A PROOF OF CONCEPT IMPLEMENTATION OF A SECURE E-COMMERCE AUTHENTICATION SCHEME

Carolin Latze[1], Andreas Ruppen[2], and Ulrich Ultes-Nitsche[3]

[1,2,3]**University of Fribourg**
Department of Informatics
Boulevard de Perolles 90
1700 Fribourg
Switzerland

[1]**carolin.latze@unifr.ch,** [2]**andreas.ruppen@unifr.ch,**[3]**uun@unifr.ch**

ABSTRACT

E-Commerce applications such as online shopping or e-banking are steadily gaining popularity in every-day life. Yet, many users are not aware of how to avoid the associated security problems: they don't choose secure passwords, check server certificates, or reliably react to browser warnings when a certificate can't be verified. Thus, protecting users from physical and social attacks such man-in-the-middle, pharming or phishing attacks is of main importance in online business. This work describes a proof of concept implementation of a secure e-commerce authentication scheme using everyday devices like Trusted Platform Modules (TPMs) and mobile phones. As the prototype has shown very promising results, the authors believe that such an authentication protocol might be a possibility in order to avoid the attacks mentioned above.

KEY WORDS

E-Commerce, Authentication, TPM

A PROOF OF CONCEPT IMPLEMENTATION OF A SECURE E-COMMERCE AUTHENTICATION SCHEME

1 INTRODUCTION

Nowadays, people use more and more e-commerce applications, like online-shopping and online-banking. Those online offers allow for attacks like phishing, pharming and man-in-the-middle attacks which try to steal the user's credentials in order to misuse his account. Advanced users would probably be able to detect those attacks, but the majority of the users are naïve users who do not think about security issues. Therefore, the technology has to help those users to avoid such attacks. In 2007, the authors of this work proposed to use trusted devices like PCs with a trusted platform module (TPM) or (trusted) mobile devices for authentication in e-commerce applications [1]. This paper presents a proof of concept implementation of that authentication protocol as well as its evaluation in a real world testbed.

The paper starts with a short introduction into the attacks to be avoided, goes on with an introduction into the proposed solution, its verification and implementation. It ends with an evalutation and a short conclusion.

2 PHISHING, PHARMING, AND MITM ATTACKS

Phishing, pharming and man-in-the-middle (MITM) attacks are the most dangerous user authentication attacks in online business today. Of course there are also other serious problems like key loggers and other kind of spyware on the user side, but those can be circumvented with an up-to-date anti-virus and anti-spy software or - for the advanced users - with intrusion detection and artificial immune systems. This work concentrates on an easy but efficient way to protect the user from phishing, pharming, and MITM attacks, which are described in detail in this section.

2.1 Phishing Attacks

Phishing attacks belong to the group of social engineering attacks. That means that they do not profit of a technical problem or bug but on careless or naïve users. Phishing relies on the idea that it is possible to talk the user into revealing her credentials by pretending to be a trusted organization

with a trustable concern. Examples are e-mails claiming that there has been a problem with the users account at bank XY and in order to clarify that, the user should submit her credentials to a side referenced by a link in the e-mail. On first sight, that link might really look like it belongs to bank XY but it does not. It belongs to an evil site whose only goal it is to collect all those user credentials. After the user entered her credentials there will usually come a message thanking the user and confirming that the problem could be solved.

2.2 Pharming Attacks

Pharming attacks have the goal to redirect the user to an evil webside without forcing the user to click on the wrong link. This type of attack relies on manipulating the DNS entries or the host file on the user's machine. The user may then type the correct URL of the online shop or bank but will be directed to an evil page looking like the desired page but whose only value is to collect user credentials. After having given his credentials, the user will see an error message and get redirected to the original page.

2.3 Man-in-the-Middle Attacks

Man-in-the-Middle (MITM) attacks are more sophisticated than pharming and phishing. As the name suggests, MITM attacks require an attacker to reside between user and the online service, she wants to reach. If the user wants to connect to the online service, she will instead connect to the MITM, who himself is connected to the online service. Instead of only stealing the user's credentials, the MITM can also modify transactions sent by the user. Imagine a user wants to connect to her bank. He will be connected to the MITM instead who forwards the user credentials to the bank (probably after having stored them). If the user then sends a transaction together with the transaction number (TAN), the attacker can easily alter the transaction as he also owns a TAN now. He will then send the altered transaction to the bank and reply with a wrong transaction confirmation to the user. Unlike phishing and pharming, this attack is also successful when using TANs and e-tokens.

3 STRONGER AUTHENTICATION IN E-COMMERCE

In 2007, the authors published their solution to protect the users against the attacks mentioned above [1]. They propose to implement an authentication protocol, that makes use of trusted devices like Trusted Platform Modules (TPMs) or mobile phones. A TPM is a small trusted chip, built into most of the computers shipped today, which has been specified by the Trusted Computing Group (TCG) [3]. This chip provides secure storage for keys and hashes and some basic cryptographic functions. Furthermore, it is the root of trust and measurements of a PC and may be identified uniquely worldwide. Such features make it extremely useful for authentication purposes. If there is no TPM available the authors of [1] propose to use a mobile phone as trusted device. To make a mobile phone really trustable, one might think about enhanced SIM cards like those from SanDisk [4] or the multimedia cards from Gemalto [5]. If neither TPM nor a trustable mobile phone is available, the authors propose to use One-Time Passwords (OTP) sent by SMS.

3.1 Authentication Using a TPM

This authentication method is the most secure of all the three mentioned above. Before ever beginning such an authentication, there has to occur an offline key exchange between the online shop and the user. This can be done using signed snail mail. The user will get a CD-ROM from the online-shop containing the shops public key and a piece of software, that seals that key to the user's TPM, generates a new user key and prints the public part. The user then has to send the printed public key back to the online shop, who will store it electronically. Additionally that CD-ROM contains the client authentication software needed to run the authentication protocol with the server in case it cannot be expected to be implemented on every mainstream operating system.

The authentication protocol itself is more or less a standard challenge/ response protocol, where the TPM calculates the challenge to the server and verifies its response. As mutual authentication is required for really secure authentication schemes, the server also has to trigger a challenge/ response protocol, where the user's TPM has to calculate the response. After both runs, both sides can be sure, they are connected to whom they wanted to be connected. The security of this protocol relies on the TPM, that does all the

security-critical tasks.

3.2 Authentication Using a Trustable Mobile Phone

As it might be possible that a user does not possess a modern computer with a TPM or that the user wants to connect to the online shop from another PC, the authors of [1] proposed another authentication using trustable mobile phones. A mobile phone becomes trustable when it either contains a Mobile Trusted Module (MTM) [6] or an extended SIM card that can run applications in its secure environment like those from SanDisk [4] or Gemalto [5].

Before starting an authentication with that device, there has to be an offline key exchange as mentioned above. This can be done using read-only memory cards, as most of the modern mobile phones have card slots. Printing the key from a mobile phone is obviously not easily possible. Therefore, the authors propose to deliver memory cards with a read-only part for the server's public key and the software mentioned above and a read-write part to store the clients key. If the user sends the memory card back using signed snail mail, it should not be possible to tamper with that key.

The authentication itself is very similar to the authentication using the TPM. The difference is that the server runs the challenge/ response protocols with the mobile phones and not with the computer, the user uses! As this does not secure the connection between the user's computer and the server, the server has to ask the user for a transaction confirmation on the mobile phones every time it receives a transaction request.

3.3 Authentication Without a Trusted Environment

There may be users that neither possess a PC with TPM nor a trustable mobile phones. But nowadays almost everybody possesses some kind of mobile phone no matter how old it is. Therefore, the authors of [1] propose to use One-Time Passwords (OTPs) as backup solution for those users. There is no offline key exchange in advance since there is no trusted medium on the user's side to store the keys. The only thing that has to be registered at the online shop in advance is the user's mobile phone number. For security reasons, this should be done using signed snail mail. Afterwards, if the user wants to connect to the online shop, he will get an OTP in a SMS on his mobile phone, that has to be sent back to the server to be really authenticated.

Later, every transaction request received by the shop has to be confirmed by the user using SMS.

4 AVISPA VERIFICATION

The AVISPA framework [7] stands for Automated Validation of Internet Security Protocols and Applications and provides a good and fast way to check protocols for security flaws.

4.1 Modeling of the E-Commerce Protocol

AVISPA provides a High Level Protocol Specification Language (HLPSL) [7] to model network protocol in order to proof their security. HLPSL models participants of the protocol as so called basic roles. In order to specify the basic roles, the protocol should be written in A-B (Alice - Bob) notation:

```
A->S: A
S->A: {Ns.S}_Ka
A->S: {Ns.Na}_Ks
S->A: {Na}_Ka
```

From that A-B notation, the basic roles can be specified easily. The following code sniplet shows The HLPSL expression of the role A (the client in our case):

```
role alice(A,S: agent,
   Ka,Ks: public_key,
   SND,RCV: channel(dy))

played_by A def=
  local
State: nat,
Na,Ns: text
  init
State := 0
  transition
0. State=0 /\ RCV(start) =|>
   State':=2 /\ SND(A)
2. State=2 /\ RCV({Ns'.S}_Ka) =|>
   State':=4 /\ Na':=new() /\ SND({Ns'.Na'}_Ks)
     /\ request(A,S,alice_server_ns,Ns')
```

```
     /\ witness(A,S,alice_server_na,Na')
4. State=4 /\ RCV({Na'}_Ka) =|>
   State':=6 /\ secret(Na',na,{A,S})

end role
```

For a detailed description of the syntax see [7].

The security goals to check are the authentication between the server and client (A) and the secrecy of the nonces exchanges during the authentication. In HLPSL the goals are specified as follows:

```
goal
secrecy_of na,ns
authentication_on alice_server_na
authentication_on alice_server_ns
end goal
```

When verifying this protocol with AVISPA against the goals, no vulnerabilities have been found:

```
SUMMARY
  SAFE
[...]
ATTACK TRACE
  %% no attacks have been found..
```

Therefore, the protocol described in Section 3 can be assumed to be secure, which means that nobody can impersonate the client to be authenticated against the server and the nonces are not released to an attacker. Obviously, there has to precede a key exchange to secure the connections, but as this work concentrated on the authentication itself, the key exchange is not part of this paper.

5 PROOF OF CONCEPT IMPLEMENTATION

In 2008, there has been done a proof of concept implementation as part of a master thesis [2]. This implementation will be described in the following sections.

5.1 The TPM Emulator

As described in Section 3 TPM is an emerging technology on the market. But even if more and more laptops ships with TPM support there are still

many machines without a hardware TPM. Besides that, developing directly on the hardware TPM can be very annoying since the only way to reset a TPM is to reboot the machine. For theses reasons M. Strasser developed a TPM emulator which comes as a Linux module [11]. But even if this TPM emulator enables older devices to use this technology, it should be clear that a software TPM will never be as secure as a hardware TPM.

5.2 TrouSerS - An Open Source TCG Software Stack

In order to keep the TPM a low cost device, the Trusted Computing Group (TCG) proposed to implement only the security critical functions inside the TPM and move the uncritical functions into the so called TCG Software Stack (TSS). One of the popular TSS implementations is called TrouSerS [9]. TrouSerS implements the TSS in C and is released under the Common Public License (CPL). It was developed by IBM. Today TrouSerS supports the TSS 1.1 specification and work on the TSS 1.2 specification has begun. Although TrouSerS deviates from the TSS specification in some points for more convenience, it is possible to compile it with strict TSS compliance.

5.3 Gammu

Gammu (GNU All Mobile Management Utilities) [10] is an open source project trying to close the gap between Linux and mobile phones. Gammu can connect to several mobile phones either over bluetooth or the USB sync cable. The mobile phone support goes from SMS sending and reading to placing phone calls and basic synchronization of the address-book and calendars. Gammu works perfectly with a Sony Ericsson K800i that we used. Gammu can be used as a normal application or in daemon mode. Furthermore, when running in daemon mode, Gammu supports two back-ends: a simple file back-end similar to mailboxes and a MySQL back-end. The latter one is interesting since no parsing is needed for reading out the content of a received SMS. Therefore, Gammu could be easily integrated into the mobile phone assisted authentication part of the protocol proposed in [1].

5.4 Implementation

All three methods described in Section 3 have been implemented. As the system should be comfortable and as transparent as possible to the user, it

was one of the goals of the implementation that the built system is integrated seamlessly into an existing system. Only few changes are necessary to an existing login procedure in order to integrate the three protocols. In fact they can be inserted right after the standard login procedure and the redirection of the client to the landing page for logged in users. The TPM and the transaction confirmation protocols are delivered as PHP code to provide the functionality of the authentication server to PHP and a little C program which does actually the binding between PHP and the authentication server written in C and the necessary PHP code to provide the functionality of the authentication server to PHP. The implementation is based on a client-server

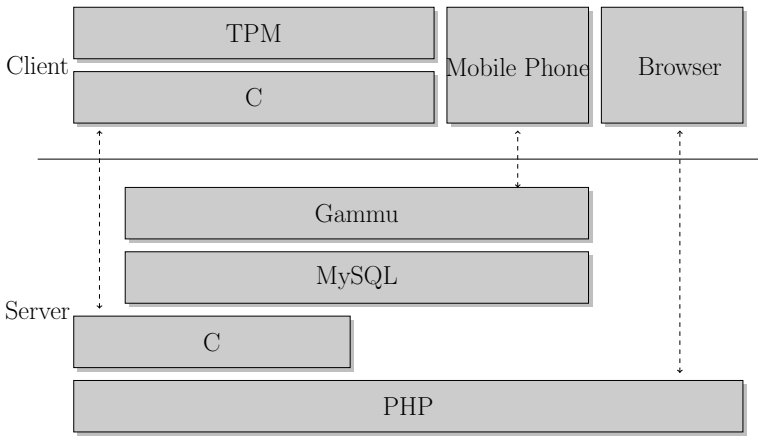

Figure 1: Layered model of the involved components

architecture, where the user of the e-commerce application plays always the role of the client. The figure 1 shows a layered model of the invovled parties. Each layer can only communicate with the layer above/below. The communication between the client and the server uses HTTP for the browser, a TCP socket for the communication with the TPM and the GSM network g for the mobile cell phone.

For the mutual authentication with the TPM and the transaction confirmation over SMS, two server parts are needed: the HTTP server and the authentication server. The HTTP server is responsible to serve the HTML pages and do the standard login procedure. The authentication server is responsible for the mutual and the strong part of the authentication. The authentication server is written in C. It provides the following functionality: Allowing the TPM to authenticate and supporting the communication with

the HTTP server. The TPM authentication is based on a message exchange over sockets. Furthermore the user session will be created in PHP. It is therefore mandatory to inform PHP about a successful authentication.

The messages exchanged for the transaction confirmation use the GSM network. Therefore the authentication server only needs to check for incoming SMS and verify if they correspond to a transaction confirmation. If this is the case, the HTTP server is informed about a successful transaction.

The communication between the authentication server and the HTTP server relies on a broadcast server. The authentication server broadcasts successful TPM authentication and successful transaction confirmations over a broadcast channel. The HTTP server can listen to this message in order to decide whether the validation has been successful or not.

The SMS one-time-password implementation is slightly different. In order to authenticate, a user has to give its credentials plus the one-time-password received on his mobile cell phone. Just as for the TPM mutual authentication and for the transaction confirmation, the user session has to be created in PHP. However Gammu only needs to send an SMS. Sending an SMS through Gammu means to execute an INSERT query on the database which corresponds to Gammu. Since the answer will be sent back through the web page, there is no need for an additional authentication server listening to incoming SMS.

5.5 Evaluation

Having a more secure way to authenticate is nice. Being transparent to the user is also nice. But all this is worth nothing if the performance is bad. By performance we understand the additional server load due to the encryption and decryption of the messages, the additional server load due to the Gammu SMS server as well as the time needed to authenticate successfully to the system. These three measures are important. The first two for the service provider: if the increase of the server load is too high, no one will provide this type of authentication. The third measure is important for the user: the advantage of e-commerce applications is the rapidity of the service. If the login decreases the performance of the system too much, it looses this advantage.

Measuring the time a login takes is quite easy: just take a stop watch and do a login. However measuring the additional CPU consumption on the server is difficult. Having an accurate measure would require doing multiple

connections from multiple clients at the same time, first without the TPM authentication and then with. Unfortunately it was not possible due to limited hardware resources, and therefore all measures have to be done using one server and one client with a TPM enabled machine, which still gives a good estimate of the resource consumption. The setup is as follows:

- The server is an Intel® Pentium® 4 3.00 GHz with 1Go RAM running an Ubuntu 32bit system.
- The client machine is an Intel® Core™2 Duo 2.26GHz with 2Go RAM, and Intel® TPM and running an Ubuntu 64bit system.

In order not to bias the experiment, only one client is authenticating to the server at a time. Besides that, the server and the client only run the necessary software to authenticate (this means a browser window and the authentication protocol on the client side and Apache, Gammu and the authentication server on the server side).

In the first test, 80 successive TPM authentications were done, but between each 5 authentications a pause of 10 minutes was accorded in order to give the TPM the time to clean up some stuff. The experience showed that the mean authentication time is 4.7 seconds and 75% of the measured times are less than 3.73 seconds. Besides that the additional server load due to the encryption and decryption of the exchanged messages stayed below 1%. Theses results are encouraging and prove that the implementation is usable.

The second and third security enhancements needs some additional hardware to enable the server to send and receive SMS. Gammu allows the sending and receiving of SMS through a cell-phone or other GSM capable device (like UMTS cards). For the following experiences a *Sony Ericsson K800i* was used as hardware back-end for Gammu. In order to evaluate the performance of the one-time-password login the following experiment was repeated 10 times and the time for each login measured:

- Enter the username and password.
- Wait for the SMS containing the one-time-password.
- Copy the received one-time-password into the browser.
- Submit the form to the server.
- Wait for the successful authentication.

The experiment showed that the mean authentication time is 19.5 seconds. Furthermore 75% of the login attempts took less than 21 seconds. Even if this is 4 times longer than the TPM authentication, this result is still quite satisfying compared to actually used systems.

The third test validated the usability of the transaction confirmation over the GSM network. Since two messages are exchanged over the GSM network the transaction confirmation takes more time than the one-time-password authentication. Experiments showed that the mean confirmation time is of 27.1 seconds. The experiment was made under the assumptions that the measurer knew how to reply to an SMS. Furthermore, the phone-number of the e-commerce application was the first number in the address-book of the cell-phone. This means that it took only four steps to insert the phone-number of the e-commerce application into the SMS. As before we can assume the time needed by the web-server to catch the submitted form and display the success page as negligible. The experiment showed that 75% of all mutual confirmations took less than 28.7 seconds. This make this system quite usable.

For the three presented systems it was proven that the measures follow a normal distribution which confirms that there are no side effect appearing with an increasing number of authentications/ confirmations. However the measures for the one-time-password and the mutual transaction confirmation only gives a rough idea of the performance of the system. In fact the receiving and sending of SMS depends on parameters relying on the GSM provider. Sending an SMS at New Year's eve will take much more time than sending the same SMS an ordinary Sunday morning.

6 CONCLUSION

This work presents a promising proof of concept implementation of a secure e-commerce authentication protocol formerly proposed by the authors. As the evaluation has shown, the presented protocol is usable in practice even in that early non-optimized state. The implementation proves to be done easily and is transparent to the user. Obviously the protocol also introduces a new degree of complexity as it requires either a TPM or a mobile phone. Furthermore the offline handshake option requires some work before connecting to a service. Therefore the e-commerce providers have to calculate the risk of those attacks for their application. It is clear that online-banking applications need more security, like using the TPM, than simple online-shops

that might be secured sufficiently using the one time password. Concluding one can say, that the proposed protocol is a promising approach in order to implement a user-friendly and secure e-commerce authentication method.

References

[1] C. Latze and U. Ultes-Nitsche. Stronger Authentication in E-Commerce. How to Protect Even Naive Users Against Phishing, Pharming and MITM Attacks Communication Systems, Networks, and Applications (CSNA 2007), Beijing, China, October 2007

[2] A. Ruppen Enabling stronger authentication mechanisms in todays e-commerce applications Master Thesis, April 2009

[3] The Trusted Computing Group. Trusted Computing Platform Alliance. Main Specification Version 1.1b Trusted Computing Group 2003 Available at https://www.trustedcomputinggroup.org/specs/TPM

[4] SanDisk SanDisk Mega SIM (tm) Available at http://www.sandisk.com/oem/productcatalog(1271)-.aspx

[5] Gemalto Gemalto Multimedia SIM Available at http://www.gemalto.com/telecom

[6] The Trusted Computing Group. TCG Mobile Trusted Module Specification Trusted Computing Group 2008 Available at https://www.trustedcomputinggroup.org/specs/mobilephone/

[7] The AVISPA Project http://www.avispa-project.org/

[8] AVISPA Web Tool http://www.avispa-project.org/web-interface/index.php

[9] TrouSerS http://trousers.sf.net

[10] GNU All Mobile Management Utilities http://www.gammu.org

[11] TPM Emulator http://tpm-emulator.berlios.de/