Man-in-the-browser

Man-in-the-browser refers to a cyberattack method that has been around since 2006. As the name suggests, it involves a malicious actor secretly tapping into a user's browser to access their private information. Furthermore, it enables them to tinker around with their victim's website or engage in other devious exploits, such as financial fraud.

Examples of real-world Man-in-the-browser Trojans include Zeus, Carberp, SpyEye, and Clampi, to name but a few.

Man-in-the-browser attack methods

Malicious actors use various tactics to carry out Man-in-the-browser attacks. In the sections below, we explore each tactic in detail.

Man-in-the-browser through Trojans

In this approach, cybercriminals implant a Trojan horse in a would-be victim's PC operating system. In this case, they usually use email phishing as the primary vehicle to trick a user into installing the trojan which, in turn, allows them to spy on users.

A typical operation involves sending users emails with malicious attachments. Upon their clicking, these files attach themselves to a user's PC's system library, commonly used by top browsers to access the Internet. In doing so, they enable a malicious actor to read a user's information or change anything on their browser.

An excellent example of such malware is ZeuS, a sneaky Trojan that has led to massive losses since its inception in 2006. In particular, ZeuS has infected millions of computer systems worldwide, resulting in the loss of billions of dollars. Its other banking variants include the Citadel Trojan, which has affected 11 million companies, leading to a $500 million loss in damages. In comparison, its Emotet Trojan version has led to the loss of $1 million per attack since its introduction.

Another way the malware has been passed around is through drive-by downloads and spam campaigns. In the first case, a cybercriminal typically sends a malicious file in a downloadable form to an unsuspecting user—who then accidentally clicks on the file to spread the malware. Similarly, the latter involves spamming a user's mailbox with unsolicited emails promising differing "benefits"; for instance, discounts upon clicking on a link leading to an infection.

The third way the ZeuS Trojan is spread across the web is through JavaScript and AJAX. Most Zeus developers prefer this method since XML, asynchronous JavaScript, or AJAX code work alongside X-Frame option headers. And in so doing, cybercriminals can oversee and command servers to create new forms within banking sites, which allows them to access personal

information and user passwords. Besides that, these languages make it difficult for an ordinary user to identify the malware. As such, they are a popular choice when configuring botnets.

Other means through which the Zbot is spread include pay-per-install services, social media messaging services, and instant messengers. The awful thing about the malware is that it mutates an infected system into a bot in a botnet. This makes life easier for bad actors, as they can rent out hacked systems to their associates to continue playing the game.

Man-in-the-browser through browser extensions

In this approach, malicious actors use an insecure browser extension or a Browser Helper Object—a malicious user script. Here, an attacker usually maneuvers malware past a browser's security features to intercept communication between a website and a user. This allows them to alter a user's financial transactions or change their website's appearance, among other nefarious deeds.

Man-in-the-browser through API hooking

The third way for an attack is by using API Hooking. In this setting, the man-in-the-browser acts as the middleman between executable applications and their libraries. Notably, the MITB strategically hooks up the internet connect function in winnet.dll. This enables a cybercriminal to change what pops up on a user's browser—the operation mimics HTML rewriting. Even worse, the fraudster can decide to change a user's website to make it appear authentic, but with misleading information.

Man-in-the-browser through SSL stripping

Last but not least, we find Man-in-the-browser attacks that make use of SSL stripping. Here, cybercriminals downgrade a website’s connection from HTTPS to HTTP, making it more vulnerable to attacks. The action makes all the communications unencrypted, as a result, setting the stage for man-in-the-browser attacks.

How to detect and prevent Man-in-the-browser attacks

From an expert standpoint, detecting and preventing man-in-the-browser attacks is an uphill task. How they spread and operate tells it all. As an example, it is difficult to stop them by using standard antivirus software or firewalls meant for virus protection. Furthermore, cybercriminals can use extensive resources to launch an attack, giving them an upper hand. They can also develop Man-in-the-browser attacks that spread exponentially making them difficult to control.

Preventing man-in-the-browser attacks: strategies for companies

First off, they can use Out-of-Band Verification. Here a different device, such as a phone can be used to verify what's passed on to a PC. This isn't foolproof, though, as the malware can wait to

strike at prime time—after authentication, which makes it difficult to stop the infection in the first place. Therefore, companies can integrate a biometric identification system to make the method more effective.

Another way to monitor and intercept such attacks is through behavior analysis. This process involves an in-depth study of user behavior to help identify abnormal account activity. These can allow bank employees to validate transactions, for instance. It also allows weeding out potential criminal activity or actions that don't match a user's profile.

Finally, is the use of client-side security measures. Such measures include ensuring the integrity of every website component and preventing sensitive data from being exfiltrated by attackers. A robust client-side security solution provides protection against man-in-the-browser attacks regardless of the approach used by threat actors.

Preventing man-in-the-browser attacks: strategies for end-users

To protect against SSL stripping attacks, users can always check their browser's address bar. Doing so enables them to spot any connection via an unencrypted HTTP protocol. Alternatively, they can install the HTTPS Everywhere extension, ensuring HTTPS communication round the clock. On top of that, the extension can help to prevent third parties from demoting their connections to HTTP.

Users can also ensure that their local area network is free from unauthorized parties and is secure. In essence, an SSL breakthrough heavily depends on local network accessibility. Big organizations can as well install robust firewalls to get the job done.