SSL State of the Union

Sander Temme – Senior Product Line Manager
Thales e-Security – sctemme@apache.org

THALES

Everybody's Going SSL

https://www.eff.org/deeplinks/2013/11/encrypt-web-report-whos-doing-what

Agenda

- The SSL Protocol
- Recent Security Developments
- How Open Source Has Stumbled
- Setting up a Test PKI
- Questions?

When I say "SSL" I also mean "TLS"

THALES

www.thales-esecurity.com

SSL Protocol Considerations

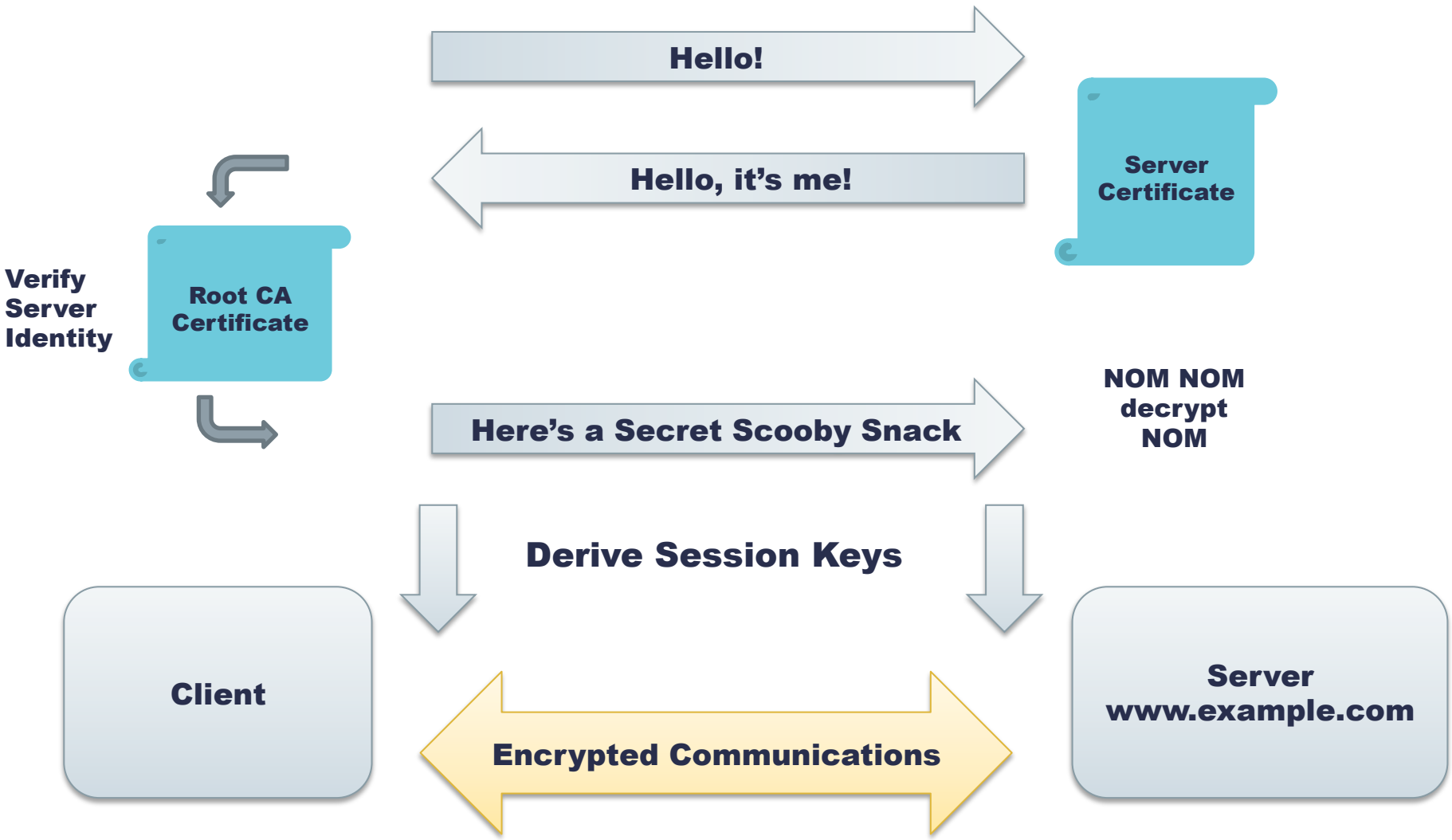
What Does SSL Do For Us?

Authenticates

Encrypts

THALES

Protocol Overview



THALES

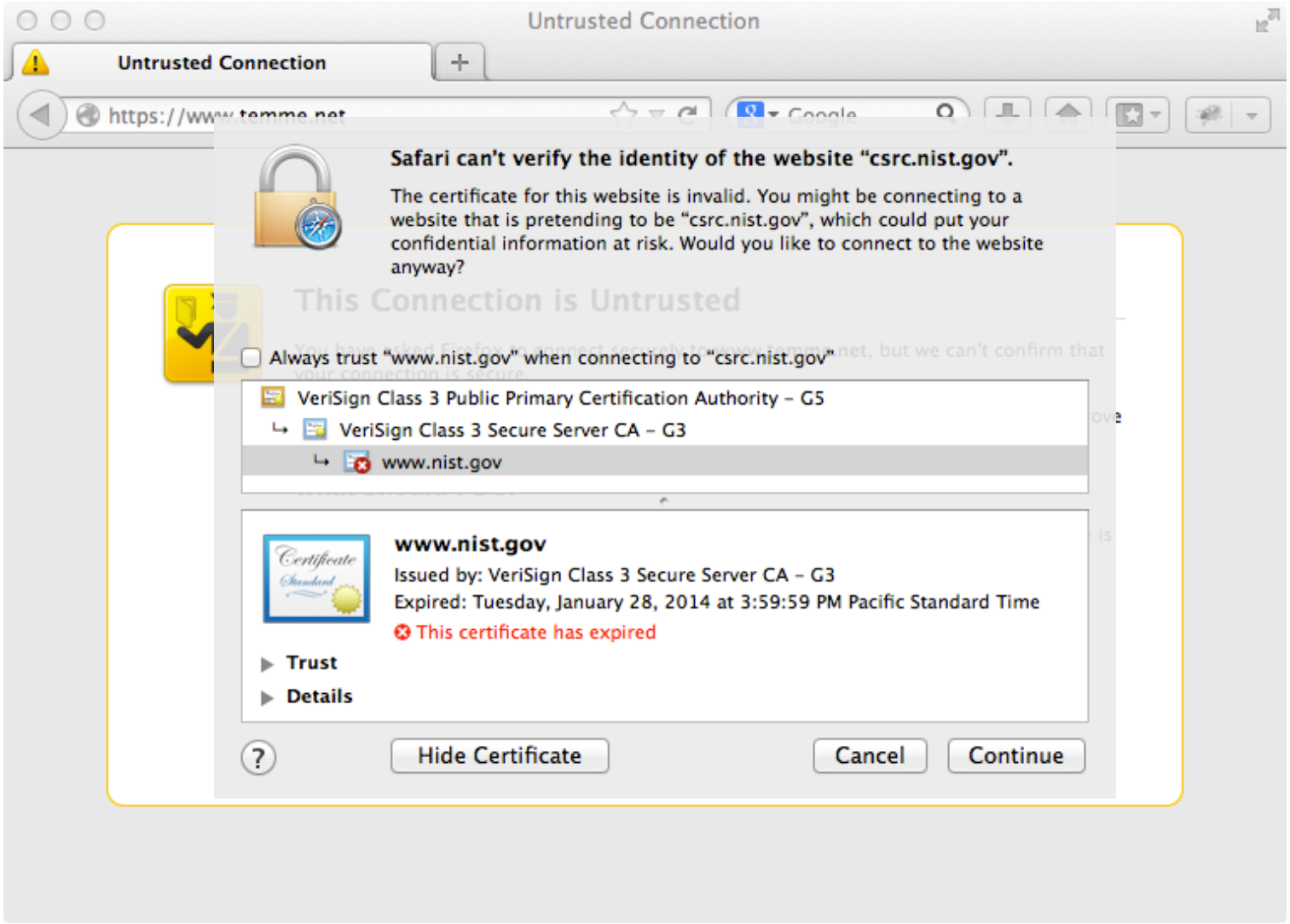
Limitations

Only in transit

Must do Certificates Well

Blind Trust in Certificate Authorities

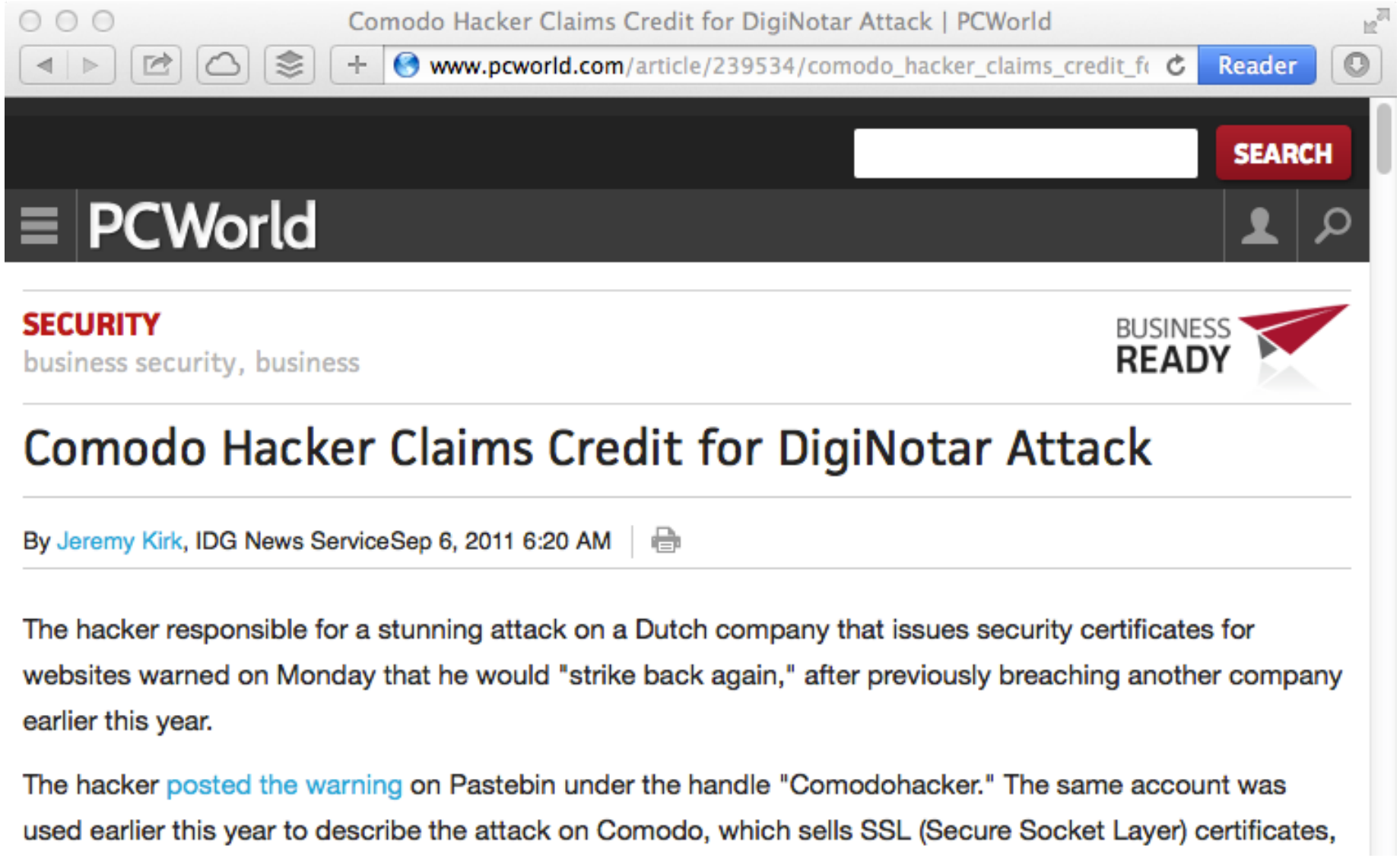
Limitations: Certificate Issues



THALES

Security Triad

Trust in Certificate Authorities?



THALES

www.thales-esecurity.com

Recent Security Developments

Protocol Attacks

BEAST, CRIME, TIME, Oh My!

We Are Not Cryptographers

- We consume crypto
- Sift through the headlines
- Most attacks Responsibly Disclosed

THALES

Attacks are Good

Daylight as a Disinfectant

Flaws get fixed

Everyone benefits

But painful

- Need a plan

THALES

This Week's Joy: Heartbleed

Bug in heartbeat in OpenSSL

Fixed in 1.0.1g; 1.0.0, 0.9.8 not affected

Exploits virtually undetectable

Unpleasant surprise

- Became public April 7, 2014
- No advance warning
- Vendors are scrambling
- We all get to update!

To wit

- New keys, get certificates reissued
- Old certificates revoked
- Assess and respond to possible intrusion

http://heartbleed.com

https://access.redhat.com/security/cve/CVE-2014-0160

Increase Trust in Certificates

DANE

Certificate Transparency

DANE: DNS-based Authentication of Named Entities

Associate Server Hostname with Certificate

TLS Association Records in DNS

Secured by DNSSEC

IETF®

http://csrc.nist.gov/groups/ST/ca-workshop-2013/presentations/Barnes_ca-workshop2013.pdf

THALES

Certificate Transparency

Central registry of valid certificates

Domain owners publish, check

Client computers “gossip” worldwide

https://www.certificate-transparency.org

http://csrc.nist.gov/groups/ST/ca-workshop-2013/presentations/Kasper_ca-workshop2013.pdf

“New” Protocol Developments

Session Tickets

OCSP Stapling

Perfect Forward Secrecy

Session Tickets

Server encrypts session state

Hands to client

Resume: client hands back encrypted session

httpd 2.4 can do this

OCSP Stapling

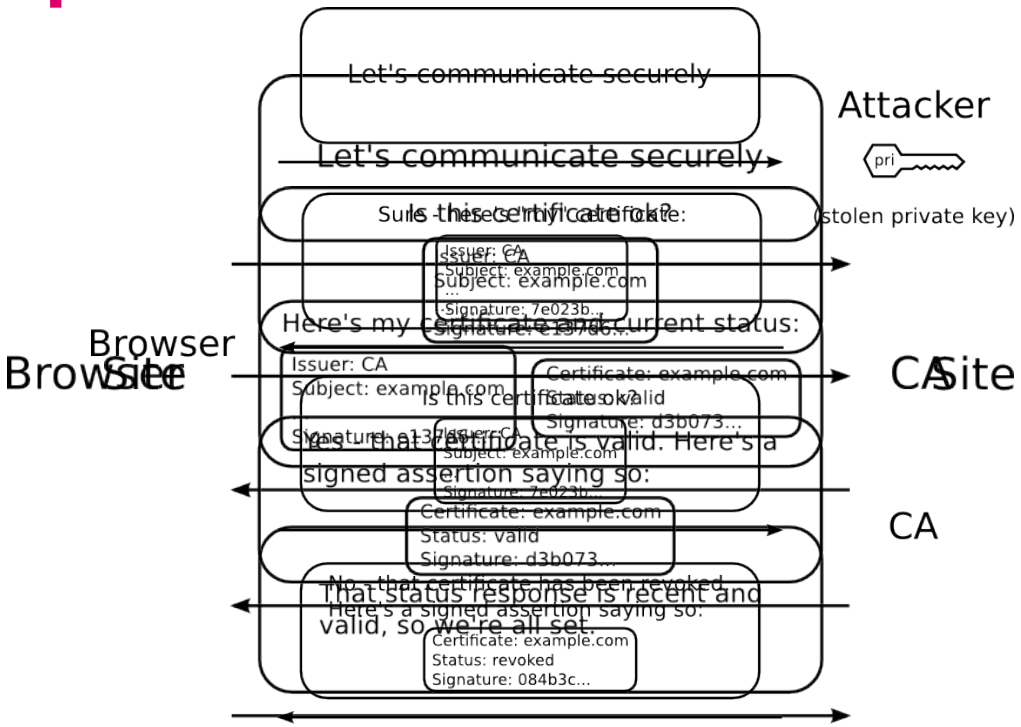
Online Certificate Status Protocol

Usually checked by client

Server obtains OCSP Response

Hands to client

httpd 2.4 can do this



https://blog.mozilla.org/security/2013/07/29/ocsp-stapling-in-firefox/

THALES

Perfect Forward Secrecy

No more Secret Scooby Snack

Diffie-Hellman key agreement

Server private key signs DH key

https://blog.twitter.com/2013/forward-secrecy-at-twitter

THALES

www.thales-esecurity.com

How Open Source has Stumbled

The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software

Martin Georgiev
The University of Texas at Austin

Subodh Iyengar
Stanford University

Suman Jana
The University of Texas at Austin

Rishita Anubhai
Stanford University

Dan Boneh
Stanford University

Vitaly Shmatikov
The University of Texas at Austin

ABSTRACT

SSL (Secure Sockets Layer) is the de facto standard for secure Internet communications. Security of SSL connections against an active network attacker depends on correctly validating public-key certificates presented when the connection is established.

We demonstrate that SSL certificate validation is completely broken in many security-critical applications and libraries. Vulnerable software includes Amazon's EC2 Java library and all cloud clients based on it; Amazon's and PayPal's merchant SDKs responsible for transmitting payment details from e-commerce sites to payment gateways; integrated shopping carts such as osCommerce, ZenCart, Ubercart, and PrestaShop; AdMob code used by mobile websites; Chase mobile banking and several other Android apps and libraries; Java Web-services middleware—including Apache Axis, Axis 2, Codehaus XFire, and Pusher library for Android—and *all* applications employing this middleware. Any SSL connection from any of these programs is insecure against a man-in-the-middle attack.

The root causes of these vulnerabilities are badly designed APIs of SSL implementations (such as JSSE, OpenSSL, and GnuTLS) and data-transport libraries (such as cURL) which present developers with [illegible] cations. The main purpose of SSL is to provide end-to-end security against an active, man-in-the-middle attacker. Even if the network is completely compromised—DNS is poisoned, access points and routers are controlled by the adversary, etc.—SSL is intended to guarantee confidentiality, authenticity, and integrity for communications between the client and the server.

Authenticating the server is a critical part of SSL connection establishment.[1] This authentication takes place during the SSL handshake, when the server presents its public-key certificate. In order for the SSL connection to be secure, the client must carefully verify that the certificate has been issued by a valid certificate authority, has not expired (or been revoked), the name(s) listed in the certificate match(es) the name of the domain that the client is connecting to, and perform several other checks [14, 15].

SSL implementations in Web browsers are constantly evolving through "penetrate-and-patch" testing, and many SSL-related vulnerabilities in browsers have been repaired over the years. SSL, however, is also widely used in *non-browser software* whenever secure Internet connections are needed. For example, SSL is used for (1) remotely administering cloud-based virtual infrastructure and sending local data to cloud-based storage, (2) transmitting cus-

Apache Projects Implicated

- Axis
- Axis2
- HttpClient 3.x
- LibCloud
- ActiveMQ
- CXF

But Mostly Fixed

Axis – EOL

Axis2 – Workaround

HttpClient 3.x – EOL

LibCloud – Fixed

ActiveMQ – Fixed?

CXF – Fixed

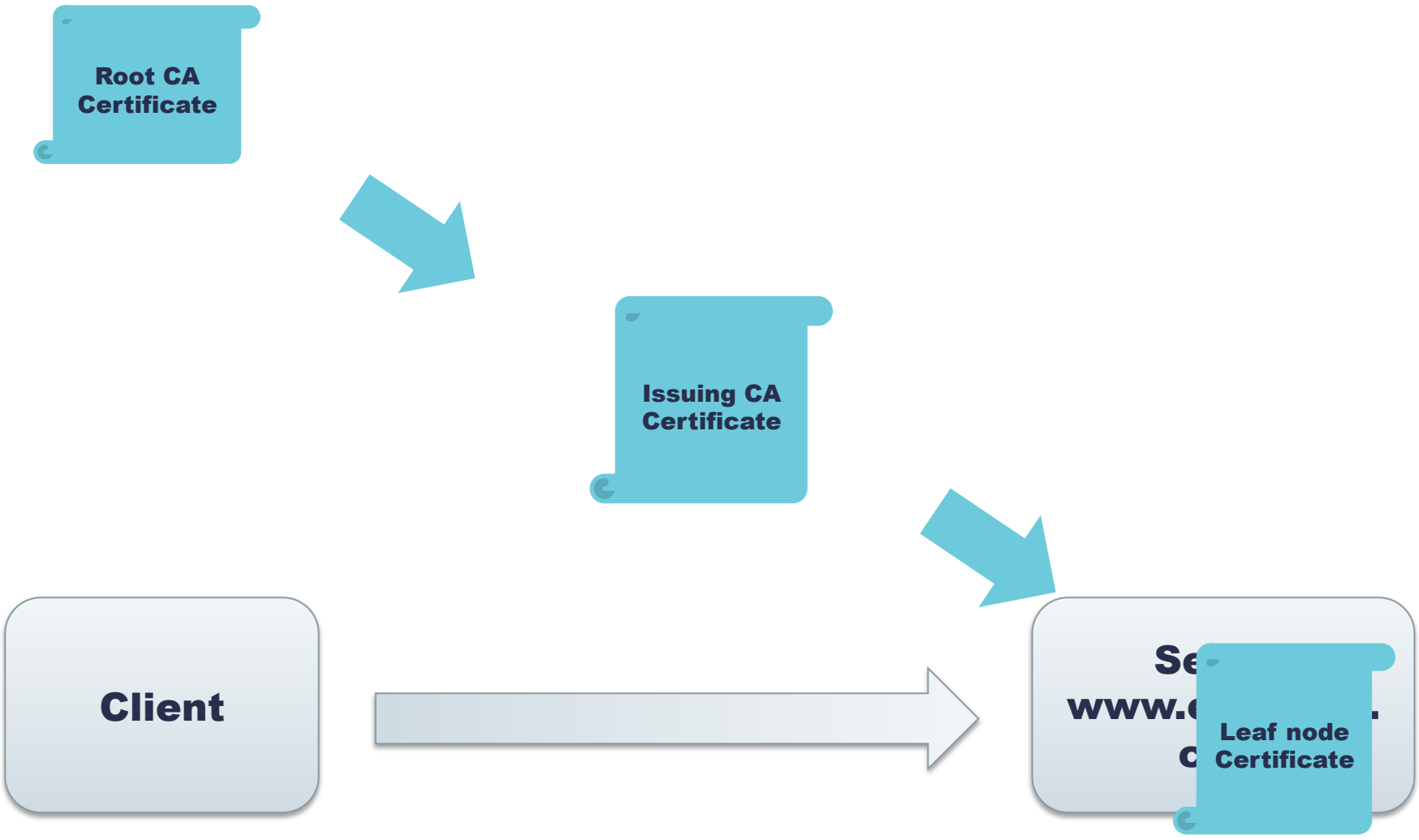
THALES

www.thales-esecurity.com

Setting Up a PKI-let

Test Your Deployment with Real Certificates

Certificate Hierarchy



THALES

OpenSSL CA

https://github.com/sctemme/pkilet

Simple script + config

Quick CA hierarchy for testing

Usage

```
./pkilet.sh -newroot

./pkilet.sh -newissuing

./pkilet.sh -newleaf localhost

openssl s_server -accept 4433 \
    -key leaves/localhost_key.pem \
    -cert leaves/localhost_cert.pem \
    -CAfile issuingCA/cacert.pem -www

curl --cacert rootCA/cacert.pem https://localhost:4433/
```

Conclusion

The State of our Union is Strong

Continued Vigilance is needed

Supporting SSL is table stakes now

Embrace and Test your SSL Support

Questions and Discussion

http://www.slideshare.net/sctemme

sctemme@apache.org

Follow @keysinthecloud on Twitter