

JANUSⁱ Verse Man-In-The-Browser

Chuan Pei Chenⁱⁱ - FrontOne Ltd

With reference to the article “Concepts against Man-in-the-Browser Attacks” by security expert Philipp Gühring (<http://www2.futureware.at/svn/sourcerer/CAcert/SecureClient.pdf>), Man-In-The-Browser (MITB) attack may progress thusly. The typical MITB attack utilizes steps as described in the following text:

1. The trojan infects the computer's software, either OS or Application.
2. The trojan installs an extension into the browser configuration, so that it will be loaded next time the browser starts.
3. At some later time, the user restarts the browser.
4. The browser loads the extension.
5. The extension registers a handler for every page-load.
6. Whenever a page is loaded, the URL of the page is searched by the extension against a list of **known sites** targeted for attack.
7. The user logs in securely to for example <https://secure.original.site/>.
8. When the handler **detects a page-load for a specific pattern** in its targeted list (for example https://secure.original.site/account/do_transaction) it registers a button event handler.
9. When the submit button is pressed, the extension **extracts all data from all form fields** through the **DOM** interface in the browser, and remembers the values.
10. The extension **modifies the values** through the DOM interface.
11. The extension tells the browser to continue to submit the form to the server.
12. The browser sends the form including the modified values to the server.
13. The server receives the modified values in the form as a normal request. The server cannot differentiate between the original values and the modified values, or detect the changes.
14. The server performs the transaction and generates a receipt.
15. The browser receives the receipt for the modified transaction.
16. The extension detects the <https://secure.original.site/account/receipt> URL, **scans the HTML for the receipt fields**, and **replaces the modified data** in the receipt **with the original data** that it remembered in the HTML.
17. The browser displays the modified receipt with the original details
18. The user thinks that the original transaction was received by the server intact and authorized correctly.

The above article clearly demonstrated MITB's threats that make secure socket layer security – the widely deployed and depended security useless. Does JANUS Vulnerable to MITM attacks? How JANUS will defense, defeat or remove the possibility of such an attempt as described above?

No, based on the techniques MITB use today, JANUS is not vulnerable to MITB attacks. JANUS has been designed to **prevent** pattern attacks such as MITB, because JANUS employs the patented Message Dissemination which can communicate discretely. (We recommend reader to read the above article in its entirety as to see the impact of MITB to existing security technologies and effectiveness of counter measures thus far)

With reference to point 6 and 8, MITB attacks are based on known pattern or patterns, say more advanced attacker may even able to update these online that make it even more powerful, hence the concern for the internet community and see it as the emergent threat. In contrast, JANUS does not provide data pattern, all useful item names are dynamically generated for the session only. There is no data pattern visible at the browser end which render MITB attacks (6, 8, 9, 10, 16) irrelevant.

With reference to point 9, 10 and 16, MITB has the capability to intercept data in the form and modifying the contents before they are sent to the service provider for processing and replace with the original data to avoid user detection during the confirmation stage, which is very clever, effective and very dangerous. JANUS, in another hand has number of measures which remove the hooks that MITB is required to launch a successful attack:

1. All user data are entered outside the form (9), after the client side data processes are completed use the client side script (JavaScript as an example) that is customized by JANUS for each web page, before a form is generated to post the data to the server.
2. Data are further protected by a hashing with One-Time-Password.
3. Even though MITB could capture the data in the form (9) after the post command it issued, it is useless because all data are has random name; it cannot distinguish the difference amongst all data hence unable to modify it (10);
4. It cannot replace the modified data (16) with the original to avoid user detection because the service provider has created a set data with different names in reply;
5. Assuming the MITB modified some data, the service provider will certainly reject it because it will fail data integrity check;
6. When a data integrity check is made by making a direct communication attempt to JANUS server, this communication cannot be stopped, and failure to complete this communication will invalidate the transaction as JANUS is expecting a message.
7. Dynamic naming of all elements stops the automatic script injection into web pages.

The above mentioned paper discussed many possible angles and approaches which only raised the barrier of attacks. From the above analyst, JANUS demonstrates its capability to **deny** all forms of Man-In-The-Middle attacks known today include MITB, knowing it will not processes self-learnt intelligence in a foreseeable future.

ⁱ JANUS, developed by FrontOne is a transaction-security solution, built with Message Dissemination & Human Factors Authentication technologies - two patents are registered in NZ, pending in US, EU and other countries.

ⁱⁱ The author of this article is the software architect of FrontOne Limited and the Inventor of both above patents.