V INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS "BIHTEL 2004"

E-banking Fat Client Security Analysis

Saša Mrdović
sasa.mrdovic@etf.unsa.ba

Abstract: *The paper examines the security of e-banking fat client. Strong authentication built in fat client is not always applied to overall system. Fat clients store data locally, and that data might not be protected with the same strong authentication. It is possible to bypass such a fat client and access e-banking data stored locally directly, and in this way effectively reduce data security to the level of security provided for local storage. The paper suggests the use of cryptographic data storage. It would ensure overall security to be equal to the strong authentication required to use fat client.*

Keywords: *e-banking, fat client, information security, cryptography*

1. INTRODUCTION

Electronic banking has become a standard offering from the banks worldwide. It enables users to make monetary transactions from the comfort of their homes or any other place that has internet access, what would traditionally require a trip to the bank.

The security always had an important role in computer and banking systems. When those two systems are combined, as in electronic banking, security becomes a crucial priority. Any security breach could result in direct financial loss for a client or a bank or both.

There are two main types of client applications used for e-banking, thin and fat client. Thin client application, usually web browser, requires no special installation on user-client side and it provides only user interface layer while all the logic and data reside on bank server.

Fat client e-banking includes full blown application installed on user-client side. This application includes user interface, logic and some local storage for data. Fat client may have a more comfortable user interface and a better error handling on client side. Also, it is a very convenient solution for clients with a slow or unreliable internet connection.

Local storage of data could be weak point from the security point of view for of e-banking fat client unless it is protected in the same way as the access to bank server, which is usually not the case. This paper will examine in more detail potential security weaknesses of fat client e-banking. It will point to the issues that need to be considered and show some real world examples that do not provide advertised information security. The paper will suggest an approach that ensures adequate information security through confidentiality, data integrity, authentication and non-repudiation.

2. PROBLEM

The main problem that will be considered in this paper is a false sense of security that e-banking customer using fat client may have. If the fat client uses advanced method of authentication, like digital certificates stored on smart cards, customer might have impression that complete system has advanced security. In reality fat client data stored locally may not have to have the same level of protection, and very often they do not.

Reliable customer authentication is imperative for e-banking. Effective authentication can help banks reduce fraud, reputation risk, disclosure of customer information, and promote legal enforceability of their electronic agreements. Methods to authenticate customers are:

- Passwords and PINS
- Digital certificates & PKI
- Physical devices such as tokens
- Biometric identifiers

In practice big US banks still relay on the oldest and the least secure method of using username and passwords [1][2]. Big European banks usually offer two methods, passwords and digital certificates [3][4]. Banks in Bosnia and Herzegovina tend to accept only advanced forms of authentication, digital certificates and tokens.

Most of the system security is based on proper client authentication. Authenticated clients are allowed to view their accounts data and make financial transaction from their accounts. This is especially true for thin clients that implement almost no logic and keep no account data on their system.

E-banking fat clients, sometimes referred to as offline banking software, enable users to create multiple payment orders offline and transmit them all together to the bank. Advanced solutions offer layers of authorization, convenient for companies. Certain users can enter payment data that needs to be signed by users authorized for that operation before it is transferred to the bank what can, again, be done only by users with authorization for transfer. Any type of fat client has to store data locally. Locally could mean local computer where client application is running or server within LAN on client side. Important is that it is not stored on the bank server but on the client system.

Access to offline banking application usually requires some form of authentication. Most often this is the same method used for client authentication to bank server. It is assumed that all access of data and their manipulation is performed through the application and therefore it is as secure as authentication method used. Since the data is stored locally it could be possible to access it directly, bypassing authentication. If the security of the locally stored data is lower than application access security, the overall security of the system would be reduced to the level of protection of the locale storage.

Various versions of local storages for e-banking fat client data will be now presented showing possible ways of their access without going through process of authentication. Then, possible malicious changes to stored data with corresponding effects will be considered.

2.1. Unauthorized access to local storage

First case that is presented is one that initiated this paper. Fat client requires certificate stored on smart card protected with a PIN, what represents a very strong two-way authentication method. Data that is entered through this protected application is stored in local MS Access database. This Access database is password protected. Implemented solution effectively reduced system security from strong two-way authentication to simple password protection. It is important to point out that simple search on the Internet returns number of free tools for MS Access lost password recovery that could effectively be used to open password protected MS Access database [5]. Now the data protection is reduced to simple file protection. Anyone that can access the file that holds Access database can access this data. Depending on the operating system used there is little or no protection from anyone with physical access to the computer where database resides. Any Windows 9x OS provides virtually no file access protection for anyone with physical access. Windows NT (2000,XP, 2003) OS line provides some protection that again can be removed by anyone with physical access using tools freely available on the Internet [6]. Even Linux/Unix OS can not protect their files from knowledgeable or just well informed user with physical access [7][8]. This is all mentioned in order to show how perfectly good cryptographic protection could be reduced to pure physical protection due to faulty implementation.

Possible consequences of this unauthorized access to banking data will be considered later. Now some basic variations to the presented case will be examined.

Local banking data can be stored not on a computer where fat client is running but somewhere else on the local network, either other workstation or a server. In any case the level of protection depends on the level of the protection offered by the network not the e-banking fat client.

Local banking data could also be stored in different way from the presented. It could be stored in a file on the computer with fat client, or anywhere on the network. Level of file protection provided by client or network operating system will define security of e-banking data. Data could also be stored in a database, different from MS Access, for instance MySql, MS SQL Server or Oracle. Again data security depends on database security not on fat client security.

Fat client must store data locally and security of those data will not be determined by fat client security but by the weaker of the two: local storage security and fat client security.

There is still an open question: who might have access to any type of local storage that would enable misuse. First consideration is whether the local storage should be considered safe or if the persons with direct or network access are trusted. If they are, then there is no need to protect the fat client at all. In reality there are many cases where persons with access to computer or network should not have access to e-banking data. On shared home computer family members might not want to share their financial data. In a company environment not all users with

network access are allowed to view, change or transmit financial data. In both cases we have users with legitimate and adequate access to computer and network that could, using some effort, gain them unauthorized access to e-banking fat client data.

Another potentially dangerous group are entities from outside, for example anyone on the Internet, that do not have legitimate access to local storages but could obtain one. It is not extremely difficult to obtain illegal access to a computer that is connected to the Internet. If one is determined enough, there are tools and exploits available that do not require special knowledge, executives and scripts ready to go. There is a term 'script kiddies' for users of those tools. SecurityFocus BugTraq mailing list [9] publishes at least 30 new security vulnerabilities every week. For instance, the most recent critical Microsoft security vulnerability in JPEG processing could allow remote code execution [10]. A malicious business competitor could use this vulnerability to create special JPEG picture file that would be mailed as a business correspondence that would enable him access to local storage.

This all might sound somewhat paranoid, but the point is that electronic banking fat clients should provide the security their makers advertise. If a fat client uses advanced forms of authentication, e.g. digital certificates, it should not relay on security of local system where fat client is installed. There are cryptographic ways to protect data that do not depend on operating system, file or database protection. This will be explained further in the part of the paper that discusses possible solutions to security problems presented.

2.2. Possible consequences

After possible ways of gaining unauthorized access to e-banking fat client locally stored data are explained, possible consequences of this access are presented.

Consequences will be explained as violations of combination information and cryptographic security. Information security is usually expressed through, so called CIA triad of, confidentiality, integrity, and availability. Cryptographic security includes confidentiality, data integrity, authentication and non-repudiation.

2.2.1. Confidentiality

Confidentiality is a service used to keep the content of information from all but those authorized to have it [11]. Financial data should be private, and e-banking application should enforce privacy. Unauthorized access to that local storage where e-banking data is stored would enable the intruder to read that data. This clearly violates confidentiality.

2.2.2. Data integrity

Data integrity is a service which addresses the unauthorized alteration of data [11]. E-banking clients must be sure that data they enter or get through fat client application were not changed in any way on the way to and from the bank. Intruders that gain unauthorized access to local e-banking data storage might be able to change data in that storage. In this way data entered through fat client application could be altered before it is sent to the bank. Existing payment orders might be deleted or changed or even new orders might be created.

It might seem difficult to alter data in local storage without knowledge of data structure and logic built in the fat client application that is supposed to manage this data. A malicious user could easily obtain e-banking software by becoming bank customer and analyze various local storage data changes caused by user actions in the fat client. He could learn how to change amounts or account numbers on the existing payment orders in the local storage after they are entered through application before they are sent to the bank.

2.2.3. Availability

Availability is a service that ensures that information systems and the necessary data are available for use when they are needed. Malicious user that gains access to e-banking local storage could make data unavailable without need for any knowledge about data meaning or structure. Simple deletion or alteration of records or even the whole storage could render the data and application useless.

2.2.4. Authentication

Authentication is a service related to identification. This function applies to both entities and information itself [11]. Only authenticated users should be able to view financial data and make monetary transactions. E-banking fat clients require authentication before they can start. Unauthorized access to local storage, as explained earlier, completely bypasses fat client authentication. For this kind of unauthorized access malicious user might have to go through, or break, local storage authentication. Point is that he does not have to authenticate to the e-banking software.

2.2.4. Non-repudiation

Non-repudiation is a service which prevents an entity from denying previous commitments or actions [11]. Bank and its clients should not be able to deny transactions executed

through an e-banking application. Without a proper authentication and ensured data integrity it would be very difficult to prove if bank client actually created and authorized transaction and even if he did, if the transaction data are the same data that client entered. Since data integrity and authentication of fat client are already shown not to be valid in case of unauthorized access to local storage e-banking data, non-repudiation would not hold in case of any dispute.

3. SOLUTION

There are certain measures that e-banking fat client users could take in order to minimize possible negative effects of insecure default setup. Local e-banking data storage has to be made as safe as possible. Safe means something that is not as easily accessible as mentioned MS Access password protected database. Local storage needs to be on a physically safe location, on a computer with limited access to trusted persons only. Proper operating system file system protection needs to be enforced. Systems need to be properly patched and virus protected. This is not a solution but self-defense measures to lessen the danger. The solution has to come from the bank and fat client software makers.

The first solution would be to do nothing to improve security of local storage data. In this solution actual e-banking fat client authentication should be changed to correspond to the authentication required for access to local storage. In this case the overall security of the system would be more obvious to the customers and there would be no false sense of stronger system security than there actually is. There would be no security through obscurity that is generally considered to be bad practice although very often used by software vendors. If big US banks consider username – password protection to be strong enough there might not be need for anything else. It is of course possible to clearly explain in user documentation that advanced authentication method used for fat client authentication does nothing for data protection. This is something that can not realistically be expected from banks.

Real solution would be to use cryptography to protect data in local storage. This solution is particularly convenient for systems that already use certificates for fat client authentication. Certificates stored on smart cards represent very safe two-way authentication method but can be used to provide other information security services like confidentiality, data integrity and non-repudiation. Availability could not be directly ensured in this way but with other services properly implemented overall availability could be indirectly improved.

Encryption of data stored in the local storage would provide confidentiality. Data would be legible to authorized users only so the privacy would be ensured. Digital signing of data stored in local storage would ensure data integrity and non-repudiation. Any change to data in local storage that bypasses fat client would deem digital signature invalid. Since digital signatures could be created only by using private key securely stored on smart card and protected with PIN, fat client user could not deny transactions and their content in case of dispute.

4. CONCLUSION

This paper attempts to raise the awareness that strong authentication does not have to translate to strong security. Bank clients using e-banking software should be fully informed on the overall security of the system they are using. Banks and their software makers should implement secure products especially when it is possible with available tools and when money is directly at stake.

It is generally considered that there are three thing needed for any illegal activity, including electronic illegal activity, means, motive and opportunity [12]. Means are in this case the mentioned tools for unauthorized data access that are there, nicely catalogued on the Internet and ready to go. Motives for illegal access to financial data are obvious since the money is one of the main motives for any activity. Opportunity is something that e-banking software makers could and should influence. By building secure applications from the cryptographic security blocks they have readily available number of opportunities could be kept to the minimum.

4. REFERENCES

[1] Chase web site http://www.chase.com/ (accessed 4.10.2004.)

[2] Wells Fargo web site http://www.wellsfargo.com/ (accessed 4.10.2004.)

[3] UBS http://www.ubs.com (accessed 4.10.2004.)

[4] Deutsche Bank http://www.db.com/ (accessed 4.10.2004.)

[5] NirSoft - Access PassView 1.12 http://www.nirsoft.net/utils/accesspv.html (accessed 5.10.2004.)

[6] Offline NT Password & Registry Editor. http://home.eunet.no/~pnordahl/ntpasswd/ (accessed 5.10.2004.)

[7] Red Hat Linux Manual – Problems after installation http://www.redhat.com/docs/manuals/linux/RHL-8.0-Manual/install-guide/s1-trouble-after.html (accessed 5.10.2004.)

[8] OpenBSD – FAQ – 8. General questions http://www.openbsd.com/faq/faq8.html#LostPW (accessed 5.10.2004.)

[9] SecurityFocus BugTraq mailing list http://www.securityfocus.com/archive/1 (accessed 6.10.2004.)

[10] Microsoft Security Bulletin MS04-028 http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx (accessed 6.10.2004.)

[11] A. Menezes, P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[12] L.Rogers, "Means, Motive, and Opportunity", Infosec Outlook, June 2000 Volume 1, Issue 3