Session Hijacking in Wireless Networks

An introduction to web application vulnerability

Manmohan PV
http://www.newscamp.org

Introduction

The term session hijacking refers to the exploitation of a valid computer session - sometimes also called a session key or Id - to attain unauthorized access to information or services in a computer system. In particular, it is used to refer to the theft of the magic cookie used to authenticate a user to a remote server. It has particular relevance to web developers, as the HTTP cookies used to maintain a session on many web sites can be easily stolen by an attacker using an intermediary computer or with access to the saved cookies on the victim's computer

Many web sites allow users to create and manage their own accounts, logging in using a username and password (which may or may not be encrypted during transit) or other authentication method. In order that the user does not have to re-enter their username and password on every page to maintain their session, many web sites use session cookies: a token of information issued by the server and returned by the user's web browser to confirm its identity. If an attacker is able to steal this cookie, they can make requests themselves as if they were the genuine user, gaining access to privileged information or changing data. If this cookie is a persistent cookie, then the impersonation can continue for a considerable period of time. Of course, session hijacking is not limited to the web; any protocol in which state is maintained using a key passed between two parties is vulnerable, especially if it's not encrypted. Thanks Wiki!!

Terminology

One of the greatest advantages of using Mozilla Firefox browser is that you can get a lot of valuable plug-ins to explore the application. Tamper Data, Cookie Culler, AEC Cookie Editor are a few of them. A brief introduction to all of the tools above, Tamper Data is used for tracking the communication between the browser and the server and the usage makes it possible to intercept and modify the data on fly and re-submit to the server. Where as Cookie Culler can be used to view and delete the cookies. AEC Cookie Editor plug-in gives even more control over the cookie where you can replace its contents and test your web application for vulnerability.

Hypertext Transfer Protocol (HTTP) is a communications protocol used to transfer or convey information on intranets and the World Wide Web. Its original purpose was to provide a way to publish and retrieve hypertext pages using an unique identifier. HTTP is stateless protocol. The advantage of a stateless protocol is that hosts do not need to retain information about users between requests, but this forces web developer to use alternative methods for maintaining users' states. For example, when a host would like to customize content for a user while visiting a website, the web application must be written to track the user's progress from page to page. A common method for solving this problem involves sending and requesting cookies.

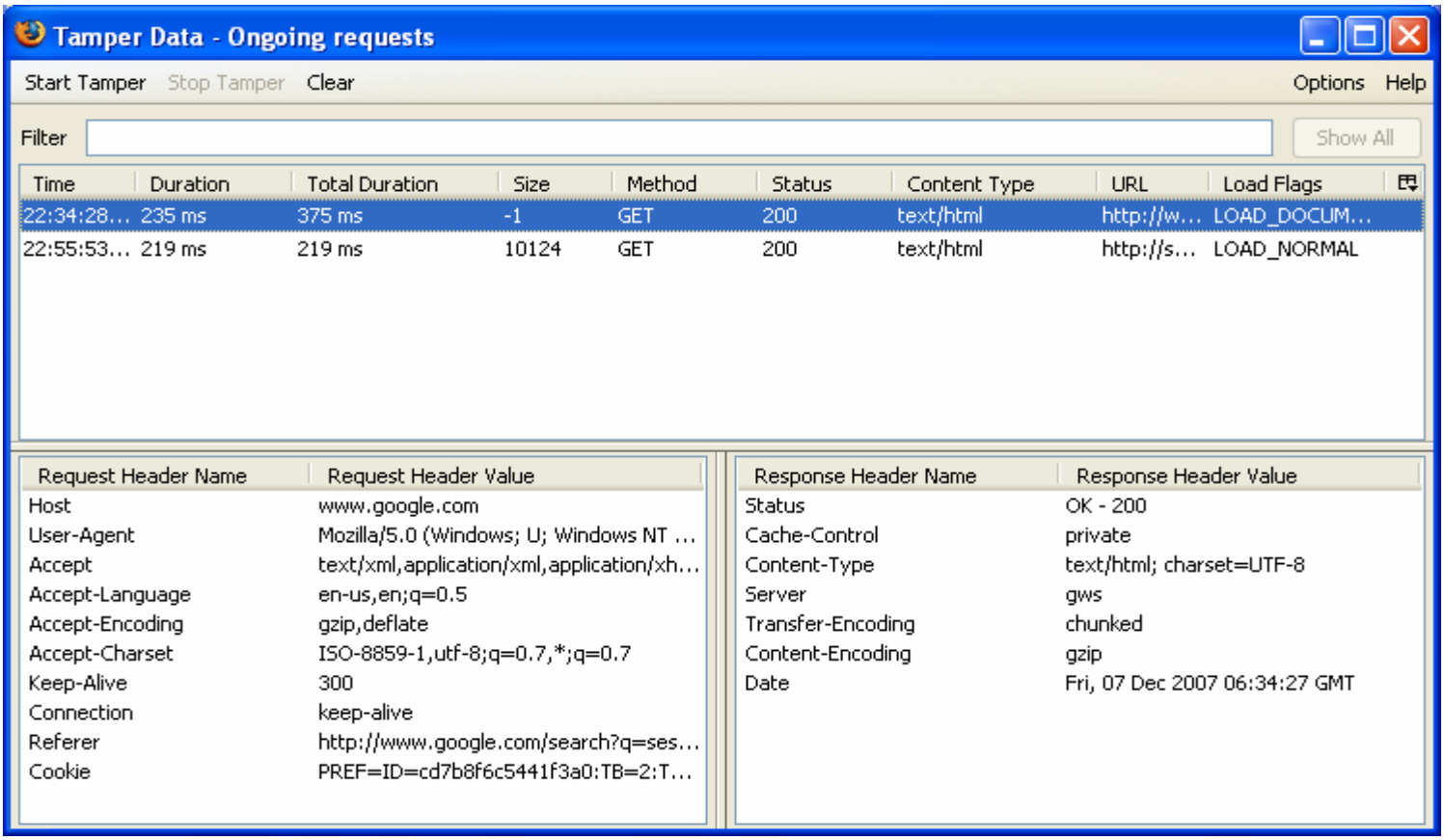
Allowing users to log in to a website is another use of cookies. HTTP cookies are used by Web servers to differentiate users and to maintain data related to the user during navigation, possibly across multiple visits. Users typically log in by inserting their credentials into a login page; cookies allow the server to know that the user is already authenticated, and therefore is allowed to access services or perform operations that are restricted to logged-in users.

The cookie setter can specify a deletion date, in which case the cookie will be removed on that date. If the cookie setter does not specify a date, the cookie is removed once the user quits his or her browser. As a result, specifying a date is a way for making a cookie survives across sessions. For this reason, cookies with an expiration date are called *persistent*. As an example application, a shopping site can use persistent cookies to store the items users have placed in their basket. This way, if users quit their browser without making a purchase and return later, they still find the same items in the basket so they do not have to look for these items again. If these cookies were not given an expiration date, they would expire when the browser is closed, and the information about the basket content would be lost.

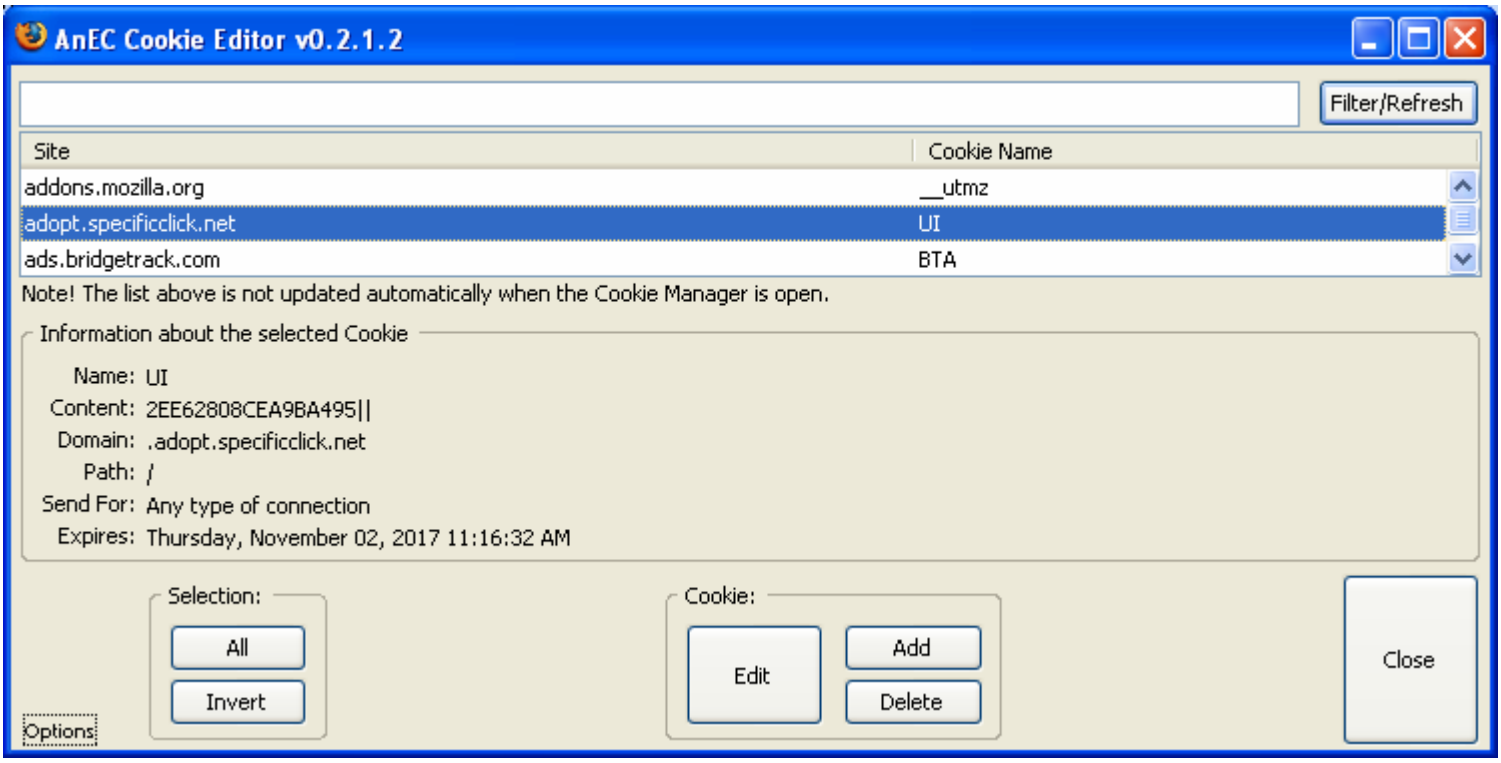
HTTPS is not a separate protocol, but refers to the combination of a normal HTTP interaction over an encrypted Secure Sockets Layer (SSL) or Transport Layer Security (TLS) connection. This ensures reasonable protection from eavesdroppers and man-in-the-middle attacks while transferring data through wired or wireless networks. This never prevents a sniffer from getting the data, but interpreting the data will not be easier as it would be in encrypted form. But in case of HTTP communication, sniffers can eavesdrop the communication channel and get most out of the browser-server interaction and this would be lot easier in case of wireless networks which are not password protected.

An Example

Let us investigate it through some of the tools mentioned above. Here, TamperData shows the ongoing communication where i did a search for "session hijacking" in google. You would able to see both the request and response header information and GET, POST parameters and thier values.



AnEC Cookie Editor shows the information about cookies in the system which explains how a web server tracks the clients using cookies.



All cookies and other information are sent through the same communication channel using HTTP or HTTPS protocols.

Exploitation Scenario

Most of the web applications use both HTTP and HTTPS protocol for their communication depending up on the sensitivity of the information sent through the communication channel. HTTPS is costly and time consuming protocol used for sending username/password, financial and other important information. Once the information is sent it switches back to HTTP mode.

Normally if you look at a web application, it drops cookies as soon as you visit the page for the first time and the cookies may contain the session id. This ID is an important piece of information which would be used for tracking the session once the user logs in. Most of the people used to neglect the importance of this ID information in the unsecured communication channel. Even after authenticating through a secure channel, they use the same session ID for tracking. This is where the vulnerability exists. Sniffing of a network is a easy task, Plenty of hardware and software are available in the market to sniff the information actively or passively. Especially in wireless networks sniffing is like eating a piece of cake.

Most of the web application grants access to sniffer if he has the session Id of an already logged-in user and he would able to access the important information of the victim. This vulnerability exists with most of the commercial web site which provide various services. Email Providers, Shopping Sites, etc are all vulnerable to this type of attack if they have implemented authentication mechanism this way.

Protection

It is possible to protect the session by dropping another cookie after the user authenticate and maintain that cookie value in the HTTPS session or changing the value of the session id cookie is another option. Even if the sniffer gets the cookie information from the unsecured HTTP session he would not able to login to the system since he does not have the secure token dropped in HTTPS session. This will ensure that the user session is protected from session hijacking in wireless networks.

References

- *Applied Cryptography By Bruce Schneier*
- *Email Security By Bruce Schneier*
- *Web Hacking - Attacks and Defense from Addison Wesley*
- *Hacking-The Art Of Exploitation*
- *Network Programming with Perl from Addison Wesley*