Journal of Telecommunications System & Management

Bassiouni, J Telecommun Syst Manage 2012, 1:3
DOI: 10.4172/2167-0919.1000e105

Editorial Open Access

Protecting Cookie-Based Internet User Sessions

Mostafa Bassiouni*

Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, USA

The use of cookies to store useful information about an Internet session is reaching a universal level. Cookies are used in almost all Internet user sessions including social media (e.g., Facebook, LinkedIn, Google+, Twitter, MySpace), media streaming (e.g., YouTube, Metacafe), retail industry (e.g., Amazon, Dell, Walmart) and various other applications (e.g., eBay, AAA). Cookies in these applications are used for many purposes including session and transaction authentication, selecting display mode, tracking of shopping cart contents, identification of user's preferences, and tracking browsing behavior.

Cookie hijacking attacks, also called Side jacking attacks, can be possible when sessions are not secure. The best way to overcome session hijacking is to use the secure protocol HTTPS (i.e., HTTP over SSL or TLS) throughout the lifetime of the session. The HTTPS protocol, which is currently used by web-based banks and reliable e-commerce applications, guarantees a fully secure session by encrypting users' login information as wells as the data of all transactions between login and logout. Due to the costly nature of this approach, many developers use HTTPS only in the login and initial authentication then revert to the approach of using cookies stored locally at the client's side to carry out authentication throughout the remainder of the session. Session cookies used for authentication store information about the identity of the client as well as a shared secret which indicates that the user has been authenticated in the initial login.

Since session cookies are transmitted over unsecure links (especially wireless links) using the unsecure HTTP protocol, they are susceptible to malicious attacks. An attacker can sniff out session cookies and consequently, take over a client's identity. This allows the attacker to gain the same level of access over the web application that the original user has. By hijacking session cookies, it becomes possible for the attacker to impersonate the victim and interact with the application server without proper authorization.

There has been a recent surge in the number of cookie sniffing tools and the number of research reports and security-related blogs that alert the Internet community to the serious security risks of cookie hijacking. Below, I give two examples of these tools.

Fire sheep [1] is a free open source Firefox extension released in October 2010. With Fire sheep installed, a user could potentially capture cookies from other users of an open Wi-Fi network, and use them to hijack their sessions at supposedly secure sites such as Face book and Twitter.

Face Niff [2] is an Android application released in May 2011. Face Niff makes it easy to hijack other people's sessions using an Android smart phone with root access. The Android application is claimed to work over any private Wi-Fi network using any of the common security protocols, including WEP, WPA-PSK, WPA2-PSK, or no security at all. FaceNiff can be used to intercept web session profiles and easily hijack sessions for Face book, Twitter, YouTube, Amazon, MySpace, Nasza-Klasa, blogger, etc.

Almost all the cookie sniffing tools have been developed with good intention. The developers of these tools have stated that the purpose of their tools is to demonstrate the vulnerability of public Wi-Fi and Web 2.0 applications to cookie-sniffing, raise awareness about the dangers of cookie hijacking, and most importantly put pressure on service providers in order to adopt more rigorous security policies and offer robust authentication protocols to protect the people who depend on their services.

Many security experts have not welcomed the release of free cookie sniffing tools. Some experts voiced their opinion by saying that the right to free speech doesn't extend to inciting criminal action, or providing tools to help break the law; just because a developer can write code to prove a point doesn't mean he/she have to release it. Although potentially harmful, these tools have produced some positive results and have generated increased interest in research for protecting session cookies. Below, I elaborate on two positive aspects of the cookie sniffing tools.

The first positive result is that some well established web sites started to offer more security services to protect user sessions against cookie hijacking. For example, Face book responded by offering optional HTTPS services to its users and recommended the users to consider enabling this option if they frequently use Face book from public Internet access points found at coffee shops, airports, libraries or schools.

The second positive result is the increased interest in providing security services to web sites that do not use the costly HTTPS protocol. For example, a plug-in for Firefox [3], has been created to notify users of vulnerable sites. Basically, the software helps the user avoids sites that do not provide adequate security and pose a threat to user privacy. Another example is the Rolling Code protocol [4], intended to prevent or reduce the harm of cookie high jacking. The protocol utilizes the initial secure HTTPS authentication to exchange a shared secret between the server and the user. For every transmission made from the client to the server, the client sends a cookie code to the server. The server validates the code using the shared secret stored at the server. In essence, the protocol is like the rolling code technology used to prevent perpetrators from recording a code and replaying it to open a garage door.

A more promising approach is the use of one-way hash chains in protecting session cookies without encryption. The hash chain can be used to generate one-time authentication tokens derived by repeatedly applying the hash function on an initial secret selected by the user. Due to the pre-image resistance property of the hash function (e.g., SHA-1), the hash values are distinct and can be used to represent one-time authentication tokens stored in the cookies transmitted from the client to the server. Once the authentication token is used and stored in a cookie, it cannot be used again in future transactions. This effectively protects against replay attacks.

***Corresponding author:** Mostafa Bassiouni, Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida 32816-2362, USA, E-mail: bassi@cs.ucf.edu

Received September 14, 2012; **Accepted** September 17, 2012; **Published** September 19, 2012

Citation: Bassiouni M (2012) Protecting Cookie-Based Internet User Sessions. J Telecommun Syst Manage 1:e105. doi:10.4172/2167-0919.1000e105

One difficulty with the one-way hash chain approach is that the number of transactions expected during the lifetime of the session must be estimated at the time of login. If the number of transactions is overestimated, the authentication in the early steps will suffer from an unjustified large computational overhead. If the number of transactions is underestimated, there will be the undesirable synchronization overhead of using HTTPS secure communication to establish a new secret and a new number for the remaining transactions. A clever way to reduce the severity of this problem is the protocol proposed in [5]. In this protocol, hash chains are arranged in two dimensions. In the first dimension, there is a single hash chain to compute the initial seeds for the multiple hash chains in the second dimension. The two-dimensional hash chain scheme [5] achieves significant improvement over straightforwardly configured one-way hash chain schemes.

It should be mentioned that all of the above solutions provide limited protection against session hijacking attacks. For example, one-way hash chain protocols cannot protect against worm-based attacks such as the nefarious Koobface worm which has repeatedly targeted users of Face book, MySpace, and Twitter. Koobface, whose name is Face book scrambled, is used by cyber creeps to hijack social media accounts without hijacking session cookies. The Koobface attack arrives in the form of a message from a friend (with Torjan code) asking to download a special video player to view a video. The download triggers an automated program that sends copies of the same viral message to all of the victim's friends, while turning full control of the victim's PC over to the attacker.

In conclusion, attackers will continue to find ways to defeat security protocols against session hijacking and the research will continue to improve these security protocols and enable them to protect against new forms of attacks.

References

1. Butler E (2010) FireSheep: Cookie Snatching Made Simple. ToorCon Conference, San Diego, CA 22-24.
2. Ponurkiewicz B FaceNiff- A new Android download application.
3. Riley RD, Ali NM, Al-Senaidi KS, Al-Kuwari AL (2010) Empowering Users Against Sidejacking Attacks. SIGCOMM'10.
4. Cashion J, Bassiouni M (2011) Robust and Low-Cost Solution for Preventing Sidejacking Attacks in Wireless Networks using a Rolling Code. Proceedings of the 7th ACM International Symposium on QoS and Security for Wireless and Mobile Networks 21-26.
5. Alabrah A, Bassiouni M (2012) A Hierarchical Two-Tier One-way Hash Chain Protocol for Secure Internet Transactions to appear in Proceedings of IEEE GLOBECOM 2012-Communication and Information System Security Symposium, Anaheim, California 3-7.