W3C Workshop on Transparency and Usability of Web Authentication

Position Paper
Peter Lipp
IAIK Graz University of Technology

My position: I am extremely glad that the W3C starts tackling an extremely important issue. (Enough? ☺)

The Web followed the tradition other of internet-protocols - features fist, security later; well, I guess this was not limited to the internet alone. However, we now facing problems that surfaced even after security had been part of the game for a while. Many of the them fall into the *social engineering* category, which makes it a difficult one to solve technologically.

Currently, technology would be available to help solving the problems:

- TLS can be used to authenticate the website (and the user)
- S/Mime can be used to authenticate email; messages which are the source for the pfishing-attack (or contains more active malware) will likely never be signed.

So far - so good. But which of the average user

a) really understands what TLS or S/Mime are
b) checks if TLS is active when entering valuable information in the browser and
c) doesn't click OK on every message box popping up on his screen and he isn't sure what it sais…
d) would realize that the pfishing message was not signed but should have been, since his bank or ebay or paypal told him, they would always sign all of their messages (as they tell him now, they will never ask for his pin etc….)
e) would be able to get a certificate for his client software in the first place without having either an expert guiding his hands

Well, there are some, but …. So the question of usability is really the important one. How can we assist those users as much as possible without assuming any special abilities they have? Let's brainstorm…

- There has always been the dilemma between security and usability. Web browsers always remark: "you are submitting private information over an insecure channel" and users always click "don't dare to bother me again, bastard, I know what I am doing". Disabling the ability to turn off that feature would be nasty for users like myself, who tend to believe they know what they are doing (hmm, makes me wonder…). So all features must be automatic and side-effect free if they can't be disabled.
- A browser could parse the webpage to check hints for what site it seems to present to the user and compare this to the URL it came from. This is very hard to do properly and likely to result in an ongoing race condition between the software developers and the pfishers.
- A mail client could compare the Link-Text with the host the URL points to. Similar problem as above - very hard to do properly and likely to result in an ongoing race condition between the software developers and the pfishers.

- Rigid rules for submitting information; browsers might only post information within a form to a trusted site. Technically easy, practically – for general use - difficult. Would require almost every site to become a trusted one or be blocked from receiving information. Ok, if certificates are for free. But then they tend to be worthless, unless somebody sponsors this. And the changeover would take a while …

- If we are content with a solution that doesn't solve the problem on a general level, but provides a solution for select cases, a trusted platform can be used for transactions with select services. This trusted platform might be a virtual machine running a dedicated environment using TPM-technology. This platform only allows connections to a set of services; service providers can subscribe to be part of that game and can be thoroughly checked to ensure proper credibility or whatever is required. Serious services like banks, ebay, paypal will happily subscribe to the service. We hope.
  This trusted platform could also be a hardware token running a web server the user connects to and the token connects to the service required in a secure way. The token will contain a crypto chip making the whole stuff pretty secure. (sounds so easy, maybe somebody has done so already?)
- I don't see an immediate solution for the general case, unless every user uses strong authentication with tokens …

Looking forward to see other's views….