Firesheep: Disclosing Dangers

By Girum Ibssa

CSC 300: Professional Responsibilities

Dr. Clark Turner

June 3, 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 [15]. 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" [24].

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" [16]. Yet the use of Firesheep may be illegal in the US and beyond [28]. 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 [22].

Contents

1 Facts **1**

1.1 Description of Firesheep 1

2 Research Question **1**

2.1 Relevance 1

3 Extant arguments **2**

3.1 In Favor of Firesheep's release 2

3.1.1 Code as "free speech" 2

3.1.2 Mozilla's support of Firesheep 2

3.2 Against Firesheep's release 2

3.2.1 Real-world use of Firesheep may be illegal 2

3.2.2 Firesheep desired to violate privacy 2

4 Analysis **2**

4.1 Why is the SE Code of Ethics applicable to this problem? 2

4.2 Argument 1: Disclosure 3

4.2.1 SE Code 1.04 3

4.2.2 Actual or potential danger 3

4.2.3 Disclose 4

4.2.4 Substituted SE Code 1.04 5

4.3 Argument 2: Volunteer professional skills 6

4.3.1 SE Code 1.08 6

4.3.2 Volunteer 6

4.3.3 Professional skills 6

4.3.4 A Good Cause 6

4.3.5 Substituted SE Code 1.08 7

4.3.6 Conclusions drawn 7

4.4 Argument 3: Ensure an appropriate method 8

4.4.1 SE Code 3.05 8

4.4.2 The Project: Ensure Correct Usage of the HTTPS Protocol 8

4.4.3 An appropriate method 8

4.4.4 Substituted SE Code 3.05 8

4.4.5 Conclusions drawn 8

5 Conclusion **9**

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 [16].

“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” [16]. Butler continues with his description of the “problem that has been talked about to death” and how it remains to be solved [16].

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 [15].

Some of those websites have fixed the security hole (by authenticating the entire site with HTTPS) that Firesheep relies on to work [19]. However, a strong portion of websites simply didn’t fix the security flaw, weeks and months after the release of Firesheep [19].

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 (June 3, 2013) from GitHub.com, a website hosting free open source programs [15]. At its time of release, several websites simply refused to implement HTTPS site-wide, allowing Firesheep to work on a large number of sites [16]. 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 [20].

Session hijacking allows users of Firesheep to impersonate ”logging in” as their victims on the sites that it works on [16]. 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" [18]. He defends by saying that code is a form of free speech, and that we have a Constitutional right to free speech [18].

3.1.2 Mozilla's support of Firesheep

Mozilla themselves supported Firesheep at its time of release. 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 [29]. 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" [29].

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 [28]. 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" [28]. 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" [28].

3.2.2 Firesheep desired to violate privacy

Firesheep is built specifically to allow untrained users to hijack browser sessions from unsuspecting victims [16], 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" [22].

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" [22]. 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 [15]. Firesheep is a software system directly written by Eric Butler, maintained on his open source GitHub account [15]. Eric Butler (the software engineer) has therefore contributed by direct participation (programming himself) in the design of a "software system" (Firesheep) [15].

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[22].

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 [22].

4.2.2 Actual or potential danger

The first part of this section to be discussed is the actual or potential danger that results from session hijacking. 'Actual' is defined as "existing in act or fact" [5]. 'Potential' is defined as "capable of being or becoming" [6]. 'Danger' is defined as "liability or exposure to harm or injury; risk; peril" [14].

The dangers of session hijacking are described by Butler: "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" [16].

Since Firesheep works by intercepting the "cookies [that] are basically shouted through the air" [16], 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.

The websites affected by the dangers of HTTP session hijacking at the time of Firesheep's release included websites as large as Google, Facebook and Yahoo! [15]. Google's Google+ social network has 343 million users [31], Yahoo's email service has 281 million users [32], and Facebook has over 1 billion users [11]. Even today, Firesheep works on sites like Stack Overflow, which over 2 million registered users at the time of writing [21].

Potential danger: Facebook session hijacking

Session hijacking has several potential dangers [16]. At the time of its release, Firesheep allowed users to perform session hijacking on social networks such as Facebook and Google+ [16]. A session hijacker would take control of your Facebook HTTP session (and thus your Facebook profile) for the duration of that session [16]. He could post updates to your Wall, your News Feed, or your friends' News Feeds. He could delete friends from your Friends list, or worse, he could record the information of some of your more vulnerable Facebook friends to do any sort of criminal activity he wanted.

Facebook's News Feed contains timestamped location-tagged photos of your friends in it [3]. Let's say that a session hijacker just hijacked your Facebook session while you were sitting in Starbucks. He sees that your friend John Doe just uploaded (five minutes ago) a picture of his lunch while on vacation in Miami [3]. Two weeks before this Miami vacation, John Doe became the owner of a new dog. He uploaded a picture of this new dog – this picture was taken two weeks ago and was geotagged at his house [3]. The session hijacker reverse geocodes those coordinates via Google Maps [13] and determined it to be 258 Chorro Street in California. John is still in vacation in Miami [3], his house in California is empty [13], and the stranger sitting in Starbucks hijacking your Facebook session [16] is wondering how much of John's stuff he can haul out of John's house while John is in Miami.

Potential danger: Stack Overflow session hijacking

Today, Facebook has correctly authenticated its entire site via HTTPS, but some websites still refuse to fix the issue. Take Stack Overflow for instance [20]. Stack Overflow's parent company Stack Exchange still refuses to authenticate their whole website with HTTPS, meaning that Firesheep still works on it today. Stack Overflow is a website for programmers with questions about programming that other programmers can answer [21]. The site persists how much help you've given other people over your account's lifetime, and publicly displays how reputable you are in your posts [21].

Imagine I'm at a hackathon in San Francisco [4] and you're an experienced 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 Stack Overflow users, but I have the reputation level of a novice, while yours boasts years of good reputation built up from always lending a helping hand. Since we're on the same open Wi-Fi network, I decide to double-click your name on Firesheep [16]. I've just hijacked your session [16] thanks to Stack Overflow not authenticating their website through HTTPS [20] and now have control of your Stack Overflow account [21]. I can do anything I want as you now [16]. I can distribute incorrect advice to people for recreational purposes [21]. I can allocate Stack Overflow 'Bounty' to myself [21], rewarding my own Stack Overflow account with the good reputation that you've built up on your account [21]. Or, if I get bored, I can just delete your account altogether [21]. This could all potentially be done while hijacking your HTTP session via open Wi-Fi at this hackathon, and could still happen today [20].

4.2.3 Disclose

To 'disclose' something means "to make known; reveal or uncover" [7]. By definition, this requires that HTTP session hijacking was (relatively speaking) an unknown problem before the release of Firesheep.

At the time of Firesheep's release, the users who were most vocal about HTTP session hijacking were in the minority [17]. 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 [17].

Other tools attempting to expose this HTTP session hijacking problem came and went [17]. "Very little has changed after each of these tools were released. They got their media hype, and then people forgot or didnt care. For the most part, the tools were only used by tech-savy people, hackers and geeks" [17].

Session hijacking had been brought up as an issue within the security community since 2004 [17]; software companies simply ignored the issue up until Firesheep's release [17]. Firesheep was another software tool aimed to demonstrate (disclose) just how vulnerable websites were to HTTP session hijacking [16].

4.2.4 Substituted SE Code 1.04

Disclose (demonstrate via software) to appropriate persons ... any session hijacking danger to the user, the public ... that they reasonably believe to be associated with software [22].

Conclusions Drawn

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" [16]. 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" [16].

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' [22].

The appropriate persons to disclose this session hijacking problem to are the majority of the general public. To be sure, session hijacking was a known problem to a small subset of the public, and it needed to be disclosed to a much larger audience.

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 [15]. Firesheep was so significant simply because it offers a much simpler GUI than traditional session hijacking solutions. 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.

To use Facebook as an example of how cynical companies were about the HTPPS problem: Facebook makes most of their money from advertising revenue [12]. It follows then that 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. With companies as large as Facebook ignoring the issue, the only way to create a change (at the time) was for the issue to come to light to the non-technical majority of the Internet's users.

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 [22].

4.3.2 Volunteer

To 'volunteer' something means "to offer oneself or one's services for some undertaking or purpose" [8]. Eric Butler wrote Firesheep in his spare time and without pay [16], thus offering his services to the community for free. He also open sourced to the public, allowing others extend it by writing their own handlers for it [16]. His purpose for writing Firesheep, as documented in its source code, was to demonstrate HTTP session hijacking vulnerabilities [15].

4.3.3 Professional skills

A 'professional' is "a person who is expert at his or her work" [9]. 'Skills' are "the ability, coming from one's knowledge, practice, aptitude, etc., to do something well" [10].

The professional skills, in this case, are the skills Eric Butler used to write an effective GUI for Wireshark (the existing session hijacking software) [1].

An Effective GUI

Butler states "Firesheep is doing the exact same thing as these other tools [Wireshark], but with a simpler user interface... Because of its simplicity, Firesheep has already succeeded in demonstrating the risks of insecure websites to a much wider audience than any previous tool, in a single day" [17].

He backs this claim with numbers: "Since being released just over a day ago, Firesheep has been downloaded over 129,000 times. Firesheep has consistently been one (if not more) of the 'Top Tweets' on Twitter, on top of Hacker News, was at one point the #10 trending search on Google in the US, and is the second suggestion on Bing when you start typing fire. Firesheep has been mentioned on countless blogs and news sites in numerous languages, and has received almost universal praise" [17].

Results of Butler's professional skills

Qualified software hackers used software like Wireshark before the existence of Firesheep [1]. Butler provided his professional skills by integrating Wireshark within Firefox and writing it to be mostly autonomous [16]. What was once a process reserved for only the most expert of software crackers accustomed to software like Wireshark [1] became a one-click 'buddy-list' for session hijackers to choose their victims from [16]. Overnight, Wireshark turned into Firesheep, with its 129,000 users and multitude of media coverage [17].

4.3.4 A Good Cause

"Good cause is a legal term denoting adequate or substantial grounds ... to take a certain action, or to fail to take an action prescribed by law. What constitutes a good cause is usually determined on a case by case basis and is thus relative" [25].

Substantial grounds to take a certain action

‘Good cause’, by definition, is calculated on a case by case basis. In the case of Firesheep, Eric Butler believed (and wrote in the release notes of Firesheep) that the actual and potential dangers of HTTP session hijacking compelled him to create and release Firesheep [16]. The objective evidence of HTTP session hijacking indeed having dangers is presented in Argument 1 of this report.

4.3.5 Substituted SE Code 1.08

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

4.3.6 Conclusions drawn

Butler forcing the hand of the websites responsible for HTTP session hijacking vulnerability to fix their security flaw should be interpreted as ‘good cause’: the resulting HTTPS fix eliminates potential session hijackings from occurring on that website [16].

Prior to Firesheep, popular websites were rampantly vulnerable to session hijacking at the time of Firesheep’s release [15]. 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” [19].

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 [26].

Put differently: the release of Firesheep quickly led to the security fix that was so desperately needed in the industry. There was an overhead cost to society of having Firesheep run rampant in the weeks it took for websites to fix this flaw (see Argument 1’s analysis). This overhead resulted in a brief 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 fix of session hijacking resulted in the removal of the dangers of session hijacking to society, and in turn resulted in an increase in net happiness to the general public. That net positive result accumulates for every year that session hijacking is not the problem it was from 2004-2010 [17].

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 Firesheep GUI on top of Wireshark. By this logic, Butler was not only ethical in releasing Firesheep to the general public, but was 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 The Project: Ensure Correct Usage of the HTTPS Protocol

Butler described in his blog post accompanying Firesheep that the goal of Firesheep (his project) was to "demonstrate just how serious [HTTP session hijacking] is" [16]. In particular, Butler took issue with large websites using the HTTPS protocol incorrectly: "sites fail to protect you because after youve authenticated, you're issued a cookie that identifies you throughout your browsing session, but if you think about it [hijacking that cookie is] just as good as your username/password for 99% of the time" [17].

Butler also took issue with another misuse of HTTPS: "some sites support full encryption everywhere, but don't implement it properly by failing to set the 'Secure' flag on authentication cookies, negating most of the benefits and leaving users at risk" [17].

Thus, Butler's primary motivation before releasing Firesheep was to assure that websites claiming to use HTTPS used it correctly to protect their users [17].

4.4.3 An appropriate method

For Butler to correct websites' usage of the HTTPS protocol, he had a few choices of methods [27].

Alternative 1: Wait for the software vendor to fix the flaw

The first option Butler had in trying to correct usage of HTTPS is to tell people about the issue, and wait for websites to begin posting a fix [27]. However, security experts had been warning websites of the dangers of HTTP session hijacking for six years before the release of Firesheep [17].

Alternative 2: Expose the flaw

The other option Butler had to correct site usage of HTTPS was to expose the flaw [27]. The key in staying within White Hat boundaries here is "determining what a reasonable time period is to wait for a fix from the software company" [27]. Butler decided that six years was long enough [16], and that the appropriate method was to expose the flaw [17].

4.4.4 Substituted SE Code 3.05

Ensure an appropriate method (software designed to expose the flaws of site HTTP vulnerabilities) is used to correct the usage of the HTTPS protocol for user safety.

4.4.5 Conclusions drawn

The appropriate method to solve this problem was to release of software designed to expose HTTPS usage flaws [16], and thus accelerate the rate of response to HTTP session hijacking vulnerabilities [16].

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 [17]. That is to say, companies had been outright ignoring this problem for (at least six [17]) 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 time to 'raise the stakes' a little and force the hands of these non-compliant software companies [17]. If verbal signals for those six years turned out to not be an appropriate method for companies as (evidently) ignorant as this, then exposure of the gravity of the HTTP vulnerability was certainly the appropriate method to choose.

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 [22]: he had to raise the stakes a little and release Firesheep to the public [27]. Software engineers under this code have an ethical responsibility to ensure that an appropriate method is used [27]; SE Code 3.05 is yet another perspective on why Butler was ethical and ethically obligated to release Firesheep [22].

5 Conclusion

A common theme throughout this discussion has been Eric Butler's ethical responsibility to do something about the rampant HTTP session hijacking vulnerability among popular websites. It was, as Eric said, a problem that was talked about to death [16]. Yet, very few software companies were willing to spend the engineering hours fixing it until Butler trivialized the session hijacking process [17]. The Code of Ethics describes situations like this in sections 1.04, 1.08, 6.08 and 1.05 [22], 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 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 something morally sound according to the Code of Ethics. To paraphrase section 1.04 once again: "if you see some danger associated with some software, disclose the danger" [22].

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] "Web photos that reveal secrets, like where you live," August 2011. [Online]. Available: http://www.nytimes.com/2010/08/12/technology/personaltech/12basics.html?_r=0

Facebook geotags your photos

[4] "Angelhack san francisco," June 2013. [Online]. Available: http://angelhack.com/

A hackathon in San Francisco

[5] "Dictionary.com unabridged," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/actual

[6] "Dictionary.com unabridged," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/potential

[7] "Dictionary.com unabridged," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/disclose

[8] "Dictionary.com unabridged," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/volunteer

[9] "Dictionary.com unabridged," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/professional

[10] "Dictionary.com unabridged," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/skills

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

Facebook's first quarter results

[12] "Facebook revenue exceeds estimates on mobile advertising," May 2013. [Online]. Available: http://www.bloomberg.com/news/2013-05-01/facebook-revenue-exceeds-estimates-on-mobile-advertising.html

Facebook makes most of its money on ad revenue

[13] "The google geocoding api," June 2013. [Online]. Available: https://developers.google.com/maps/documentation/geocoding/

Google's Reverse Geocoding API

[14] "Online etymology dictionary," Jun 2013. [Online]. Available: http://dictionary.reference.com/browse/danger

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

Source code and documentation for Firesheep

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

Eric Butler's personal blog post detailing Firesheep.

[17] ——, "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

[18] ——, "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

[19] ——, "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

[20] 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.

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

Live query on Stack Overflow's current user base.

[22] 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'

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

[24] 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.

[25] J. R. N. Henry Campbell Black and J. M. Nolan-Haley, "good cause: Black's law dictionary." [Online]. Available: WestPub.Co.p.476.ISBN0-314-88536-6.

Definition of 'good cause'

[26] R. Hursthouse, "Virtue ethics." [Online]. Available: http://plato.stanford.edu/cgi-bin/encyclopedia/archinfo.cgi?entry=ethics-virtue

[27] S. Keating and D. Barrett, "Ethics in black, white and grey: University of advancing technology." [Online]. Available: http://www.uat.edu/academics/Ethics_in_Black_White_and_Grey.aspx

Describes options that White Hat ethical hackers have about security flaws

[28] 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

[29] ——, "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

[30] 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"

[31] 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

[32] 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

[33] 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

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

UK Article on the legality of Firesheep