Casey Gowrie
COMP116
Final Project

# Session Hijacking and the Cloud

**Mentor: Ming Chow**

## Table of Contents

## Abstract

Cloud Computing is a hot topic in tech currently. Many of the large technology companies offer cloud-based services, as do many smaller cloud-computing companies. Many people store a lot of data in the cloud. But not many people know how the cloud actually works. This project will provide a brief overview on how things are stored in the cloud and analyze session hijacking, one of the top vulnerabilities in the cloud. Session hijacking is the use of a valid session key to gain access to the computer system, here the system where a lot of data is stored. Hijacking is a common cloud vulnerability as all of your data is stored on a server, which is often accessed through a web application, that you don't even know the location of, so attackers can gain a wealth of information with very little trace. I will explore how session hijacking gets access to data, possible ways to detect attacks, and protect against them.

# 1. Introduction

## 1.1 What Is The Cloud?

The term cloud storage refers to storing data on an off-site file system hosted by a third party. The files are stored in what can basically be described as a remote database, as opposed to your local machine. Some cloud services are specific and store photos, emails, etc. whereas some are broader and can store many types of data. Data is often stored on multiple servers as a backup if one server fails.

Let us examine how a cloud service works:

1.) The service provider (company hosting the cloud) has a data server connected to the Internet over a network.

2.) A client sends information, the files to store, to the server, which then copies and stores the information.

3.) Clients then access data through a UI (often a web application), such as Google Drive

   a. When a client wants to access data, they send a request to the server, which then returns the information to be edited.

   b. The client can then send back the updated file, which the host makes a new copy of.

## 1.2 The "Cloud Multiplier Effect"

In September of 2014, the Ponemon Institute surveyed over 1000 IT practitioners and revealed that cloud usage is perceived to increase risk of data breach. The report

quantified the expected cost of a data breach in the cloud. The rate of adoption, growth of mobile cloud usage, and ease and speed of sharing through the cloud all contribute to a "multiplier effect," which causes expected cost to increase.

The survey said that the risk multiplier is about 3 times. If the average cost of a data breach of 100,000 records is 20 million and there is a 12 percent chance of this happening within two years, the expected cost is 2.4 million. Let's say we see a 50 percent increase in usage over the next year, then that number would increase two and a half times, to over 5 million, based on the findings on the multiplier effect.

### 1.3 Session Hijacking

When a user logs on to a web service, such as the cloud, a session is created. This user session keeps track of user information, including a session ID, to authenticate user requests for data. When the user logs out, the session is ended, and the session ID no longer makes valid requests. Session data is normally stored within a cookie or as parameters in the URL.

Session hijacking involves the attacker taking over a user session by obtaining a valid session ID. The attacker can then pretend to be the authorized user and make requests as that user. In the cloud, this could lead to a breach of the victims' information that is stored there, making cloud services particularly vulnerable.

Every analysis of security in the cloud lists session hijacking as one of the top three threats to it. Because of this, we will be taking a more in depth look at session hijacking and some of the defenses.

## 2. To the Community

Cloud computing is the future of technology. People continue to move more and more data to the cloud and away from local storage. Since a lot of personal data is now stored on the cloud, a breach can expose a lot of information. Also, lots of companies are starting to use cloud based services, especially smaller businesses. So it is extremely important that people at least understand some of its vulnerabilities.

Many people use the cloud. In July 2013, Apple reported 350 million iCloud users; that's just one service provider. Few of these people even know how data is stored in the cloud, let alone the risks associated with a breach.

66 percent of business professionals have said their organizations use of cloud storage worsened their ability to protect sensitive information. For example, Drop Box was breached just a couple of months ago, with hundred of passwords released and thousands more threated. Session hijacking is one of the biggest security vulnerabilities to the cloud, making it a likely attacking point.

## 3. How a Session Hijacking Attack Happens

A successful session hijacking attack generally follows this pattern: A legitimate wireless station authenticates itself to an access point, meaning the user creates a valid session by logging in. Next, the attacker performs some type of scanning or sniffing attack to gain session data, which is likely found in the browsers' cookies. It is also possible for attackers to brute force session data, as some IDs are created in a more linear manner. Once s/he has valid session information, the attacker sends a disassociation management frame, which is a MAC frame used to manage associations, to the wireless station with a MAC address to be spoofed, allowing HTTP communication to be sent to the attackers instead. After getting this message the real (client's) wireless station terminates the connection to the access point, leaving the attacker with authenticated access to the application, such as a cloud service, where they would then have access to all of the clients information.

This basic outline, of how session hijacking attacks are carried out, allows us to better understand some of the defenses and detection mechanisms relating to the attack and why they can be effective.

## 4. Detection and Defenses

In reality, defenses of session hijacking are useful to any web application that requires authentication and has user sessions, the cloud included. The cloud, today, is the hottest web application of this type. I will discuss where to look for session hijacking attacks and then some of the common defenses and tradeoffs of each one. When working in the cloud, other people's information is being stored on the providers

network. Defenses are important, but attacks are inevitable. In the event of the breach, it is critical to notify users that the data has been compromised. Thus, detecting a breach becomes critical as well, as many attacks may be undetected for large periods of time.

## 4.1 Detection

Below, we will briefly discuss some existing detection methods and some proposed ideas.

### 4.1.1 Existing Methods

A lot of existing methods relate to WLAN intrusion detection systems, as this is a prominent source of attack.

One current system is the monitoring of MAC frame sequence numbers. If there is a significant change, it is a possible sign of intrusion, as the attacker is likely now authenticated and making requests with a different sequence number.

Also, one can do MAC address verification amongst registered users.
Open source products, such as WIDZ are designed to detect rogue access points and wireless stations using similar but a little more in-depth techniques than described above.

Another way to detect attacks is to analyze the lead up to the attack. Packet sniffers can be used to pay attention to repeated ARP updates (watching for spoofing), frames with different MAC addresses, and ACK attacks, which is a

way for the attacker to reset the connection. If one ore more of these attacks are observed, hijacking should be investigated.

### 4.1.2 Proposed Models

The existing models are all effective, except that they do things based on easily calculable things that can be spoofed, such as known MAC addresses and frame sequence numbers (can easily be predicted or eavesdropped). Many proposed methods try to detect based less predictable, more randomized parameters. Examples from Gill, Smith, Looi, and Clark (2005), include monitoring received signal strength, which likely differs greatly from attacker to client, and monitoring round-trip times of the handshake between the client and the service provider, both much harder to mimic. Based on the experiments they did, both seem to be less spoof-able and acceptable methods of detection.

Ideally, there are no break-ins to detect. Although that is wishful thinking, we will now move to specific defenses, so we can hope to lower the chances of these break-ins.

## 4.2 Defense Techniques

First, we will discuss defenses that can be used by service providers to prevent session hijacking. Then, we move to what clients can do to help protect themselves.

### 4.2.1 HTTPS

Many services use SSL to protect login pages, but once the client logs-in do not use encryption. This allows attackers to "Sidejack" a session if they are on the same network by examining the unencrypted HTTP messages.

HTTPS is a combination of the standard HTTP protocol and the SSL (Secure Sockets Layer) protocol, which provides secure cryptographic communication over the Internet. HTTPS can be used from end-to-end, for the entire session, to prevent attackers from sniffing packets to obtain a valid session ID.

At Toorcon 12, Eric Butler released a Firefox add-on called Firesheep that exposed the vulnerabilities of services, such as Gmail, using HTTP after the user had logged in. An attacker could easily download the add-on and start capturing information. Then anyone who logged on to a vulnerable service on the same network could be session hijacked with the click of a button. Firesheep works by sniffing the HTTP packets and stealing cookie data, which will contain session information. These cookies are then copied to the attackers browser to falsely authenticate the attacker as another user.

In a blog post, Butler discussed the only effective fix was full end-to-end encryption (not just a log-in time) with HTTPS. Many services, including Google and Yahoo now do not even offer a normal HTTP option and encrypt all requests.

### 4.2.2 One-Time Cookies (OTC)

HTTPS generally works well to protect cookies, but there are still ways to expose them, and session IDs can be brute forced. Also, HTTPS can be challenging for smaller companies with largely distributed systems because of financial and performance effects. In 2012, Dacosta et. Al. proposed one-time cookies for session authentication.

OTC does not require state synchronization, which could be costly. Through a WordPress plug-in, the authors showed that OTC had a negligible overhead for web applications.

OTC prevents attacks by signing each user request with a session secret that gets securely stored in the users browser. Protecting each user request prevents an attacker from simply brute forcing a session ID to make any request desired as a user.

### 4.2.3 SessionShield

Cross-site scripting (XSS) is another common method for attackers to steal a session ID. XSS can be used to trick the browser into executing malicious code, which may send a users session data to the attacker's computer. SessionShield is a lightweight client-side protection mechanism against XSS session hijacking attacks presented by Nikiforakis et al.

SessionShield is a proxy outside the browser that inspects all incoming/outgoing requests. It will help to protect a user against session hijacking attacks that

manifest as XSS, even if the web application didn't mitigate potential problems properly. The service strips session identifiers out of incoming HTTP requests (from the server) and stores them locally in its own database. Then, in outgoing requests (from the client), SessionShield checks the recipient information (assuring that it is the Web Application), and then adds the stripped information back into the request.

This is especially useful for clients, as it is the only software defense for clients that I have come across. For clients that are moving data on to the cloud and accessing it through web applications, it adds another layer to security despite what the vendor may use.

### 4.2.4 Network Security

There are limited things that one can do on the client side to prevent against the attacks, as service providers must take most measures. Insecure HTTP communication is susceptible to session hijacking, as it makes it easy for attackers to sidejack a session. Public networks are, therefore, very vulnerable. So, if a client limits use of unprotected networks, they can help defend from attacker capturing communication that will contain the session ID.

## 5. Proof of Concept: SSL and HTTPS

For my proof of concept, I created a screen capture video attempting to perform two session hijacking attacks, more specifically the form known as "sidejacking". After testing various session hijacking programs, such as Hamster-Ferret, Paros,

SessionThief, and even Firesheep, I came across a FireFox plug-in, called GreaseMonkey that could be used in conjuction with Wireshark to sniff out session data and inject cookies to hijack a session.

For the first Hijack attempt, I used StackExchange. While I realize this is not a cloud service, it does not use end-to-end encryption and, thus, offers a good representation of what a cloud service that didn't would act like. I was able to capture packets from login, grab cookies, and inject them on another browser to hijack my own session.

For the second Hijack, I tried Gmail, which stores emails on its servers, a cloud service. Because Gmail uses HTTPS for all requests, I was unable to see their cookies in packet details, meaning I could not easily sidejack their session.

The hijack test was a good representation of how cloud services can improve security measures by using full end-to-end SSL encryption for all requests. In performing the hijacks, I observed that most cloud base services have convert to constant HTTPS within the last couple of years, and don't even offer the option to not encrypt requests, most taking it away last year.

The video can be found at:

https://www.youtube.com/watch?v=EIHOPopei0U

## 6. Conclusion

Session hijacking is a vulnerability in many web applications that permit user sessions, but has come even further to the forefront with the growth of cloud applications.

Cloud applications often use web applications and store a lot of a users data. Because of this, users stand a large risk with the possibility of session hijacking. A successful session hijacking attack on the cloud will expose a lot of user data to the attacker.

Detection is difficult, because through various spoofing, the attacker can make the connection to the server look very much like that of a previous client. Defenses include MAC address verification and MAC frame sequence numbers, but could be strengthened by adding more variability in parameters that are checked, such as signal strength.

Ideally, prevention is better. Most preventative measures are left to the service providers. Defenses include full end-to-end encryption using HTTPS. A few years ago, a majority of applications only used HTTPS upon authenticating log-ins, which allowed sniffing of HTTP packets to release session data. With SSL encrypting request in more places now, an attacker can't discern cookie setting in plain site. Also, packages like one-time cookies can be used to thwart attacks by changing the way requests are authenticated.

Users do have a few opportunities to protect themselves. SessionShield is lightweight user protection that can correct XSS vulnerabilities in websites by protecting session identifier data. Clients can also do things as simple as using protected wireless to prevent network sniffing that may expose session data.

All in all, with careful steps taken on both sides of the requests, hijacking can be minimized and help the cloud keep user data safe. With cloud use growing and the multiplier effect active in the eyes of IT professionals, we need to take as many steps possible to prevent and detect hijacking issues in the cloud. Luckily, cloud service providers realize the importance of protecting the data they store, and are paying careful attention to vulnerabilities that can be exploited from the Internet interaction.

## Works Cited


Butler, Eric. "Firesheep." *{codebutler}*. N.p., 24 Oct. 2010. Web. 10 Dec. 2014.

Dacosta, Italo, et al. "One-time cookies: Preventing session hijacking attacks with stateless authentication tokens." *ACM Transactions on Internet Technology (TOIT)* 12.1 (2012): 1.

Gill, Rupinder S., et al. "Passive techniques for detecting session hijacking attacks in IEEE 802.11 wireless networks." (2005): 26-38.

Halton, Wolf. "Security Issues and Solutions in Cloud Computing." *Wolf Halton OpenSource Security*. N.p., 25 June 2010. Web. 10 Dec. 2014.

MacMillan, Douglas. "Dropbox Blames Security Breach on Password Reuse." *Digits RSS*. WSJ, 14 Oct. 2014. Web. 26 Oct. 2014.

Mervat Adib Bamiah* et al. / (IJAEST) INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES Vol No. 9, Issue No. 1, 087 - 090

Nair, Suthish. "How to Avoid Session Hijacking in Web Applications." *TechNet*. Microsoft, 25 Aug. 2013. Web. 26 Oct. 2014.

Nikiforakis, Nick, et al. "SessionShield: Lightweight protection against session hijacking." *Engineering Secure Software and Systems*. Springer Berlin Heidelberg, 2011. 87-100.

"Protecting Your Users From Session Hijacking." *EPiServerWorld*. 1 Jan. 2014. Web. 11 Dec. 2014. <http://world.episerver.com/Documentation/Items/Tech-Notes/EPiServer-CMS-6/EPiServer-CMS-60/Protecting-Your-Site-From-Session-Hijacking/>.

Rouse, Margaret. "Session Hijacking (TCP Session Hijacking)." *What Is ?* 1 Sept. 2009. Web. 26 Oct. 2014.

"The Cloud Multiplier Effect." *Cloud Security Alliance*. N.p., 2014. Web. 9 Dec. 2014. <https://blog.cloudsecurityalliance.org/wp-content/uploads/2014/09/NS-Data-Breach-EU-IG-00.png>.

Trusted Computing Group. "Cloud Computing and Security - A Natural Match." *Trusted Computing Group*. 1 Apr. 2010. Web. 11 Dec. 2014. <http://www.trustedcomputinggroup.org/files/resource_files/1F4DEE3D-1A4B-B294-D0AD0742BA449E07/Cloud Computing and Security Whitepaper_July29.2010.pdf>.

Whitaker, Andrew, and Daniel P. Newman. *Penetration testing and network defense*. Pearson Education, 2005.