Global Information Assurance Certification Paper

Secure Authentication on the Internet

GSEC Gold Certification

Author: Roger Meyer

Adviser: Carlos Cid

Accepted: 04 April 2007

© SANS Institute 2007, Author retains full rights.

Table of Contents

1 Abstract

Malicious applications targeting financial account information have increased dramatically over the last years. The number of online applications is growing strong. The ease of use of the Internet and the growing user base make a perfect target for criminals. Attacking thousands of users is achievable with only one click.

The methods used by these criminals vary immensely, but they have one thing in common: they are getting more and more sophisticated. With these increasing threats, governments are issuing stronger legislations and companies are realizing that their current systems can not thwart current attacks anymore.

To counter these threats, current authentication systems have to be adopted. Not only the criminal side has made advances in the last years. The security industry developed new mechanisms and protection systems to thwart even the most sophisticated attacks.

This paper covers current Internet authentication mechanisms and possible attacks. It helps the reader to understand todays issues with authentication mechanisms. To understand the attack vectors, one has to know the current attack trends. Authentication systems can be classified according to their resistance against common attacks. Ten different authentication systems will be introduced and classified accordingly.

Applying state-of-the-art authentication mechanisms described in this paper, current attacks can be thwarted and users are given a technology they can trust again and safely operate on the Internet.

2 Introduction to Authentication

2.1 What is Authentication?

The term authentication describes the process of verifying the identity of a person or entity. It is the process of determining whether someone or something is, in fact, who or what it is declared to be. Authentication is part of most online applications. Before a user can access its email account, its online banking account or its favorite online shopping account, it has to identify and authenticate itself to the application. The most common form of authentication is done through the use of passwords.

Before describing the process of the authentication, we explain some terms. In this context, AAA is often used. AAA stands for Authentication, Authorization and Accounting. It is important to know the differences between those terms:

Authentication: the confirmation that a user is who it is claiming to be.

Authorization: the process to determine whether the user has the authority to issue certain commands.

Accounting: measuring the resources a user consumes during access.

Identification: Identification is the process that enables recognition of a user described to an automated data processing system.

The login process consists of the following steps. To be

recognized by an application, the user has to identify itself. Identification is achieved through the presentation of its credentials. The next step – authentication – essentially verifies a user's claimed identity. Once authenticated, the authorization defines what a user can see and do in the application. The accounting process is keeping track of user actions during all steps.

This process – like most security processes – is a chain. If one chain element breaks, the entire chain falls apart. This emphasizes the importance of the authentication system. Authorization and accounting are only possible if users are correctly identified and authenticated.

2.2 The need for secure Authentication

Internet usage and online applications are experiencing spectacular growth. Worldwide, there are over a billion Internet users at present. A big reason for the success of the Internet is the simplicity and that you can access the applications from anywhere. This growth in popularity has not gone unnoticed by the criminal element – the simplicity of the HTTP protocol makes it easy to steal and spoof identity. The business liability associated with protecting online information has increased significantly and this is an issue that must be addressed.

Online fraud has become a major source of revenue for criminals all over the globe. This has made detecting and preventing these activities a top priority for every major company. Most disturbing is the recent increase in the number of attacks and the evolution of their techniques. As repeatedly noted by the Anti-Phishing Working Group[1], Phishing attacks happen at a

1 The Anti-Phishing Working Group (APWG) is the global pan-industrial and law enforcement association focused on eliminating the fraud and identity theft

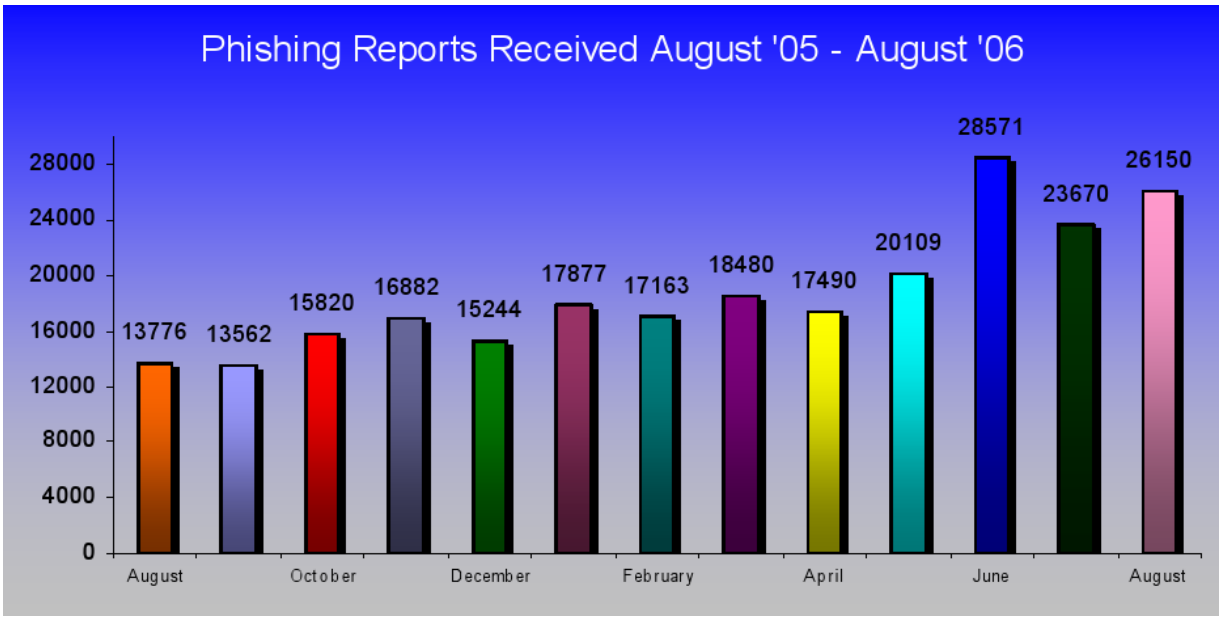

Illustration 1: APWG: Phishing Reports Received between August 2005 and August 2006 [Image: Anti-Phishing Working Group (APWG: Phishing Activity Trends Report, 2006)]

constantly increasing rate:

- The total number of unique Phishing reports submitted to APWG in August 2006 was 26150.
- This represents an increase of nearly three thousand attacks from July 2006 and is the second most ever recorded. (APWG: Phishing Activity Trends Report, 2006)

Around 10 years ago there were only a very limited amount of online applications. This has changed rapidly with the growth in Internet users. Today, almost everything that can be done offline has an online counterpart. This goes from simple email access, to paying your bills online, managing your funds with online banking and reporting your taxes online. All those applications have one thing in common, they need a secure authentication prior the user can access a service.

that result from phishing, pharming and email spoofing of all types. http://www.antiphishing.org/

2.3 The Challenges in secure Authentication

The security of an application is always a trade off between a high level of security and more usability. The more security is added to an authentication system (pass phrases instead of passwords, multiple authentication tokens), the lower will be the acceptance rate of the users and the usability will decrease. It is a big challenge to find the most secure authentication system which is accepted by the users.

Users always want new applications and features with easy to use interfaces. At the same time they are worried about the increasing dangers. Moreover, new legislations are pushing manufactures and companies to protect the privacy of their clients.

The increasing mobility of the users is another important factor. The users want to access their applications not only with their desktop at home, but also in their office, on vacation with their PDA and everywhere with their cell phone. Those needs pose significant requirements to the security of applications.

These broad demands from the users create a wide range of attack vectors. Phishing, identity theft, spyware, malware, keyloggers, javascript attacks, and generally untrusted consumer platforms all make the traditional means of password-based authentication ever more complicated.

2.4 Regulatory Issues

In October 2005, the Federal Financial Institutions Examination Council (FFIEC), a US government agency, issued what it called guidance (FFIEC, 2005) but which looks much like a mandate. Starting in January 2007, financial institutions must provide consumers of online financial services with strong authentication (minimum two-factor authentication).

Despite these strong measures, the reaction to the FFIEC guidance was quite low. Bruce Schneier, an internationally renowned security technologist and author, who wrote about the failure of two-factor authentication already in March 2005 (Schneier, 2005), pointed out that this will not help. He says that two-factor authentication will not mitigate identity theft, because it is not an authentication problem – it is a problem with fraudulent transactions. He continues saying that "authenticating the person is not the way to proceed, the problem is fraudulent transactions. [...] To mitigate that risk, we need to concentrate on detecting and preventing fraudulent transactions. We need to make the entity that is in the best position to mitigate the risk to be responsible for that risk. And that means making the financial institutions liable for fraudulent transactions."

Despite this statement a robust authentication helps preventing many problems with fraudulent transactions later. And, smart authentication mechanisms can be extended to authenticate transactions to obtain maximum security (see section 2.5).

2.5 Transaction Authentication

The ultimate goal we are trying to prevent is financial loss. Yet, the focus is laid on the user authentication, instead of the transaction. Paul McGowan, in his essay "Gone Phishing...", brings it to the point: "While it is incredibly difficult to prevent the bad guys from stealing access credentials (especially with browsers like Internet Explorer around), it is actually much simpler to prevent your money disappearing off to some foreign country." (McGowan, 2005) Bruce Schneier explains in his countless blog entries, that "the solution is not to better authenticate the person, but to authenticate the transaction." (Schneier, 2006)

Transaction authentication can be achieved through either

passively monitoring transactions and flagging suspicious transactions for additional authentication or actively requesting additional authentication in the application.

Reactive back-end systems that monitor for suspicious behavior are of the first kind. They can be rule-based (e.g. all transactions with amounts over a certain limit) or based on a risk score (e.g. transactions to a different country and users logged in from a foreign country). Flagged transactions can get bumped to second-factor authentication – usually, a call on the telephone, something the user has.

Active systems are directly built in the online application. Some or all transactions require additional authentication like a one time password (OTP) or a transaction authentication number (TAN)[2]. It is critical that the transactional authentication be cryptographically distinct from the session authentication mechanism or the attacker will try to get the user to re-authenticate for the session. This is only achievable with a tamper proof, secure channel from the user to the application. An additional device (the user owns) builds a cryptographically secured channel to the web server and displays arbitrary messages on its display, usually detailed information of the transaction (see Illustration 2).

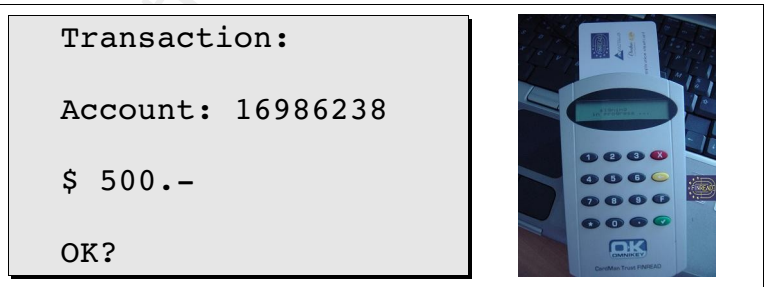

Illustration 2: Example of what a transaction could look like (Image: FINREAD, FINancial Transactional IC Card READer, a European Commission funded initiative http://www.finread.com/)

2 Transaction authentication number (TAN) http://en.wikipedia.org/wiki/TAN_(banking)

The process works as follows. The user initiates a transaction which requires authorization for this transaction. The critical transaction details (like the amount and the recipient) are securely transferred and displayed on the trusted display of the reader. Given that the user explicitly approves the transaction via the secure keypad, the reader signs the transaction with the Smart Cards signature key. By executing critical operations on the trusted reader device and by involving the user via the trusted reader interfaces, content manipulation attacks can be eliminated.

The devices permitting to sign a transaction will be introduced in the next section 3 (Authentication Systems).

3 Authentication Systems

Authentication methodologies are numerous and range from simple to complex. The level of security provided varies based upon both the technique used and the manner in which it is deployed. The most prevalent form is probably the authentication with a user name and a password. Unfortunately it is also one of the most insecure methods. There is an unlimited range of variations of how a user can be authenticated to a web application. Some of the most popular ones are going to be described in the following.

Authentication methods can involve up to three factors:

1. Knowledge: something the user ***knows*** (e.g. a PIN or a Password)
2. Possession: something the user ***has*** (e.g. a Smart Card or a USB Token)
3. Attribute: something the user ***is*** (e.g. biometric characteristics like a fingerprint or the pattern of the eye)

According to these three factors, authentication schemes can be divided in single factor and multi factor authentication:

1. Single Factor Authentication relies on one factor only. Basic user name/password authentication for example is based on something you know.

2. Multi Factor Authentication is based on two or more factors. This can be accomplished through software (e.g. a software certificate), hardware (e.g. Smart Card or USB token) or any out-of-band approaches of one time passwords (e.g. via SMS or E-Mail). These methods have varying levels of security and impose different levels of inconvenience to the end user. An example is an ATM card. The card represents something you have, the PIN represents something you know, hence it is a two factor authentication.

Additionally, the notion of Mutual Authentication is used in this context. Mutual Authentication gives the user a simple way to verify that they are really connected to the intended online institution before providing sensitive information.

One reason phishing attacks are successful is that unsuspecting users cannot determine they are being directed to spoofed Web sites during the collection stage of an attack. The spoofed sites are so well constructed that casual users cannot tell they are not legitimate. Web site operators can aid users in differentiating legitimate sites from spoofed sites by authenticating their Web site to the user.

Techniques for authenticating a Web site are varied. The use of digital certificates coupled with encrypted communications (e.g. Secure Socket Layer, or SSL) is one; the use of shared secrets such as digital images is another. Digital certificate authentication is generally considered one of the stronger authentication technologies.

In the following sections, the most common forms of authentication will be detailed. The next section will list the possible attacks for each of those methods.

3.1 Shared Secrets (Passwords)

Shared secrets (*something a person knows*) are information elements that are known or shared by both the user and the authenticating entity. Passwords and PINs are the primary means of ensuring security and providing access to online applications today. Experience shows that users choose bad passwords. Common choices include the user's real name, login name, date of birth, and simple dictionary words. There are plenty of studies that show that users choose bad passwords and reveal them easily. The latest was a study conducted by Infosecurity Europe 2004 which found that 71% (of respondents) were willing to part with their password for a chocolate bar (Infosecurity Europe, 2004). It also showed that 34% of respondents volunteered their password when asked without even needing to be bribed. The reason why passwords are insufficient is simple: They are too easy for a determined thief to capture.

3.2 One Time Passwords (OTP)

A one time password is – in comparison to static passwords – a dynamic password that changes for every usage. There are two ways a one time password system may work:

- The list can be generated randomly, and a copy kept by the user and the system.
- The list (or, more likely, a specific entry from the list) can be generated on demand by the user and validated by the system.

In the following four methods (scratch lists, time-based,

challenge-based and out-of-band transmission) are explained.

3.2.1 Scratch List

Scratch or grid cards (*something a person has*) contain randomly generated numbers and letters arranged in a row and column format. The size of the card determines the number of cells in the grid. They are cheap and placing the OTP on a wallet-sized plastic card makes it durable and easy to carry and thus very popular. This type of authentication requires no training and, if the card is lost, replacement is relatively easy and inexpensive.

	A	B	C	D	E	F	G	H	I	K
1	YCNv	SBLS	BNMK	84DJ	LMEM	POLK	QICM	M6SC	BSOE	56FG
2	8KUH	CFRE	UJNH	TGBH	PLO9	UKJM	BMZR	QAR5	TBD3	MSFT
3	P9T5	M8QQ	KVR6	7HDF	MVTM	NBGH	6ZDF	AS2S	DED5	POK9
4	VBNM	YXCV	EDC5	SDFG	TZHG	EQTR	7ZH6	4RF6	098O	MKJ6
5	QAXA	SHDK	KDME	XNBL	DG6A	AX5T	POZR	NM4D	14ME	56LP
6	YCNv	SBLS	BNFK	84DJ	OKU9	POLK	QACM	M6SC	BSOE	56FG
7	8KUH	CFRE	UJNH	TGBH	R5TG	UKJM	BMZR	QAR6	TBD4	MSFT
8	P9T6	EATJ	KVR7	7HDF	MVTE	NBGH	6ZDF	AS2S	DED6	TGHZ
9	VBNM	YXCV	EDC6	SDFG	TZHG	EQTR	7ZH7	4RF7	098O	MKJ7
10	QAXA	SHDK	KDME	XNBL	MUT9	AX5T	POZR	NM4D	14ME	VSM9

Illustration 3: Scratch Card

Used in a multi factor authentication process, the user first enters his or her user name and password in the established manner. Assuming the information is entered correctly, the user will then be asked to input, as a second authentication factor, the characters contained in a randomly chosen cell in the grid. The user will respond by typing in the data contained in the grid cell element that corresponds to the challenge coordinates.

The problem with the scratch list is that if the list is compromised, it can then be used for future access. The usability depends a lot on the acceptance rate of the clients. Some find it easy to use, other might see it as a very cumbersome technology.

Illustration 4: Two RSA SecurID® tokens

3.2.2 Time-based

Time-based password systems, such as the

RSA SecurID[3], are another type of one-time passwords. In such a system, the password changes every minute or so, by an algorithm known to the system and the user's authentication device (*something you have*).

This device is typically a small card with a liquid crystal display readout for the current password. It is very easy to use as there is no button to press.

The algorithm to create the password is based on a function of a key and the current time. The key is pre-set at the fabrication of the device. Further, the token has an internal clock. A timer creates a new password in a pre-defined time-range (usually every 60 seconds), using the current time and the key as its input. The function itself is usually proprietary, but is basically a form of the following:

password = encrypt(TIME with KEY)

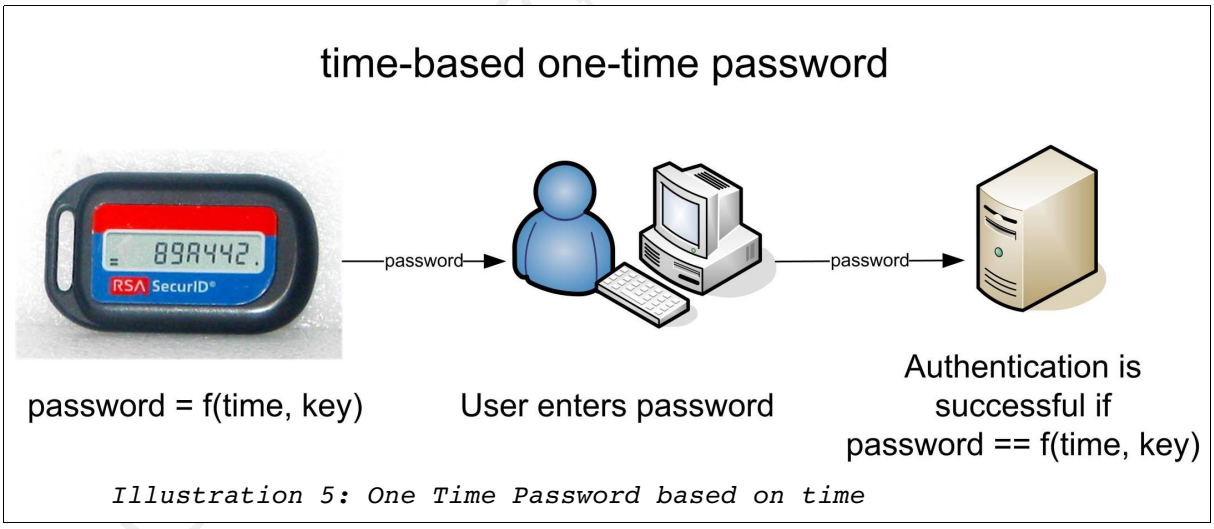

Illustration 5: One Time Password based on time

One challenge of this system is the synchronization of the time (CryptoCard Corp., 2006). The authenticating server has to cope with the drift of the internal clock of the device by computing several passwords with different times (+/- a couple of seconds). Therefore, multiple password will be accepted as correct. Another problem might be the pre-initialized key by the

3 RSA SecurID Authentication: http://www.rsasecurity.com/node.asp?id=1158

manufacturer. Therefore, the manufacturer has a copy of each secret key for every token ever sold. The lifespan of time-based tokens is between 24 and 60 months, at which time the token stops working and has to be replaced. This increases the costs as the tokens and the license are rather expensive. (Chevassut & Siebenlist, 2005)

3.2.3 Challenge-Response Based

A Challenge-Response based system works as follows. When the user attempts to log in, the system generates a random challenge. The user unlocks his token with his PIN, and then enters the challenge. The token calculates a response and displays the result. The user sends the result back to the system as his response to the challenge. All that has passed over the wire (and thus all that could be snooped on) is the random challenge and the encrypted result, not the user's PIN (to unlock the token) or the key.

In comparison to the time based system, here the time is replaced by a challenge, and the SmartCard/Token is optionally protected by a PIN. Instead of using the time as input for creating the password, a server generated challenge is used:

password = encrypt(CHALLENGE with KEY)

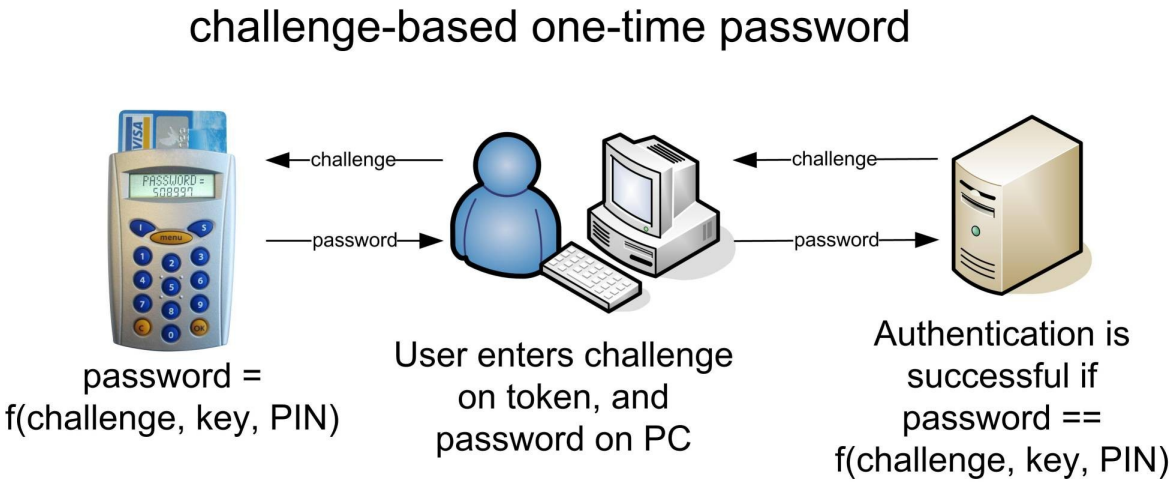

Illustration 6: One Time Password based on a challenge

The system is based on the fact that with the possession of the challenge and the password, it is unfeasible to recover the key or the PIN, or to produce valid responses (passwords) for other challenges.

Challenge based system are considered to be slightly more secure as time based systems as the challenge is controlled by the authenticating party and the additional PIN for unlocking the Card/Token.

3.2.4 Out-of-band Transmission

An out-of-band transmission uses a channel different from the one the user is using to initiate the transaction. This separate channel gives an additional layer of security, as a potential attacker has to intercept both channels. Usually, the out-of-band channel is used for a second authentication factor like a one time password (OTP).

Out-of-band transmissions can be an e-mail, an SMS (Short Message Service) to a mobile phone, a call or a fax. For example, after the institution receives the transaction request, an SMS is sent back to the originating individual with the details of the

transaction including a one time password, which the user has to enter to confirm the transaction. (StrikeForce, 2004)

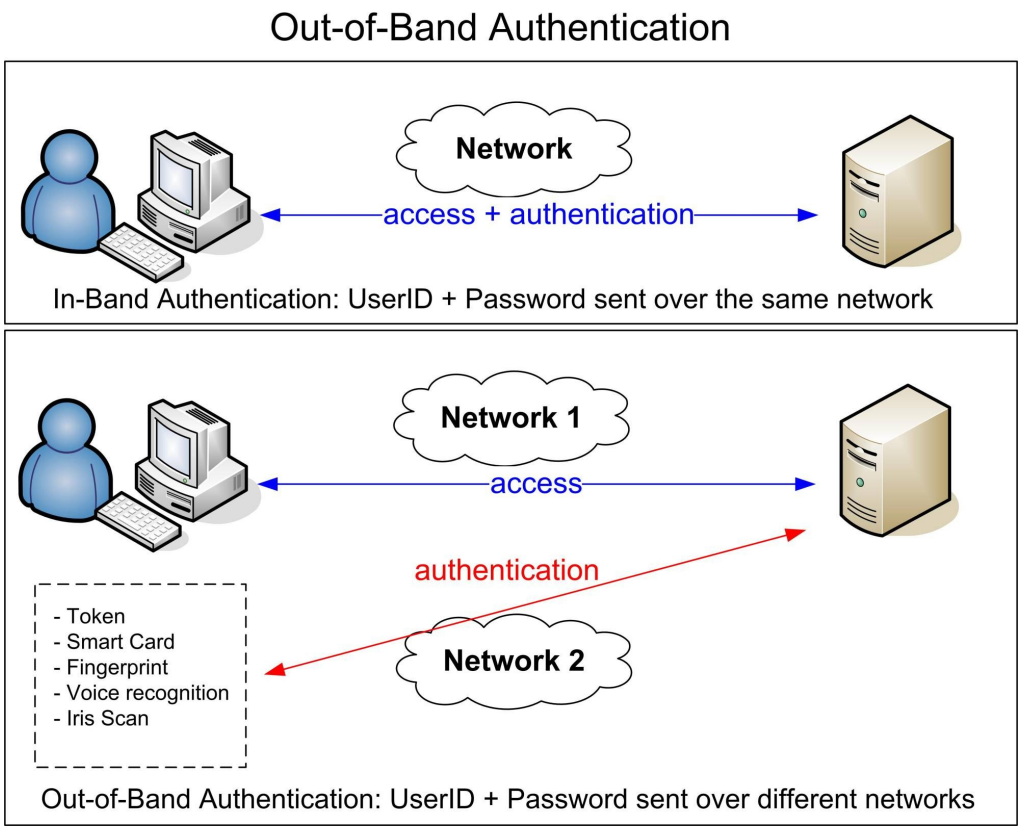

Illustration 7: Out-of-Band methodology creates a separate channel for authenticating a user's credentials

3.3 Software Tokens (SSL Certificates)

A soft token is a software version of a hardware token (see section 3.4). It is a piece of software which is sent to the users' device, such as a cell phone, a PDA, or a laptop. They can be used as the only or as an additional authentication token (*something the user has*).

On the one hand, soft tokens are flexible and less expensive than hardware-based solutions. On the other hand, however, soft-tokens have to deal with (at least) two security problems:

1. Soft tokens are inherently vulnerable to malware and keylogger attacks. Malware can, for example, read out cryptographic keys. Keylogger attacks typically try to retrieve the user credentials when they are typed in.

2. Soft tokens are vulnerable to visual spoofing attacks. (Oppliger, Hauser, Basin, Rodenhaeuser, & Kaiser, 2006)

Both problems are difficult to solve. Soft tokens are most commonly used in the SSL (Secure Sockets Layer) protocol. SSL offers the advantage to optionally authenticate not only the web server, but also the client. Using SSL client authentication is a way of authenticating a client to a server. This form of authentication does not occur at the application layer, it occurs during the connection handshake (transport layer) using SSL certificates. This mutual authentication provides a defense against phishing and similar attacks.

Client SSL certificates are "personal" certificates for each user. In this method, we can authenticate users based on "*something you have,*" using a certificate and a private key corresponding to the client certificate, as well as "*something you know,*" which would be a pass phrase to the private key. When the client browses to a server which requires a client SSL certificate the browser will request directly the pass phrase for the certificate.

Advantages:

- Strong two-factor authentication in a highly secure software implementation
- Convenience with merging of SSL certificates onto a user's personal device and eliminating the need to carry another item
- Support for a wide range of computing platforms and

devices

Disadvantages:

- Can only be used on the host where the software resides
- Vulnerable to malware, keylogger and visual spoofing attacks

3.4 Hardware Tokens

Tokens are physical devices (*something the person has*) and may be part of a multi factor authentication scheme. Two types of tokens are discussed here: the USB token device and the smart card. The devices have the ability to store digital certificates that can be used in a public key infrastructure (PKI) environment and can perform cryptographic operations.

The USB token device plugs directly into a computer's USB port and therefore does not require the installation of any special hardware on the user's computer. Once the USB token is recognized, the user is prompted to enter his or her password (the second authenticating factor) in order to gain access to the computer system.

A smart card is the size of a credit card and contains a microprocessor that enables it to store and process data. To be used, a smart card must be inserted into a compatible reader attached to the user's computer. If the smart card is recognized as valid (first factor), the user is prompted to enter his or her password (second factor) to complete the authentication process.

Smart cards are hard to duplicate and are tamper resistant; thus, they are relatively secure for storing sensitive data and credentials. Smart cards are easy to carry and easy to use. Their primary disadvantage as a consumer authentication device is that they require the installation of a hardware reader and associated

software drivers on the consumer's home computer.

In the following, the properties of hardware tokens are discussed independently of their interface (USB, serial port, optical transmission). Before that, the different Smart Cards and Smart Card readers have to be classified.

3.4.1 Smart Card Classification

A Smart Card often contains a microprocessor (to process data) and memory (to store data) and is complying with ISO 7816 standard[4]. They can be classified on basis of *card components*, *card interface*, and *smart card OS (Dhar, 2003)*.

The components can be divided into *Memory Cards* and *Chip Cards*. Memory Cards are the most common and least expensive cards. They contain a EEPROM (application data) and a ROM (data storage) part. The security logic on the card controls the memory access after a secret code has been provided.

The Chip Cards contain a microprocessor. Additionally to the EEPROM and the ROM part they contain RAM (volatile memory used by the processor) and a CPU which is the heart of the card and carries out the varying instructions. This makes the card more expensive than memory cards. Their capability of executing instructions like encryption and decryption make them ideal for credit and financial cards.

Card interfaces can be contactless (e.g. RFID), contact (requiring a reader) or a a combination of them. In the course of this paper we are only considering contact cards.

There are many different Operating Systems for smart cards. The new trend in smart card operating systems is the JavaCard[5] Operating System. JavaCard OS was developed by Sun Microsystems.

4 Smart Card Standards http://www.smartcardbasics.com/standards.html
5 Java Card technology http://java.sun.com/products/javacard/

Java Card OS is popular because it gives independence to the programmers over architecture and Java OS based applications can be used on any vendor of smart cards that support the JavaCard OS.

The question of Public Key vs. Symmetric Key encryption is usually a decision of scalability vs. speed (Kaliski, 1998). The management of public keys is much easier, but symmetric cryptography is up to 100 times faster than asymmetric cryptography. A hybrid approach is usually a good compromise, where the smart card establishes a symmetric key via server's public key. For the user authentication, the key distribution and the data protection a symmetric key will be used.

3.4.2 Reader Classification

Smart cards need a way to interact with their users. Since there is no built-in display capability in most cards, the card reader must take on this responsibility. Any display used during critical transactions, such as transferring money, needs to have two properties: the display must be trustworthy, and it must be unspoofable. Making sure a terminal presents proper and trustworthy information to a user is really a trust issue. Consumers will use a PC to interact with the smart card. The problem is that PCs are notoriously insecure. If your PC cannot be trusted, how can you believe that what it is telling you on behalf of your smart card is correct? In fact, one excellent reason for using smart cards in the first place is that PCs cannot be trusted. The reasoning goes that it is better to store secrets like PINs, sensitive personal data, and private keys on a smart card than on a PC. That way, if the PC is compromised, your secrets cannot be so easily stolen. (McGraw & Felten, 1999)

What is needed is a trusted display. The following table lists the reader classes and their features. The classification of

card readers into classes 1 through 4 is, unfortunately, not based on any specification according to ISO[6], or other, similar organizations. It is commonly used though (KOBIL, 2003).

	Class 1	*Class 2*	*Class 3*	*Class 4*
Description	simple contact unit	including PIN pad	including display	including extra auth-entication module
PIN entry	-	key pad	key pad	key pad
UI (User Interface)	-	LED	LCD display	LCD display

Table 1: Classification of card readers

Class 1 readers are simple contact units to establish a connection between a PC and a smart card. There is no logic on the unit.

Illustration 8: Class 1 reader

Class 2 readers come with a key pad to safely enter the PIN to unlock the smart card. Once the user has entered the PIN, the reader "unlocks" the smart card and the PC communicates directly with the smart card.

In addition to class 2 readers capability, **Class 3** readers are equipped with a LCD screen to display arbitrary text. Here, the PC is communicating with the reader device and may send encrypted text from the online application to the reader, which decrypts it and displays it on the screen so the user can verify it (transaction signing).

Illustration 9: Class 3 reader

Class 4 readers include a security module with their own identity. This module signs the transactions, which are sent over this terminal. This ensures that only approved readers can make transactions. Transactions can be traced back to which reader made

6 International Organization for Standardization http://www.iso.org/

what transaction.

3.4.3 Class 1: without SSL Client Authentication

A class 1 hardware device consists of a Smart Card and a simple Smart Card reader which is connected to the PC. This usually requires some additional software to be installed on the client PC. The system is based on PKI, the private key is securely stored on the Smart Card. Before the user can login to his web application he has to unlock the Smart Card with a PIN, which he enters on the PC. This is a multi factor authentication; the PIN is the first factor and the Smart Card the second.

3.4.4 Class 1: with SSL Client Authentication

This system uses the same hardware as the class 1 reader described in the previous section, except that additionally the private key on the Smart Card is used to build a mutual authenticated SSL connection with a client certificate. This mutual authenticated connection prevents Man-in-the-Middle attacks and authenticates the user to the web site.

3.4.5 Class 3: with SSL Client Authentication

A class 3 reader has a PIN pad and a small display. The PIN pad is used to enter the client PIN. This prevents trojans and keyloggers that might be installed on the client's PC to record the PIN. The display is used to send arbitrary, encrypted information from the server to the client. This is especially used for transaction signing. Transaction signing allows that every transaction (or some transactions consolidated into one) optionally has to be confirmed (signed) by the client. As we have a secure channel from the server to the client hardware reader, the display is a tamper-proof and secure way to display transaction information and have the client confirm each

transaction.

3.4.6 Tokens with optical Transmission

Next to hardware readers connected directly to the users PC to set up a secure link, there are other means of setting up a secure channel. One way is an optical transmission[7]. For this, the user's screen is used as a one way transmission of strong encrypted data into the token. The server emits a dynamic graphic/picture which is read by the optical interface of the reader. A special optical interface technology is used that allows the transmission of an encrypted message over any type of screen technology. This secure channel is most useful if applied with transaction signing, where the user has to confirm each transaction (which is going to be displayed on the screen along with a OTP).

4 Attacks

Internet authentication methods can be classified according to their resistance against common attacks. According to (Hiltgen et al., 2006), attacks can be divided into two common types: offline credential stealing attacks and online channel breaking attacks.

1. **Offline credential stealing attacks** try to gather a user's credentials. This can be accomplished by some malicious software such as a virus or trojan horse on the user's PC or by tricking the user to voluntarily reveal his credentials through phishing attacks.

2. **Online channel breaking attacks** are the most sophisticated attacks today. Instead of trying to get hold of a user's credentials, messages between the client PC and the

7 The AXSionics Authentication System http://www.axsionics.com/

server are unnoticeable intercepted. This can happen when the intruder is masquerading as the server to the client and as the client to the server, respectively.

Based on the above taxonomy, the different authentication methods are going to be classified accordingly. Illustration 10 shows the possible places of attack. In the past we have seen lots of attacks on the server side. Recently, attacks have shifted away to the user, the user's PC and the network. The future will focus on the client side (the user's PC). Some very sophisticated trojan horses have appeared already – more on trojan horses later.

Let us review some of the methods attackers can use. These are only some of the possible scenarios.

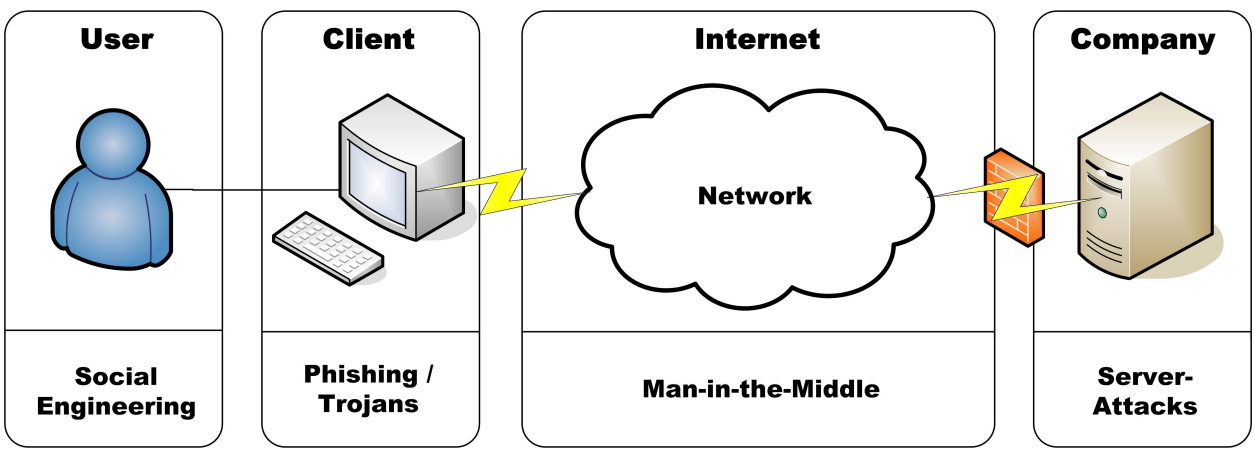

Illustration 10: Types of attacks

4.1 Phishing

Phishing is simply a form of modern day social engineering that employs both technical and non-technical methods for the purpose of extracting personal, confidential information. Confidential information can be a user name, login ID, account number, password or any information which leads to a certain user.

Social-engineering schemes use "spoofed" e-mails to lead consumers to counterfeit websites designed to trick recipients into divulging financial data. They have a technical and a non-

technical part. The technical part mostly consists of an e-mail with a spoofed sender address of the targeted institute and an authentic looking body, asking the user to react. The user then has to click on a link which takes him to the fake attacker's website, tricking users into entering their credentials into some faked web form. The success rate of such phishing emails are astonishing high and are at least partially founded because most users actually do not know how to reliably identify a genuine web site/server (Schechter, Dhamija, & Ozment, 2007).

Conventional phishing is an offline attack, as the attack aims at fraudulently gathering a user's credentials. However, Phishing attacks can also be online attacks: The attacker collects the users credentials, uses them immediately to login to the application and executes the fraud transaction immediately. Such attacks, executed in real-time will be seen more often in the near future.

Studies by the Anti Phishing Working Group (APWG) done in 2004 have concluded that phishing attacks are likely to succeed with a chance of 5% on all message recipients. (APWG, 2004) To defend against this type of attack common sense and user education is enough in most cases.

Authentication systems which are based on static passwords (like scratch lists) are especially vulnerable to phishing attacks. Dynamic codes automatically generated by devices are less susceptible to this form of attack. (Wüest, 2005) (Ollmann, 2004)

4.2 Man in the Middle

A man in the middle attack is one in which the attacker intercepts messages and then retransmits them, substituting his own message for the requested one, so that the two original parties still appear to be communicating with each other. In a man

in the middle attack, the intruder uses a program that appears to be the server to the client and appears to be the client to the server. The attack may be used simply to gain access to the message, or enable the attacker to modify the message before retransmitting it.[8]

There are a few ways for man in the middle attacks to work. One is to put up a look-alike malicious site which is basically a proxy to the actual website. When the victim logs in with proper credentials, the attacker can simply ride on that established online session. Note that even a challenge-response type of token would work in this case because the attacker (or the man in the middle) is passively observing the connection between the server and the victim. The challenge will reach the victim, who will then send in the response. The attacker simply proxies the traffic until the session is established and then sends the fraudulent transaction.

MITM attacks can also happen locally on the victim's PC with a trojan type of attack. Local MITM attacks can support remote MITM attacks with a local redirection (by manipulating the hosts file for example which maps domain names to IP addresses) to increase the success rate.

8 Definition of man in the middle attack at searchsecurity.com: http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci499492,00.html

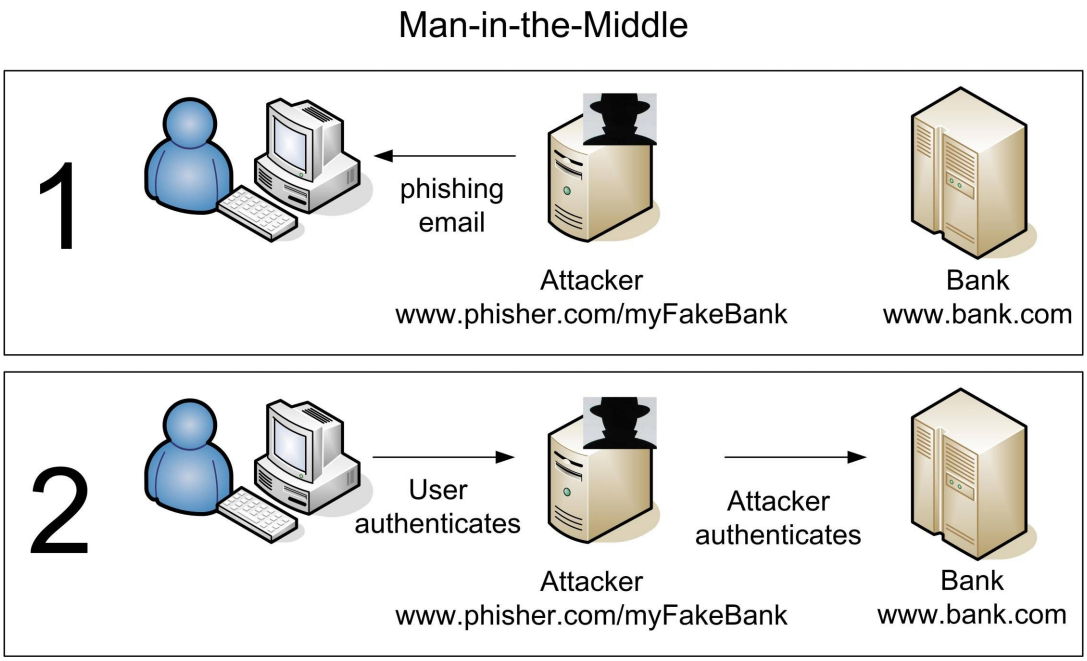

Illustration 11: Man-in-the-Middle

1. Phisher sends fake E-Mail to victim directing them to fake website

2. User authenticates to fake website and Phisher instantly logs in to real website with the user's credentials

The challenge in remote MITM attacks relies on the faking of the actual website and the SSL certificate. SSL certificates are issued to a certain domain name (specified in the CN – the Common Name[9]). If the domain name does not match the one in the certificate, the browser will issue a warning. Likewise if the certification is not signed by a trusted Certificate Authority (CA) like Verisign, the browser will issue a warning. But as most users can not tell the difference between a self signed SSL certificate and a signed one or just simply ignore all warnings, most attacks will still success. (Schechter, Dhamija, & Ozment, 2007)

Mechanisms to detect MITM attacks are achievable with mutual authentication, for example through client authentication on the

9 SSL Certificates HOWTO: What is SSL and what are Certificates? http://tldp.org/HOWTO/SSL-Certificates-HOWTO/x64.html

server with SSL certificates.

4.3 Malware (Trojan Horses, Viruses)

Traditional Trojan Horses and Viruses are very common today. These trojans generally steal credentials by logging the keystrokes or fake login forms and then send the credentials to a collection site for later exploitation by the attackers. Most such malware are offline credential stealing attacks. However they can operate in real-time and become online attacks by sending the collected information in real-time to the attacker which can then login with the still valid credentials.

Another possible attack method arises when malware sitting on a users PC submits automated transactions. The trojan would have to detect the user logging in and then submits the transaction using the user's session.

Generally, as malware is installed on the victims machine, it has access to all input and output on the system and thereby access to saved passwords, codes and certificates.

4.3.1 Advanced Malware

Todays probably most sophisticated malware come as extensions to browsers. Microsoft's Internet Explorer allows with Browser Helper Objects (BHO)[10] to add new features as plugins. The counterpart for Firefox are Extensions[11]. Trojans masqueraded as plugins to browsers are so powerful because they operate inside the browser. Once installed, they have all the resources to see and manipulate any traffic the user sees and enters. There are currently trojans available (Websense, 2006) that wait for the user to post personal information to a monitored website, capture

10 Browser Helper Objects (BHO): http://en.wikipedia.org/wiki/Browser_Helper_Object
11 Firefox Extensions: https://addons.mozilla.org/en-US/firefox/browse/type:1

the submitted data and send it back to the attacker.

For such kind of malware, it is even possible to send fraudulent transactions without sending any data back to the attacker. This is possible by intercepting the user's post request and replacing the relevant fields like the amount and the recipient account number. The malware will then also replace the confirmation page sent back to the user with the correct, faked data. Once such kind of malware is installed, the user has almost no possibility to verify or recognize fraudulent activity.

This kind of attack is classified as an online channel breaking attack and can only be defeated through the introduction of transaction signing authentication. This can be accomplished by having the user acknowledging each transaction through a tamper-proof, out-of-band channel (see section 2.5).

5 Comparison

This section compares the resistance of the introduced authentication types in section 3 against the previously listed attack vectors. Illustration 12 graphically displays a comparison of the discussed authentication methods.

Passive Phishing attacks are most successful with static passwords. When dynamic codes and passwords that are generated by a device on demand are used, such attacks become impossible as phishers try to get hold of already existing credentials. All discussed authentication types provide optimal protection except static passwords and scratch/grid cards. Grid cards provide a slightly better security compared to scratch cards as the user does not know which grid is next. If an attacker is asking a user to enter the next 10 grid card entries (on a card with 100 entries) on a fake website, chances are 10% to catch a valid entry. If the grid is randomly chosen and each one is used only

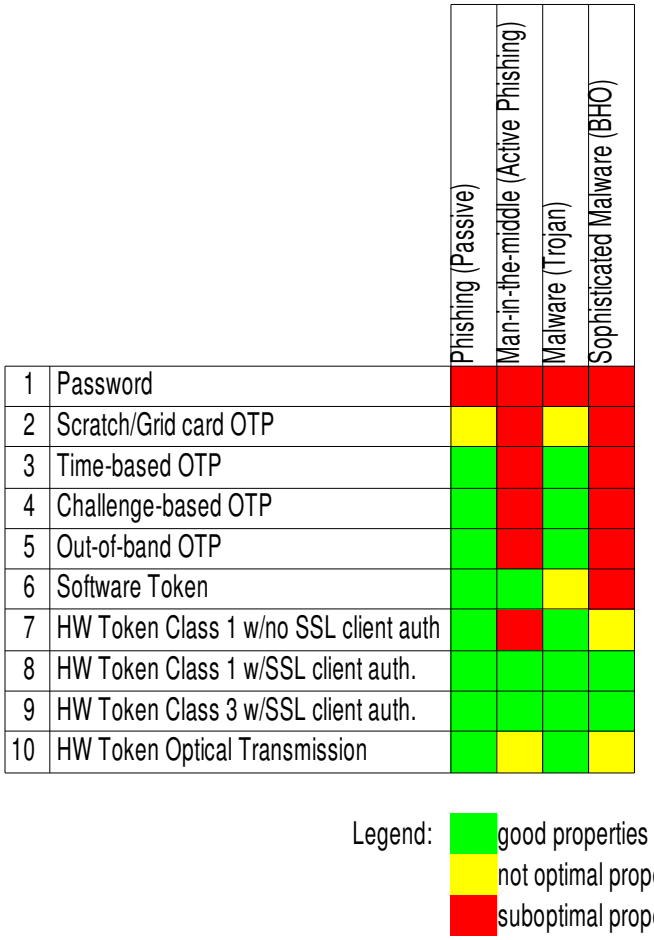

Illustration 12: Comparison of authentication types

once, the chances that the attacker will pick a valid entry are much lower.

To protect against **Man in the Middle (Active Phishing)** attacks, a user has to be given a tamper-proof way to authenticate the server. Todays websites, secured with the SSL protocol, rely on the user checking the server's SSL certificate to verify its authenticity. Authentication systems relying on the user are the static password system, all OTP systems and the hardware token / Smart Card class 1 reader which does not use the client certificate to verify the website. The hardware token with the optical transmission is a special case in that it does not establish a mutually secured channel. As long as this system is only used to authenticate the login itself, it does not provide

more security than OTP systems. MITM attacks can only be defeated with the extension to authenticate the transaction – where a secure channel from the web server to the client is established.

As **malware** runs on the victim's computer, authentication systems based on software tokens are most vulnerable to such attacks. Malware can easily read such tokens saved on the victim's machine. Grid cards and static passwords do not provide optimal properties because of their long validity of the credentials. All other authentication systems are well protected against malware.

Sophisticated malware can only be 100% thwarted with hardware tokens which provide SSL client authentication. The token with optical transmission is again a special case. Only with transaction authentication this attack can provide good properties. For all the other authentication types it is very difficult to defy against this kind of advanced malware.

6 Conclusion

The number of online applications is skyrocketing. No company wants to miss the opportunity to introduce its business to the online world. But this rush comes with a price: the security of new applications is mostly neglected – especially the authentication.

The use of single-factor authentication, such as user name and password, is inadequate for guarding against account fraud and identity theft in sensitive online services (FDIC, 2004). The introduction of additional authentication provides an added level of security.

Todays attacks are mostly offline-based phishing attacks. But attackers are adapting to new authentication systems very fast. Many online services are still vulnerable to basic offline

password stealing attacks.

The biggest problem is the contaminated client systems. To thwart todays most sophisticated trojans, an authentication system has to be secure against online channel breaking attacks.

Another challenge is mutual authentication. Cryptographic protocols in use such as SSL/TLS can be extended to support strong mutual authentication. Depending on the desired security, the private keys and certificates can be stored on tamper proof hardware devices (Smart Cards, USB tokens).

To achieve the maximum security, an authentication system has to thwart online channel breaking attacks and be optionally extensible to transaction authentication.

7 References

Hiltgen, A., Kramp, T., & Weigold, T. (2006). Secure Internet Banking Authentication. IEEE Security and Privacy. vol. 04, no. 2, pp. 21-29.

FDIC. (2004). Putting an End to Account-Hijacking Identity Theft. Federal Deposit Insurance Corporation, Division of Supervision and Consumer Protection, Technology Supervision Branch.

Wüest, C. (2005). Phishing In The Middle Of The Stream - Today's Threats To Online Banking. Symantec Security Response, Dublin.

Ollmann, G. (2004). The Phishing Guide. NGSSoftware Insight Security Research.

Anti-Phishing Working Group, (August, 2006). Phishing Activity Trends Report

Schneier, B. (2005). Two-factor authentication: too little, too late. Communications of the ACM. Volume 48 , Issue 4, 136.

Schneier, B. (2005). The Failure of Two-Factor Authentication.

McGowan, P. (2005). Gone Phishing...

Schneier, B. (2006). Fighting Fraudulent Transactions.

Dhar, S. (2003). Introduction to Smart Cards. Auerbach Publications.

Schechter, S., Dhamija, R., Ozment, A. (2007). The Emperor's New Security Indicators: An evaluation of website authentication and the effect of role playing on usability studies

CRYPTOCard (2006). Time-Based vs Event-Based Two-Factor Authentication

KOBIL Systems GmbH (2003). KOBIL Smart Card Terminals

StrikeForce Technologies, Inc. (28 June 2004). Out of Band Methodology

Chevassut, O., Siebenlist, F. (2005). Secure (One-Time-) Password Authentication for the Globus Toolkit

McGraw, G., Felten, E. (1999). Securing Java. Chapter Eight: Java Card Security: How Smart Cards and Java Mix. John Wiley & Sons, Inc

Oppliger, R., Hauser, R., Basin, D., Rodenhaeuser, A., Kaiser, B. (2006). A Proof of concept Implementation of SSL/TLS - Session-Aware User Authentication

FFIEC. (October 2005). Guidance on Authentication in Internet Banking Environment. Federal Financial Institutions Examination Council (FFIEC).

Infosecurity Europe 2004. (2004). Infosecurity Europe 2004 Information Security Survey. http://www.passwordresearch.com/stats/study57.html

Kaliski, B. (1998). Some Perspectives on Smart Card Cryptography. RSA Laboratories.

APWG (2004). APWG Press Release 20-Apr-2004.

Websense (2006). Websense® Security Labs Alert: new phishing Trojan which installs itself as an Internet Explorer Browser Helper Object (BHO)