

- Creating a Certificate Signing Request:

Using Our Private Certificate Authority: Client Certificate

- Signing the Certificate Signing Request:

```
[pca]$ openssl ca –spkac certs/client.csr –out certs/client.cert –days 365 -config
openssl.cnf
Using configuration from openssl.cnf
Enter PEM pass phrase:
Check that the SPKAC request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
CommonName            :PRINTABLE:'Employee Certificate for Paul Weinstein'
emailAddress          :IA5STRING:'pdw@waubonsie.com'
organizationName      :PRINTABLE:'Waubonsie Consulting'
organizationalUnitName:PRINTABLE:''
stateOrProvinceName   :PRINTABLE:'California'
localityName          :PRINTABLE:'Richmond'
countryName           :PRINTABLE:'US'
Certificate is to be certified until Oct  7 04:16:40 2003 GMT (365 days)

Write out database with 1 new entries
Data Base Updated
```

Using Our Private Certificate Authority: Client Certificate

- Configuring Apache:

```
#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
SSLCACertificatePath /usr/local/apache/conf/ssl.crt/ca.cert

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
SSLVerifyClient require
SSLVerifyDepth  10
```

Using Our Private Certificate Authority: Certificate Revocation List

- Revoking an Existing Digital Certificate:

```
[pca]$ openssl ca -revoke certs/client.cert -config openssl.cnf
Using configuration from openssl.cnf
Enter PEM pass phrase:
Revoking Certificate 10.
Data Base Updated

[pca]$ openssl ca -gencrl -out ca.crl -config openssl.cnf
Using configuration from openssl.cnf
Enter PEM pass phrase:
```

Publishing the Private Certificate Authority

- Setting MIME-type in Apache:

```
#
#   Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
```

Using Our Private Certificate Authority: Certificate Revocation List

- Configuring Apache:

```
#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
SSLCARevocationFile /usr/local/apache/conf/ssl/ssl.crl/ca.crl
#SSLCARevocationPath /usr/local/apache/conf/ssl/ssl.crl
```

Review

- The Basics:
  - Review of Digital Certificates
  - A Private Certificate Authority in Action
- The Nit and Gritty
  - Creating a Private Certificate Authority
  - Publishing the Private Certificate Authority
  - Using Our Private Certificate Authority

Citation

Hirsch, Frederick *Introducing SSL and Certificates using SSLeay*. <http://www.pseudonym.org/ssl/wwwj-index.html>.

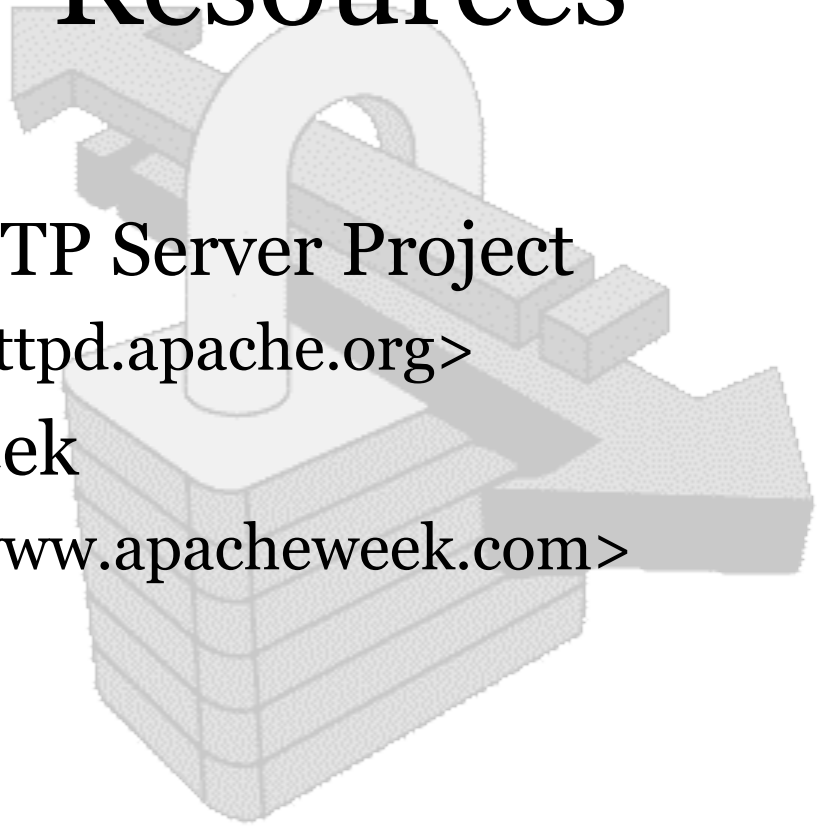
Mobily, Tony, et al. Professional Apache Security. Birmingham: Wrox Press, 2003.

Weinstein, Paul, et al. Professional Linux Security. Indianapolis: Wrox,, 2006.

Resources

- This Presentation:
  - <http://www.weinstein.org/work/presentations/apacheconeu06/pca/> (HTML)
  - <http://www.weinstein.org/work/presentations/apacheconeu06/pca.pdf> (PDF)

Resources



- Apache HTTP Server Project
  - <http://httpd.apache.org>
- Apache Week
  - <http://www.apacheweek.com>

Resources

- mod_ssl Project, <http://www.modssl.org>
  - Mailing Lists, List Archives:
    - <modssl-announce@modssl.org>
    - <modssl-users@modssl.org>
    - <http://marc.theaimsgroup.com/?l=apache-modssl>
- OpenSSL Project, <http://www.openssl.org>
  - Mailing Lists, List Archives:
    - <openssl-announce@openssl.org>
      - <http://marc.theaimsgroup.com/?l=apache-modssl>
    - <openssl-cvs@openssl.org>
      - <http://marc.theaimsgroup.com/?l=openssl-cvs>
    - <openssl-dev@openssl.org>
      - <http://marc.theaimsgroup.com/?l=openssl-dev>
    - <openssl-users@openssl.org>
      - <http://marc.theaimsgroup.com/?l=openssl-users>

Any Questions?