

Leveraging Certificates for Improved Security

Joe St Sauver, Ph.D.

joe@internet2.edu or joe@uoregon.edu

InCommon Certificate Program Manager and
Internet2 Nationwide Security Programs Manager

Internet2 Fall Member Meeting, Raleigh NC
1:15-2:30 Eastern, October 4th, 2011, Room 302A

<http://pages.uoregon.edu/joe/leveraging-certificates/>

Disclaimer: the opinions expressed in this talk are those of the author, and are not necessarily those of any other entity.

I. Introduction

Where I'm "Coming From"

- **Full Disclosure:** I want folks to know that my responsibilities now include serving as InCommon's Certificate Program Manager, in addition to my continuing responsibilities as Internet2's Nationwide Security Programs Manager. If that potentially "biases" my interest in certificate-related security topics, well, you've been officially advised. :-)
- **This is NOT a "Sales Pitch:"** Yes, InCommon does run a Certificate Program for the community, and many of you participate in that program. Welcome today, and thank you! If you're not currently participating, but you buy a lot of certificates, try figuring out how much you spend on them. Compare that to http://www.incommon.org/cert/cert_fee.html If it makes sense for you to participate, we'd love to have you become part of the program too. If it doesn't, that's also cool. Obviously you should do what makes sense for your school.

Today's Session

- ***Session Emphasis:*** For the most part, I'd like this session to stay focused on technical/general certificate-related issues. If there are business/admin questions that relate specifically to the InCommon Certificate Program, I'd be happy to visit with you about those, but let's do that after today's session.
- ***Session Format:*** I'd like this to be an interactive session, so although I've prepared material to go over, feel free to chime in if you've got comments on any of the topics I raise – the value of this session will increase if everyone is actively engaged.
- ***Technical Level:*** Since Member Meeting audiences always include an interesting mix of managerial and technical attendees, I've tried to provide "backfill" where necessary, but I've tried to also include enough technical material to keep the session interesting for those with a primarily technical focus.

How This Talk Relates to Other Member Meeting Certificate-Related Sessions

- ***SSL/TLS Certificates: Giving Your Use of Server Certificates A Hard Look (Yesterday, 4:30-5:30 PM)***
“What are the security issues we face when it comes to hardening and improving the way we run existing SSL-enabled web servers?” That is, “How can we run our existing web servers more securely?”
- ***This Talk --*** “Are we using certificates as broadly as we can? Or are there additional/new ways we could or should use SSL server certificates to improve our security? And what new frontiers are opened by use of client (personal) certs?”
- ***PKI BOF (Tomorrow Noon-1PM)*** -- An interactive community-driven session, lead by Jim Jokl of U Virginia. Topics include the recent two factor auth survey and client certificate installers among others. Please attend!

Another Way of Thinking About This Talk

- Assume you've just joined the InCommon Certificate Program
- You now have the ability to issue unlimited certs for all of your domains, as well as the ability to issue client (personal) certificates. You could do one of two things:

Option 1: Replace your existing certificates with certificates from the InCommon Certificate Program as they expire, but otherwise do nothing new.

Option 2: Replace your existing certificates (as in Option 1), but ALSO look for new or different ways you might be able to use certificates at your site, including potentially client certs.

- This talk focuses on Option 2 since folks presumably already know how to do Option 1. :-)

II. Types of SSL Certificates, Including Extended Validation ("Green Bar") Certs

Consider Replacing Your Regular SSL Certificates with Extended Validation SSL Certificates

The Term "Certificate" Can Be Misunderstood

- Most users have no idea that SSL "certificates" even exist.
- If a typical person heard the word "certificate," they might mentally conjure up an image that looks something like this:

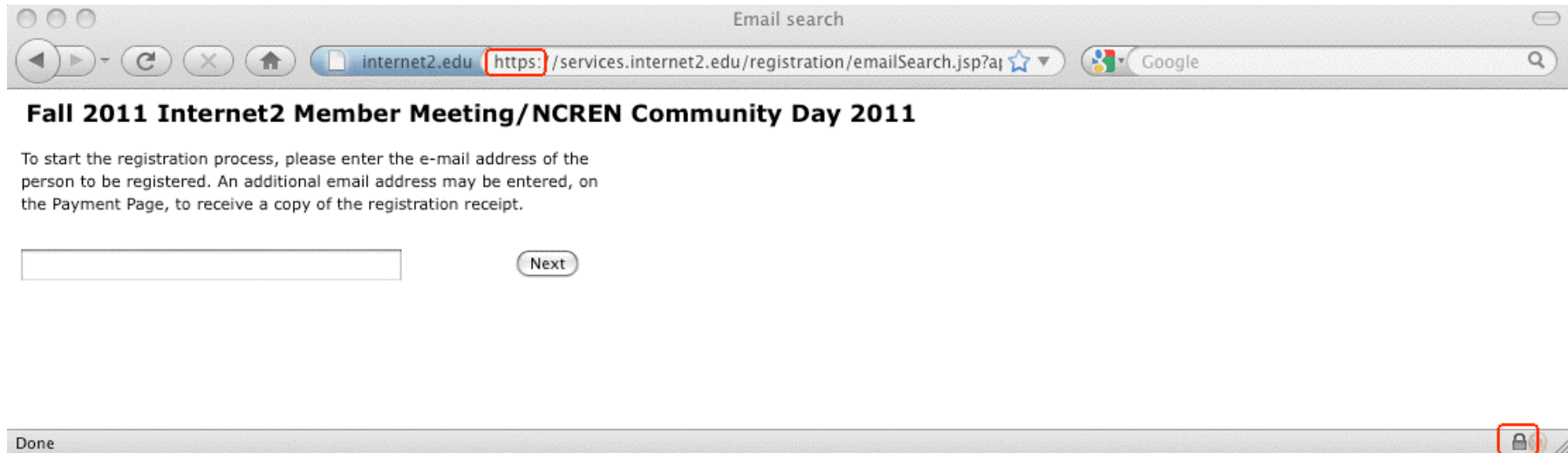


- Information technology people, on the other hand, if they hear "certificates," might first wonder if you're discussing staffers who've obtained professional credentials such as the CCIE (Cisco Certified Internetwork Expert), or CISSP (Certified Information Systems Security Professional).

Those *Aren't* The Sorts of Certificates We're Going To Talk About Today

- Today, we want to focus on cryptographic certificates.
- Those sort of certificates are often referred to as "X.509 certificates" or "SSL certificates" or "web server certificates."
- If cryptographic certs do get noticed, it's usually when a user visits a secure web site (such as the one we all visited to register for this meeting).

A Sample Secure Web Page



- In Firefox, a variety of user interface elements are meant to "cue" or help the user to notice that this is a special secure web page, and not just a "regular" one. Notice:
 - the blue field with the domain in the address bar
 - the https URL prefix (with an "s", standing for "secure")
 - the little padlock on the bottom margin of the window

Those Cues are Subtle, Can Be Overlooked or Faked, And May Confuse Some Users

- Many folks won't even notice that the "s" on https, or the presence of that little padlock, when visiting a secure page.
- Sometimes, phishers attempt to trick users by adding a "padlock" favicon.ico (<http://en.wikipedia.org/wiki/Favicon>) to a web site they're abusing, hoping that users will think that the padlock icon they're seeing means that this is a secure site, even though it isn't.
- The prominent blue-colored area near the web address is hard to overlook, but will users understand what that color is meant to connote?
- What if the user was curious, and wanted more information? For example, what if users started clicking on some of those user interface cues, looking for more information?

More Information *IS* Potentially Available...

- While looking at that web site in Firefox, users can click on the blue area to the left of the web address, asking to see "More Information" and then ask to "View Certificate:"



Note: This particular cert predates the InCommon Certificate Service – when it expires in a month of so I'm confident that it will be replaced with a certificate from the InCommon Certificate Program. :-)

But Users Usually Won't Bother Looking at Cert Info

- After all, why should they? Secure web sites "work" even if users doesn't look at the certs, eh?
- Also, certificate-related information panes prominently feature all sorts of obscure/intimidating/cryptic numbers and even basic vocabulary that gets used in unexpected or confusing ways (for example, consider the terms "Common Name" or "Fingerprint" as shown on the preceding slide)
- Even if a user did look at that information, but then had questions, where would they go to get those questions resolved? Secure sites typically are pretty much a "take it or leave it" proposition, right? You really only have two choices: use it, or don't use it.

Users May Have Little Choice But to Accept

- A recent study by Holz, Braun, Kammenhuber and Carle* found that less than 1-in-5 of the certificates they observed had both a correct host name and a valid certificate chain, allowing the cert to be shown to be cryptographically valid.
- The rest of the time, presumably one of the following is true: 1) appropriate certificate checks aren't occurring, 2) users are proceeding notwithstanding substantial cert problems, or 3) an awful lot of sites using certs won't be accessible!
- In truth, the problem may even be worse than that – not only do users accept certs they shouldn't, they routinely draw unsound inferences from a site's use of an SSL certificate...

* Preprint shared on the Randombit Cryptography mailing list 29 Sep 2011, see <http://lists.randombit.net/pipermail/cryptography/2011-September/001658.html>

The Default Mental Thought User Process

- Many users have been implicitly or explicitly **mis-trained**, and now -- almost as an article of faith -- many seem to **incorrectly** believe that:
 - “As long as you see https in the address bar... or you see that little padlock down on their web browser's bottom margin... or you see the blue colored field up near the web address... Then you can proceed to 'safely' enter passwords, credit cards, etc., on that site.”
- That is, of course, completely **crazy**.
- Peter Gutmann of the University of Auckland has some classic examples of “interesting” sites that have valid certs (and some real “mainstream” sites that have invalid ones) in “PKI as Part of An Integrated Risk Management Strategy for Web Security,” EuroPKI 2011, http://www.cs.auckland.ac.nz/~pgut001/pubs/pki_risk.pdf

What Certificates Can (and Can't) Do

- Securing web sites through use of cryptograph certificates was meant to accomplish three things:
 - protect your information from eavesdropping
 - ensure that your information hasn't been altered in transit
 - reassure you that you're dealing with the site you really wanted to deal with, and not a fake/imposter site
- That's **NOT** what users think they're getting from a site secured with a certificate.
- **Users have a far simpler (wrong!) notion:**
SSL "Secured" Site? --> The Site Must Be Trustworthy
(fair, honest, able to be relied on, etc)
- SSL certs are actually like the services of a notary public. A notary public certifies that she saw you sign, and that your government ID matches your name and signature, NOT that the contract she saw signed was a fair or worthwhile one.

Put Simply: Certs Are NOT About "Reputation" or "Trust"

- **Good** people and organizations can get certificates. Really **bad** people and organizations (including criminals!) can **also** get (at least some types of) certificates (see Gutmann's talk, mentioned previously in this section).
- It would be, and is, a critical/terrible mistake to assume that just because a web site has a valid SSL certificate, that that site is trustworthy! Many users do NOT "get" this, so this is a point worth stressing if you talk with them about certs
- Technically, certs themselves don't even provide protection against eavesdropping or tampering, that's done by the cryptographic key pair that's behind the certificate.

Certs Bind *Identities* (Of One Sort or Another) To *Cryptographic Key Pairs*

- What are those “cryptographic key pairs” we just mentioned? Behind every certificate there lives:
 - a public key that can be freely shared with anyone, and
 - a corresponding (secret!) private key.
- The public key forms the foundation for each Certificate Signing Request (CSR). The CSR gets forwarded to the certificate authority (CA), which then **validates the identity of the requesting entity** (one way or another), issuing a certificate that cryptographically signs the public key, binding the requesting entity's identity to it.
- What varies from cert type to cert type is the type and thoroughness of the validation process that gets employed. That validation can vary from extensive (in the case of extended validation certs) to nothing at all (self-signed certs).

Four Levels of Identity "Validation"

1) "Self-Signed" certs: these certificates **haven't been validated** by a broadly recognized certificate authority.

You have no assurance whatsoever that those cryptographic credentials really belong to who you think they belong to. Maybe they do, maybe they don't. You simply don't know.

2) Domain Validation (DV) certs: an automated email with a unique code gets sent to an email address associated with the domain for which a cert has been requested. This email goes to a standard role account (such as postmaster) or a point of contact address listed in the domain's whois. The certificate requesting party then follows the instructions in that email ("click on this link") to demonstrate "control" over that **domain name**. DV certs are probably the most common type of SSL certificate. InCommon, however, doesn't issue them. DV certs bind a certificate to the applicant's domain name.

Four Levels of "Validation" (continued)

3) Organizational Validation (OV) certs: OV certs, such as the ones that InCommon does issue, require a more careful verification of the applicant's business identity and applicant roles. Use of an OV cert will typically not result in any prominent indicator or mark that signifies that status in most browsers, although if you check the certificate manually, you will normally see the organization name (and *not* just a domain name) actually listed in the cert.

4) Extended Validation (EV) certs: Extended validation certs are the least common sort of cert. Before an EV cert gets issued, a more thorough investigation of the identity of the applicant gets conducted per the requirement developed by the Certificate Authority and Browser forum. Sites that use an EV cert **will** display the organization's name in a "green bar" in current generation web browsers.

EV Cert Indications Vary From Browser To Browser

OTA - EV SSL Certs Resources

Online Trust Alliance (OTA) (US) <https://otalliance.org/resources/EV/index.html>

See the EV Difference in Various Web Browsers

The image displays five browser windows, each showing the Online Trust Alliance (OTA) website. The browser icons are listed on the left: Internet Explorer, Firefox, Chrome, Opera, and Safari. The screenshots illustrate how each browser displays the EV certificate information for the URL <https://otalliance.org/>. Internet Explorer shows 'Online Trust Alliance (OTA) (US)' in the address bar. Firefox shows 'Online Trust Alliance (OTA) (US) https://otalliance.org/'. Chrome shows 'Online Trust Alliance (OTA) [US] https://otalliance.org'. Opera shows 'Trusted otalliance.org'. Safari shows 'Online Trust Alliance (OTA)' and 'Google' in the address bar.

Choice of A Cert Validation Type

- Why do some sites use self-signed certs, while others use DV certs, OV certs, or EV certs? Three factors normally drive site selection of one or the other: cost, convenience and consumer confidence.
- **Cost:** self-signed certs are free. EV certs are quite cheap. OV certs are usually more expensive (since they involve manual processing/validation of applicant details). EV certs, involving the most thorough review, are the most expensive.
- **Convenience:** You could make your own self-signed certs. EV certs, since they are normally processed on a wholly automated basis, can be issued in near real-time. OV certs normally take longer (since they involve manual processing). EV certs take the most time (and paperwork!) of all.
- **Consumer Confidence:** Hopefully this increases from self-signed certs to DV certs to OV certs to EV certs.

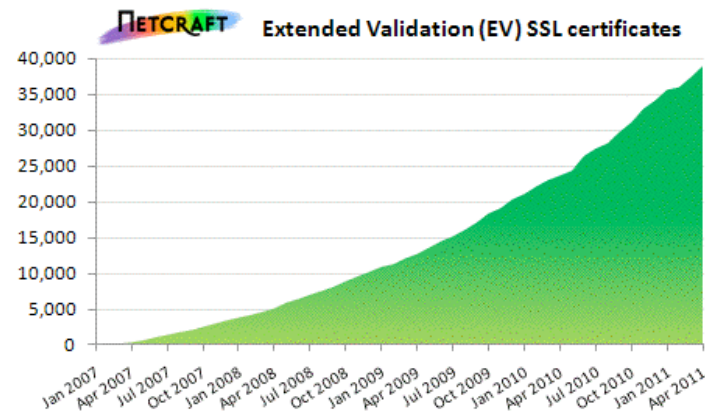
EV's Role: Reversing A "Race To The Bottom"

- For a long time there weren't all these types of certs. In the old days, there were only self signed certs, and carefully verified commercial certs (similar to today's OV certificates).
- Then somewhere along the line, certificate authorities began to compete on price. If you're competing on price and have very thin profit margins, you can't afford to do extensive validation of each applicant and still make a profit. You need to substitute automation -- and many CAs did. They began to simply verify that you had control over a *domain name*, rather than that you controlled a *business entity*. Importantly, in many ways, these cheap DV certs had little that tangibly distinguished them from higher quality/higher cost OV certs.
- **Extended validation certificates were created in an effort to claw back consumer confidence, particularly for banks and other prime phishing targets. BUT...**

Netcraft survey indicates slow adoption of Extended Validation SSL certificates

By Dancho Danchev | April 26, 2011, 4:05am PDT

Summary: According to the latest Netcraft SSL Survey, Extended Validation SSL certificates still only account for 2.3% of all valid third party certificates analyzed by the company.



According to the latest Netcraft SSL Survey, **Extended Validation SSL certificates** still only account for 2.3% of all valid third party certificates analyzed by the company. The steady, but slow adoption is attributed to both, pricing and site verification concerns. The survey finds that extended validation SSL certificates are most prevalent on high traffic or financial web sites, and are used to further establish a trusted relationship between the web site and the visitor.

Restricting the survey's sample to the busiest 1,000 websites in the world, 81 sites accepted HTTPS connections and presented a valid SSL certificate. Nearly a third of these certificates used Extended Validation – a far higher proportion than the 2.3% share of all certificates.

Why The Slow EV Rollout? Issue One: Cost

- While EV certs do a nice job of potentially improving confidence in critical sites, extended validation certificates are still quite uncommon because they have traditionally been **expensive** to obtain (hundreds of dollars per cert), and many sites simply couldn't afford to obtain them.

- **Key Point/Good News:**

Cost has **ceased** to be an impediment to deploying EV certificates, at least for sites participating in the InCommon Certificate Program, because **extended validation certificates are included in the InCommon Certificate Program at no additional charge.**

EV Issue Two: Paperwork

- Even though the InCommon Certificate program can make cost a non-issue for Cert Service subscribers, getting extended validation certs WILL still require your site to do some paperwork, including typically producing a "lawyer letter."
- Details of the restrictions associated with EV certs, and what's required in terms of paperwork can be seen here: <https://www.incommon.org/cert/evcerts.html>
- I wouldn't let the potential paperwork deter you from applying for an EV cert – it really isn't THAT bad (and besides, those of us in academia excel in processing paperwork, right :-))


EV Issue Three: User Awareness

- The final issue that has inhibited EV cert adoption has been a lack of user education and awareness.
- Even with a change in color on the browser address bar, many users don't "get" what that color implies, or why extended validation certificates are better than a regular SSL certificate.
- Nonetheless, despite those issues, we ARE seeing higher education sites deploying EV certificates, both from the InCommon Certificate Program and from other sources. A few examples of universities that are using EV certs follow...

A University Site That Uses an EV Cert

Cornell University Web Login

Cornell University (US) [https://web1.login.cornell.edu/?](https://web1.login.cornell.edu/) Google

 Cornell University

CUWebLogin

NetID:

Password:

[What is this?](#)
[I forgot my password!](#)
[I don't have a NetID, now what?](#)


To log out, you must Exit or Quit your browser.

Caution: Always check your browser's address bar before you enter your NetID password to make sure the address starts with https://web*.login.cornell.edu/ (where web* is either web1, web2, web3 or web4).

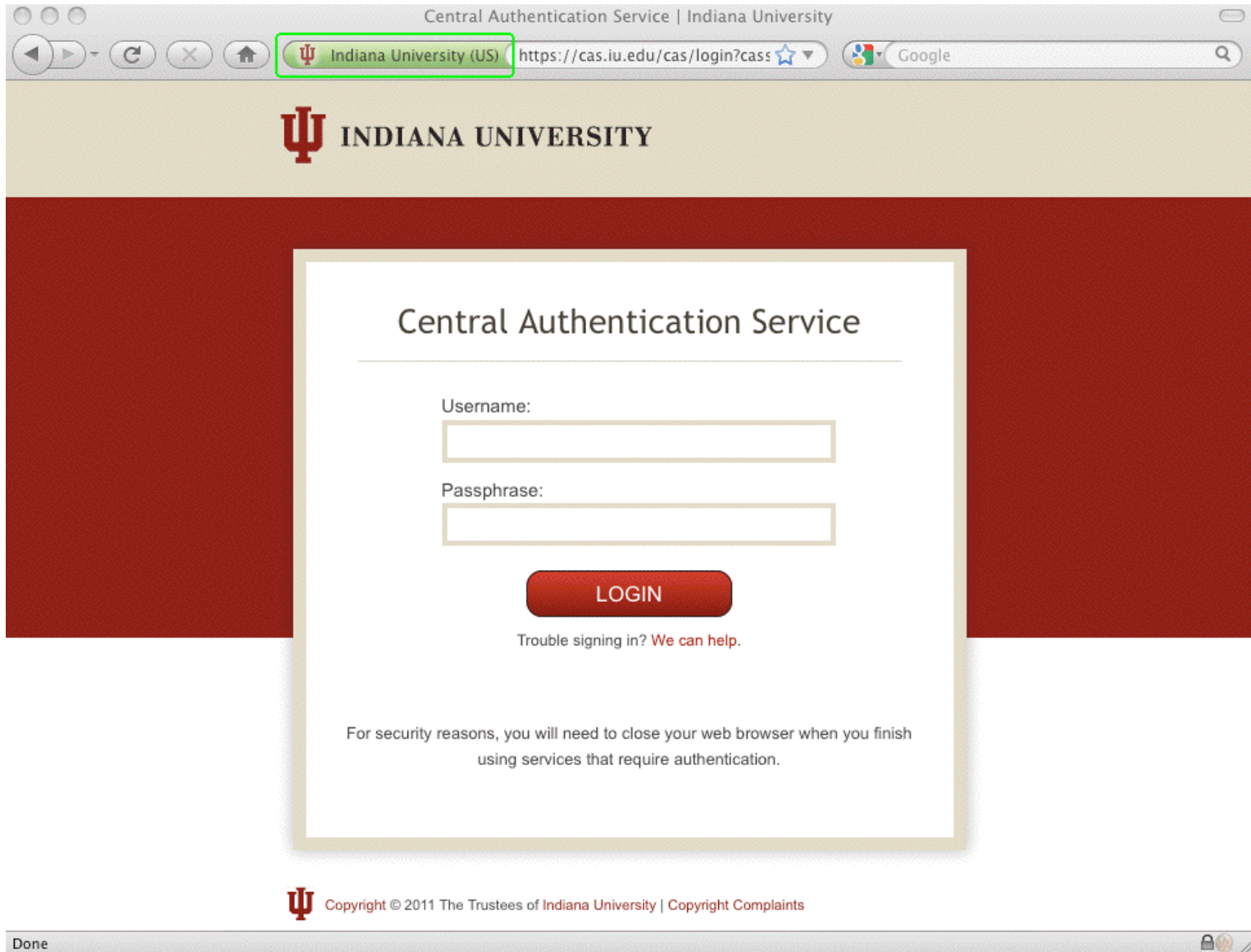
CUWebLogin is a component of Cornell University's central authentication service. If you are unsure of the authenticity of any online University service, please contact security@cornell.edu.

This service and the services to which it provides access are for authorized use only. Any attempt to gain unauthorized access, or exceed authorized access, to online University resources will be pursued, as applicable, under campus codes and state or federal law.

© 2008 Cornell University. All Rights Reserved.

Done 

A Second Example Using An EV Cert




And A Third Example Using an EV Cert

The screenshot shows a web browser window titled "Princeton University Authentication Service". The address bar contains "https://fed.princeton.edu/" and the page title is "Princeton University (US)". The page features the Princeton University logo and a "Help" link. A prominent orange banner reads "Central Authentication Service". Below this is a login form with fields for "NetID" and "Password", a "LOGIN" button, and a checkbox for "Prompt me before logging into other CAS protected sites." A "Change your Password" link is located at the bottom right of the form. The footer contains the copyright notice "© 2010 The Trustees of Princeton University".

Princeton University Authentication Service


Princeton University (US) https://fed.princeton.edu/ Google

 PRINCETON UNIVERSITY [Help](#)

Central Authentication Service

NetID

Password



LOGIN

Prompt me before logging into other CAS protected sites.

[Change your Password](#)

© 2010 The Trustees of Princeton University

Done

And A Final Example Using an EV Cert



Is YOUR School Using EV Certs? Should You Be?

- If you have a security-critical web site that collects or uses:
 - Passwords or cookies for authentication or “login”
 - Personally identifiable information (such as SSNs)
 - Financially sensitive information (such as credit card info)
 - Medical information (e.g., HIPAA-covered information)
 - Grades or other FERPA-covered student records

Or, if you have sites that may be a priority target for spoofing, such as wireless authentication or VPN sites, then yes, I think you SHOULD be using Extended Validation ("green bar") certificates for those sites.

- **"If it's a site that matters, GO GREEN!"**

Coming Back to EV Issue 3 For A Second: Have You Trained Your Users About EV Certs?

- It is probably unrealistic to expect users to know about and understand extended validation certificates on their own – you should consider an awareness program to help make your users aware of the role and implications of extended validation certificates when they encounter them.
- You should particularly train them that if they customarily see a green bar cert for a critical site, but then one day they suddenly don't, they should STOP and find out what's going on. Did they perhaps make a typo or otherwise accidentally go to the wrong site? Or, is something sinister happening?
- EV certs have one other advantage: Criminals like to work from the shadows and hide their identity so law enforcement can't track them down and hold them accountable. It's hard or impossible for a criminal to do that AND obtain an EV cert.

Are EV Certs Cryptographically “Stronger?”

- No. EV certificates, just like self-signed certs, EV certs, or OV certs, all have 2048 bit signatures these days. They are not cryptographically stronger.
- EV certs ARE procedurally stronger when it comes to establishing who’s “behind” those certificates.
- EV certs are also less common, which helps to reduce the chance that a look-alike site will have a “green bar” cert the way a real site might.

Weren't A Number of EV Certs Issued in the Recent DigiNotar Incident?

- Yes. When Dutch CA DigiNotar B.V. was compromised, the hacker issued a variety of certificates, including a number of unauthorized “extended validation certificates.”
- As a result of that incident, all known mis-issued DigiNotar certs have been revoked. DigiNotar’s root certificates have also been eliminated from the default list of trusted CAs in popular browsers and operating systems.
- For more information, see:
 - “DigiNotar Damage Disclosure,”
<https://blog.torproject.org/blog/diginotar-damage-disclosure>
 - “DigiNotar Public Report, Version 1,” [English language]
<http://tinyurl.com/diginotar-report>
 - “VASCO Announces Bankruptcy Filing by DigiNotar B.V.,”
<http://tinyurl.com/diginotar-bankruptcy>

Recapping This Section

- Users generally don't understand what certs can do and can't do for them, and as a result they may "trust" sites that they shouldn't. Help educate them!
- Certificates bind identities to cryptographic credentials; the strength of that binding is a function of the validation done when that certificate is requested.
- Extended validation ("green bar") certificates were created to reverse the domain validation "race to the bottom," but because of cost, paperwork and a lack of user familiarity, are still relatively uncommon
- InCommon Certificate Service participants can obtain extended validation certificates at no additional cost
- Most universities should be deploying EV certs for their "sites that matter," and some are, in fact, already doing so.

III. Secure All Network Services, Not Just HTTP

*Broaden Your Use of Certificates:
Use SSL to Encrypt All Network Services
(Except for Times When You're Using SSH Instead)*

Classic Cert Usage: Web Servers, Only

- In the old days, many sites would often only use a certificate to secure their most critical web site, such as an online ordering site where credit card information would get entered.
- From that simple beginning – perhaps just one or two web servers per domain – certificate usage has expanded at many sites to include **ANY** web server that asks the user to login with a username and a traditional password. Objective? Protect the user (including privileged users!) from having their username and password sniffed.
- Classic examples of those sort of additional web servers:
 - institutional ERP (administrative) systems
 - campus webmail systems
 - departmental web servers
 - many other web servers

Don't Forget Your Auxiliary Domains & Test Servers

- When you're thinking about your school's web servers, you probably tend to think about the production web servers in your school's main dot edu domain.
- Don't forget about other domains that your school may also own and use, including domains that may be in dot com, dot org, dot net, dot info, etc. Some of you may even have multiple dot edu domain names for the same school, grandfathered from before that practice was prohibited by the dot edu domain admin folks. If you're getting your SSL certificate through the InCommon Certificate Service, you can get SSL certs for all those school-owned domains, just as you would for hosts that use your primary dot edu domain name.
- Also remember your test and development web servers – since there's no incremental cost, you might as well use real web certificates there as well.

Issue Host Specific Certs, Not Wildcard Certs

- Historically, some sites may have obtained a single wildcard certificate (e.g., for *.example.edu) that would work for any server in that domain. This was a cheap option that also minimized cert-ordering-related administrative hassles.
- However, using a shared wildcard cert means that the binding between the cert and any system's identity is weaker than if the cert had been minted for one and only one domain name.
- It also means that if ANY web server using that cert is compromised, and the associated private used with that cert ceases to be trustworthy, ALL web servers on campus using that cert would need to obtain and install a replacement cert.
- Therefore, our recommendation to you, particularly if you participate in the InCommon certificate program, would be to issue host specific certs for individual hosts rather than sharing domain-wide wildcard certs!

SSL Certs CAN BE, and ARE, Used On More Than Just Web Servers...

- While SSL certs are most commonly used on secure web servers, they can be used to secure other protocols, too. For example, users may retrieve their email via POP or IMAP, or submit email via the email submission port (port 587), and when they do that they will routinely be supplying their username and password over the network (where it could potentially be sniffed if sent unencrypted)
- As a result, these services also began using SSL certificates to protect passwords and other sensitive information. This is normally referred to as “Transport Layer Security” or “TLS”
- Some services, such as mail server to mail server (MTA-to-MTA) mail transfers, also ended up doing opportunistic encryption with certs even if those exchanges didn't involve sending passwords in clear text over the wire.

Some Protocols with RFC-Specified TLS Support

- Message Submission: RFC 2476
- POP and IMAP: RFC 2595
- LDAP: RFC 2830
- SMTP: RFC 3207
- XMPP (“Jabber”): RFC 3920
- NNTP (“Usenet News”): RFC 4642
- Syslog: RFC 5425
- SNMP: RFC 5591

Additional Protocols Known To Use Certs

- Many other application also use TLS. For example:
 - 802.1x Wireless Authentication
 - OpenVPN can use TLS
 - some implementations of Samba/SMB/CIFS support TLS (the key factor is often LDAP support for TLS)
 - selected VoIP implementations support TLS (see http://en.wikipedia.org/wiki/Comparison_of_VoIP_software)
- All this said, https, pops, and imaps remain the most common SSL/TLS encrypted protocols.
- What about telnet (or rlogin/rsh) and ftp (or rcp)?

ssh and sftp: TOFU, No Certs

- An interesting exception to the general rule that "SSL/TLS will get used to secure network services" is that:
 - telnet/rlogin/rsh (the remote login "terminal" protocols) got replaced with ssh (secure shell, see RFC4250-RFC4256)
 - ftp/scp (the file transfer protocols) got replaced with sftp/scp (secure ftp/secure copy, based on ssh)
 - X11 sessions normally get forwarded over ssh, not TLS
 - secure rsync leverages ssh for security, not TLS
- ssh differs from ssl in many ways, including the fact that ssh does NOT use certificates to establish trust. ssh is a TOFU ("Trust On First Use") protocol, tracking and alerting the user to any subsequent change in cryptographic keys that occurs.
- While ssh and ssl are totally different protocols, many ssh implementations do rely on SSL cryptographic libraries. For example, OpenSSH is built using the OpenSSL libraries.

Using Dedicated Secure Ports vs. StartTLS

- Coming back to SSL/TLS, there is a point of confusion that sometimes arises: in the old days, SSL-secured versions of protocols would often run on their own dedicated ports, e.g.:
 - SMTPS would run on port 465*
 - IMAPS would run on port 992
 - POPS would run on port 995The only way you even could connect to those ports was via an SSL-secured connection. No SSL? No connection!
- Modern practice is different. Now you simply connect to the normal port associated with a service (e.g., port 25 for SMTP, 143 for IMAP, or 110 for POP), and then issue the STARTTLS command when you want to “go encrypted.”

- * Note that port 465 was *never* officially assigned by IANA for SMTPS; it officially belongs to URD according to IANA!

TLS Isn't Perfect – For Instance, Sometimes StartTLS Negotiation May Fail, and TLS Only Provides Encryption “In Flight” not “At Rest”

- You should be alert to the possibility that TLS enabled services may not require TLS to happen for a connection. If TLS protection can't be negotiated, what should happen? Some administrators may want to refuse to establish the connection (protecting the password); others may want to fall back to an unencrypted connection. What does your site do?
- While TLS provides end-to-end protection while network traffic is being transmitted, it does **not** protect that traffic before it is transmitted or after it has been received.
- Later in this talk, we'll talk about using S/MIME for encrypting email payloads and other files. S/MIME will protect encrypted message bodies or other files, even when those messages are "at rest" (e.g., have been received and stored)

Enabling SSL/TLS Support in MTAs

- If you want to enable SSL/TLS in major mail transfer agents, including the MTA your school probably uses, the process is pretty simple. Instructions for some of them can be found at:
- **Exim:** http://www.exim.org/exim-html-current/doc/html/spec_html/ch39.html
- **Microsoft Exchange:**
<http://support.microsoft.com/kb/829721>
- **Postfix:** http://www.postfix.org/TLS_README.html
- **Sendmail:** <http://www.sendmail.org/~ca/email/starttls.html>

Enabling SSL/TLS Support For POP/IMAP

- It is also easy to add SSL/TLS support for your POP/IMAP servers. For example, see:
- **Courier:**
<http://www.courier-mta.org/imap/INSTALL.html#install>
- **Cyrus:** <http://delouw.ch/linux/Postfix-Cyrus-Web-cyradm-HOWTO/html/cyrus-config.html>
- **Dovecot:**
<http://wiki2.dovecot.org/SSL>
- **UW IMAP:**
<http://www.washington.edu/imap/IMAP-FAQs/index.html#3.6>

Enabling SSL/TLS Support For LDAP

- Likewise, you can use SSL/TLS to secure directory services, such as LDAP:
- **389 Directory Server:**
<http://directory.fedoraproject.org/wiki/Howto:SSL>
- **Active Directory:**
<http://support.microsoft.com/kb/321051>
- **OpenLDAP:**
<http://www.openldap.org/faq/data/cache/185.html>

Bottom Line for This Section

- Is your campus using SSL/TLS everywhere you might be?
- If you were to sniff your traffic at your network border, would you be seeing any sensitive traffic that's still being passed in plain text?
- If you are still seeing sensitive traffic in plain text, begin working to deploy SSL/TLS security on those unencrypted network services.

IV. HTTP Strict Transport Security (HSTS)

*Most Web Pages Are Currently Served Unencrypted.
The New Default? Strive to Make It "Always Use https."*

Certificates: They're Not Just For Critical Web Content Anymore

- For a long time, most sites only deployed certs for critical content, leaving the vast majority of routine web traffic flowing over the network unencrypted.
- Why? The usual answers (valid or not) were all or some of:
 - we do encrypt login info, that's the only real worry, right?
 - no secrets are being served; users don't care; why bother?
 - why buy expensive certificates when they aren't needed?
 - it's a hassle obtaining, installing and maintaining certs
 - we don't want to have to accept the "performance hit" associated with doing encryption and decryption on traffic
 - debugging problems will be harder if the traffic is encrypted
 - encrypted traffic can't be cached or proxied
 - incoming encrypted traffic can't be scanned for malware
 - let me look into that/I'm too busy/I'll do it "real soon now"

Then, The World Encountered Firesheep...

- If you're not familiar with Firesheep, see <http://codebutler.com/firesheep> (24 October 2010)
- Firesheep is an application that does a nice job of demonstrating that encrypting just the user's login session is not enough (at least if a web site relies on cookies for authentication and access control): even if an attacker couldn't capture your username and password, they could still capture an unencrypted cookie, and after that, the attacker would then have full control of your account.
- This wasn't a new vulnerability, but creation of Firesheep made it apparent to everyone that this was a practical (rather than theoretical) worry.
- The only real solutions? Don't rely on cookies to carry critical security information -- **or** encrypt everything with https.

HTTP Strict Transport Security (HSTS)

- What we really need is a way for sites to declare that ALL traffic for their domain MUST be sent via https, and ONLY via https.
- If we just wanted to ENCOURAGE use of https on a site, a formal protocol isn't absolutely necessary. Any site could simply decide to start using https to secure the web pages on their site, and voila, it could be done. However, it's easy for a user to accidentally request a page via http (instead of https), or for a web programmer to mistakenly link to an unencrypted local web page rather than an encrypted one.
- Fortunately, HSTS provides a way to say that a site MUST use https only. See: "HTTP Strict Transport Security (HSTS)," Hodges (Paypal), Jackson (CMU) & Barth (Google), tools.ietf.org/html/draft-ietf-websec-strict-transport-sec-02 (expires February 6th, 2012)

Enabling HSTS

- Enabling HSTS on a web site that uses Apache is pretty easily done, see the description at "Adding HTTP Strict Transport Security (HSTS) to Apache Virtual Hosts," <http://linux.dashexamples.com/2011/08/adding-http-strict-transport-security-hsts-to-apache-virtual-host/>
- The one required additional HSTS header must be sent via an https: page – that header will be ignored if it is sent via an unencrypted http page. As a result of that requirement, and also due to limited browser support, OWASP emphasizes use of a 301 permanent redirect instead: www.owasp.org/index.php/HTTP_Strict_Transport_Security (note that while approach work with any browser, it doesn't rule out use of self-signed certs or other HSTS corner cases)
- That OWASP page also provides pointers to recipes for enabling HSTS on IIS, NGINX, and other web servers, too.

Browser Support for HSTS

- To be candid about one disappointing point: browser support for HSTS is not currently really where I'd like it to be, and that's a shame, because browsers play a key role in recognizing and enforcing use of the HSTS protocol.
- ***The good news?*** HSTS **is** at least currently enabled in recent versions of Firefox and Chrome (e.g., see for example <http://www.chromium.org/sts>). Another helpful point: even if you use a browser that doesn't support HSTS, that browser's non-support of HSTS shouldn't actively break anything.
- ***The bad news?*** There are major/important browsers that don't currently have support for HSTS: Internet Explorer, Safari, and Opera do not have support for HSTS at this time. (Likewise, I don't believe that HSTS has made it into many mobile device web browsers). Talk to your vendors!

Name-Based Virtual Hosting and https Usage

- One other consideration: when it comes to regular (non-https) hosting, many sites use name-based virtual hosting. In name-based virtual hosting, dozens or even hundreds of domains may get hosted on a single shared IP (e.g., see for example <http://httpd.apache.org/docs/2.2/vhosts/name-based.html>)
- Traditionally, secure web sites needed IP-based hosting, with each secure web site residing on a dedicated address. If you have lots of secure web sites, doing IP-based hosting could rapidly deplete your pool of available addresses.
- Server Name Indication ("SNI") eliminates that requirement if you're running a current secure web server and browser. See <http://wiki.apache.org/httpd/NameBasedSSLVHostsWithSNI>
- *Caution:* Some browsers on some operating systems do not have support for SNI. Current versions of Firefox are generally SNI-safe on most all current operating systems.

Mixed Scripting and Mixed Display

- While you're tightening things up and promoting use of https everywhere, you may particularly want to note the problem of "mixed scripting," where an https page loads a script, cascading style sheet or plugin resource over an insecure (http, instead of https page).
- Also bad: when an https page loads an image, iframe or font over http, a related if somewhat less serious problem that's sometimes called "mixed display".
- A nice summary posting on this issue is available at "Trying to End Mixed Scripting Vulnerabilities," June 16, 2011, <http://googleonlinesecurity.blogspot.com/2011/06/trying-to-end-mixed-scripting.html> (the comments to that post bring up some interesting examples of prominent sites that apparently have issues in this regard)

Quick "Take Aways" For HSTS

- **The old default:** unencrypted http for most web pages, with SSL/TLS security only where it's absolutely needed.
- **The new default:** plan on using encryption (https) everywhere.
- **Things to check:**
 - Can my web server software support SNI? If not, do I have enough IPv4 address space to do IP-based hosts for all my HSTS-enabled hosts? Should I be requesting more?
 - Are we recommending a browser that supports SNI (as well as HSTS)? Are there any old legacy Windows XP user desktops that we might need to get updated?
 - If we moved to HSTS, would we encounter mixed scripting or mixed display issues?

VI. Speaking of Browsers...

*Attacks Targeting SSL/TLS Directly Exploit User's Browsers.
Be Sure Your Users Have the Tools And Know-How They
Need to Chart and Travel A Safe Trail Online...*

We've Been Largely Talking About What We Should Be Doing On The Server Side, But What About Your Users' Browsers?

- Some recommendations won't exactly be surprising.
- For example, when it comes to dealing with secure sites, just as with pretty much anything else, browsers need to be kept patched up-to-date, including any browser plugins. Encourage use of Secunia PSI/CSI/OSI (see secunia.com) or at least <http://www.mozilla.org/en-US/plugincheck/>
- You may wonder what cert-related stuff needs patching or updating in the browser itself. Well, Browsers include lists of trusted root certificate authorities, and they also include hardcoded blacklists of certificates which are known to be untrustworthy. When a DigiNotar-type incident occurs, browsers need to update those lists. See, for example, Firefox's list of recent security updates...

Firefox's List of Recent Security Issues

Impact key:

- **Critical:** Vulnerability can be used to run attacker code and install software, requiring no user interaction beyond normal browsing.
- **High:** Vulnerability can be used to gather sensitive data from sites in other windows or inject data or code into those sites, requiring no more than normal browsing actions.
- **Moderate:** Vulnerabilities that would otherwise be High or Critical except they only work in uncommon non-default configurations or require the user to perform complicated and/or unlikely steps.
- **Low:** Minor security vulnerabilities such as Denial of Service attacks, minor data leaks, or spoofs. (Undetectable spoofs of SSL indicia would have "High" impact because those are generally used to steal sensitive data intended for other sites.)

Fixed in Firefox 7

- **MFSA 2011-45** Inferring Keystrokes from motion data
- **MFSA 2011-44** Use after free reading OGG headers
- **MFSA 2011-43** loadSubScript unwraps XPCNativeWrapper scope parameter
- **MFSA 2011-42** Potentially exploitable crash in the YARR regular expression library
- **MFSA 2011-41** Potentially exploitable WebGL crashes
- **MFSA 2011-40** Code installation through holding down `Enter`
- **MFSA 2011-39** Defense against multiple Location headers due to CRLF Injection
- **MFSA 2011-36** Miscellaneous memory safety hazards (rv:7.0 / rv:1.9.2.23)

Fixed in Firefox 6.0.2

- **MFSA 2011-35** Additional protection against fraudulent DigiNotar certificates

Fixed in Firefox 6.0.1

- **MFSA 2011-34** Protection against fraudulent DigiNotar certificates

Fixed in Firefox 6

Example of One of Those Specific Advisories



Mozilla Foundation Security Advisory 2011-35

Additional protection against fraudulent DigiNotar certificates

Impact: High
Announced: September 6, 2011
Product: Firefox, Thunderbird, SeaMonkey

Fixed in: Firefox 6.0.2
Firefox Mobile 6.0.2
Firefox 3.6.22
Thunderbird 6.0.2
Thunderbird 3.1.14
SeaMonkey 2.3.3

Description: As more information has come to light about the attack on the DigiNotar Certificate Authority we have improved the protections added in [MFSA 2011-34](#). The main change is to add explicit distrust to the DigiNotar root certificate and several intermediates. Removing the root as in our previous fix meant the certificates could be considered valid if cross-signed by another Certificate Authority. Importantly this list of distrusted certificates includes the "PKIOverheid" (PKIGovernment) intermediates under DigiNotar's control that did not chain to DigiNotar's root and were not previously blocked.

References:

- [Interim Report September 5, 2011: DigiNotar Certificate Authority breach "Operation Black Tulip"](#)
- https://bugzilla.mozilla.org/buglist.cgi?bug_id=683261,683449,683883

Wouldn't OCSP & CRLs Handle Revocation?

- Certificate revocation is supposed to be handled through the use of the Online Certificate Status Protocol (OCSP, RFC2560) and through the use of Certificate Revocation Lists (CRLs, RFC5280) -- online blacklists of revoked certificates.
- However, not all browsers on all operating systems enable OCSP and CRL status checking by default. Even if OCSP and CRL checking is enabled by default, some users may choose to disable it (for perceived speed, perceived privacy, or for no good reason whatsoever). Given that reality, it would be a mistake to assume that all users will always know that a particular certificate should not be relied on.
- Recent browser updates are intended to help make sure that everything possible has been done when a compromise may be potentially particularly disruptive, including hard blacklisting known bad certs inside the browser code itself.

What About Smart Phone/Tablet Browsers?

/www.mobilityfeeds.com/mobility-feed/2011/09/diginotar-bogus-certificates-make-android-and-apple-user:

DigiNotar bogus certificates make Android and Apple users vulnerable to privacy attacks

Eight days after the discovery that a fraudulently issued Web credential [actively targeted Iranians](#) as they accessed their Gmail accounts, millions of people who rely on Google and Apple products remain vulnerable to similar attacks.



The inaction of Google in updating its *Android* Operating System (OS) and Apple in making changes to its *iOS* and *Mac OS X* is even more striking given a report issued Monday that found that a security breach on Dutch firm DigiNotar minted at least 530 additional counterfeit certificates for domains including [addons.mozilla.org](#), [Skype](#), and various Microsoft update sites.

While updates issued over the past week have protected users of the major browsers and Email clients, users of Google *Android*-based devices, *iPhones*, *iPads*, and Apple *Safari* on *Mac* remain susceptible unless they take special precautions.

"Apple is characteristically quiet again when it comes to security and at a time when its users need their help most of all," Andrew Storms, Director of Security Operations at [nCircle](#), wrote in an Email. *"Users are left going the unofficial route looking for experts outside of Apple to tell them how to protect themselves."*

Apple has steadfastly declined to comment on unpatched security vulnerabilities in its products.

Developers of Google's *Chrome* browser have done a good job of communicating the risks users face from the fraudulently issued *DigiNotar* certificates.

In the past week, as additional information has come to light, they have issued two updates designed to prevent the bogus credentials from being accepted by the browser when users encounter them.

Google officials have been considerably more inert when it comes to threats the certificates pose to users of *Android*, the world's most widely used smartphone OS.

A Google spokesman declined to comment for this post.

Android users who want to take security matters into their own hands can install the latest version of [WhisperCore](#), a privacy app that will block most SSL certificates signed by *DigiNotar*.

Detecting Changes in Cert Usage on Firefox

- If you routinely use ssh, you know that if/when a server changes its keys, ssh notices and warns you that something's awry. Paraphrasing: "The credentials you saw last time are not the same as the credentials you're being given this time. Watch out! Someone may be doing a man-in-the-middle attack against you!" That's potentially a very helpful alert.
- A similar process doesn't happen when it comes to web browsers and secure web sites. You might see one certificate today, and a completely different certificate tomorrow, and as long as both are validly signed, your browser won't complain.
- CertificatePatrol is a Firefox browser plugin that helps expose those sort of changes for the https websites you visit.

Mozilla Corporation (US) <https://addons.mozilla.org/en-US/firefox/addon/certificate-patrol/> Google



Certificate Patrol 2.0.12

by [Carlo v. Loesch](#), [Gabor Adam Toth](#), [20after4](#)

Your browser trusts many certification authorities and intermediate sub-authorities quietly, every time you enter an HTTPS web site. This add-on reveals when certificates are updated, so you can ensure it was a legitimate change.

[+ Add to Firefox](#)

★★★★☆
42 user reviews
16,910 users

[Add to collection](#)
[Share this Add-on](#)

Enjoy this add-on?
The developer of this add-on asks that you help support its continued development by making a small contribution.

[Contribute](#)
\$10.00 suggested



About this Add-on

Your web browser trusts a lot of certification authorities and chained sub-authorities, and it does so blindly. "Subordinate or intermediate certification authorities" are a little known device: The root CAs in your browser can delegate permission to issue certificates to an unlimited amount of subordinate CAs (SCA) just by signing their certificate, not by borrowing their precious private key to them. You can even buy yourself such a CA from [GeoTrust](#) or [elsewhere](#).

It is unclear how many intermediate certification authorities really exist, and yet each of them has "god-like power" to impersonate any https web site using a [Man in the Middle](#) (MITM) attack scenario. [Researchers at Princeton](#) are [acknowledging this problem](#) and recommending Certificate Patrol. Revealing the inner workings of [X.509](#) to end users is still deemed too difficult, but only getting familiar with this will really help you get in control. That's why Certificate Patrol gives you insight of what is happening.

[Add-on home page](#)

[Support site](#)

Version 2.0.12 [Info](#)

September 15, 2011

Released under [Mozilla Public License](#), version 1.1

An Example of An Anomaly Detected by CertificatePatrol: The Googleplex, Temporarily Out of Sync?

(apis.google.com)

Info: This certificate has a wildcard for several hostnames.

Old Certification Hierarchy:

- Builtin Object Token:Equifax Secure CA
- Google Internet Authority
- *.google.com

New Certification Hierarchy:

- Builtin Object Token:Equifax Secure CA
- Google Internet Authority
- *.google.com

Issued To:

Common Name (CN): *.google.com
Organization (O): Google Inc
Organizational Unit (OU):
MD5 Fingerprint: - FD:D0:18:00:2C:61:18:F7:9A:F6:D7:D5:8E:CA:29:48
+ EC:F2:0C:91:0D:AA:48:E8:E1:73:6E:8E:C2:AD:C5:FC
- 56:F6:A9:A9:D2:ED:FD:1A:B2:F9:63:7E:D3:51:AC:56:B3:59:A9:8D
+ B3:93:D0:5C:A0:7D:03:45:95:62:EC:18:1A:EA:BD:01:52:84:98:06
SHA1 Fingerprint:
Validity:
 Too many pop-ups? Try checking authority only for this domain

Ignore this website Reject Accept

General Details

This certificate has been verified for the following uses:

- SSL Server Certificate
- Email Signer Certificate
- Email Recipient Certificate

Issued To

Common Name (CN) *.google.com
Organization (O) Google Inc
Organizational Unit (OU) <Not Part Of Certificate>
Serial Number 51:A9:99:AD:00:03:00:00:2E:74

Issued By

Common Name (CN) Google Internet Authority
Organization (O) Google Inc
Organizational Unit (OU) <Not Part Of Certificate>

Validity

Issued On 8/11/11
Expires On 8/11/12

Fingerprints

SHA1 Fingerprint B3:93:D0:5C:A0:7D:03:45:95:62:EC:18:1A:EA:BD:01:52:84:98:06
MD5 Fingerprint EC:F2:0C:91:0D:AA:48:E8:E1:73:6E:8E:C2:AD:C5:FC

Set Nickname...

Close

General Details

This certificate has been verified for the following uses:

- SSL Server Certificate
- Email Signer Certificate
- Email Recipient Certificate

Issued To

Common Name (CN) *.google.com
Organization (O) Google Inc
Organizational Unit (OU) <Not Part Of Certificate>
Serial Number 68:F3:F0:28:00:03:00:00:2F:F6

Issued By

Common Name (CN) Google Internet Authority
Organization (O) Google Inc
Organizational Unit (OU) <Not Part Of Certificate>

Validity

Issued On 9/4/11
Expires On 9/4/12

Fingerprints

SHA1 Fingerprint 56:F6:A9:A9:D2:ED:FD:1A:B2:F9:63:7E:D3:51:AC:56:B3:59:A9:8D
MD5 Fingerprint FD:D0:18:00:2C:61:18:F7:9A:F6:D7:D5:8E:CA:29:48

Set Nickname...

Close

Pruning Browser Root Cert Stores?

- The DigiNotar incident has also made some parties recommend some, um, unconventional strategies, including:

"[...] configuring the enterprise browser platform so as to reduce the number of root CAs the enterprise relies upon. First weed out those root certificates that no one recognizes. [...] Second, weed out those root certificates that are used rarely or not at all. [...] Third, for those CAs that remain, take a few moments to interact with the CAs and determine their practices with respect to RAs and their other affiliates. [continues]"

"From the Experts: SSL Hacked!", Corporate Counsel, Law.com, Sept 28, 2011, tinyurl.com/pruning-root-certs

What's The Big Deal About Having "Lots" of Default Trusted CAs?

- Each and every default-trusted certificate authority can potentially issue a perfectly valid (looking) certificate for any domain. Those valid (looking) certificates can then be used by attackers trying to man-in-the-middle your traffic.
- If you have over a hundred and fifty CAs that you trust by default, people worry that that's "too many," and that one or more of them may in fact be insecure or untrustworthy.
- The "obvious" (if hugely difficult) solution to this problem is to remove the "obscure" or "unneeded" CAs from that default set, as the author on the preceding slide suggests.
- In reality, however, that's a task that's fraught with many problems.

Before Giving That Strategy A Try (If You Do...)

- Be sure you can restore the default trust anchors, just in case you end up removing something that you wish you hadn't.
- Recognize that most people have little or no basis for recognizing or assessing trust anchors for retention or potential removal decisions. You might try saying, for example, "I'm only going to keep big American CAs," but you might be surprised at how many commonly used/critical web sites use certs from less common overseas CAs.
- If you bump into a site of that sort after you've pruned the trust anchor that would normally validate it, you (or your users!) will then need to exercise your own best judgment: is this a cert I want to permit, or not? Absent extensive personal investigation, mistakes will inevitably be made, and users will end up accepting certs they shouldn't be accepting and rejecting certs that they shouldn't be rejecting. Ugh!

If You Feel You *Must* Prune Trust Anchors

- One strategy that **might** work would be to compare the trust anchors recognized by major operating systems and applications, keeping only those that are common to all members of that reference set. Put another way, if ALL common operating systems and browsers trust a particular CA, you might decide you might as well do so, too.
- However, if you do that, what's your plan for keeping that set of local trust anchors current over time? Do you really want to make trust anchor maintenance your regular hobby?
- You also need to figure out what you're going to do if a trust anchor that you're nervous about has intermediate certs that are cross-certified by a trust anchor you do like... this may be more complex than you might think!
- My recommendation? PLEASE **resist** the urge to manually tweak the default operating system/browser trust anchors!⁷²

Certificate Stapling

- As mentioned on the preceding slides, currently any CA you trust can issue a seemingly valid certificate on behalf of any domain. Wouldn't it be swell if sites could specify that their site will always and only use certs from one vendor, and that any cert that might be seen from some other vendor should NEVER be trusted for their site?
- *The Good News?* This is precisely one of the use cases described by the DANE effort in the IETF. See Section 3.1 of <http://tools.ietf.org/html/draft-ietf-dane-use-cases-05>
- *The Bad News?* The DANE work relies on deployment of DNSSEC (which is only beginning in many parts of the net).
- At the risk of asking you to check and potentially work on Yet Another Thing, how **IS** deployment of DNSSEC coming at your campus? (Yes, this stuff really does all interlock nicely, doesn't it?) If you're not looking at DNSSEC, you should be

Moxie Marlinspike's "Convergence"

- You may also hear about Convergence, a Firefox add-in that uses a network of "notaries" who collectively and anonymously tell you if what you're seeing for a site's certificate is consistent with what they've seen for a site's certificate. (see <http://convergence.io/index.html>)
- This sort of check is potentially helpful if your worry is that someone may be trying to do a local "man-in-the-middle" attack against you. If they attempt to do this, the cert you'll see will differ from the cert that the rest of the world will see, and Convergence will hopefully alert you to that.
- Some in the community have expressed concerns about the ultimate scalability of the Convergence model; see, for example: "Why Not Convergence?"
<http://www.imperialviolet.org/2011/09/07/convergence.html>
- For now, I suggest that sites just test/evaluate Convergence.

Browser Exploit Against SSL/TLS Tool (BEAST)

- The technical media has been all atwitter recently about BEAST, a browser-based attack exploiting long-known (heretofore theoretical) vulnerabilities that exist in widely deployed and routinely used versions of SSL/TLS.
- Unfortunately, the community still hasn't really converged around a practically workable solution to this vulnerability yet.
- One of the nicest summaries I've seen of what browser vendors are thinking about is: "Browsers Tackle the 'BEAST' Web Security Problem," September 29th, 2011, http://news.cnet.com/8301-27080_3-20113530-245/browsers-tackle-the-beast-web-security-problem/ (or try <http://tinyurl.com/beast-summary> if you prefer).
- For now, I think the best advice I can give you on this one is to continue to monitor this vulnerability.

Browser Recommendation Recap

- Be sure users keep their browsers patched up to date
- OCSP and CRL checking should be enabled, and should remain enabled, imperfect as those protocols may be.
- If you use Firefox, run CertificatePatrol
- Think twice before messing around with the default trust anchor set that present in your operating system or browser
- Try to find at least a little time to work on DNSSEC at your site, and learn about what's going on in the IETF DANE working group
- If interested, experiment with Convergence (however I would NOT recommend abandoning traditional CA-based models in favor of SSL notary-based approaches at this point in time)
- Monitor ongoing work relating to effectively countering the vulnerabilities targeted by BEAST

Shifting Gears...

VII. Client Certificates

*Also Known As "Personal Certificates," "S/MIME Certificates,"
"PKI Certificates," "PKCS#7 Certificates," etc., etc., etc.*

Many Of You May NOT Be Familiar with Client ("Personal") Certificates

- Normally, when people think of certificates, they think of SSL web server certs, such as the certs we've been discussing up until this point. Client certs are different: they get issued to people, not servers.
- Client certs bind an individual's online identity to cryptographic keys controlled by that individual.
- Client certs are most often used with S/MIME to help make email more secure, making it easy for users to sign and encrypt the body of the mail messages they may send.
- Client certs can also potentially be used as a "2nd factor" for authentication purposes, augmenting or supplanting traditional passwords for securing online resources such as confidential web content, and for other purposes.

Client Certificate Are *Not* Something New

- Even though you may never have heard of client certs, they're not new. Much of the foundation work underlying client certs can be found in the RSA PKCS ("Public Key Cryptography Standards") documents, which date to 1993 – that's eighteen years go!
- So why aren't people seeing client certs in widespread deployment?
- Well, first of all, even if client certs aren't particularly common in higher education, client certs ARE being broadly deployed in at least some other environments, such as in the federal government. There, you find them included as part of "HSPD-12" "CAC" and "PIV" cards...

HSPD-12 and Federal CAC/PIV-I Cards

- On August 27th, 2004, then-President George W. Bush issued "Homeland Security Presidential Directive 12," (see <http://www.idmanagement.gov/documents/HSPD-12.htm>) mandating the establishment of a common identity standard for federal employees and contractors.
- As a result, the federal government (and approved commercial contractors acting on the government's behalf) have already collectively issued millions of "Common Access Cards" ("CACs") and "Personal Identity Verification-Interoperable" ("PIV-I") smart cards.
- "First responders" alone (as defined in HSPD-8) may ultimately require issuance of over 25.3 million such cards. (see http://www.dhs.gov/xlibrary/assets/Partnership_Program_Benefits_Tax_Payers_Public_and_Private_Sector.pdf)
- That is ***NOT*** a toy-scale cert project by any means!

CURRENT STATUS – HSPD-12

- *HSPD-12 Credentials Issued as of June 1, 2011:*
Credentials issued to Employees: **4,151,358 (88%)**
Credentials issued to Contractors: **842,946 (81%)**
(Total credentials issued: 4,994,304 (87%))
- *Background Investigations Verified/Completed as of June 1, 2011:*
Background investigations completed for Employees: **4,128,415 (87%)**
Background investigations completed for Contractors: **886,137 (85%)**
(Total investigations verified/completed: 5,014,552 (86%))
- 18 federal credential issuance infrastructures are in operation nationwide
- 59 system integrators and 592 products on GSA Approved Products and Services List

Agency specific status may be located at:
http://www.whitehouse.gov/omb/e-gov/hspd12_reports/

* US Military Personnel are included in Employee Numbers

Source: http://www.idmanagement.gov/presentations/HSPD12_Current_Status.pdf

Why Are The Feds Using Client Certs? If You Need LOA-4, They're Basically Your Only Practical Option

- NIST 800-63 Version 1.0.2 (see csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf) says:

"Level 4 – Level 4 is intended to provide the highest practical remote network authentication assurance. Level 4 authentication is based on proof of possession of a key through a cryptographic protocol. Level 4 is similar to Level 3 except that only “hard” cryptographic tokens are allowed, FIPS 140-2 cryptographic module validation requirements are strengthened, and subsequent critical data transfers must be authenticated via a key bound to the authentication process. The token shall be a hardware cryptographic module validated at FIPS 140-2 Level 2 or higher overall with at least FIPS 140-2 Level 3 physical security. By requiring a physical token, which cannot readily be copied and since FIPS 140-2 requires operator authentication at Level 2 and higher, this level ensures good, two factor remote authentication."

Some Federal High Security Applications That Use Client Certs May Be Surprising

Controlled Substance Ordering System Homepage
http://www.deacom.gov/ordering.html

Drug Enforcement Administration | Office of Diversion Control
E-Commerce Program

About Electronic Ordering

Electronic controlled substance orders are placed using a software program that has been approved for CSOS. Typically, this software is available through a wholesaler and may be implemented into their ordering Web site. This software includes functionality to digitally sign the purchase order using the purchaser's CSOS digital certificate issued by DEA. A CSOS Certificate may be installed into multiple software programs and may also be transferred to multiple ordering computers.

The diagram illustrates the CSOS E-commerce process flow. At the top is the U.S. Department of Justice Drug Enforcement Administration logo. Below it, a central yellow oval labeled 'CSOS Enabled Ordering Software' contains two blue boxes: 'Pharmacy/Purchaser' on the left and 'Wholesaler/Supplier' on the right. Five numbered steps are shown with arrows: 1. Enrollment (from Pharmacy/Purchaser to DEA), 2. Digitally Signed Order (from Pharmacy/Purchaser to Wholesaler/Supplier), 3. Certificate Validation (from Wholesaler/Supplier to DEA), 4. Controlled Substance(s) Supplied (from Wholesaler/Supplier to Pharmacy/Purchaser), and 5. Transaction Reported (from Wholesaler/Supplier to DEA).

1. An individual enrolls with DEA and, once approved, is issued a personal CSOS Certificate.
2. The purchaser creates an electronic 222 order using an approved ordering software. The order is digitally signed using the purchaser's personal CSOS Certificate and then transmitted to the suppliers. The paper 222 is not required for electronic ordering.
3. The supplier receives the purchase order and verifies that the purchaser's certificate is valid with DEA. Additionally, the supplier validates the electronic order information just like it would a paper order.
4. The supplier completes the order and ships to the purchaser. Any communications regarding the order are sent electronically.
5. The order is reported by the supplier to DEA within two business days, .

Client Certs Can Even Be Secure Enough for Use in Conjunction with National Security Systems

- See the "National Policy for Public Key Infrastructure in National Security Systems," March 2009 (<http://www.cnss.gov/Assets/pdf/CNSSP-25.pdf>) makes it clear that client certs even form the foundation for NSS uses:
 - "(U) NSS operating at the unclassified level shall obtain PKI support from the established Federal PKI Architecture.
 - "(U) NSS operating at the Secret level shall obtain PKI support from the NSS-PKI.
 - "(U) The NSS-PKI hierarchy shall rest on a Root Certificate Authority (CA) operated on behalf of the national security community in accordance with policies established by the CNSS PKI Member Governing Body. The NSS-PKI Root CA shall serve as the anchor of trust for the NSS-PKI."
- TS/SCI ("JWICS") counterpart of the NSS-PKI? IC-PKI.

Note, However: NOT All Client Certs ARE The Same

- Imagine two dramatically different "sorts" of client certs:
 - One sort might be issued to anyone who asks for it, perhaps based on a user-completed web form and a confirmatory email message sent to a user-supplied email address. This personal cert might be stored in the user's browser and/or operating system, used without requiring a PIN for access, and be easily exported and re-imported onto other devices. This is a *lower security* client cert.
 - Another type of client cert might require use of a tamper resistant physical hardware token or smartcard with a PIN. In this approach, keys are generated on the token or smartcard, and in a non-exportable way. The cert that's associated with those keys only gets issued to the user in person, after "identity proofing" the user. Escrow policies may apply, too. *This* is a higher security client cert.

What Does Your Site Need?

- Do you want a relatively casual credential that can be used on an ad hoc/experimental basis to potentially improve the security of campus email messages, or perhaps to control access to internal web sites?
- Or do you need a carefully controlled credential for very serious purposes, where a breach could have expensive consequences? Granted, your school is not part of the military, or part of the intelligence community, or a law enforcement agency, or a pharmacy ordering DEA-controlled narcotics, but at the same time, you may want or need to take special care to secure access to sensitive campus systems.
- For example, what if you needed to secure access to your school's core routers and switches, or privileged accounts on your school's most sensitive administrative systems, or your site's authentication (or backup server) infrastructure?

Are There Specific Authentication Issues You're Facing?

- Do you worry that regular passwords may be easily guessed, or targeted for brute force attacks?
- Are clear text passwords getting sniffed?
- Are users being socially engineered into disclosing their passwords in response to phishing scams?
- Is crimeware capturing passwords as they're being entered on the keyboard, or is it extracting passwords from integrated password stores where users may have saved them in apps?
- Are users are using the same password at secure on-campus sites as well as at other insecure off-campus ones?
- Are users sharing their passwords with friends or relatives?
- Do users just have too many passwords to remember? Or are they fed up with increasingly long/stringent password requirements, periodic required password changes, etc.?
- Are auditors pressuring you for a more secure solution?

Next Question: What Can Your Site Afford?

- While client certificates are available at no incremental cost as part of the InCommon certificate program, *deploying* client certificates may still have associated costs. For example:
 - If you decide to use USB-format hard PKI tokens and/or smart cards, you will need to budget for their cost (and for any required driver software or mandatory vendor support).
 - If you decide to use smart cards, you'll need card readers
 - You may need a token provisioning/management system
 - You may need a central directory to store credentials
 - You may need a locally-written or locally-tailored installer to handle installing certs and/or middleware on user systems.
 - Systems will need to be tailored to permit cert authentication
 - If you do careful vetting of your user's identity at credential issuance time, you'll need one or more employees to do that.
 - Users will need documentation, training, and support

Have You Considered Potential Alternatives?

- For example, if you're planning to use client certs for S/MIME, have you also considered using PGP/Gnu Privacy Guard? If someone were to ask, "Why are you going to use S/MIME with client certs instead of PGP/GPG," what would you say?
- If you're planning on using client certs as a 2nd factor authentication solution, have you also considered, instead:
 - using free ssh private key-based authentication?
 - using a 2nd channel (typically smart phone based) solution? (see for example <http://motp.sourceforge.net/> or <http://code.google.com/p/google-authenticator/> or even commercial options such as <http://www.duosecurity.com/> (free for up to ten users, and for open source projects))
 - using inexpensive hardware one time password tokens? (\$5 tokens: <http://www.entrust.com/savings/index.cfm>)
 - or what about biometric methods, instead?

Get A Little Experience, First

- It's sometimes tempting to "swing for the bleachers," trying to hit a grand slam the first time you're up to bat, when in fact the prudent thing might be to make sure you just get on base. This is true for client certs, as for baseball.
- I'd like to urge you, before you embark on a major all-campus project involving client certs, or even a pilot scale project that might involve some of your most sensitive systems, to first spend a little time just experimenting with client certs.
- Get free client certs for yourself, and for your team members.
- Use them for relatively low impact activities, such as signing your email, while you gain familiarity with them.
- Try purchasing and using hardware tokens or smart cards. What works? What doesn't work on your devices or in your environment? In an experimental environment, you've got the freedom to push the envelope without worrying *too* much.

Does This Mean "Joe Says Don't Do Client Certs?"

- NO! Emphatically no! We WANT you to get experience with client certs. PLEASE try client certs! Client certs are a terrifically powerful technology that can potentially transform your online world. PLEASE PLEASE PLEASE try them!
- However, we don't want you to try them the wrong way, have a bad experience, and then end up forswearing them forever.
- Please start slowly. Begin by obtaining a client certificate and using it for S/MIME email message signing and encryption as an initial experiment.
- See the following section...

VIII. Using A Client Cert To Sign and Encrypt Email On A Mac With Thunderbird

*Signing and Encrypting Email Is
A Classic Client Cert Use Case*

Scenario: Email Is Critical, But Insecure

- Professor Anderson is administering a large distributed science experiment, while also continuing to teach both a large introductory lecture class and a graduate seminar in her department. She also serves on the Promotion and Tenure Committee. In doing that work, Anderson relies on email for things such as:
 - Approving requests for time on the project's facilities and purchase requests
 - Assigning readings for her classes and consulting with grad students
 - Discussing discoveries and new inventions with fellow faculty members
 - Exchanging personnel files and personnel assessments.

Anderson is concerned that her regular email isn't really sufficiently secure for some of these purposes. She worries that someone could "forge" email, and while pretending to be her, maliciously try to cancel one of her classes or approve an unauthorized purchase. She also worries about transferring confidential personnel-related materials, or sharing information about new patentable inventions, by unencrypted email. Professor Anderson wished there was some way she could digitally sign her email, to prove it really came from her, and some way that she could encrypt sensitive information she needs to send or receive.

Suggested Solution: S/MIME With Client Certs

- Professor Anderson and her correspondents can use S/MIME signed and encrypted email to exchange messages that are protected from forgery and eavesdropping.
- What's minimally needed:

Email Software Supporting S/MIME: most popular email solutions (Thunderbird, Outlook, Apple Mail, Gmail, etc.) either directly support S/MIME, or can be made to support S/MIME through use of a third party add-on. Note that BOTH sender and receiver need to be using an email client that supports S/MIME if the parties want to be able to exchange encrypted messages, or validate signed messages.

A Client Certificate: In order to be able to digitally sign her mail, and to be able to exchange encrypted content, Professor Anderson needs to obtain a client certificate. Client certificates are available at no charge, and typically require only the completion of an online web form, and then clicking on a link sent to the user's email address.

Let's walk through this process, assuming use of Thunderbird on a Mac.

Getting A Client Cert

- If your university is a subscriber to the InCommon Certificate Service, your university's cert administrator or registration authority should be able to issue you an invitation for one.
- Even if your campus isn't doing client certs (yet) or your campus isn't a subscriber to the Certificate Service, you can still get a one year personal certificate for test/trial use from: <http://www.instantssl.com/ssl-certificate-products/free-email-certificate.html>
- Use Firefox to collect your certificate by following the instructions that will get sent to your email address from the CA. When you collect your certificate from the CA, it will automatically be installed in Firefox on the system you're using at that time. ***NOTE: Do NOT accidentally collect and install your personal cert on a shared system (such as a machine in a computer lab or cyber cafe)!***

Exporting Your New Cert From Firefox (You'll Only Need To Do This Once)

- Once you've collected your certificate, you need to export it from Firefox so you can import it into Thunderbird.
- Assuming you're using Firefox on a Mac, go to Firefox --> Preferences --> Advanced --> Encryption --> View Certificates --> Your Certificates.
- Highlight your certificate by clicking on it, then click Backup. Save the file as a PKCS12 file with a name of your choice, such as myprivatekey.p12 You will need to provide a strong password for this file because it contains your private key.
- ***Note: Never share that file! Save a copy of that file on a CD or thumb drive somewhere safe, such as in your safety deposit box. Don't forget to also save a copy of the password you'd need to access the contents of that file!***

Install Thunderbird (If You Don't Already Have It)

- If you don't already have Thunderbird, install a copy of it from <http://www.mozilla.org/en-US/thunderbird/>
- Configure Thunderbird to use your email account:
 - In Thunderbird, go to Thunderbird --> File --> New --> Mail Account...
 - You'll need to know:
 - Your email address and password
 - The name of your outgoing mail server (this is often known as the SMTP server)
 - The name of your incoming mail server (this is often known as your IMAP server).
- ***Caution:*** do not use POP (instead of IMAP) unless specifically advised by local support staff to do so. POP will download your email onto your local workstation, and may delete it from the server unless you tell Thunderbird to "leave mail on server." If you need help, check with your IT help desk.

Importing Your Cert Into Thunderbird (You'll Only Need To Do This Once)

- In order for Thunderbird to be able to use the cert you exported from Firefox, you'll need to import it.
- In Thunderbird, go to Thunderbird --> Preferences --> Advanced --> Certificates --> View Certificates --> Your Certificates. Click Import.
- Select the S/MIME private key you exported on the preceding page. You'll need to provide the password you used when saving that file.

Choose The Certificate You Want To Use For Your Email (You'll Only Need To Do This Once)

- You'll only have one certificate at this point, but you still need to "select" it for Thunderbird to use it.
- In Thunderbird, go to Thunderbird --> Tools --> Account Settings. In the left hand column, click on your account. Click View Settings For This Account. Click Security. In the Digital Signing pane, click Select. Select the certificate you imported. You will also be asked if you would like to use the same certificate for encrypting and decrypting messages sent to you. Click Yes. Click Okay to close the window.
- You're now ready to try sending a signed message.

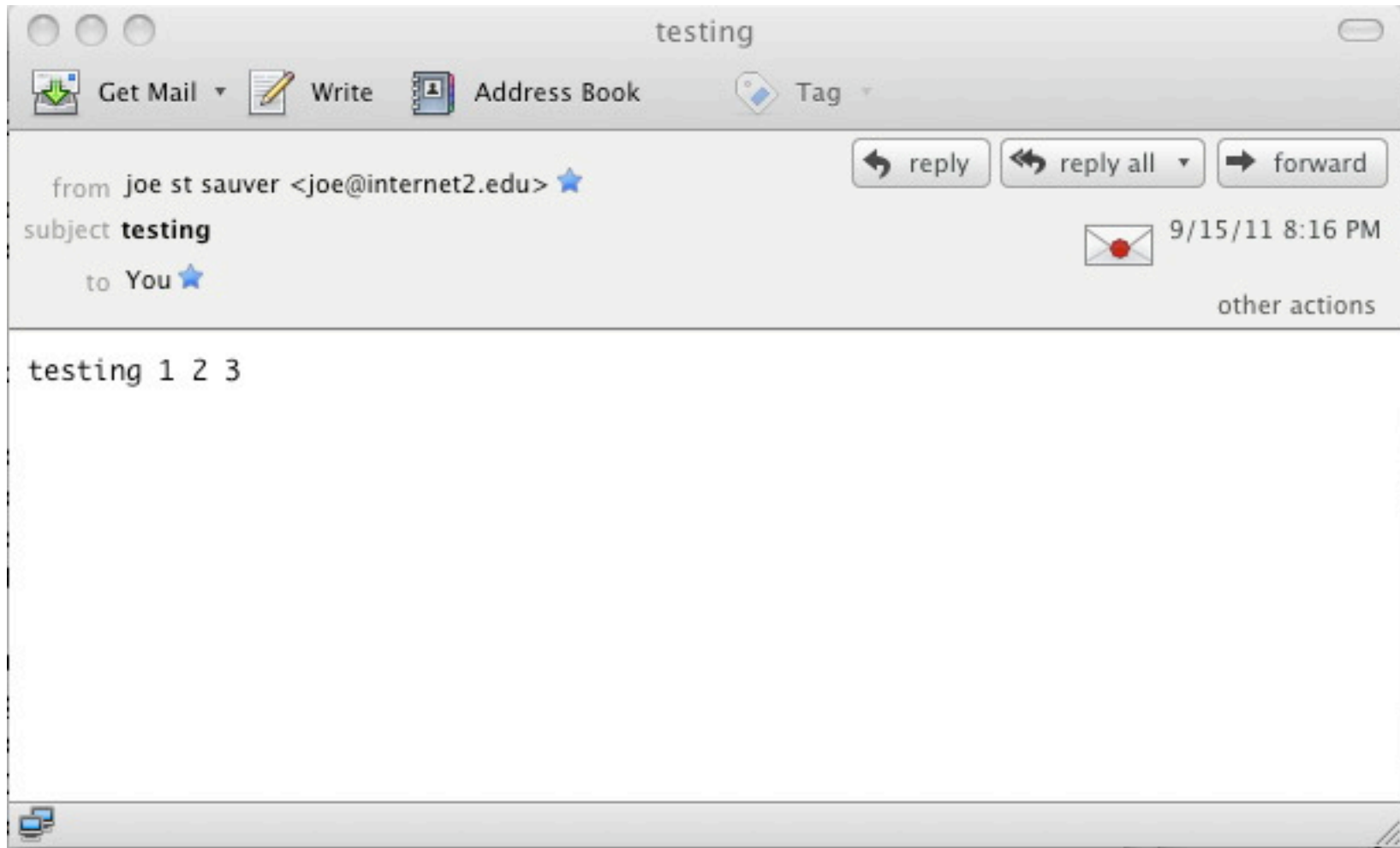
Sending A Signed Message in Thunderbird

- 1. *Begin Writing a Message As You Normally Would in Thunderbird.*** In Thunderbird, go to Thunderbird --> File --> New Message, and create a new test email message to a friend or colleague. Once you've finished writing it, do NOT send it yet -- you still need to cryptographically sign it.
- 2. *Digitally Sign The Message.*** In Thunderbird, go to Thunderbird --> Options --> Digitally Sign This Message.
- 3. *[Optional] Check That This Message Will Be Sent Digitally Signed.*** In Thunderbird, in the window where you're composing your message, click on the yellow security padlock and confirm that "The contents of your message will be sent as digitally signed" says "Yes." Click Okay.
- 4. *Send the Message.*** In Thunderbird, where you're composing your message, click Send.

A Couple of Notes About Message Signing

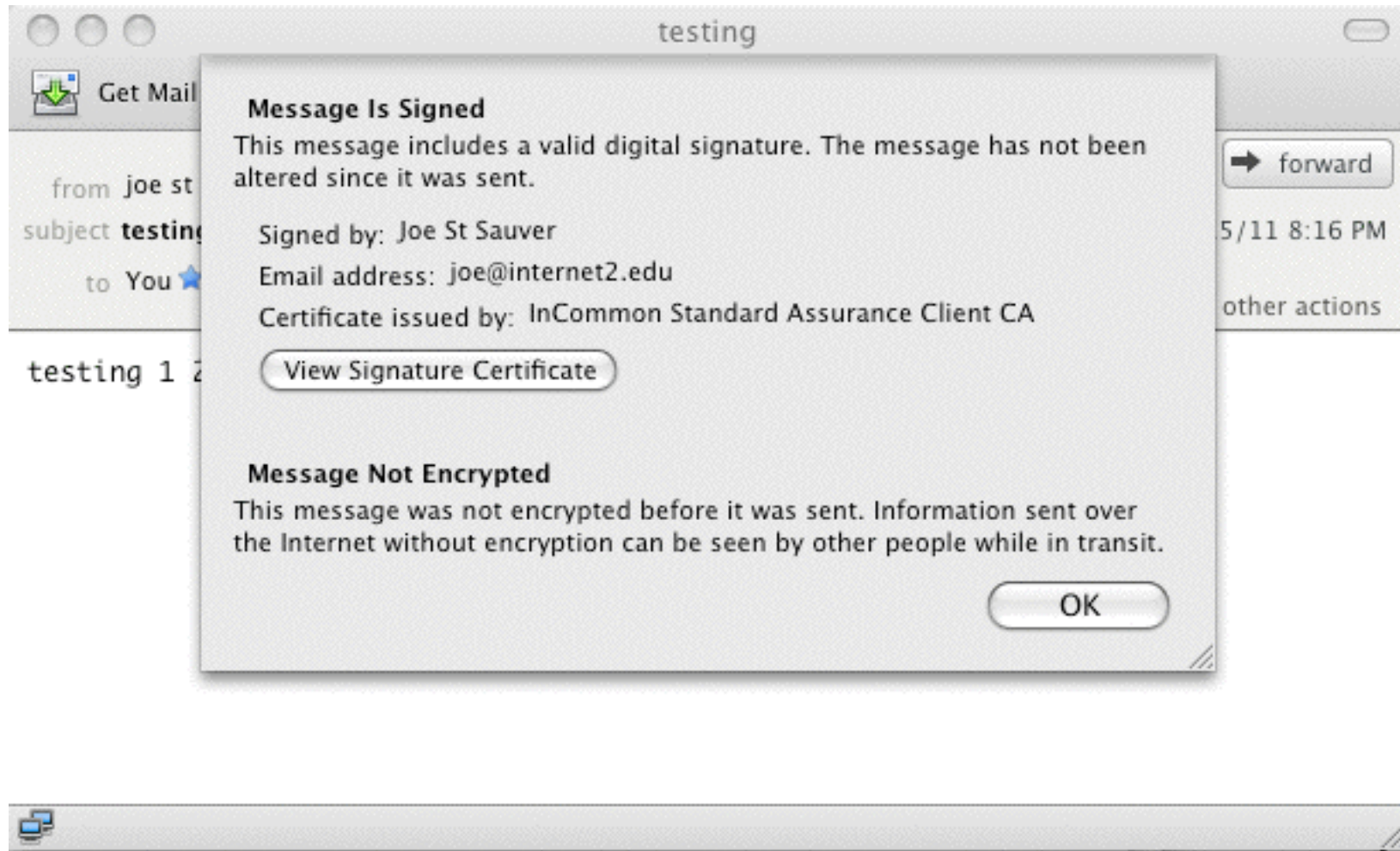
- **Note #1:** When your friend or colleague receives your cryptographically signed message, they'll get the body of the message as they normally would, but they will **also** receive an additional S/MIME signature. If they are S/MIME enabled, their email client will automatically check and confirm your message's cryptographic signature, and show a little red seal. If they're not S/MIME enabled, they can simply ignore the S/MIME signature (it will be an attachment called smime.p7s).
- **Note #2:** When your correspondent receives your S/MIME signed message, they will automatically get your S/MIME public key. At that point, if they are S/MIME enabled, they will be able to send encrypted mail to you. If you would like to send encrypted mail to them, have them begin by sending you a signed email so you'll automatically get their S/MIME public key.

What Does A Signed Message Look Like In Thunderbird?



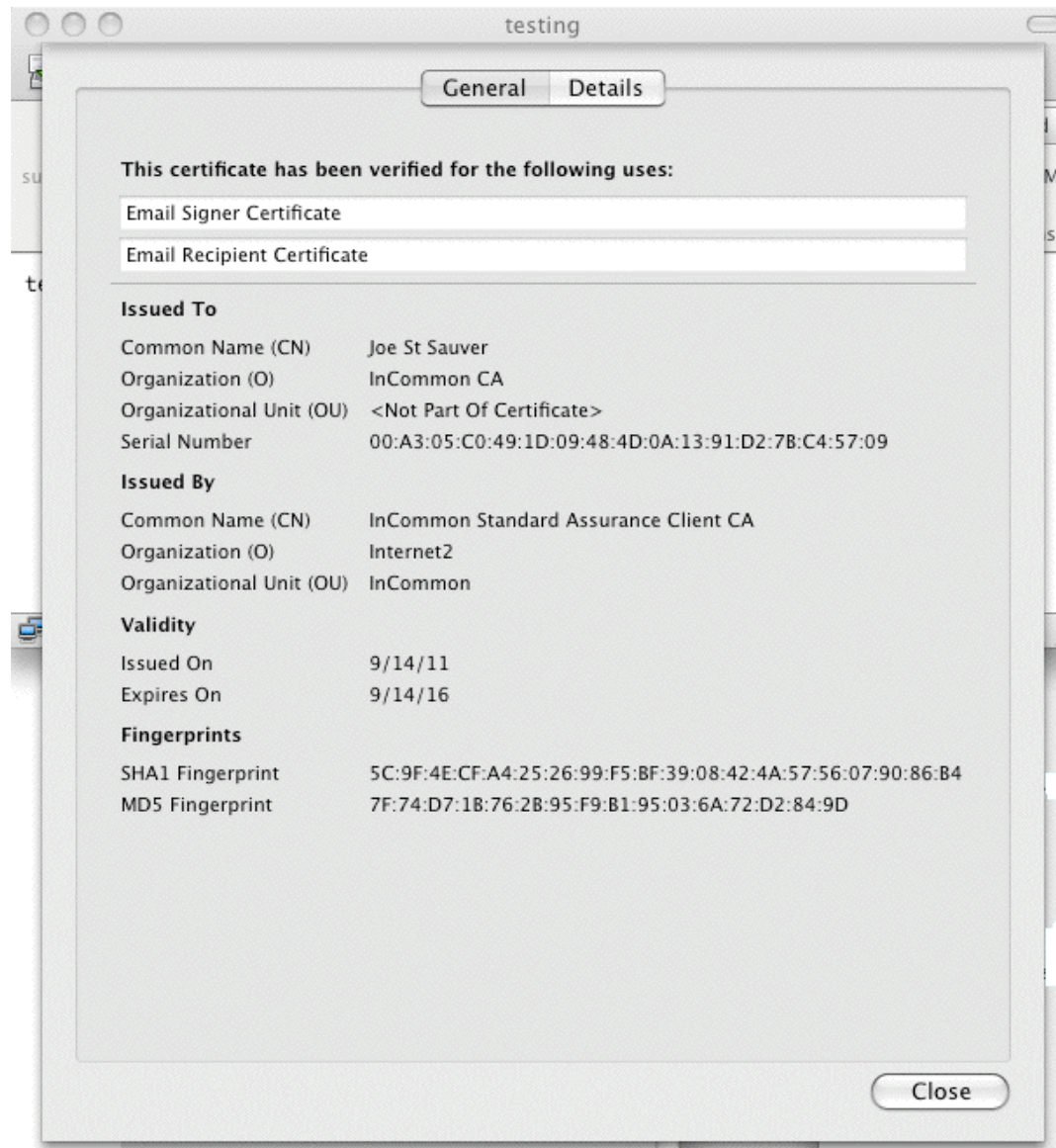
Note the little red seal on the "envelope"...

If You Click on That Little Envelope



Remember: only the message BODY gets signed...

If You Ask To View The Signature Certificate



Remember: Only The Message Body Gets Signed!

- Sometimes users incorrectly assume that the "whole" message gets signed. It doesn't. Things like the message Subject:, Date:, and From: field CAN be modified and the message WILL still have a valid signature.
- Therefore, do NOT rely on what you see in the message headers, including the apparent sender, the listed subject, or the apparent time the message was sent.
- **ONLY THE MESSAGE BODY** gets signed, and hence, that's the **ONLY THING** that's protected against tampering or other manipulation.

B. Encrypting A Message With Thunderbird

- Once you've received at least one signed message from your correspondent, you'll then be able to send them an encrypted message...
- **1. Begin Writing a Message As You Normally Would In Thunderbird.** In Thunderbird, go to Thunderbird --> File --> New Message, and create a new test email message to a friend or colleague. Once you've finished writing it, do NOT send it yet -- you still need to encrypt it.
- **2. Encrypt The Message.** In Thunderbird, go to Thunderbird --> Options --> Encrypt This Message. (If you like, you can also optionally click Digitally Sign This Message).

Encrypting A Message (Cont.)

- **3. [Recommended]** In Thunderbird, in the window where you're composing your message, click on the yellow security padlock and confirm that "The contents of your message will be sent encrypted" says "Yes." Click Okay.
- **4. Send the Message.** In Thunderbird, in the window where you're composing your message, click Send.
- **Some Potentially Important Caveats:**
 - We recommend that you always double check that the message you think is being sent encrypted IS encrypted (see step 3, earlier on this slide)
 - Only the body of the message (and not the To:, From:, Date:, or Subject: of the message) will get encrypted.
 - Any malware scanning your site might normally do, will NOT be able to be done to S/MIME encrypted messages!
 - Lose/forget your key? You won't be able to decrypt...

“What If I Use Gmail’s Web Interface?”

- Gmail works with Thunderbird over IMAP as well as via its native web graphical user interface. You can access your free Gmail web email account using Thunderbird, signing and encrypting messages as previously described.
- However, if you don’t want to forgo the Gmail web interface, you can also try installing a free-for-personal-Gmail-accounts Firefox plugin called Penango that will enable you to sign and encrypt mail from within Gmail’s web interface. See <http://www.penango.com/>

You should be aware that while Penango appears to work within the web email interface, it actually injects your signed or encrypted mail to Gmail separately, alongside the web email session, not from within it.

Encryption and Key Escrow vs. Signing and Non-Repudiation

- By state law (or local policy), some users may be required to escrow their encryption keys so that if they are unable to decrypt a critical encrypted message or document, for whatever reason (hit by a bus?), it can still be recovered.
- Those same organizations may simultaneously require that signing keys be strictly non-repudiable, so that a digital signature, once made, cannot be subsequently disavowed.
- These seemingly contradictory requirements may make the process of issuing and sharing keys more difficult (for example, normally a user gets the key needed to encrypt traffic to a user via a signed message, but in this case the keys used for signing are not the same as the keys used for encryption). Multiple client certs are thus sometimes needed, adding to the likelihood that confusion may result.

Disambiguating Multiple Certs

- Client certs get associated with the user's email address. How, then, do you issue both an escrowed encryption cert AND a non-repudiable signing cert using the same address?
- The InCommon Certificate program allows more than one key to be issued to the same email address. However, to avoid confusion, some sites are trying "+ addressing." For example, a user might have certs with the email addresses:

-- jsmith+signing@example.edu

-- jsmith+encryption@example.edu

Mail to either of those addresses, or mail to just plain jsmith@example.edu , will all get automatically delivered to the same account at most sites, just as if the +whatever part of the address didn't exist.

Sharing Public Keys On An Ad Hoc Basis

- In cases where the normal pair wise key exchange mechanism won't work (such as situations where different certs must be used for encryption and for signing), you need some way for users to share their public keys.
- In an enterprise, this is often done via an enterprise directory.
- On the Internet, however, attempts to create a giant central global S/MIME directory that lists "everyone", or even just a directory of directories, have never really worked out.
- To address this problem, I've written and am currently testing a simple web-based public key server for S/MIME that's superficially modeled on the key servers that are routinely used for sharing PGP public keys. If you're a member of InCommon or Internet2 and you'd like to try this simple key server at your site, I'd be happy to share that beta code with you, provided you send along feedback/suggestions.

IX. Client Certs and Coping With Multiple Devices Per User, Users in Labs on Shared Systems, Users With Mobile Devices, Etc.

While It's Convenient To Assume That Client Certs Can Just Live In A User's Dedicated Workstation, That's Not How Most of The World Looks/Works These Days

“What If I Use Multiple Systems?”

- If you use one system at work and another one at home, one option would be to export your cert from your main system, carrying it home on a USB key or CD, and then reimporting it into your system there.
- You could try using different certs on each device, but that's a mess if you need to decrypt content encrypted for one of your systems while actually using a different one.
- A more secure option would be to get a PKI hard token or smart card, saving your cert onto that device rather than saving your cert into your computer's operating system or browser. Smartcards are shaped like a credit card or ID card, but include a cryptographic processor as well as just memory. They connect to the computer via an external card reader.
- PKI hard tokens look like USB thumb drives, but the device actually includes the “brains” of a smart card along with an integrated card reader with a USB interface.

Should I Pick A Hard Token or Smart Card?

- Which of the two formats might work best for you?
 - Smart cards may be a nice option if you need an integrated cert storage device AND a physical ID card.
 - PKI hard tokens, however, eliminate the need for card readers, and can be obtained in a format integrating both a PKI hard token AND a one time password (OTP) capability.
- What's your budget per token (or per smart card and card reader)? Hard tokens or smartcards (with readers) may cost ~\$50 or more each, plus required software and support.
- How many tokens or smart cards are you going to deploy? Deployment processes that might work fine for relatively small projects may not scale to campus-wide deployments.
- Does it matter if you'll need to install software drivers?
- Do you need to support Macs as well as Windows systems?
- Do you also need to support tablets or smartphones?

The Mac Problem

- Pretty much every smartcard or hard token will work with current versions of Microsoft Windows – that's the easy part. Finding smartcards or hard tokens that will also work with Macs may be harder, but critically important in higher ed.
- One example of a hard token that will usually work with Macs is the Safenet (formerly Aladdin) eToken Pro. (There are multiple Safenet products with similar names, but subtly different capabilities and levels of compatibility, so be sure to confirm that you are ordering a Mac-compatible product that also supports 2048 bit client certs)
- eToken Pro tokens are available from a variety of resellers, including the big guys such as CDW. Don't forget to also buy the required software and support package for your tokens.
- The best source for info on smart cards on the Mac is probably <http://militarycac.com> While its focus is officially on CAC cards on the Mac, much of its info is generally helpful.

Slick-Sided Mobile Devices and Hard Tokens

- Setting aside Mac challenges for a second, you want to also think about how you'll integrate hard tokens or smart cards with mobile devices that your users may have, such as the iPad, the iPhone, Android devices, Blackberries, etc.
- The problem is that most hard tokens, and most smart card readers for that matter, connect via USB. Some portable devices may not have a readily accessible USB port into which you can plug a hard token or smart card reader.
- The solution? You can buy so-called Bluetooth smartcard readers (sometimes also known as "CAC sleds") to allow BlackBerries or selected other mobile devices to access smart cards via secure Bluetooth, but they may cost \$200+. See www.apriva.com/products/iss/authentication/reader
- Android? iPhone? See <http://www.biometricassociates.com/products-baimobile/smart-card-reader-iphone-android.html>

X. What Else Can I Do With Client Certs, Besides S/MIME Signed and Encrypted Mail?

*There Are Many, Many, Many Potential
Client Cert Use Cases*

Signing Stuff (Other Than Just Using S/MIME)

- Client certs can do lots more, including signing documents...
- Signing **Microsoft Word documents** (Windows only), see <http://pages.uoregon.edu/joe/signing-a-word-document/>
- Need to sign documents on a Mac? Try **OpenOffice**: <http://tinyurl.com/openoffice-signing>
- Adobe has an extensive guide to securing PDFs, including use of digital certificates for **signing PDFs**, see: <http://tinyurl.com/adobe-signing>

Encryption Using Client Certs (Other Than S/MIME)

- **PGP Whole Disk Encryption** (see the datasheet linked from <http://www.symantec.com/business/whole-disk-encryption>)
- **Microsoft Windows Encrypted File System**
<http://technet.microsoft.com/en-us/library/bb457116.aspx>
- **IPsec VPNs** (Most IPsec VPNs are deployed without use of client certificates, however at least some VPNs can be configured to use client certificates if desired — see, for example, <http://www.strongswan.org/> and <http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/DCertPKI.html>)

Authentication Using Smart Cards/Client Certs

- **RedHat Enterprise Linux** Smart Card Login
See <http://tinyurl.com/redhat-smartcards>
- **Windows Active Directory** Login with Smart Cards
See <http://support.microsoft.com/kb/281245>
- **OpenSSH authentication** (via third party X.509 patches)
<http://roumenpetrov.info/openssh/>
- **Mac OS X** has deprecated native support for smart cards, but third party providers do still offer support, see <http://smartcardservices.macosforge.org/> and <http://www.thursby.com/mac-enterprise-management-high-security-smart-cards.html>

Authentication Using Client Certs (cont.)

- Controlling access to web content served by **Apache**
http://httpd.apache.org/docs/2.0/ssl/ssl_howto.html#allclients
- Controlling access to web content served by **Microsoft IIS7**
<http://technet.microsoft.com/en-us/library/cc732996%28v=ws.10%29.aspx>
- Controlling access to **wireless networks** via EAP-TLS, including configuring **Eduroam**. See

http://www.cisco.com/en/US/tech/tk722/tk809/technologies_white_paper09186a008009256b.shtml and

<http://www.internet2.edu/presentations/jt2011summer/20110710-hagley-eduroamtutorial.pdf>

Client Certificates Can Even Potentially Be Used For Building Access Control Purposes

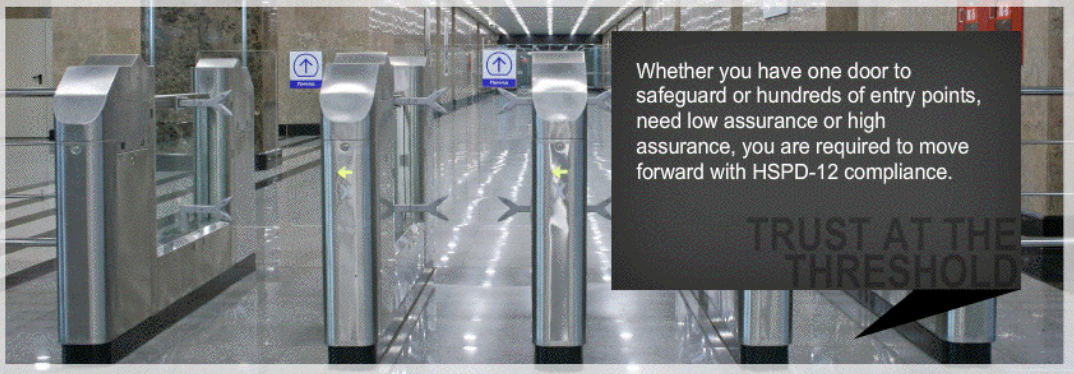
CAC Card Readers | PIV Card Readers | Single Door Access Solutions for Military and High Security applications

http://www.bridgepointsystems.com/

BRIDGEPOINT™ *Trust at the Threshold™*

Company Products and Solutions Markets Served Our Advantage FAQ Partnership News Resource Center Contact

You are here » Home



Whether you have one door to safeguard or hundreds of entry points, need low assurance or high assurance, you are required to move forward with HSPD-12 compliance.

TRUST AT THE THRESHOLD™

■ BridgePoint, the leaders in providing proven, trouble-free PIV and CAC Readers and solutions for HSPD-12 compliance.

Whether you need to upgrade an existing physical access system or install a new card reader system with PKI verification and certificate validation, our best-in-class solutions cover the spectrum for strong authentication. For nearly a decade, the Army and Navy have relied on our CAC card access and CAC card authentication solutions. BridgePoint's CAC and PIV readers help the nation's top government agencies ensure "Trust at the Threshold™."

Our products are securely designed and built in the USA, ensuring robust performance backed by exceptional support. For over eight years BridgePoint has supported the DoD Common Access Card program with CAC card readers that meet the GSA-APL Evaluation Program.

[Click here to learn more about our TWIC, CAC, and PIV Reader Solutions](#)

I NEED TO //

- Understand HSPD-12 and related standards-now and in the future
- Know whether our building needs low assurance or high assurance.
- Upgrade my existing system to use CAC readers and PIV readers
- Learn about installing a new trusted PACS with PKI authentication
- Integrate strong authentication PKI into my PACS
- Create a single door CAC and PIV access system
- Learn about PIV-I solutions for non-Federal issuers

XI. Don't Forget About Policies, Governance And Potential Legal Issues

Client Certs (The Technology) Need to Be Supported By Appropriate Policies and Governance Structures

- In looking at successful deployments of client certs, such as the federal government's HSPD-12 CAC/PIV card project, one of the things I'm struck by is that its success is not just a technological thing, it's a sign that appropriate policies were developed by the community.
- If you're planning on doing a major client cert project, please be sure you are also considering the policy implications of moving to client certs, not just the technology issues.
- Your deployment of client certs will also be facilitated if you have local governance structures that understand and can support and advise you on your work.

Be Sure To Keep University Counsel In The Loop, Too

- Why? Well, let me give you one closing example... strong cryptography is export controlled by the U.S. Bureau of Industry and Security, including being subject to the "deemed export" rule. If you plan to issue client certificates to all campus users, remember that some users, such as individuals from North Korea, Sudan, or a relatively small list of other embargo'd countries, may not be eligible for access to strong cryptographic technologies, including potentially client certificates. For more on this point, please consult with your attorney regarding the provisions of the "Deemed Export" rule. As a starting point, see <http://www.bis.doc.gov/deemedexports/deemedexportsfaqs.html>
- Increased use of encryption for official records, may also raise long term institutional record management issues, or potentially FOIA issues for some schools.

Thanks for the Chance To Talk Today!

Are There Any Questions?