Term Paper: Firesheep

By Girum Ibssa

CSC 300: Professional Responsibilities

Dr. Clark Turner

May 24, 2013

Abstract

Firesheep is an extension for the Firefox web browser that wraps Wireshark (an existing piece of session hijacking and packet sniffing software [1]) in a simple GUI [4]. Created by Eric Butler and Ian Gallagher, Firesheep's motivation was described by its creators: "We're bringing up this tired issue to remind people of the risks they face, especially when on open WiFi networks, and to remind companies that they have a responsibility to protect their users. To drive this point home, we are releasing an open source tool at ToorCon 12 which shows you a 'buddy list' of people's online accounts being used around you, and lets you simply double-click to hijack them" [14].

Was it ethical to release this software? Eric stated in his release of Firesheep that "It's extremely common for websites to protect your password by encrypting the initial login, but surprisingly uncommon for websites to encrypt everything else...This is a widely known problem that has been talked about to death, yet very popular websites continue to fail at protecting their users" [5]. Yet the use of Firesheep may be illegal in the US and beyond [16]. I will show that Firesheep was indeed ethical to release primarily due to SE Code 1.04, the requirement for Software Engineers to fully disclose any potential software danger to the public [12].

Contents

1 Facts

1.1 Description of Firesheep

Eric Butler describes in his blog: "When logging into a website you usually start by submitting your username and password. The server then checks to see if an account matching this information exists and if so, replies back to you with a 'cookie' which is used by your browser for all subsequent requests [5].

"It's extremely common for websites to protect your password by encrypting the initial login, but surprisingly uncommon for websites to encrypt everything else. This leaves the cookie (and the user) vulnerable. HTTP session hijacking (sometimes called "sidejacking") is when an attacker gets a hold of a user's cookie, allowing them to do anything the user can do on a particular website. On an open wireless network, cookies are basically shouted through the air, making these attacks extremely easy" [5]. Butler continues with his description of the "problem that has been talked about to death" and how it remains to be solved [5].

Qualified hackers were already able to perform session hijacking like this well before the release of Firesheep, using the program 'Wireshark' (which Firesheep is built on top of). [1].

Firesheep works by allowing users to write custom "handlers" for it to allow it to work on webpages of their choice. Firesheep's source code currently includes "handlers" for several popular websites including: Amazon, Basecamp, bit.ly, Enom, Facebook, FourSquare, Github, Google, Hacker News, Harvest, The New York Times, Pivotal Tracker, Twitter, ToorCon: San Diego, Evernote, Dropbox, Windows Live, Cisco, Slicehost, Gowalla, Flickr and Yahoo [4].

Some of those websites have fixed the security hole (by authenticating the entire site with HTTPS) that Firesheep relies on to work [8]. However, a strong portion of websites simply didn't fix the security flaw, weeks and months after the release of Firesheep [8].

2 Research Question

Was it ethical for Eric Butler to release Firesheep to the general public as a means of forcing websites to improve their flawed security?

2.1 Relevance

Firesheep can be downloaded today (May 24, 2013) from GitHub.com, a website hosting free open source programs [4]. At its time of release, several websites simply refused to implement HTTPS site-wide, allowing Firesheep to work on a large number of sites [5]. Today, Firesheep still works on a few sites that haven't switched to using HTTPS site-wide, including the entire Stack Exchange network of websites [10].

Session hijacking allows users of Firesheep to impersonate "logging in" as their victims on the sites that it works on [5]. The consequences of this can be anything from posting false Facebook status updates to outright deleting a person's Stack Overflow profile. Eric Butler's free software simplifies this process to the point where normally unqualified people may perform these acts, increasing the vulnerability of typical users worldwide.

3 Extant arguments

3.1 In Favor of Firesheep's release

3.1.1 Code as "free speech"

Eric Butler himself argues in favor of the release of Firesheep. In his article "Firesheep, a week later: Ethics and Legality", Butler states outright that "it is nobody's business telling you what software you can or cannot run on your own computer" [7]. He defends by saying that code is a form of free speech, and that we have a Constitutional right to free speech [7].

3.1.2 Mozilla's support of Firesheep

Mozilla themselves support Firesheep. Firefox (the browser for which Firesheep is an extension for) features an internal blacklist of extensions that it does not allow to work [2]. Mozilla, the creators of Firefox, specifically decided not to blacklist Firesheep [17]. Mike Beltzner, director of Firefox, praised its release: "[Firesheep] demonstrates a security weakness in a number of popular websites, but does not exploit any vulnerability in Firefox or other Web browsers" [17].

3.2 Against Firesheep's release

3.2.1 Real-world use of Firesheep may be illegal

The actual use of Firesheep, however, may be illegal [16]. Federal wiretapping laws state that it's not illegal "to intercept or access an electronic communication made through an electronic communication system that is configured so that such electronic communication is readily accessible to the general public" [16]. However, Jonathan Gordan, partner at Los Angeles law firm Alson and Bird, states that "when people are accessing their social network [account], they have an expectation that whatever they're doing is governed by the privacy settings in that network", and that a open Wi-fi network does not qualify as "readily accessible to the general public" [16].

3.2.2 Firesheep desired to violate privacy

Firesheep is built specifically to allow untrained users to hijack browser sessions from unsuspecting victims [5], which directly contradicts Section 1.03 of the Code of Ethics: "1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment" [12].

4 Analysis

4.1 Why is the SE Code of Ethics applicable to this problem?

The Code defines software engineers as "those who contribute by direct participation to the...design, development...of software systems" [12]. Did Eric Butler contribute by "direct participation" to the design of some "software system"?

Firesheep is the name of software written in the C++ programming language that wraps existing packet sniffing software (Wireshark) in an easy to use, one-click GUI extension for the Firefox browser [4]. Firesheep is a software system directly written by Eric Butler, maintained on his open source GitHub account [4]. Eric But-

ler (the software engineer) has therefore contributed by direct participation (programming himself) in the design of a "software system" (Firesheep) [4].

Since he contributed by "direct participation" to "the design" of a "software system," Eric Butler is defined by the ACM Code of Ethics to be a "software engineer," and is therefore bound to adhere to the rules of the Code[12].

4.2 Argument 1: Disclosure

4.2.1 SE Code 1.04

Disclose to appropriate persons ... any actual or potential danger to the user, the public ... that they reasonably believe to be associated with software [12].

4.2.2 Appropriate persons (to disclose to)

The appropriate persons to disclose this session hijacking problem to are the majority of the general public. As I will explain later in this argument, session hijacking was a known problem to a small subset of the public, and it needed to be disclosed to a much larger audience.

4.2.3 Dangers

The dangers of session hijacking are simple: "It's extremely common for websites to protect your password by encrypting the initial login, but surprisingly uncommon for websites to encrypt everything else. This leaves the cookie (and the user) vulnerable. HTTP session hijacking (sometimes called "sidejacking") is when an attacker gets a hold of a user's cookie, allowing them to do anything the user can do on a particular website. On an open wireless network, cookies are basically shouted through the air, making these attacks extremely easy" [5].

4.2.4 (Endangered) Users

The victims of potential session hijacking are simply any users of the affected websites (e.g. Facebook, Google) that are so unfortunate as to be physically near any session hijacking predators.

Users logged into any one of these sites are 'affected' by these sites' inability to use site-wide HTTPS to prevent session hijacking.

4.2.5 Dangers associated with non-site-wide HTTPS authentication

Since Firesheep works by intercepting the "cookies [that] are basically shouted through the air" [5], any users who are on the same open WiFi network as Firesheep users are potential victims of the dangers associated with non-site-wide HTTPS authentication.

4.2.6 Substituted SE Code 1.04

[Eric Butler shall] Disclose to the general public ... any session hijacking dangers to the users ... that they reasonably believe to be associated with non-site-wide HTTPS authentication [12].

4.2.7 Argument 1 Analysis

The release of Firesheep can be seen as a programmer's attempt to 'show by doing'; Butler chose to release Firesheep as a catalyst for the seriousness of the HTTPS problem.

Attempting to alleviate this HTTPS problem by popularizing session hijacking via a beautiful GUI (and thus openly catalyzing the problem) is an excellent form of danger disclosure to the general public.

Session hijacking was already possible well before the release of Firesheep through software such as Wireshark [1]. Firesheep itself is actually just an extension of Wireshark [4]. Firesheep was so significant simply because it offers a much simpler GUI than traditional session hijacking solutions.

The 'affected websites' to be spoken of at the time of release included websites as large as Google, Facebook and Yahoo! [4]. Google's Google+ social network has 343 million users [19], Yahoo's email service has 281 million users [20], and Facebook has over 1 billion users [3]. Even today, Firesheep works on sites like Stack Overflow, which over 2 million registered users at the time of writing [11].

What can you do with session hijacking? At the time of its release, Firesheep worked on the social networks Facebook and Google+. I, a session hijacker, could easily take control of your Facebook session (and thus your Facebook profile) and post any changes I want to your Wall, News Feed, or friends' News Feeds. I could delete some of your friends, or what's worse, I could record the information of some of your more vulnerable Facebook friends to do any sort of criminal activity I wanted.

Imagine another scenario: Facebook's News Feed contains timestamped location-tagged photos of your friends in it. I just hijacked your Facebook session, and see that your friend John Doe just took a picture five minutes ago of a sweet margarita that he's sipping with his girlfriend in Miami. John Doe also took a picture of his new dog – this picture was taken two weeks ago and was geotagged at his house, which I looked up in a public database to be 258 Chorro Street in California. John's in Miami, his house in California is empty, and I'm the stranger sitting in Starbucks hijacking your Facebook session wondering how much of John's stuff I can lug out of his house while he's gone.

Today, Facebook has correctly authenticated its entire site via HTTPS, but some websites still refuse to fix the issue. Take Stack Overflow for instance [10]. Stack Overflow's parent company Stack Exchange simply doesn't want to authenticate their whole website with HTTPS, meaning that Firesheep still works just fine on it today. Stack Overflow is a website for programmers with questions about programming that other programmers can answer. It persists how much help you've given other people over your account's lifetime, and publicly displays how reputable you are in your posts. Some people take this reputation very seriously.

Imagine I'm at a hackathon in San Francisco and you're a hot-shot programmer who also happens to be at this hackathon. We're both on the same open Wi-Fi connection here at this hackathon. We're both on Stack Overflow, but my reputation level on the website is novice level, while yours boasts years of good reputation built up from always lending a helping hand. Well, we're on the same Wi-Fi network and I decide to double-click your name on Firesheep. I've just hijacked your session thanks to Stack Overflow not authenticating their website through HTTPS: I now have control of your Stack Overflow account. I can do anything I want as you now. I can "troll" help-

less novice programmers with incorrect responses for a quick laugh. I can send a Stack Overflow 'Bounty' to myself, rewarding my own, real Stack Overflow account with a ton of the good reputation that you've built up on your account. Or, if I get bored, I can just delete your account altogether. I never really liked you for winning this hackathon anyways, right?

To paraphrase SE Code 1.04: 'if there's serious danger associated with some software, you're morally obligated to tell the right person what's going on' [12]. Eric Butler developed Firesheep as a concrete example of how serious the HTTPS problem was: "Today at Toorcon 12 I announced the release of Firesheep, a Firefox extension designed to demonstrate just how serious this problem is" [5]. That is to say, Firesheep was released specifically with the intent to educate the general public of the security holes many major websites had concerning non-authenticated HTTP sessions. In the blog post accompanying the release of Firesheep, Eric Butler made it very clear that he believed that non-HTTPS session hijacking was a serious problem: "This is a widely known problem that has been talked about to death, yet very popular websites continue to fail at protecting their users" [5].

The problem with non-HTTPS session hijacking at the time was that the users who were most vocal about HTTPS were in the minority. Companies like Facebook went years without dealing with the problem, claiming that the problem simply wasn't worth the engineering hours that it would take to fix it [6]. Session hijacking had been brought up as an issue within the security community since 2004 [6]. To use Facebook as an example: why would Facebook care if they make most of their revenue from the ads targeted at the other 99% of their users? That is to say, Facebook had no financial incentive to heed the warnings from the small security community (made of engineers like Butler) who were originally so vocal about the issue. The only way for the something to change was for the issue to come to light to the other 99% of Facebook.

Therefore, Eric Butler had an ethical responsibility, by SE Code 1.04, to create and release of Firesheep to the general public in order to disclose the dangers of session hijacking to the public.

4.3 Argument 2: Volunteer professional skills

4.3.1 SE Code 1.08

Be encouraged to volunteer professional skills to good causes [12].

4.3.2 Professional skills

Professional skills in this case are simply the skills Eric Butler had in writing an effective, beautiful GUI for Wireshark (the existing session hijacking software) [1].

4.3.3 Good causes

Butler's release of Firesheep was considered, at least from his point of view [5], a good cause. This will be explained fully in the analysis portion of this argument.

4.3.4 Substituted SE Code 1.08

[Eric Butler should] be encouraged to volunteer his skills writing an effective GUI for Wireshark to the good cause of preventing session hijacking.

4.3.5 Argument 2 Analysis

Any qualified software cracker could have used Wireshark before; Butler simply provided his professional skills by integrating Wireshark right into Firefox and writing it to be mostly autonomous. What was once a process reserved for only the most expert of software crackers accustomed to software like Wireshark became a one-click 'buddy-list' for session hijackers to choose their victims from [5]. Overnight, Wireshark turned into Firesheep. Firesheep gained 129,000 users in those 24 hours, garnering many "Top Tweets" on Twitter and turning into the #10 most searched Google query in the United States [6]. Butler's professional skills turning Wireshark into Firesheep are certainly impressive in their own right.

But was it a 'good cause'? In its simplest, Butler forcing the hand of the websites responsible for HTTPS session hijacking vulnerability to fix their security flaw should be interpreted as 'good cause' because the resulting HTTPS fix entirely eliminates potential future session hijackings from occurring.

Prior to Firesheep, popular websites were rampantly vulnerable to session hijacking at the time of Firesheep's release [4]. The list of vulnerable websites is listed in Section 1.1, and included sites as popular as Google and Facebook. Additionally, "the risks of insecure websites have been known for years, yet little has been done about what has become an increasingly widespread problem" [8].

Victims of session hijacking ultimately have their 'happiness' taken from them in exchange for the happiness of the culprit. Preventing the session hijacking altogether results in a net gain in happiness of the society and is thus a good cause, according to utilitarian ethics [15].

Put differently: the release of Firesheep quickly led to the security fix that was so desperately needed in the industry. There was certainly an overhead societal cost of having Firesheep run rampant in the weeks it took for websites to fix this flaw (see Argument 1's analysis). This overhead would have resulted in a net loss in happiness to the general public. However, the overall result of Firesheep was that non-HTTPS session hijacking mostly became a thing of the past. The disappearance of session hijacking as a problem results in net happiness to the general public, and that net positive result accumulates for every year that session hijacking is not the problem it was from 2004-2010 [6].

The ACM Code of Ethics therefore applies to Eric Butler as he was indeed encouraged to volunteer his GUI skills to this good cause. Similar to Argument 1, the rampancy of vulnerable HTTP websites at the time essentially necessitated Butler's intervention by releasing the Firefox GUI over Wireshark. By this logic, Butler was not only ethical in releasing Firesheep to the general public, but was entirely encouraged to do so on a basis of the utilitarian aspects of the Code of Ethics, Section 1.08.

4.4 Argument 3: Ensure an appropriate method

4.4.1 SE Code 3.05

Ensure an appropriate method is used for any project on which they work or propose to work.

4.4.2 An appropriate method

The appropriate method to solve this problem was the release of (controversial) software designed to accelerate the rate of response to HTTP session hijacking vulnerabilities.

4.4.3 (Software) Project

The project Butler was working on here was essentially the well-being of the HTTP protocol as a whole. If not that, then the 'project' was at least this particular vulnerability of the protocol.

4.4.4 Substituted SE Code 3.05

Ensure an appropriate method (software designed to accelerate the rate of response to HTTP vulnerabilities) is used to promote the well-being of the HTTP protocol as a whole.

4.4.5 Argument 3 Analysis

As was stated in previous arguments in this report, HTTP session hijacking was a problem that was talked about to death among security professionals for years [6]. That is to say, companies have been outright ignoring this problem for (at least six [6]) years with no intention of changing their stance at the time of Firesheep's release. For a person in a position like Eric's it would be easy to argue that after six years of fruitless warnings, it was about time to 'raise the stakes' a little and force the hands of these non-compliant software companies. If verbal signals for those six years turned out to not be an appropriate method for companies as (evidently) ignorant as this, then a 'show by doing' demonstration of the gravity of the HTTP vulnerability was certainly an appropriate method.

Following the guidelines of SE Code section 3.05, Butler thus had no choice but to use a more appropriate method to deal with this known issue: he had to raise the stakes a little and release Firesheep to the public. Software engineers under this code have an ethical responsibility to ensure that an appropriate method is used; SE Code 3.05 is yet another perspective on why Butler was ethical and ethically obligated to release Firesheep.

5 Conclusion

A common theme throughout this discussion has been Eric Butler's ethical responsibility to do something about the rampant vulnerability that websites had to session hijacking. It was, like Eric said, a problem that was talked about to death. Yet, very few software companies were willing to spend the engineering hours fixing it until Butler trivialized the session hijacking process. The Code of Ethics describes situations like this in sections 1.04, 1.08, 6.08 and 1.05 [12], and the Code states that Butler had a moral responsibility to do something about it. His release of Firesheep was ethical.

Software engineers in similar positions should consider the approach Butler used in garnering attention towards an issue: let your code speak. If some piece of software is the cause of some major danger to people, software is also likely to be a solution to said danger. This was especially true in Butler's case, and can be true for many software engineers who are in a position to do some-

thing morally sound according to the Code of Ethics. To paraphrase section 1.04 once again: "if you see some danger associated with some software, do something about it" [12].

References

[1] "About wireshark." [Online]. Available: http://www.wireshark.org/about.html

Wireshark official website

[2] "Extensions.blocklist.enabled." [Online]. Available: http://kb.mozillazine.org/Extensions.blocklist.enabled

Firefox's own details of how its blacklist works, and what it was for.

[3] "Facebook reports first quarter 2013 results," May 2013. [Online]. Available: http://investor.fb.com/releasedetail.cfm?ReleaseID=761090

Facebook's first quarter results

[4] E. Butler, "Firesheep source code (github)." [Online]. Available: https://github.com/codebutler/firesheep/wiki/Handlers

Source code and documentation for Firesheep

[5] ——, "Firesheep," October 2010. [Online]. Available: http://codebutler.com/firesheep/?c=1

Eric Butler's personal blog post detailing Firesheep.

[6] ——, "Firesheep, a day later," October 2010. [Online]. Available: http://codebutler.com/firesheep-a-day-later/

Eric Butler's personal blog; his thoughts on Firesheep the day after release

[7] ——, "Firesheep, a week later: Ethics and legality," November 2010. [Online]. Available: http://codebutler.com/firesheep-a-week-later-ethics-and-legality/

Eric Butler's personal blog post detailing the ethics and legality of Firesheep

[8] ——, "Firesheep, three weeks later: Fallout," November 2010. [Online]. Available: http://codebutler.com/firesheep-three-weeks-later-fallout/

Eric Butler's personal blog post detailing the fallout of Firesheep

[9] M. W. Dictionary, "Responsibility definition."

[10] S. Exchange, "Why doesn't the stack overflow team fix the firesheep style cookie theft?" November 2010. [Online]. Available: http://meta.stackoverflow.com/questions/69171/why-doesnt-the-stack-overflow-team-fix-the-firesheep-style-cookie-theft

A post on Stack Exchange asking why they outright refuse to use HTTPS site-wide.

[11] ——, "Stack overflow users," May 2013. [Online]. Available: http://stackexchange.com/leagues/1/week/stackoverflow

Live query on Stack Overflow's current user base.

[12] A. for Computing Machinery, "Software engineering code of ethics and professional practice." [Online]. Available: http://www.acm.org/about/se-code

The ACM Code of Ethics that this paper uses to define 'ethics'

[13] R. Garner, "Moral philosophy: A systematic introduction to normative ethics and meta-ethics."

[14] S. Gibson, "Security now! episode 272 (transcript)," October 2010. [Online]. Available: http://www.grc.com/sn/sn-272.htm

Transcript of a podcast in which Leo Laporte and Steve Gibson discuss Firesheep.

[15] R. Hursthouse, "Virtue ethics."

[16] G. Keizer, "Is it legal to use firesheep at starbucks?" November 2010. [Online]. Available: http://www.computerworld.com/s/article/9194159/Is_it_legal_to_use_Firesheep_at_Starbucks_

Article on the legality of Firesheep

[17] ——, "Mozilla: No 'kill switch' for firesheep add-on," October 2010. [Online]. Available: http://www.computerworld.com/s/article/9193420/Mozilla_No_kill_switch_for_Firesheep_add_on?taxonomyId=17&pageNumber=1

Computerworld article detailing that Mozilla either can't or won't disable Firesheep from working in Firefox using Firefox's blacklist mechanism

[18] B. Kennish, "Widgetjacking: Why more social widgets mean less secure wi-fi," November 2011. [Online]. Available: https://blog.disconnect.me/widgetjacking

A blog discussing "life after Firesheep"

[19] M. McGee, "Report: Google+ now 2nd-biggest social network worldwide," 2012. [Online]. Available: http://marketingland.com/report-google-now-2nd-biggest-social-network-worldwide-31908

A post detailing Google's Google+ statistics

[20] R. Molla, "Gmail finally beats hotmail, according to third-party data [chart]," 2012. [Online]. Available: http://gigaom.com/2012/10/31/gmail-finally-beats-hotmail-according-to-third-party-data-chart/

A post on detailing the popularity of email clients

[21] S. J. Purewal, "Firesheep's a huge hit with amateur hackers," October 2010. [Online]. Available: http://www.pcworld.com/article/208773/Firesheeps_a_Huge_Hit_with_Amateur_Hackers.html

News post about Firesheep being popular among unqualified hackers

[22] TechnoLlama, "Is firesheep illegal?" November 2010. [Online]. Available: http://www.technollama.co.uk/is-firesheep-illegal

UK Article on the legality of Firesheep