Hardware Token Requirements Discussion

November 16, 2011 - Results of the InCommon 2-Factor Authentication Survey

The final results of the InCommon 2-Factor Authentication Survey are available here.

June 22, 2011 Call -- PKI Hardware Tokens topic

1. Form factor: at least USB in the size/shape of a typical flash drive is required. A family of products that uses the same chip and is available in other form factors is preferred. Smart Card and USB-format with flash being to other desired scenarios.
2. FIPS 140-2 Level 2 certification is required.
3. Support for Windows and Macintosh required. Linux an important plus. Windows means both 32 and 64 bit platforms.
4. Windows mini-driver and Microsoft Smart Card CSP strongly preferred over a required CSP software installation.
5. Macintosh support for direct keychain integration is preferred over PKCS-11 and Tokend
6. Support for a minimum of three certificates is required. More strongly desired.
7. RSA key generation and support for 2048 bit keys required.
8. The private key must NOT be exportable from the device.
9. Site configurable policies for PIN length and complexity, token timeout, and PIN retries before blocking are required.
10. End users must not be able to change policies or parameters WRT minimum PIN and timeout.
11. Wireless access to the token (contactless access) highly desirable.

Wireless access to the smart card - e.g., contactless access is desired.

June 8, 2011 PKI Subcommittee Call on PKI Hardware Tokens

Pre-call notes #1

1. Form factor: are both smart-card and USB important? USB is more universal (on computers) but smart-cards can be used for things like physical access, copy machine, etc., and probably have longer "insertion life" than USB devices. (I assume we're not addressing certs on cellphones, etc.)
2. OS support: We should try to avoid the need for any added software on user platforms. I'm told that there are now standards that allow modern OSs to autoconfigure when seeing a particular device, much like they do with external HDs, printers, etc. Is this a good assumption? What OS(s) support this?
3. What's in the x509 object(s) on the device? That has been a serious problem in the past if the content is intended to be understood by a broad set of unrelated services. I think there needs to be several (at least) options but I would like to encourage use of the SIA field to hold a pointer to an appropriate attribute service for the Subject. The AIA field also should contain a pointer to a repository of all valid CA certs in which the Subject name is the Issuer of this cert. One x509 content option should be a Subject name that contains only a unique (to the CA) abstract identifier. Other options would include more about the individual, including email address. Etc.
4. Characteristics of the smart chip:
    a. Is the private key exportable?
    b. Can it hold multiple x509 objects? Other objects?
    c. What happens when the x509 object expires? Can the device content be re-written?
    d. What length PIN is used? Does it have a PIN timeout? Is the timeout fixed or changeable by the user?
    e. Can the PIN be reset without re-issuing the x509 object(s)? If so, this presents a vulnerability. If not, ...
    f. What happens if the device is lost or destroyed? Especially in the case where the SN is only an abstract identifier since the aggrieved person will have to prove that s/he was the correct holder of that identifier...

Pre-call notes #2

1. Direct support for Mac and Linux preferred over provides source code and its your issue to deal with

Pre-call notes #3

1. OS support for Windows (32 & 64-bit). Use of the Windows Smart Card CSP and mini-driver preferred.
2. OS support for Macintosh. Direct Keychain integration preferred over a PKCS 11 and TokenD
3. Ability to import PKCS-12
4. Ability to generate 2048 bit key pairs; private keys are not exportable
5. Token management solution that handles any needed formatting, remote PIN resets, archive of admin keys, etc
6. Willingness to work with Comodo on token management solution and certificate provisioning
7. Discounts would be nice
8. Form factor choices (but at least simple usb and small). Dual flash memory and PKI Token a potential plus
9. Site configurable token policies (PIN complexity, PIN retries before blocking, inactivity timeout, etc).
10. Inability of user to change site PIN policies
11. Simple user PIN changes from native OS
12. Auto OS certificate registration and deregistration on removal a plus
13. Support for multiple certificates a plus but in practice for normal users, not all that great
14.

Call Discussion

1. FIPS 140-2 Level-2 required
2. Token management software is capable of supporting InCommon Silver at a minimum
3. Support for at least three certificates, more are preferred
4. USB flash drive form factor required, card option with same chip preferred, bluetooth capability a plus
5. (a) Template control from Comodo to be able to select CSPs in a secure way and in parallel (b) an integrated stack for both token management and certificate provisioning.
6. Prefer 1-3, require 3-1 (XP,Vista,7) 3-2 and 3-3, must not do 1-4-a, require 3-9,

7. Linux support is important
8.