Online banking and man in the browser attacks, survey of the belgian situation

Jérôme Dossogne Olivier Markowitch
Université Libre de Bruxelles
Fac. Sciences, Dept. Computer Sciences
Boulevard du Triomphe - CP212, 1050 Bruxelles, Belgium
jerome.dossogne@ulb.ac.be olivier.markowitch@ulb.ac.be

Abstract

In this paper we review a non-exhaustive list of online banking systems used in Belgium in regard of man in the browser attacks. We focus our attention to the signature on each transaction and suggest simple solutions that would prevent and/or detect attacks attempts.

1 Introduction

Nowadays, most financial institutions propose online accesses to some of their services; called online banking, net banking, ... the security of these systems (identification/authentication, signature of transactions, ...) are based on different tools.

Meanwhile, *man in the browser attacks* become increasingly worrying [6, 5, 4, 3, 1] . In this kind of man-in-the-middle attacks, the user's web interface of the online banking systems is under the control of the attacker.

In this paper we review the online banking systems of the main belgian banks in regards to such attacks. We propose reasonable solutions to detect when an attack happens as well as to prevent them.

2 A man in the browser attack based on the signature of a transaction

When considering online banking services, we can formalize a transaction and its notation as such: $A_U \xrightarrow[D]{M,C} A_E$, U wants to send the amount M from his account A_U to the account A_E of another entity E with a communication C at the time D. The signature of U on the transaction $A_U \xrightarrow[D]{M,C} A_E$ and expressed with the notation $\sigma = \text{Sign}_U(A_U, A_E, M, C, D)$ would be considered as a proof of the identity of U and of his request to perform the transaction $A_U \xrightarrow[D]{M,C} A_E$. This set of variables on which the signature should depend is the minimum set we recommend for all systems of signature. Depending on the signature protocol, it can be advised to sign more parameters, for example a random value R generated by the online banking service to prevent replay attacks, i.e. $\sigma = \text{Sign}_U(A_U, A_E, M, C, D, R)$.

It is obvious to anyone that signing an unfilled check is, regarding security, hazardous. The same can be said about the behavior of a bank's customer that sign a check without paying attention to the amount. The digital equivalent to these situations happens when the user performs a signature on a document which value is not dependent of all the parameters/sections/elements of the document.

If the document is a transaction and if such signature is not computed using all the elements of the transaction, it is possible for an individual in control of the user's

browser's view and or behavior to modify that aspect and still forward the same signature coming from his victim to the bank. Such attacks can be, for example, conducted by impersonating the bank's website (phishing) or even by distributing or installing a modified version of the web browser used to access such service. Since the signature is computed using only the secret (a PIN code, a password or a code out of list distributed to the user) and parts of the transaction, for example $\sigma = \text{Sign}(U, M)$ without using A_U, A_E, C nor D, another transaction could be created by an attacker with modified values A'_U, A'_E, C' and D' and σ would be a valid signature for the second transaction too. In our case, such attacks can take as targets the customer's browser and its rendering function, it's cache or even configure the browser to use a proxy which will alter the pages while the user browses the website.

To protect the user against a modified version of it's web browser, it is usually suggested to use anti-malwares softwares. Such suggestions are quite relevant but we have to keep in mind that these protection techniques are far from perfect. Moreover, recent polls [7, 8] on websites with a computer oriented content and communities, reveals on increase since last year from 15.06% to 15.84% of their community that does not use any antivirus program. Therefore, the security of an online banking service should not suppose that every of its customer are shielded against software based attacks such as virus and spywares.

Our objective is to bring the importance of signing a transaction using all the parameters of the transaction (A_U, A_E, M, C and D) to the attention of the reader.

To secure further online banking services, it is possible to detect attempts of attacks on their customers. Indeed, it is possible to include other aspects in the signature such as the IP address of the user and of the server. An educated user would be able to detect if it is his own or not. Otherwise the name of the Internet provider (obtained through a reverse dns lookup request) would already help and force the attacker to use the same Internet provider as his potential victim.

When signing all the elements for each transaction is considered too fastidious, i.e. of a too high cost for the user, it is possible to ease this process with the creation of pre-approved list. For example, a user could create a pre-approved list made of A_E's and decide that any transaction that does not sign the A_E variable must have an A_E included in that list. This limit the scope of an attack since the attacker could still modify the value of A_E in the transaction but only to one present in the contact list. When creating or adding items to such list, it would be mandatory for the user to sign each request or else an attacker would simply have to insert his own account number in the list.

3 The Belgian situation

We propose to look at the main Belgian net banking services at the light of the attacks such as the ones discussed in section 2. The four first banks we present, Dexia, Fortis, KBC and Centea are using an unconnected security token called Digipass in different ways, we will show the consequences of their choices. ING is using another kind of device which they also call Digipass. The Deutsche Bank is using a Code Card and the Keytrade bank requires the use of an RSA SecurID device to authenticate the user.

3.1 Dexia

The Dexia online banking service[10], called *Dexia Direct Net*, requires the use of a standalone unconnected smart card reader equipped with a numeric keypad and a screen (called Digipass). When a customer completes a transaction, he is asked to sign the transaction through his Digipass wherein the user inserts his bank card and with the knowledge of the card's PIN code.

However a user U signs a transaction $A_U \xrightarrow[D]{M,C} A_E$ using that service, the website does not ask the user U to sign the whole transaction but only the amount M and a number R which appears random to the user. This allows an attacker to modify the value of some parameters (A_E for example), i.e. $\sigma = \text{Sign}_U(M, R)$. While this apparently random value could well be dependent on the transaction's parameters it can only increase the security of the transaction by protecting the system against replay attacks. Indeed, even if that value was dependent on A_E for example, an attacker could easily modify A_E, ask the website for another apparently random value and submit it for signature to the user who will sign it without noticing the difference between the received random and a genuine value.

To improve the level of security, a signature σ involving all the parameters of the transaction (A_U, A_E, M, C and D) should be considered, $\sigma = \text{Sign}_U(A_U, A_E, M, C, D, R)$.

The service also offers the possibility to memorize transactions in a list L for a future reuse. These transactions can be partially defined, leaving some parameters undefined (A_U, M, D and C for instance). When a user wishes to add an item to that list the website does not ask the user U to sign the whole transaction but only a number which appears random to the user. This allows an attacker to modify the value of all the transaction's parameters.

To improve the level of security, a signature $\sigma = \text{Sign}_U(A_U, A_E, M, C, D)$ involving all the parameters of the transaction (A_U, A_E, M, C and D) should be considered when performing a request to insert such a transaction in the list L.

After applying the improvement suggested for the creation of L, a cost effective way to improve the overall security of the service could be to allow a user to sign only the parameters that are not already pre-approved. For example, if there is an entry in L with only $A_E = x$ and $M = y$, when the user U ask to execute a transaction $A_U \xrightarrow[D]{M=y,C} A_E = x$, the service will only ask him to produce a signature involving (A_U, C and D) $\sigma = \text{Sign}_U(A_U, C, D, R)$ since the couple $A_E = x$ and $M = y$ were already pre-approved.

3.2 BNP Paribas Fortis

The BNP Paribas Fortis online banking service[11], called PC banking, requires the use of the same device called Digipass by Dexia. When a customer completes a transaction, he is asked to sign the transaction through his Digipass wherein the user inserts his bank card and with the knowledge of the card's PIN code.

When a user U signs a transaction $A_U \xrightarrow[D]{M,C} A_E$ using that service, the website asks the user to sign the parameters M and A_E, i.e. $\sigma = \text{Sign}_U(M, A_E)$. To prevent itself against replay attacks, the server will deny any transaction with the same M and A_E in a short interval of time.

This system is an improvement from the system proposed by Dexia since the destination account A_E cannot be modified anymore by an attacker without the user's knowledge and approbation. While the two parameters M and A_E are now protected, A_U, C and D are still subjects to the attack we mentioned in section 2.

Therefore, it is still possible to improve the level of security by producing a signature σ involving all the parameters of the transaction (A_U, A_E, M, C and D), i.e. $\sigma = \text{Sign}_U(A_U, A_E, M, C, D)$.

Moreover, when a user wishes to sign multiple transactions at once, as the system permits it, the signature generation protocol changes and falls back to the system used by Dexia and is, in such case, subject to the same vulnerabilities and recommendations we mentioned for Dexia's online banking service.

3.3 KBC

The KBC online banking service[12], called KBC online, requires the use of the same device called Digipass by Dexia. When a customer completes a transaction, he is asked to sign the transaction through his Digipass wherein the user inserts his bank card and with the knowledge of the card's PIN code.

When a user U signs a transaction $A_U \xrightarrow[D]{M,C} A_E$ using that service, the website only asks the user to sign a challenge R which enables the server to prevent any kind of replay attack, i.e. $\sigma = \text{Sign}_U(R)$.

However, as you can see, neither A_U, M, C, D, or A_E are part of the signature generation. Therefore, the man in the browser attack we described in section 2 can be applied on any of those parameters.

To improve the level of security, a signature σ involving all the parameters of the transaction (A_U, A_E, M, C and D) should be considered, $\sigma = \text{Sign}_U(A_U, A_E, M, C, D, R)$.

3.4 Centea

The Centea online banking service[13], called Centea-Online, has the same requirement and behavior regarding the signature of a transaction than the KBC online banking service.

3.5 ING

The ING online banking service[14], called Home'Bank, proposes two methods for its identification's process. The first requires the use of ING's software called security module and the other requires the use of a device with a screen and a numeric keypad called Digipass. Depending on the method used, ING has different policies. More restrictions on the maximum amount to be transfered per transaction or per week are applied when the security module is used than when the Digipass is used.

3.5.1 Using the security module

We will now study the first method proposed by ING. To log in and execute a transaction, the user has to launch the security module and ask, via the Home'Bank website, the authentication process that calls the security module which is listening to the port 1234 on the localhost ip address (127.0.0.1). The security module then asks the user to provide his password. From this point on, the user is considered as logged in and can browse the Home'Bank to ask of a transaction to be executed. After filling in all the fields needed for a transaction and asking the website to validate it, the security module is called and prompt the user for his password. If the password is correct the transaction is executed.

When called, the security module executes its task using a safety file created earlier by the module. To create the safety file, the module ask the user to provide a code distributed by ING.

It is here unclear of what is really signed since the communications between the website and the security module are not visible to the user. Therefore, the mentioned attack can clearly be applied by modifying the user's web browser's behavior. One would simply have to display values for the transaction's parameters different than the ones sent to the security module.

In general, security systems where the communications are under the solely control of the browser are unsecure. Such systems are obviously subject to attacks where the installed browser can be modified without the user's knowledge. As we explained in

section 3.1, the user knows what he signs* but he does not sign enough parameters for the system to be fully secure. On the other side, in ING's case, the user does not know what he signs at all, which leaves open a possibility for an attacker in control of his web browser to modify all the parameters of the transaction.

It is possible to prevent the use of a modified browser by trying to enforce the use of certified and signed programs conjointly with the security module. However, the use of an external device that cannot be modified without the user's knowledge, that are distributed and verified by the bank and work on challenge-response base is better than any security module installed on a complex and unsecure environnement like a personal computer. More generally, any software present on a computer is potentially a source of weaknesses since it can be attacked/modified. Such environment should therefore be considered as more insecure.

3.5.2 Using ING's Digipass

The second method for ING's website's identification process requires the use of a standalone unconnected device with a digital screen and a numeric keypad called Digipass by ING. However, Contrary to the Digipass used by Dexia, Fortis, KBC and Centea, ING's Digipass is not a smart card reader and therefore does not require the use of the user's bank card.When a customer completes a transaction, he is asked to sign the transaction through his Digipass and with the knowledge of the Digipass's 5 digits PIN code.

When a user U signs a transaction $A_U \xrightarrow[D]{M,C} A_E$ using that service, the website only asks the user to sign a 12 digits value R, i.e. $\sigma = \text{Sign}_U(R)$.

To improve the level of security, a signature σ involving all the parameters of the transaction (A_U, A_E, M, C and D) should be considered, $\sigma = \text{Sign}_U(A_U, A_E, M, C, D, R)$.

3.6 Deutsche Bank

The Deutsche Bank online banking service[15], called Online Banking, requires the use of card, called Code card, associated to the user, identified by a serial number and used for the array of 5x8 ($[A; E] \times [1; 8]$) codes of four 10-based digits displayed on its surface.

Prior to performing a transaction, the user has to use his login, password and the Code card to identify himself with the website. In this case, no unconnected security device are used, the password is verified by the website itself. When a user U confirms a transaction $A_U \xrightarrow[D]{M,C} A_E$ using that service, the website asks the user one of the 40 possible codes present on the Code card. In this case, as you will by now have guessed, all the parameters of the transaction are subject to man in the browser attacks.

Moreover, if a user loses his password, he simply has to call the helpdesk of the company and identify himself. The human operator will ask him 1 of the 40 codes of the Code card (for example A3). Afterwards, a new password will be sent by mail. Notice that the Code card will not be renewed even if the user just revealed a part of his secret (i.e. one code) on an unsecure channel to an unindentified human (the operator could be dishonest, the call might have been redirected or someone might have been listening to the line or to the user's room door).

Since the website awaits for the code only for a limited amount of time, the browser can easily reveal all the secret codes. Indeed, if the user takes too much time to answer, the website will ask another code. Therefore, the browser could slow himself down when asked to submit the confirmation code to the website. The browser could

*Even though, even if the user inputs the amount in his Digipass to generate the signature, he has no proof that the values he inserted in his device were really used to generate the signature.

do so by faking a slow network and/or computer without the user suspecting anything. Moreover, the browser could even faking the sending of the code to the website and ask the user to enter several codes each times he wishes to authenticate with the online banking service.

One might argue that the Deutsche Bank's website, using virtual keyboards, is protected against malwares trying to capture the codes. However, it has been demonstrated that such systems are not secure [2]. Moreover virtual keyboards are sometimes even less secure than regular keyboard regarding an attacker in the same room who can watch the user interact with the website since the user interact slower with the virtual keyboard than the real one.

3.7 KeyTrade

The KeyTrade online banking service [16] requires the use of a standalone RSA SecurID device [9] equipped with a screen (called Keytrade ID) of which the serial is associated to his login.[†] Prior to performing a transaction, the user has to authenticate with the website using the knowledge of his login, password and a code, called Keytrade ID code, generated with his Keytrade ID. Afterwards, the user can complete the various fields required for his transaction and confirm that transaction using a secondary password, which is different from the authenticating password.

In other word, once again, the code inserted by the user in the browser which will be sent to the bank's website to confirm the transaction does not depend on the parameters of the transaction. Therefore, a modified browser could send false informations to the bank's website using the same code of confirmation without the user's knowledge.

4 Conclusion

In conclusion, none of the systems we had the opportunity to observe did protect all the parameters of the transaction. To the best of our knowledge, the one protecting the most each transaction is the one provided by BNP Paribas Fortis which only protects the amount to be transferred and the account of destination in case of a single transaction. We discovered the existence of identification systems for video-games that ensure a stronger security than the one used by some online banking systems. For instance, a version of the identification system of Activision-Blizzard's Massively Multiplayer Online Role-Play Game, World of Warcraft (used by approximatively 11 millions paying subscribers i.e. around the size of the whole Belgian population) has a procedure similar to the KeyTrade online banking system which appears to be more secure than the Code card system of the Deutsche Bank since the pool of confirmation codes asked when a user wishes to identify himself is not limited to the contrary of the system used by the Deutsche Bank.

References

[1] Philipp Guhring, Concepts against Man-in-the-Browser Attacks, Financial Cryptography, FC++ number 3, 2007 number 3, 2006

[2] Ryan Naraine, Hacker demos how to defeat Citibank's virtual keyboard, `http://blogs.zdnet.com/security/?p=195`, 2007 (last checked 21th march 2010)

[3] Stan Hegt, Analysis of Current and Future Phishing Attacks on Internet Banking Services, Master's Thesis, Eindhoven University of Technology, 2008)

[†]Other authentication methods are being developed by KeyTrade as can be seen on their website.

[4] Diego Alejandro Ortiz-Yepes, Enhancing authentication in eBanking with NFC enabled mobile phones, Master's Thesis, Eindhoven University of Technology, 2008

[5] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, Your Botnet is My Botnet: Analysis of a Botnet Takeover, Technical report, University of California, 2009

[6] Avivah Litan, Where Strong Authentication Fails and What You Can Do About It, Gartner Research, ID Number: G00173132, 2009

[7] PCInpact.com, Les INpactiens et les anti-virus, `http://www.pcinpact.com/sondages/detail/189.htm` (last checked 18th march 2010)

[8] PCInpact.com, Les antivirus et vous, `http://www.pcinpact.com/sondages/detail/135.htm?action=vote&reponse=0` (last checked 18th march 2010)

[9] RSA.com, Solution Brief: RSA SecurID Two-factor Authentication, `rsa.com` (last checked 16th march 2010)

[10] Dexia, Dexia Direct Net, `https://directnet.dexia.be` (last checked 21th march 2010)

[11] BNP Paribas Fortis, PC banking, `https://www.fortisbanking.be` (last checked 21th march 2010)

[12] KBC, KBC online, `https://www.kbc.be/` (last checked 21th march 2010)

[13] Centea, Centea-Online, `https://www.centea.be/` (last checked 21th march 2010)

[14] ING, Home'Bank, `http://homebank.ing.be/` (last checked 21th march 2010)

[15] Deutsche Bank, Online Banking, `https://secure.deutschebank.be/` (last checked 21th march 2010)

[16] KeyTrade Bank, `https://secure.keytradebank.com/` (last checked 21th march 2010)