Web Application Security of Money Transfer Systems

1 Alaa Hussein Al-hamami, 2 Fadi Ali Oqla Najadat, 3 Mohammed Saad Abdul Wahhab

1 Prof. of Database Security, Dean of College of Computer Sciences and Informatics, Amman Arab University, Zip Code: 11953, P.O.B: 2234, Amman, Jordan
2 IFRAD for Research and Development Co., Zip Code: 11953, P.O.B: 542299, Amman Jordan
3 Master in computer science, College of Computer Sciences and Informatics, Amman Arab University

1 Alaa_hamami@yahoo.com, 2 fadi_najadat@yahoo.com, 3 mohamed85_saad@yahoo.com

ABSTRACT

Information security science started to play a vital role in our life and became an important issue used for judging on any system about either its success or failure. E-banking applications for transferring money are considered as one of the most important applications that banks nowadays are taking care about; maintaining its process validity and accuracy as a necessity for the health of the transferring process to transfer the correct amount into the right receiver.

E-money transferring process can be attacked by hackers through using different malwares and viruses for changing the transfereee information and the transferring amount, one of them called “Silent Banker” and is considered one of the most important threats that appealed a global senior banks’ concern around the world because of its high capability in penetrating the most powerful security banking systems and the ability to use different tools to do so, which cost banks large and painful losses.

This paper proposes a solution to the “SilentBanker” problem through blocking possible security vulnerabilities that SilentBanker can penetrate the security system through. The necessary tests were held through this thesis to prove the validity of the proposed solutions.

Three main phases are presented which are: phase (0): Lock the browser, phase (1): Encryption and phase (2): Decryption; all are combined in order to introduce the best results in preventing the SilentBanker attacks. Results showed that the banking side presents major role in the detection process as checking whether the transfer process was successful so to successfully transfer the amount and without any error through, or to inquire both bank and client sides about the failures if not.

Keywords: SilentBanker; MitM, MitB, WAS; SSL, ST; SA; TA; System.Security.CodeAccessPermission; Encryption/Decryption.

1. INTRODUCTION

A well known e-Banking Threats of Silent banker Trojans to be represented as a "state-of-the-art in online banking fraud" (which is designed to steal banking credentials and account information of bank customer by taking one of as an Hyper-Text Markup Language (HTML) injection coding). The case is when sale of certain products through a Web application occurs, SilentBanker will instill particular malicious HTML inside the client browser for defying any security protections inside any browser and theft passwords and account number [1].

Web Application Security (WAS) is the automatic securing process for the HTML web pages when browsing by any server. Web applications security is a key business issue and biggest threat for organizations that concerns in intellectual property, critical client data, trade secrets, and such interest resulted from loss of confidential data in the first place. Loss of company’s reputation, then financial penalties obligations, and Lack of awareness of any Web application security breaches can cause huge damages to the company’s reputation and brand image and as a result on the whole business itself [2, 3].

Trojan SilentBanker is danger because of a special feature to avoid two-factor authentication, the capability to perform both Man-in-the-Middle (MitM) attacks and Man-in-the-Browser (MitB) attack, it works without users encrypting and then resending it to the attacker database, back-door potential and screenshots on computers [1, 3].

This paper is divided into several sections as follows: Section I gives an introduction about the web application security of money transfer systems, section II reviews some of the related works about this project, section III explores the methodology of work that contains the main three phases of designing system security engineering process model with their results, evaluating the design security software, security permissions access coding and making strong encryption/decryption, section IV explores results that include on the client side, in case of no penetration, in case of any penetration and the proof for the lock status, section VI illustrates conclusions of the work and section VII explores some of the future improvements.

2. RELATED WORKS

The available e-banking payment methods nowadays are highly insecure and cyber frauds through produced high financial sufferers and critical information losses and business reputation and brand image shaking due to latest innovative online banking threats like phishing, Pharming, and new Trojans malwares generations that are now using various infection vector tools that can overcome the available protection methods [4, 5, 6, 7].

Silent Banker threat is a very dangerous type caused in huge financial losses in more than 400 of large United
States (US) banks and others worldwide where many articles, web journals and anti-virus websites had recently wrote about. More specifications for the unique characteristics of SilentBanker as to be represented as a very dangerous threat which was determined specifically by anti-threats and viruses software's solution companies first in 2007 through the Symantec corporation report titled "Symantec Internet Security Threat Report Trends for July–December 06 [8, 9, 10].

Anti-Virus Companies are providing nowadays better understanding for Banking Trojans Malware problems through producing anti-viruses detection and removals solutions, such as MacAfee which presents various Infection Method for SilentBanker, [11, 12, 13].

Securing Online-Banking Services require a “trusted terminal” on the client-side for sensitive Input/output information transactions using the Transport Layer Security (TLS) cryptographic security protocol that can establish the encryption keys between the browser of clients’ computer and bank server, and can use a separate device for a client “lock-in” form for a specific service provider that seems to be a good solution for cryptography layers’ security but not enough”. Transactions mechanisms include: “(a) user identifications (IDs) and passwords, (b) Public key infrastructure and Digital certificates, (c) Digital Signatures [20], (d) Secure Socket layer (SSL), and various (e) Payment Protocols”. The public key infrastructure (PKI) technique is providing the necessary “confidentiality, non-repudiation, authentication, and data integrity” for electronic transactions encryption [14].

3. METHODOLOGY

This paper proved that the traditional security techniques are ineffective, and should be improved continuously while the SilentBanker Trojan Threat is presented as a difficult dilemma to be solved since the facts about its characteristics are existing. In this paper, the PHP Web development Programming language is used for defining the SilentBanker threat problem and promotes the necessary “security solution” for employing theories into practice.

Figure 1 shows the necessary phases for designing the “System Security Engineering Process Model” starting by defining the analyzes methodology of security problem features for the desired web application security; designing the process phases, formalizing and customizing it according to the web application security specifications, and finally implementing security testing and code review on a small scale as prototyping.

3.1 Phases of designing System Security Engineering Process Model

3.1.1 Phase 0: Developing the Suitable Security System Model for Money Transferring System Web Application

In phase (0) the suitable security system Model is developed for the web application of money transferring system through making the necessary analysis for the security problem for developing the appropriate software design as a necessary stage for drawing the necessary structural analysis for setting the system security engineering process that draws the attack tree and determines the necessary checklist and classify the security requirements according to the desired risk levels; the security system model can be drawn now and setting also the required Security Target (ST) for the problem and Threat analysis implementation in Structured Analysis (SA). Security level analysis is shown in figure 2 below.
Phase (0) Results determined the Security gaps, weaknesses, and vulnerabilities of Banking Web Application Security that SilentBanker Can Pass through: SQL injection hacking, XSS: Cross-site scripting, cookie Poisoning, Password Cracking and HTML injection Code as our SilentBanker case study is.

According to The System Security Engineering process; it used both benchmark methods of checklist and attack tree to make the necessary comparison within the required standards of security in a systematic way. Security Requirements can now be determined according to software requirements and the used standards specifications for defining the desired system security risk level, implemented through risk rating usage for “most sensitive information that is handled by the computer system and the minimum user clearance of logical and physical access to the system” as shown in figure 3. The ‘Attack Tree’ helps in determining the necessary requirements for designing the Security Model, and so makes the necessary improvements according to results [18].

The thesis in phase (1) tried to introduce the SilentBanker threat analysis through analyzing the SilentBanker Trojans threat; its Infection and Installation mechanism of the “User mode hooks attack” methodology thorough analyzing its security problem and the HTML Injection attack tool, its technical details as a thorough analysis for the SilentBanker problem.

Phase (1) Result shows Threat analysis (TA) with the help of Structured Analysis (SA) that begins in defining the main architecture for the system, and the scenarios for possible attack under a general level using one of most formal and well-known Security Models “Bell-LaPadula” that “(a) defines the concept of secure state and fundamental modes of access, and (b) gives rules for giving subjects access to objects with specific security policy under presenting three access modes of read only, read & write, write only”. Access rights are illustrated in Figure 4 as a “process between security layers, where subject cannot read object of higher sensitivity and where a star property means that subjects cannot write to object of lower sensitivity, the Strong star property enforces that objects cannot read or write to any other security layer under certain security functions for encrypting communications where trust relationships are analyzed using a recognized formal methods while cannot write or read for any further security layer” [17, 18, 19].

3.1.3 Phase 2: Developing the Suitable Security System Model for Money Transfer Web Applications

As the problem is now clear enough - it’s the right time now for phase (2) theoretical evaluation of the designated security module through designing the suitable security plan and implementing it for testing the security functionality and penetrating SilentBanker vulnerability [18].

After completing Phase (2), the required Security Software System for e-banking that prevents the SilentBanker threat from having the necessary access is completely designed.

3.2 Evaluating the Designated Security Software

Security Software System design should have a systematic evaluation in order to make sure whether this system has achieved the required security level or not. The illustrated V-model in Figure 4 implemented in each phase for making the required security development [18, 17]:
Figure 4: The Software Evaluation and Testing V-model [17].

The last step before real implementation is described in figure 5 and called as the Theoretical Evaluation or “security assessment” held for each security functions for a structured standards analysis that held in performing the threat analysis.

Figure 5: High-Level Model of Security Testing and Evaluation [17]

### 3.3 Security Permissions Access Coding

Security permissions access Code should be obtained from “System.Security.CodeAccessPermission” through introducing the necessary Demand method implementation that can be clearly defined using the “IPermission interface” and “IStackWalk interface” combined together. Putting into practice the “Code access permissions (not identity permissions)” is by introducing “IUnrestrictedPermission interface” for guarantee of an automatic unrestricting permission for the involved parties’ and hierarchy ranking them for through the “EncryptionPermission” as shown in Figure 6 [21].

Figure 6: System.Security.CodeAccessPermission [21]

### 3.4 Making Strong Encryption/Decryption

Encryption means to put protecting users’ privacy as a most important priority under an easy-to-use service tradeoff as shown in figures 7 and 8 through “(a) Protect data with military grade encryption, (b) Implement encryption transparently so users don’t have to deal with it, (c) Allow users to change their password without re-encrypting their data, and (d) allow IT access to data without the user’s password through business environments” [22].

Figure 7: File Encryption [22]

Figure 8: File Decryption [22]

### 4. RESULTS

Results were showed as the following:

#### 4.1 On the Client-Side

1. The client will first try to enter the e-banking screen shot –out from his/her computer.
2. While considering the “lock” icon included at the bank screen inside the down-bar (for preventing the SilentBanker entry), e-bank will ask for both the client’s username and password that can pass the security specifications inside the e-banking web application as shown in Figure 9.

4.2 In Case of No Penetration

4.2.1 Bank Authentication Scheme

3. Bank Authentication Scheme: The Bank - in its turn - will check about the client private data (of user name and password) if it is already available inside its client data list or not as shown in Figure 10.

4. After the check and ensuring that everything is correct, The Bank then will ask the client to enter the payee information in the right place.

4.2.2 Client-Entry

5. The client in his turn will enter the necessary full information, the transferee amount and currency type - as shown in Figure 11.

4.2.3 Bank Decryption and Verification processes

6. Bank Decryption and Verification processes are described in Figure 12

7. The Bank screen shows that the previous operation process is successfully done and nothing wrong is happening–as described in Figure 13.

4.3 In Case of Any Penetration

The following scenario presents on both sides:

1) Encryption and Decryption processes on Bank side:

On the Other case; the encryption and decryption processes that held by the Bank Web Application Security System showed a penetration for its web security, the
below Banking screen shot (Figure 14) showed text message ill the client that “there’s something wrong” and asked him to try to login again.

Figure 14: Bank Web Application Security System showed a penetration for its web security presented as a failure login

8. If something is wrong (if someone tries to access to the account via trying enter username or password wrong, and show it to the user (there is someone try to access to this user (n) times). As can be noticed; WAS of e-bank couldn’t encrypt and decrypt the hacked entry data; WAS of e-bank couldn’t encrypt the entry “hacked” data- so bank decided to stop the transferee process and then alarm the banking system about the hacking process. As shown below; figure 15 shows the fully decryption for adding a new client.

Figure 15: The Banking System adds a new client account for its clients’ Data Base

The successful adding for the new client is shown in Figure 19. The below Banking Screen shot also shows the possibility of Deleting clients as adding them to the bank database.

2) Bank Money-Tranfaring Process

The Bank Money-Tranfaring Process from the clients’ account and according to the determined amount is shown in Figure 16.

Figure 16: The Bank Money-Tranfaring Process

4.4 The proof for the Lock Status

Locking the Bank WAS Layers is shown inside the status bar of the web page

In case of penetration, The Lock Status will detect the penetration as sending a message of “try again later” as shown below in Figure 17. In case of the Locking is in-exists, the SilentBanker won’t be detected and that shows the SilentBanker successfully Stealth the client’ credential information of both username and password.

Figure 17: The Penetration Status was detected from the Bank WAS and a message was sent showing it

5. CONCLUSION

This thesis discussed one of most dangerous threats that e-money transfer process are facing nowadays, called the “SilentBanker” and showed related attach process while the transferring process occurred and recommended a proposed solution for preventing this threat attack and so transferring process will executes safely.

The Solution was presented through recommending a combined solution- as knowing that such threat can work through using the injection coding attach tool; locking the browser was the first defense line to use, the RSA encryption and decryption coding was the second one that uses for indentifying the payee identity- where Banking side presents major role in the detection process as checking whether the transfer process was successful so to successfully transfer the amount and without any error through, or to inquire both bank and client sides about the failures if not. This thesis tried to combine mixed stages for
introducing best results in preventing the SilentBanker attack.

6. FUTURE IMPROVEMENTS

Future work is recommended to implement this work on a large-scale for the commercial market usage; in order to do so, using Quality Function Deployment (QFD) in business analysis for determining the quality standards under a certain quality standards’ portfolio for all of structure, procedures and control - should be used for giving sufficient assurance for the customer in market in having Trusted Computing Base (TCB) documentation that can guarantee security for this application.

Work on developing the Voice over Internet Protocol (VoIP) technology usage as a one-authentication communication channel was another recommended solution for log-in procedures –without going back to the client’ computer assistant in logging- in where the SilentBanker install itself in its browser. This similar-to sell phone technique was a smart way to avoid the user browser usage.

Bearing in mind that such e-Criminals’ intelligence hackers will continue to innovate more capable attack tools -especially in money-making businesses and whatever security developer tried to invent more systems security problems will not stop nor disappear despite of all tries to! And as the new innovative trends attacks specification the react innovative solution will be recommend.

REFERENCES


