Review of Man-in-the-Browser Attack using Security Attack Scenarios

Anil Saini, Manoj Singh Gaur, Vijay Laxmi

Anil Saini
Malaviya National Institute of Technology, Jaipur, India
anil.pilani@gmail.com

Manoj Singh Gaur
Malaviya National Institute of Technology, Jaipur, India
gaurms@gmail.com

Vijay Laxmi
Malaviya National Institute of Technology, Jaipur
India
vlgaur@gmail.com

Abstract*—A Web browser is an important component of every computer system as it provides the interface to the Internet world. Browsers facilitate the web users through online services like e-mail, banking and shopping. The new unforeseen functionalities may be added to the web browsers in the form of extensions. The extensions have access to sensitive browser APIs and untrusted web page content, which may result in browser attack like Man-in-the-Browser attack. The major target of this attack is customers of Internet banking. This paper makes two major contributions. First, it presents the threat model for Man-in-the-Browser (MITB) attack. This model identifies various threats and point of attacks used by MITB attack. The major cause of MITB attack is malicious extensions and vulnerabilities found in benign-but-buggy browser extensions. In our study we find that the current browser security model is not secure enough to protect against MITB attack. Second, this paper presents the possible security attack scenarios for MITB threat model. The aim of adopting scenario based approach is to generate possible test cases for MITB attack and show how the system will react on these test cases.

***Keywords*—Threat model, Browser extensions, Vulnerabilities, Browser attack, Security attack scenarios**

I. Introduction

In the world of Internet, the web browser is the most commonly used application for the users connecting with Internet. The browser allows users to view and interact with content on the web pages. It provides users the interface to perform wide range of activities, such as, personal financial management, online shopping, social networking and professional business. Hence, the web browsers are becoming an increasingly adequate and important platform for millions of Internet users.

In order to add new functionality to the browser and enhance the user interface, a third party code called extension is added to the browser. These extensions possess capabilities, such as, cross domain network access and user's file system, which, if used improperly, pose a significant risk to the security. In [1], the author has examined security issues of functionality extension mechanisms supported by web browsers. In this the author has made the primary contribution by using code integrity checking techniques to control the extension installation and loading process.

The MITB Trojan attack is the fastest growing critical threat effecting consumers and business banking customers. The attack has the ability to intercept and manipulate any web page information and web transaction which a user submits online in real time. With this attack, innocent organizations are being targeted, resulting in large data and financial losses. Unfortunately, many security methods such as, antivirus protection, strong authentication mechanism and OS-patching are not effective against MITB attacks. The MITB attack is carried out successfully on secured channel protected with security mechanisms like SSL/PKI, two or three factor authentication. Thus, even a secured communication layer is not enough to provide effective solution against MITB attacks. This paper concerns about MITB attack, caused due to malicious and vulnerable browsers extensions. In the past, many vulnerabilities have been observed in Browser APIs and Firefox extensions [2][3].

Our major focus in this paper is to analyze the MITB attack using threat modeling and provide an effective security model for the same. The major attack vector used by the MITB attack to enter into the victim machine involves the browser extensions. Since the browser extension provides full privileges to browser internals and the user's file system, it can pose a significant security risk and reliability of the browser platform. Our extension review shows that many unsafe coding practices may results in security breach, which allows an attacker to explore vulnerabilities in such buggy codes and then exploit the system.

The remainder of the paper is structured as follows. In Section II, we present a brief discussion on the current browser extension security. We present our threat model for MITB attack in Section III. In Section IV, we present the attack scenarios to demonstrate the proposed threat model. In section V, we apply our test case scenarios to the current browser extension model. In section VI, we present a brief discussion on related work done in this area. Finally, the paper is concluded with future directions in Section VII.

II. Browser Extension Security

The objective of this paper is to analyze the security concerns in the browser extensions using threat model analysis for MITB attack. We identified various points of attack in the browser extensions used by attackers. The browser extension has given the privileged to access the critical browser components. The browser extensions make use of the Cross-Platform Component Object Model (XPCOM) to provide access to variety of components within the browser such as JavaScript, DOM and Browser APIs.

The browser extension vulnerabilities and its protection have been discussed by [4]. They propose a mechanism to protect users from benign but buggy extension by designing privilege levels for browser extension. They studied the over privileged extensions and provide least privilege system for extensions. In this paper, we review the point of attacks in browser extensions.

- *Malicious extensions.* Malicious browser extensions are being utilized just like other types of malwares. The malicious extensions in Firefox, Chrome and Safari have been created by attackers, who try to get them installed inside browser through various tricks, such as, web-based drive-by-downloads, social engineering tricks or infected attachments. The extensions which are downloaded from official web browser add-on directory are trustable and protected from malicious contents, because they undergo through review process which detects and removes malware from the directory. The example of malicious browser extension is banking Trojans which take advantage of vulnerabilities in browser extension resulting in an attack called Man-in-the-browser attack.

- *Benign-but-buggy extensions.* Even a trusted extension, downloaded from official trusted directory can have subtle vulnerabilities that expose the web browser to a serious attack from the web. Since the extension developers are not the security experts, the extensions might vulnerable to attacks originating from the malicious websites and the network. The author in [5] proposed a static information-flow analysis to find security vulnerabilities in buggy browser extensions. In this paper, we assume that the developer could write the wrong code that contains vulnerabilities. We will discuss the various vulnerable points of attack for DOM based XSS and privilege escalation attacks against non-malicious extensions in section 4.

- *Over privileged extensions.* The Mozilla Firefox extension system runs with browser's full privileges. Extensions can read and manipulate content from websites, can access browsers APIs, user's file system and network. Therefore, malicious and vulnerable extensions can cause serious security damage. There is a possible risk of a privilege escalation attack that grants web page script the full privilege of extensions. When an over privileged extensions interact with web pages, it has privilege to extract every piece of data from that web page, such as, user credentials, passwords etc. Since the information is extracted from the browser DOM within client's browser, this information is captured before it sends over encrypted channel.

III. Threat Model for the MITB Attack

In browser extension model, there are several vulnerable points of attack which an attacker can exploit to execute MITB attack. We now present a threat model for MITB attack which covers all possible attack paths (or threats). We present a threat model, shown in Figure 1, which identifies, various assets which are affected by an attack, possible threats and system vulnerabilities. In addition to that, it also provides countermeasures to mitigate potential threats in this attack.

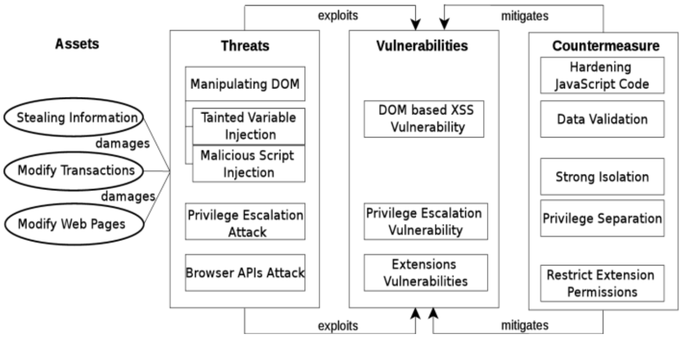

Figure 1. Threat Model for the Man-in-the-Browser Attack

The MITB Trojan infect user assets and it will install a malicious extension program into the browser. Whenever the web page is loaded, the Trojan will be activated and damages the user assets, like, it can steal user credentials, modify current transactions and web pages. The extension extracts all sensitive information, which user inputs in the web page forms and then silently modifies this information without the user noticing and suspicion. For instance, when an user performs the banking transaction on the browser infected with the malicious extension, an attacker can extract all the information from the web page fields. It then modifies every original information, such as, the transaction amount and the destination receiver account through the document object model(DOM) interface [6] and then resubmit the false information to the legitimate bank server.

This model describes various threats which are used by an attacker to exploit vulnerabilities present in an extension.

- *Manipulating DOM.* Browser DOM is a potential place of abuse, an attacker can employ techniques like runtime injection and tainted variable injection [5] to plot attacks on DOM tree. A malicious content can be injected into the web pages to manipulate and access the DOM tree.

- *Privilege escalation attacks.* The major benefit that attackers get from the browser model is from the over privileged extensions. The privilege escalation attack grants full privileges of extensions to malicious web page scripts.

- *Browser APIs attacks.* The browser extension governs access to the browser APIs. If an extension has over privileged permissions it can gain access to APIs. For example, if an attacker managed to get alter the user's bookmarks using APIs.

IV. Attack Scenarios

A scenario based approach describes the ways through which an attacker might make use of the identified threats and vulnerabilities. The primary purpose of generating security attack scenarios is to identify the known attack vectors and steps to perform the attack. In this paper, we choose scenario based approach because scenarios can be easily integrated within the development methodologies and it is the best way of representing attack descriptions. The process is divided into two major phases: first the building of scenarios and then the testing of the scenarios.

A. Building of Scenarios

This process involves two basic steps: first the identification of the possible threats that causes MITB attack and then involves the identification of possible countermeasures against the identified threats.

a) *Identification of possible threats:* During this step, the possible threats are identified as shown in earlier section in Figure 2. The intentions of an attacker are analyzed in terms of attack performed and possible attack path adopted to execute the MITB attack. For example, the possible threat for executing MITB attack could be vulnerable browser extensions, i.e. an attacker aiming to modify the web transactions by manipulating the content of the page. The attack flow between user and compromised browser is shown in Figure 2. It shows why the MITB attack can happen in the web browsers. The major cause of this attack is browser extensions which are trusted by the users. These vulnerable extensions could be possible point of attack for an attacker, as it grants full privileges of the browser components to an attacker. The attacker can exploit extension vulnerabilities to access browser APIs and DOM structure to plot various transactions and web pages.

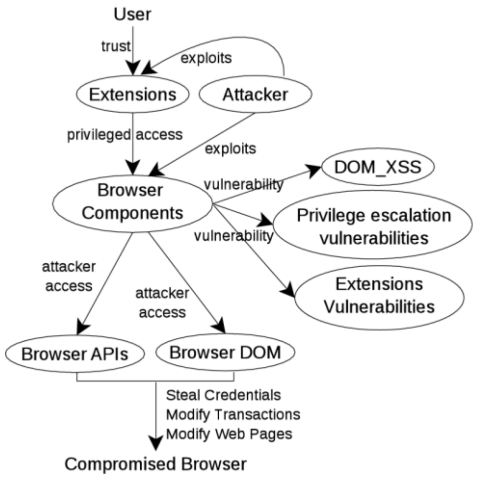

Figure 2. Why MITB Attack can Happen

The primary target of this attack is Internet banking customers, so here we have shown in Figure 3 that, how a transaction between a user and bank server is manipulated. An attacker can apply social engineering attack vector to plot malicious extension inside web browser. Now an attacker can exploit DOM and privilege escalation vulnerabilities to inject code which modifies DOM values, this way the original transaction made by user has been modified by an attacker.

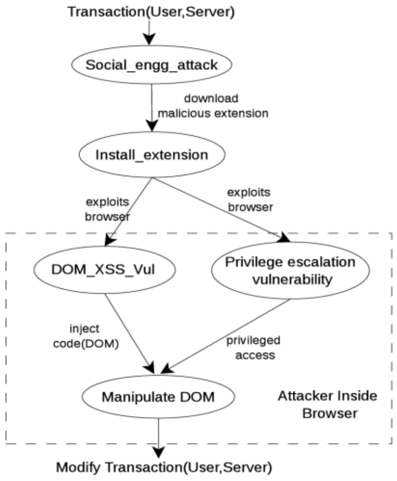

Figure 3. How MITB Attack can Happen

b) *Identification of possible countermeasures:* The second step involves the identification of the possible countermeasures to prevent the possible threat causes by the MITB attack. In this few security practices are suggested that can prevent MITB attack in the browser system. As an example, consider a vulnerable and over privilege extension which is capable of injecting the code and manipulating browser DOM. An attacker aims to exploit the extension vulnerabilities to read and manipulate the user data. However, if the users have been assigned with secure capabilities to restrict the extension permissions and isolate extension from the browser, somehow the attack can be prevented.

B. Testing of Attack Scenarios

Using the proposed threat model, the scenario based approach aims to identify the intentions of the attackers by generating the possible attack scenarios (or test cases) for the threat model and apply these scenarios to the system to see how it deals. So here we present different attack scenarios for the MITB attack threat model and show how the system will react on these attack scenarios. As derived from the analysis of threat model, the following scenarios regarding interception and modification are presented.

a) Interception attack scenario: In this attack scenario the attacker wishes to attack the privacy of the system by stealing the user information. As identified in the analysis of the threat model, privilege escalation attack grant excessive p [illegible] can extract the DOM information. Therefore, the attacker's main point of attack is to intercept the DOM information transmitted during the web transactions. The system reaction should be tested for present attack scenario using following test cases:

Test case 1: Information Stealing
Point of attack: An attacker can extract the information from the browser DOM structure.
Discussion: The current web transaction systems provide encrypted channel between the user and the server, this is the reason why an attacker would not be able to steal the information over the encrypted channel. But within the browser it can extract information from the web page fields.
Test case result: Currently the system is partially protected against the information stealing. Information can be stolen within the browser but not over an encrypted channel.
Countermeasures: Strong encryption mechanism should be adopted. Privilege separation and strong isolation mechanism should exist between the browser and the extensions.

Test case 2: Sniff Passwords
Point of attack: The attacker tries to obtain the access to the web transaction portal by using the authorization details of the victim user.
Discussion: The main objective of an attacker would be to capture the secret passwords transmitted during web transactions. Currently the extension model does not have any kind of protection for this attack.
Test case result: Currently the system fails to protect against password sniffing attacks.
Countermeasures: The hashing and encryption of the browser DOM content specially the password field could be the primary step to provide protection against the password sniffing.

Test case 3: Social Engineering
Point of attack: The attacker tries to trick people to perform some unwise action, but sometimes an attacker tries to exploit other browser or network vulnerabilities.
Discussion: The attacker tries to persuade the innocent users to provide their confidential information. The MITB Trojan displays fake pages to the user and ask the user to enter their authentication credentials in real-time to approve the transaction.
Test case result: Currently the system fails to protect against social engineering attacks.
Countermeasures: Blocking of malicious JavaScript code can be useful against social engineering tricks.

Test case 4: Session Hijacking
Point of attack: The attacker tries to intercept the information transmitted between the user and the web server.
Discussion: The attacker tries to exploit the web session between the user and the web server. It compromises the one time password code (session tokens) to gain unauthorized access to the web server. This can be performed though malicious code injection that creates a page with a fake error message and ask the user to input the One Time Password code.
Test case result: Currently the system fails to protect against session hijacking attacks.
Countermeasures: The system can be protected against code injection attacks with hardening of the JavaScript codes and the input validations.

b) *Modification attack scenario:* Modification attack scenario aims to attack on the integrity of the system. In this an unauthorized party not only gains access to the system but also tempers the user assets. As discussed in the threat model, the malicious content can be injected into web page to manipulate the DOM structure. The system reaction can be examined using following the test cases:

Test case 1: Modify web transaction
Point of attack: The attacker tries to modify the web transactions as they occur in real time.
Discussion: The attacker tries to access information from web transactions going on between the user and the server. It compromises the browser DOM structure with the help of the malicious extensions and tries to manipulate the browser DOM. The attacker can access and modify the current transactions in real time without the knowledge of the user.
Test case result: The current over privileged browser extension system fails to protect against this attack.
Countermeasures: Privilege separation and strong isolation mechanism must exists between the browser and the extensions.

Test case 2: Modify web pages
Point of attack: The attacker tries to modify the web pages using the code injection techniques.
Discussion: The attacker tries to manipulate information on web pages. An attacker is capable of injecting malicious code into a web page which modifies the original web page and then an attacker can apply social engineering attacks to steal user information.
Test case result: The current over privileged browser extension system fails to protect against this attack.
Countermeasures: The system can be protected against code injection attacks with hardening of the JavaScript codes and the input validations.

V. Applying Test Case Scenarios to the browser extension model

In order to test the security of the browser extension model for the MITB attack, two different kinds of scenarios were identified involving six different test cases. By applying these test cases many useful results were obtained about the security of the browser extension model. The brief summary of test case results are shown in table 1. It was identified that the system has failed to provide enough protection against the MITB attack, most of the attacks are successfully executed by an attacker on the client's browser. All the test case attacks are successfully executed in current browser extension model. The information stealing is partially protected, this is because the cc

an attacker can capture the information.

TABLE I. TEST CASE RESULT

Test Case	Technique	Result
Information stealing	DOM extraction	Partially protected
Password stealing	DOM extraction	Not protected
Social engineering	Exploit vulnerabilities	Not protected
Session hijacking	Code injection	Not protected
Modify web pages	Code injection	Not protected
Modify web transactions	Code injection	Not protected

VI. Related Work

Despite of serious consequences of the threat, the man-in-the-browser problem has received relatively little scientific attention. The attack was first introduced by P. Guhring [7], in which he presented the detailed description of the problem, identified the various points of attacks and the methods of attack and also suggests few possible countermeasures for the MITB attack. A comprehensive review on the browser extensions based MITB Trojan attack, poses a serious and growing threat to the online banking customers has been done by Utakrit in [8]. In this the author has analysed the MITB attack for online banking transactions and suggest few risk mitigation techniques. In [9], Dougan and Curran has presented the comprehensive study on the MITB attacks and several related Trojans, how the specific versions of the attack are executed. The author has examined the attack with reference to its control structure, data interaction techniques, and the methods for beating security.

Several solutions to protect against the Man-in-the-Browser problem have been suggested. In [10], an information security company SafeNet provides solutions for combating Man-in-the-Browser Attacks. It provides an effective out-of-band authentication solution, a separate device with keyboard and display the digitally signed user's transaction. Strong authentication solutions include both SMS out-of-band authentication and the secure browsing solutions necessary for preventing financial fraud that can result from Man-in-the-Browser attacks. The solution provided by Entrust [11] is based on active safeguards and passive safeguards. It provides two effective solutions with the fewest drawbacks, first one is out-of-band transaction detail confirmation and the other one is fraud detection that monitors user behavior, in this server-side monitoring of a user's movement is done through a banking website. In [12], a security company, Arcot provides the few countermeasures. It provides multi-factor authentication and digital signing solutions that protect against the MITB attacks while retaining ease of use, ease of management, and ease of deployment.

VII. Conclusion and Future Work

In this paper, a comprehensive study on current browser extension model is discussed and based on the existing security issues, it is found that the MITB attack pose a serious threat for many online services. The MITB attack is reviewed

ll

applicable threats and vulnerabilities exploited. We have seen the various point of attacks which may be adopted by an attacker and also identified all vulnerabilities present in browser extension. Using the MITB attack, we have seen that it is relatively easy to manipulate the user inputs, the web pages and the server responses with different techniques. We have presented the scenarios-based approach to test how the browser extension model reacts with the MITB attack. With the help of various test case attacks for our threat model, we have shown that the browser extension model does not provide enough protection against the MITB attack. In addition to that, we have also suggested few countermeasures to mitigate the MITB attack.

The most important future direction we envision is to focus on the countermeasures against MITB attacks and practical implementations of these methods. It is important to note, however, that the browser extensions are not the only way to realize man-in-the-browser attack. The browsers have other vulnerable points of attack, such as API hooking, Userscipts, and Virtualization. In future, we will focus on other vulnerable points of attack in browsers.

References

[1] Mike Ter Louw, Jin Soon Lim, V. N. Venkatakrishnan. "Enhancing web browser security against malware extensions," Journal in Computer Virology on, pages 179-195, vol. 4, August 2008.

[2] R. S. Liverani and N. Freeman. "Abusing Firefox Extensions," Defcon17.

[3] S. Willison. "Understanding the Greasemonkey vulnerability," http: //simonwillison.net/2005/Jul /20/vulnerability/.

[4] Adam Barth, Adrienne P. Felt, Prateek Saxena and Aaron Boodman. "Protecting Browsers from Extension Vulnerabilities," In Proceedings of the 17th Annual Network Distributed System Security Symposium 2010.

[5] Sruthi Bandhakavi, Samuel T. King, P. Madhusudan and Marianne Winslett. "VEX: Vetting Browser Extensions for Security Vulnerabilities," In Proceedings of the 19th USENIX conference on Security on, page 22, 2010.

[6] w3c document for DOM. http://www.w3cdom.org.

[7] P. Guhring. "Concepts against Man-in-the-Browser Attacks," http://www.cacert.at/svn/sourcerer/CAcert/SecureClient.pdf.

[8] Utakrit Nattakant. "Review of Browser Extensions, a Man-in-the-Browser Phishing Techniques Targeting Bank Customer," In Proceedings of the 7th Australian Information Security Management Conference on, page 19, 2009.

[9] Timothy Dougan and Kevin Curran. "Man in the Browser Attacks," International Journal of Ambient Computing and Intelligence (IJACI), Volume 4, Issue 1, 2012.

[10] Understanding Man-in-the-Browser Attacks and Addressing the Problem. http://ru.safenet-inc.com/uploadedFiles/About_SafeNet/ Resource_Library/Resource_Items/White Papers - SFDC Protected EDP/Man%20in%20the%20Browser%20Security%20Guide.pdf, SafeNet-2010.

[11] Defeating Man-in-the-Browser. How to Prevent the Latest Malware Attacks against Consumer Corporate Banking. http://download.entrust. com/resources/download.cfm/24002/, Entrust-2010, March.

[12] Protection Against Man-in-the-Middle Attacks. http://www.ca.com/ ~/media/Files/whitepapers/protection-from-mitm-mitb-attacks-wp.pdf, Arcot-2006.