

W3C Workshop on Transparency and Usability of Web Authentication

Position Paper

Reversed Responsibilities: Browser Authentication instead of Server Authentication

Sebastian Gajek, Jörg Schwenk
www.nds.rub.de

Abstract: As founding members of the German anti-phishing working group www.a-i3.org, we have collected and analysed phishing/pharming attacks and countermeasures proposed by German banks. These analyses show that on the one hand, users are charged with properly authenticating web servers even with new countermeasures, but on the other hand the security problems are becoming far too complex for users to even understand them.

Therefore, we propose to reverse the responsibilities: Web servers are responsible for cryptographically authentication a user's browser to protect against man-in-the-middle-attacks, and then properly authenticating themselves to the human user. This can be done with SSL client authentication, or with XML security, using the browsers cryptographic engine and the private key of the user stored therein.

The Treat: Phishing and Pharming

It is a well-known fact that phishing and pharming attacks are highly successful, despite the fact that attackers don't bother to imitate a SSL indicator at all. This leads us to conclude that SSL/TLS should be used only to encrypt traffic between client and server, and not to authenticate the server. Even if users would bother to check the SSL padlock, there is a large variety of attacks ranging from Javascript [1] to bugs in certificate handling [11].

The reasons for this are:

- The SSL padlock icon is too small.
- The SSL PKI is not understood by customers, especially the notion of trusted CAs.
- DNS naming is not properly understood. For instance, why is service.bank.com OK, but bank.service.com not?
- The possibilities of web design are not clear to customers, especially the power of Javascript.

We conclude that average users are typically overextended with the validation of security parameters. Therefore, we cannot presuppose that only a user-convenient presentation of security solves this problem. Due to the growing complexity of Internet applications and the amplified requirements on security a couple of users will always be unable to correctly "decode" the security parameters. Against this background, we claim that ideally we should shift the tasks of security to a more reliable participant. In the case of mutual authentication we stipulate to weight the security responsibility of Web servers, and in particular to ensure that the server communicates with the legitimate user and not that the user has to verify, if he communicates to the demanded services.

In the past, several solutions have been proposed.

Worthless Solutions

After many years of successful deployment of SSL, the conclusion that SSL is unsuitable for server authentication is hard to communicate. As an example, German banks introduced a security measure called *indexed TAN* (iTAN) to presumably improve the security of online-banking. This solution consists of issuing a printed list of randomly chosen *transaction numbers* (TANs) indexed by, e.g., the numbers from 1 to 100 to the customer. After logging in to his account, and after sending a financial transaction request to the bank, the customer is asked for a special TAN with index i , and has to submit the corresponding TAN to validate the request.

This solution implicitly assumes that the customer verifies the SSL certificate of the bank. If this assumption is not met, a real time MITM attack is possible. A proof-of-concept was presented by a-i3 in September 2005 [2], with work overhead of about ten hours PHP programming. Other solutions along the same lines have been proposed (e.g., eTAN [3], Verified by Visa [4]), which are susceptible in equal manner.

Lessons Learned: Reversed Responsibilities

We think that Internet technologies and attacks will become increasingly complex and sophisticated in the next years. It will thus become impossible to educate customers to enable them to authenticate web servers. However, we already have tools at hand to inverse the roles: Web servers can authenticate web browsers through cryptographic means, and then use this trusted piece of software to communicate with the user. This does not completely solve the security problems of the WWW, because Trojan horses with complete control on the input (e.g. keyboard) and output (e.g. display) of a PC can not so easily be beaten. It can however help to make the web more secure.

Short Term Solution: Using SSL Client Certificates

SSL client authentication is a feature present in standard Web browsers, however rarely used. One of the few applications known to the authors is, e.g., the renewal of e-mail certificates, when the website of the CA requests the old e-mail certificate to get all the information contained therein.

A possible solution to the phishing/pharming problem can be sketched as follows:

- The bank issues X.509 client certificates to its customers. This can be done through a web interface, which triggers the generation of the public/private key pair, sends the PKCS#10 request to the CA server, and stores the certificate in the browsers certificate store. The request can be authenticated by the customers PIN and additionally by a TAN (if available). Stronger out-of-band authentication is possible, but also more expensive.
- It is assumed that the customer only uses a small number of different computers/browsers for online-banking. To each of these browsers, a different certificate is issued to avoid export and import of private keys.
- It is further assumed that the customer has not been subject to a phishing/pharming attack lately. This can be achieved, e.g., by issuing a new PIN to customers by snail mail, which is some kind of out-of-band authentication.
- During the SSL handshake the server requests client authentication specifying the type of X.509 certificate issued by the bank's CA. The verification of the signature chain is done by the bank. If this verification is successful, the SSL connection is secured against MITM attacks.

Similar solutions already exist, e.g. Passmark employed by the Bank of America, where Cookies are used instead of client certificates. However, cookies are difficult to protect against Cross Site Scripting and DNS/ARP spoofing attacks combined with fake SSL server certificates. To put it shortly: Cookies have been invented to be sent away from the browser, private keys to stay forever in the cryptographic module.

SSL client authentication improves the trust of banks in their customers, but not the trust of customers in online banking or electronic commerce. Hence client authentication via SSL should be complemented with the authentication of the server to the customer (not to the browser!). To achieve this, the credential sent by the server to the customer must be easily verifiable by a human being. The best solution seems to us some kind of visual credential, e.g., a private picture of the customer given to the bank.

Middle Term Solution: Browser Support for XML Security

Nowadays, a straightforward proceeding is to secure the communication between two distributed systems, generally provided with means of network security. However, the data transmitted is afterwards unprotected at the target machine. As a result, messages are protected only while in transit on the wire. For example, sensitive data on a hard disk drive is not generally protected unless we apply a proprietary encryption technology. Moreover, network security technologies do not provide non-repudiation well, which is which is important for business Web services and, in particular, for any business transaction. Non-repudiation means in this context that a communicating partner can prove that the other party has performed a particular transaction. Both downsides exemplify that shifting to data-based security (in collaboration with common network security protection mechanisms) opens a gap to new applications of security, and to a far-ranging opportunity of more fine-grainly realizing security mechanisms.

For this reason, we propose to adopt XML Security technologies in Web browsers, more precisely to implement XML-based security libraries (e.g. [8]). Without doubt XML is a promising data standard and provides by now a brought range of security mechanisms, such as XML Signature [9] and XML Encryption [10], to realize security technologies on data/message layer. Moreover, next generation browser would dramatically benefit from the adaptation of XML Security technologies. The web browser could be used for a new portfolio of applications, e.g., as a trusted viewer verifying signed invoices [5].

As a mean to sign messages from the customer to the bank and to encrypt messages from the bank to the customer, the private key/X.509 client certificate pair stored in the browser can be used. Thus, the bank is able to check message authenticity of the customers' messages.

As a side effect, webmail support for S/MIME messages can be enhanced. In our XMaiL project [6], we have implemented a parser (and the corresponding inverse parser), which transforms S/MIME messages to a special XML format in such a way that the validity of the original PKCS#7 signature can also be checked in the XML format.

With properly designed XML schemas, even some kind of trusted computing can be achieved immediately:

- In Germany, smart card readers are available with their own (limited) keyboard (digits 0 to 9) and display (a few ASCII characters).
- If the data to be signed contains an element <SCR display>, which contains only a small number of ASCII characters, these characters can be extracted by an XPath expression.

- The hash value of the complete data element is then sent to the smart card reader, together with the extracted ASCII characters.
- The smart card reader displays the ASCII characters and appends the hash of these characters to the hash of the complete data element.
- This new hash value is then signed, and the signature value included in an XML signature element.

Summary

We think that customer awareness (cf. [7]) can not be increased sufficiently to protect against future sophisticated attacks. Therefore, the responsibilities for entity and message authentication must be shifted from the customer to the web service provider.

In a short term scenario, the possibilities of SSL client authentication must be used to protect e-business and e-banking on the internet.

In a middle term scenario, browsers should include the basic XML security libraries, e.g. XML Signature and XML Encryption. This may offer, together with support for the inclusion of external hardware-based security modules (e.g., smart card readers and smart cards) reliable security even in the presence of malicious software on the customer's PC.

References

- [1] André Adelsbach, Sebastian Gajek and Jörg Schwenk: Visual Spoofing of SSL Protected Web Sites and Effective Countermeasures. ISPEC2005, Lecture Notes in Computer Science, Springer, 2005.
- [2] https://www.a-i3.org/images/stories/pressemeldung/pressemeldung_itan_lang.pdf
- [3] http://www.postbank.de/ql_1101978438667/pbde_pk_home/pbde_pk_produkteundpreise/pbde_pk_serviceundkredite/pbde_pk_online_banking/pbde_pk_infoseite_itan.html
- [4] https://usa.visa.com/personal/security/vbv/?ep=v_sym_verified
- [5] Sebastian Gajek, Wolfgang Kubbilun, Michael Psarros, and Jörg Schwenk: Trustworthy Visualisation and Verification of Multiple XML-Signatures. CMS 2005, Lecture Notes in Computer Science, volume 3677, pages 311-320, Springer, 2005.
- [6] Jörg Schwenk, Lijun Liao, Lars Ewers, Wolfgang Kubbilun: Secure XMail or How to Get Rid of Legacy Code in Secure E-Mail Applications. CMS 2005, Lecture Notes in Computer Science, volume 3677, pages 291-300, Springer, 2005.
- [7] FFIEC Guidance: Authentication in an Internet Banking Environment. FIL-103-2005, October 12, 2005. <http://www.fdic.gov/news/news/financial/2005/fil10305.html>
- [8] Apache XML security. <http://xml.apache.org/security/>
- [9] XML Signature WG. <http://www.w3.org/Signature/>
- [10] XML Encryption WG. <http://www.w3.org/Encryption/2001/>
- [11] SSL Man-in-the-Middle attacks. <http://www.sans.org/rr/whitepapers/threats/480.php>