A Comparative Study of Security Level of Hotmail, Gmail and Yahoo Mail by Using Session Hijacking Hacking Test

Thawatchai Chomsiri

Faculty of Informatics, Mahasarakham University, Thailand

Summary
This research presents the results of the experimental about security level of three famous Web Mails—Hotmail, Gmail and Yahoo Mail. These three Web Mails were hacked by means of Session Hijacking. The researcher conducted this experiment on the LAN system and used information capturing technique to gain Cookies and Session ID inside Cookies. Then, Hijacking was conducted by using two Hijacking methods. The first method, which was common and easy to conduct, used only one Cookie. The second method, which was not very popular but offered high penetrating power, used all Cookies (Cookies cloned by SideJacking tools). The results show that the Web Mail with the highest security level is Yahoo Mail; the second one is Hotmail; and the Web Mail with the lowest security level is Gmail.

Key words:
Session Hijacking, Hotmail, Gmail, Yahoo Mail, Hack, Cookie.

1. Introduction

Nowadays, the use of free Web Mails is very popular since users can check their mails from anywhere. All they need are only internet connection and web browser. Meanwhile, free Web Mails also provide large capacity mailboxes to meet the need of their users. In selecting which Web Mail should be used, apart from the mailbox capacity and access speed, we need to consider about the security. However, the mailbox capacity and the access speed of the top 10 free Web Mails are close on each other, so the security level becomes an important issue we need to consider before choosing a free Web Mail to use. On the LAN system, Session Hijacking is a very popular and easily hacking method to access mailboxes of other people. It is easily done by sniffing (Capturing) Cookies, and then Session ID is used to access mailboxes. Hackers can do so by creating ARP Spoof first, and then they capture victims' Cookies. After that, browser which allow user to change the value in Cookies (i.e. using Firefox-Cookies Editor or using Opera) is used to send the Cookie/Session ID of the victims to substitute the Cookie/Session ID of the Hacker. Finally, the hacker will be able to access the e-mail system by the right of those victims.

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

This research will test and compare the security level between the three most popular free Web Mails [1] — Hotmail, Gmail and Yahoo Mail. The test will focus on

hacking by Session Hijacking method divided into two types comprising (1) sniffing Cookies and using only one Cookie which is easy and popular (If there are many Cookies, they will be tested one by one.) and (2) sniffing all Cookies and using them (Cookies Cloning) by employing a tool named SideJacking [2]. The second type has more complex steps and is not yet popular; still, it has high penetrating power.

2. Background

Websites involving membership system such as general Websites, Web Boards as well as Web Mails need to employ mechanism which enables Web Servers to know which member they are communicating with. For example, Bob and Alice are checking their e-mails at the same time. The Web Server must answer HTTP Response in order to send Bob's mailbox to Bob, not to Alice. Such mechanism can be done by assigning each user to have Session ID generated by Web Servers after users successfully logged in the system. There are several methods to send Session ID between Browsers and Web Servers, e.g. sending it together with URL, sending by using Hidden Field or sending by containing Session ID inside Cookies etc.

Sending Session ID together with URL by assigning Session ID to be Parameter, such as

http://mail.mydomain.com/mbox.php?sid=FxQ4zy3rN

is not a safe method since anybody might be able to take a peek at the monitor and bring the Session ID to use, or sniff Session ID, and then easily enter it on the Browser. Sending Session ID together with Hidden Field provides a safer level of security as nobody is able to take a peek to get the Session ID on the monitor. However, Session ID can be sniff by using some programs such as Web Scarab or Acunetix – HTTP Sniffer. Sending Session ID by containing it inside Cookies provides high security like sending it by using Hidden Field; this is the most popular method which is used by Web Mails.

This research aims at studying the security level of Web Mails and the endurance level which Web Mails play against hacking by Session Hijacking. The security and endurance levels are tested by two hacking methods as follows.

1. Hacking by changing only one Cookie—snatching a victim's Cookie and bringing it to use (If there are many Cookies, they will be used one by one.)
2. Hacking by changing all Cookies (Cookies Cloning)—using SideJacking [2] which often clone all Cookies and send them with HTTP request

The following parts elaborate details and steps of both two methods.

2.1 Session Hijacking by Imitating only One Cookie

A hacker will sniff a victim's Packets by using Sniffer or Ethereal programs; Packet wanted by hackers is one that has HTTP Request. The hacker will look for Cookies in order to take Session ID which is inside the victim's Cookies. After the hacker gets the victim's Session ID, he will log in the system by the account that he has created for hacking. After Authentication has been operated through the hacker's account, there will be Session ID which is a value. Then, he will substitute that Session ID with the victim's Session ID by editing the value in Cookies. There are several Browsers supporting value editing in Cookies such as Opera and Fire Fox which install Add Ons named Cookie Editor. In the case that some Websites use many Cookies, there must be at least one Cookie used for sending Session ID. The hacker will look for the Cookie Name which its Cookie Value is most likely to be Session ID, and bring it to test. If it is unsuccessful, the next probable one will be used for the test. The procedure is repeated until every single Cookie is tested. This method of hacking will be unsuccessful if the Websites send Session IDs by using two Cookies or more (such as dividing a Session ID into two parts and keeping them on two Cookies). Similarly, if the Websites use the technique of constantly changing the Session IDs, the hacking will also fail.

2.2 Session Hijacking by Copying All Cookies (SideJacking)

SideJacking was invented by Robert Graham [2] and presented in Black Hat 2007 Conference [3]. Function of SideJacking is to sniff victims' Request information. Hackers will gain important things existing in HTTP Request such as Cookies, URL and Parameter. SideJacking tools are capable of repeating such HTTP Request in order to get victims' HTTP Response. Moreover, SideJacking tools enable hackers to continue further links. For instance, when hackers can access victims' Mail Inboxes, they will be able to click and see e-mails of their victims. Meanwhile, they will also be able to create new e-mails by using the names of their victims and send them to anybody.

There are two programs in a SideJacking set consisting of Ferret and Hamster. Ferret will function as sniffing (capturing) the data. When a victim is sending HTTP Request, Ferret will record the captured data and keep them in files with extension ".pcap". At the same time, Ferret will detect HTTP Request and bring it to create a hamster.txt file in order for Hamster program to bring it for use. Hamster program will read the hamster.txt file, create links which the victim used to access and show them to hackers so that they can follow the same process. Furthermore, Hamster also sends Requests to Web Servers instead of Browsers by using the victim's Cookies. Hamster will communicate with Browsers by acting as Proxy on IP address 127.0.0.1, Port number 3128. To use it, hacker's browser will connect the proxy (Hamster) and browse to http://hamster/ in order to hack victims' Mailboxes.

Prior to experimenting by using both techniques 2.1 and 2.2, the process of ARP Spoof must be conducted on the Switch Network first in order for hackers to act as the Man in The Middle (MITM). However, for Wireless Network, hackers are able to sniff data without conducting ARP Spoof, but according to the field test, it is found that they cannot sniff every packet on Wireless Network. Thus, if they want to sniff 100% of victims' HTTP Request, the ARP Spoof must be conducted.

3. Methodology

The researcher conducted the experiment on Local Area Network (LAN)—both Switch Network LAN and Wireless Network LAN. Since this experiment was conducted to measure the security level of each Website against the Session Hijacking attack, so the experiment was designed for a hacker to be able to sniff all cookies in order to bring all of them to test. To allow this to happen, the ARP Spoof was included in every test although some tests were conducted on Wireless LAN. In addition, the victim's computer was controlled so as not installed anti ARP Spoof program such as Anti ARP, and Static ARP was not conducted on that computer. Meanwhile, on the Gateway Router computer, Static ARP was not conducted, and Static Port (Port Security [4],[5]) was not conducted as well.

Experimenting any Web Mail by SideJacking method, if it was found that the hacker was able to hack the system despite only one time of the experiment, the result was recorded as 'Success'. Then, another Web Mail was tested.

However, if the hacker could not hack the system, the test would be repeated in order to entirely gain Links on http://hamster/. After that, the experiment was preceded by clicking on Links until every Link was clicked. If it was found that the hacker was able to hack the system although not every Link was tested, it is regarded that the hacker could hack the system. The result was recorded as 'Success'.

In the case that every Link was clicked; the test was conducted repeatedly to ensure that every Link was tested at least 10 times within five minutes or less, timing from the first second as soon as the victim refreshed the Mailbox page (in order to control the variable concerning Cookies/Session Timeout). When the test was complete, and the hacker could not hack the system even once, the result would be recorded as 'Fail'.

For the test that only one Cookie was copied, it would be checked in order to be certain about how many Cookies are in the each Web Mail. Then, every Cookie was tested. If it was found that the hacker could hack the system although not entire Cookies were tested, the result would be recorded as 'Success'. In contrast, if all entire Cookies were tested, but it could not be hacked, each Cookie would be repeatedly tested for 10 times. If the entire process was complete, but it could not be hacked, the result would be recorded as 'Fail'.

4. Results

After the experiment, the results are as follows.

4.1 Gmail

Firstly, this Web Mail was tested by SideJacking, and it was found that the victim's Mailbox could be hacked by clicking on URL (on http://hamster/) as shown below.

http://mail.google.com/mail

For the hacking by changing Cookies one by one, it was found that there were eight involving Cookies in the Domain named as follows.

mail.google.com
google.com
www.google.com

When Cookies were tested one by one, it was found that the victim's Mailbox could be hacked by changing the Cookie named "GX" in the Domain named mail.google.com.

Hence, it could be concluded that Gmail has no resistance to hacking by Session Hijacking--both copying only one Cookie and copying all Cookies (SideJacking).

4.2 Hotmail

The experiment began with testing by SideJacking, and the result showed that the victim's mailbox could be hacked by clicking on URL as shown below (Parameter was not included).

http://by123w.bay123.mail.live.com/mail/InboxLight.aspx

http://by123w.bay123.mail.live.com/mail/ReadMessageLight.aspx

Regarding the test by changing Cookies one by one, it was found that there were 14 involving Cookies in the Domain named as follows.

by123w.bay123.mail.live.com
live.com
mail.live.com

When Cookies were tested one by one, it was found that the victim's mailbox could be hacked by changing the Cookie named "RPSTAuth" in the Domain named live.com.

However, hacking Hotmail was more difficult that hacking Gmail because there were a large number of Cookies in Hotmail which were shown on Opera and Firefox. In addition, the captured Cookies which were needed to test were more than those of Gmail.

As a consequence, it could be concluded that Hotmail has no resistance to hacking by Session Hijacking--both copying only one Cookie and copying all Cookies (SideJacking). Comparing to Gmail, however, finding Session ID in Hotmail is more difficult.

4.3 Yahoo Mail

Yahoo Mail was tested by changing Cookies one by one (from 12 entirely involving Cookies), but it was found that the victim's mailbox could not be hacked. It might be possible that Yahoo Mail uses two Cookies or more; otherwise, it might include another security mechanism. Then, Yahoo Mail was tested by SideJacking, and it was found that there were involving Links as follows.

http://us.mg3.mail.yahoo.com/ws/mail/v1/formrpc?
http://us.mg3.mail.yahoo.com/dc/launch ?.

http://us.mg3.mail.yahoo.com/dc/rs?
http://us.mg3.mail.yahoo.com/fc/fc?
http://us.bc.yahoo.com/
http://geo.yahoo.com/
http://presence.msg.yahoo.com/
http://ts.richmedia.yahoo.com/
http://www.yahoo.com/

However, after the above Links were clicked on and thoroughly tested, it was found that SideJacking tools were unable to hack the victim's mailbox. For this result, it might be possible that Yahoo Mail employs another mechanism apart from Cookies, or it might use Web Technology which has not yet been supported by SideJacking tools.

Therefore, it could be concluded that Yahoo Mail has the resistance to hacking by Session Hijacking--both copying only one Cookie and copying all Cookies (SideJacking).

4.4 Table of result

The results were concluded in Table 1 and Table 2.

Table 1: The Results of the Experiment by Means of Editing Cookies One by One

Hacking Method	Hotmail	Gmail	Yahoo Mail
Number of Cookies Which Must Be Tested	14	8	12
The Name of Cookie Which Contains Session ID	RPSTAuth	GX	–

Table 2: Results of the Experiment

Hacking Method	Hotmail	Gmail	Yahoo Mail
Using One Cookie	Success	Success	Fail
Using All Cookies (Sidejacking)	Success	Success	Fail

Success = able to hack
Fail = unable to hack

5. Conclusion and Further Study

The security level of Hotmail, Gmail and Yahoo Mail has been measured by hacking by means of Session Hijacking. The victim's Cookies and Session ID are captured on LAN, and then Hijacking is conducted in two methods. For the first method, Session Hijacking is conducted by copying only one Cookie. It is found that Yahoo Mail could not be hacked while Gmail and Hotmail could be hacked; Hotmail is more difficult to be hacked than Gmail.

For the second method, Session Hijacking is conducted by copying all Cookies (using SideJacking tools). The results show that Gmail and Hotmail could be hacked while Yahoo Mail could not be hacked.

As a result, it is concluded that the Web Mail which has the highest security is Yahoo Mail; the second one is Hotmail while the Web Mail with the lowest security is Gmail.

The next research to be conducted in the future is testing the security level of the top 10 free Web Mails by the top 10 Web Hacks. The world top 10 free Web Mails would be tested one by one by using the top 10 popular web-hacking methods[6],[7] comprising such as XSS, Session Hijacking, SQL Injection, etc. This research would be conducted in order to find the differences of the security level between 10 free Web Mails. It would be beneficial information for users regarding selecting free Web Mails to use.

References

[1] http://webworkerdaily.com/2007/05/11/web-worker-head-to-head-to-head-gmail-hotmail-and-yahoo-mail/
[2] http://www.erratasec.com/news.html
[3] Hamster tool – Sidejacking (cookie munging / man-in-the-middle), BlackHat, 28 July thru 2 Aug 2007 – Las Vegas, NV.
[4] "Cisco IOS Switch Security Configuration Guide", www.cisco.com
[5] "HTTPS Hacking Protection". Thawatchai Chomsiri. Proc. of the IEEE 21st International Conference on Advanced Information Networking and Applications (AINA-07), Volume 1, IEEE CS Press, May 2007, Niagara Falls, CANADA.
[6] http://www.owasp.org/index.php/Top_10_2007
[7] http://www.infosecurity-magazine.com/webinars/Top_ten_web_application_hack_attacks.html