SafeCrawler: Affective Visualization of Web Security

Oliver Stirling Schneider
Department of Computer Science
University of British Columbia
oschneid@cs.ubc.ca

ABSTRACT

Modern day web browsing exposes the user to a host of possible attacks. Although many automated defenses exist, sometimes we are unable to fully protect the user. In these scenarios, we encounter a fundamental tradeoff between security and user experience. In addition, there are many possible threats that require a human's eye to evaluate whether a risk even exists, such as phishing attempts. One common approach is to provide the user with a warning to give them the ability to make informed decisions. Unfortunately, warnings have a dismal track record in the security community, and the user is left more or less blind in these situations. Enter SafeCrawler, a new kind of warning system. SafeCrawler is an easy-to-use Chrome Extension that highlights unsafe elements in a web page. Unlike previous warning approaches, SafeCrawler considers the user's emotional state to be more persuasive. We hope that using affective response will make the user more cautious and willing to consider their browsing risks. In this paper, we describe the initial design and implementation of SafeCrawler, as well as preliminary feedback from users.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Design, Human Factors, Security

Keywords

affective computing, computer security, online security, visualization

1. INTRODUCTION

Computer security hosts a number of key tradeoffs. One of those tradeoffs is between security and user experience. As some threats cannot be reliably eliminated without compromising the user experience, tools often resort to warnings to provide the user with some protection without forcing them to relinquish control. Indeed, sometimes human input is necessary to identify a threat.

Unfortunately, the security community has established that warnings simply do not work [12,6,9]. Users either find them confusing, do not understand the underlying threat, or find that behaving securely is not worth the additional cost in the long run. Furthermore, expertise has little effect on how users act, so it is not enough to educate users [6, 14].

Affective computing, the consideration of affective and emotional state, has recently been making strides in a variety of scenarios. Affective state can influence decision making and make persuasive systems more successful [2]. In this way, I hope to encourage users to be cautious, which is the ultimate goal of warning systems. I propose to use affect to provide unique warning and awareness systems to the user.

I have developed a Chrome extension, SafeCrawler, that can highlight potentially unsafe HTML elements with this new type of warning. SafeCrawler uses visualizations designed to contain affective content to the user. In this way, I hope to provide a more thoughtful method of navigating the web, encouraging users to behave cautiously and not take unneeded risks.

In this paper, I will first review the related work on both online warning systems and relevant affective research. I will then describe the various visualizations used by SafeCrawler, and explain their rationale. Next, I will present the implementation details of SafeCrawler, including its extensible architecture, features, and limitations. After, I will summarize user feedback and design lessons learned from initial evaluations. Finally, I will conclude and present future directions for this work.

2. RELATED WORK

There has been a great deal of research on user browsing habits, particularly in scenarios where automated defenses are not available. Oftentimes, the onus (and blame) for safe online habits falls to users. Since the mid nineties, however, the security community has moved towards designing for end-users, rather than criticizing them [1, 15].

Cormac Herley even argues that sometimes unsafe browsing is the rational decision using utility theory [9]. Externalities, or indirect costs, often end up being more costly to the user than the estimated risk. For example, the extra time needed to verify that a website is not a phishing site adds up over the course of a year, while the direct financial cost

Symposium of CPSC 538W Final Projects (SOCFP) 2013, April 17, 2013, Vancouver, BC, Canada.

of ignoring such threats be very low on average.

Regardless, some problems must be tackled with human involvement. By making the human involvement less costly, or more trustworthy when it does appear, we stand to both protect users from possible threats and provide a fulsome web experience.

2.1 Online Security Warnings

Warning the user in an easy-to-notice way can yield benefits, but getting the user's attention is not so trivial. Users have long demonstrated "Banner Blindness" by ignoring visually salient information when looking for something specific [4]. Even animation does not help memory of banners [3]. So what are we to do?

Schechter et al. looked at users' ability to look at increasingly salient warnings for bank logins. They found that all users entered their password when the HTTPS indication was removed, 97% entered their password when the site-specific authorization images were missing, and a full 53% entered their password when a full warning page was presented [12]. This decreased to 36% when real accounts were used, a still surprisingly high number.

Egelman et al. discovered that active warnings (which interrupt the user) are more effective than passive warnings, but still achieve low results when used with an email phishing threat [6]. They based their designs on the Warning Sciences literature, particularly the Communication-Human Information Processing (C-HIP) model. This model identified failures of the warnings in two areas: low reading of warnings, and low understanding of warning content. Habituation was also observed, where users would ignore warnings they saw before. User expertise had no effect.

Sunshine et al. argue that warnings should be used as a last resort when we cannot guard against or eliminate the threat [14]. They found that in general, a variety of SSL warnings did not persuade users to act more securely. Many participants continued to view websites with expired certificates, unknown certificate authorities, or even a domain mismatch. Warnings similar to that currently implemented by Google Chrome (an interrupting full page warning). Again, expertise rarely mattered.

Shin and Lopes examined a colour visualization inspired from a traffic light to warn users about an SSLstripping attack [13]. SSLstripping is a form of man-in-the-middle attack where the attacker presents an unencrypted page to the user, intercepts it, and then makes the secure connection themselves. The visualization used green to represent encrypted elements, red for unencrypted elements that should be encrypted (such as login fields) or an invalid SSL certificate, and yellow for other unencrypted elements that aren't on a white list. Again, warnings were not effective.

Brustoloni and Villamarin-Salomon tackled the habituation problem through context-sensitive guidance, and morphic warnings (that are re-ordered whenever presented to the user) [5]. They found that the threat of someone auditing their security rationale improved security, but still 66% of their participants accepted unjustified risks. This threat of auditing is only really feasible in an organization, such as a company email client and account.

Finally, Fogg et al. used factor analysis of an online survey to find that the overall look and feel of a website contributes the most to its credibility [8]. Warning systems that leverage this fact might be more effective. Also, computers are social agents, and social agents can be more persuasive [7]. Casting the computer as an entity with agency might be an effective method for design of more effective warnings.

2.2 Affective Computing

However, none of these approaches directly considered affect in their design or analysis. Affective computing is the use or consideration of affect, ephemeral states of being such as emotion (anger, fear, anxiety) or physiological conditions (hunger, cold). A great deal of research has been carried out upon the psychology of affect, and it has been shown to have a number of effects on decision making.

When studying decision making, some of the most accurate theories come from regret theory [2]. Regret theory posits that users make decisions (sometimes suboptimal) to avoid regret or disappointment in the future. This process can be influenced by affective states. Positive affect has been shown to improve problem solving and memory of perceptual and motor skills [10]. Negative affect also plays a role. Individual who are fearful were more pessimistic about unknowns, and more prone to avoiding risks [11]. Conversely, angry individuals have been shown to be optimistic and risk-taking, similar to happy people [11]. Disgust has been shown to decrease buying and selling prices of stocks [2]. I suspect that using fear, disgust, or similar emotions might persuade users to behave in a more risk-adverse way.

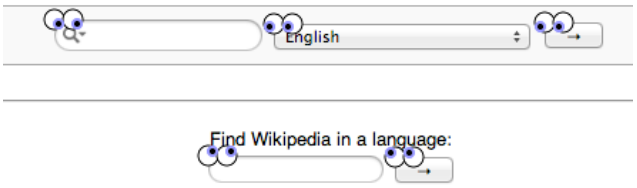
Furthermore, patterns that are established while in an emotional state tend to be carried forward [2]. This might be a key feature to counteract habituation and the ignoring of warnings. On top of this, emotional effects are enhanced when the outcome is unknown (the "surprise effect"). Given that users are often troubled by ambiguity with security messages, affect might be more effective when the explanations are highly technical.

Overall, there is a severe lack of affective considerations in the security community, and a great deal of promise for the role of emotion and affect in persuading users to behave securely or cautiously online.

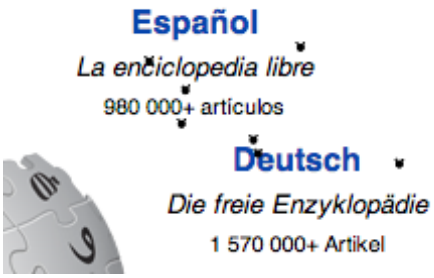
3. DESIGN

I have developed a Google Chrome extension, SafeCrawler, to explore the use of affect in online browsing security. Drawing inspiration from SSLight [13] and other SSL warnings [14], SafeCrawler highlights unsafe elements in a webpage. These elements could be identified as risky for a number of reasons. This includes unencrypted elements that should be encrypted, iframes that reference sites of ill repute, possible phishing attempts with links that have a domain mismatch, and many more threats. Each element could each be combined with a variety of visualizations designed to caution the user.

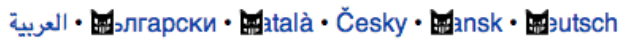
Each visualization is intended to evoke an affective response in the user. For example, using insects crawling over a link to evoke disgust (or possibly fear) with that link, which could make the user avoid clicking on it or at least motivated them to be cautious. As well, these markups are be super-imposed over a web page to provide a localized threat, and a sense of immediacy. This will affect the overall look and feel, possibly changing the user's trust in the site and again encouraging them to be more cautious. In this section, we will cover the design rationale for each of visualizations.



(a) Eyes



(b) MultiFlies



(c) Evil

Figure 1: SafeCrawler visualizations, captured from `www.wikipedia.org`.

3.1 Eyes

A pair of eyes was chosen to invoke the feeling of being watched. It also involves direct symbolism of prying eyes, suitable for communicating possible threats to personal information. This visualization was especially designed for unencrypted forms to caution users that anything they enter could be viewed in transit.

The eyes themselves are simple to not be too visually busy. The pupils follow the mouse to give the user more watched or intruded upon. Overall, the eyes are designed to be cartoony and a little amateur. Looking to Fogg's findings on website credibility [8], an amateur design might make the site feel more amateur (and thus less credible). Further, eyes might give the website a sense of agency, to make it more persuasive [7].

The eyes blink every two seconds to make them more salient, more engaging, and more amateur (by visually referencing the animated GIFs of the 1990s). Each pair of eyes on the screen has a random offset time to ensure that they do not blink in unison.

3.2 Flies

Flies were chosen as a simple representation of an insect or a swarm of insects. This includes the visual pun of buggy software, a suitable visualization for possibly infested third party sites. Flies are also meant to evoke disgust or fear in the user, encouraging them to avoid risky links. They were specifically animated with randomness to make them feel unpredictable and to enhance realism.

SafeCrawler provides two fly visualizations: "FlyDisplay" and "MultiFlies". FlyDisplay produces a single fly, while MultiFlies produces a between 3 and 5 flies, randomly chosen. In both cases, the flies are randomly distributed along the width and height of the HTML element. Flies also oscillate according to a random path, which is offset between flies to produce a swarm effect.

3.3 Evil

The "Evil" visualization is intended to be a more menacing figure that directly represents a malicious adversary. This visualization is given angry eyes and angular, horn-like ears. As well, figure's grin implies some benefit at the user's expense. The mouth moves up and down to simulate laughing or eating, either of which could be more menacing to the user.

3.4 Future visualizations

There are many visualizations that I could not implement due to time constraints, or that came to mind after preliminary feedback. There were originally plans for similar insect-like visualizations, such as ants or grubs crawling over the elements. Another original plan was to provide a colour overlay on elements, for example, desaturating them or providing a dusty layer for elements that have not been updated in a long time. These elements could be produced with the benefit of Perlin noise, but simpler approaches were used for the initial development.

The eyes are relatively benign, and the addition of more realism or eyebrows to symbolize anger or malicious intent might be valuable. That, or a more spy-like figure to imply prying eyes might be more appropriate. These are left to future work, as this space is virtually limitless.

4. IMPLEMENTATION

SafeCrawler is implemented as a Google Chrome extension with JavaScript, HTML5, and CSS. The code runs in a content script, which injects JavaScript code into any loaded webpage. Settings are stored in the extension's local storage, and accessed through message passing from the content scripts. A background script listens for content script messages, and passes them onto the content script via Chrome's message passing procedure. Content scripts run when the web page is idle.

Because the project is exploratory, the user is allowed to mix-and-match which HTML elements on the page are paired with which visualizations. This is decided through the extension's popup menu (Figure 2), which presents the available options to the user. The user is able to select a visualization for each of a variety of *targets*, sets of elements found in the web page's HTML source. Targets include:

AllLinks returns a list of every link element on the page.

AllInputs returns a list of every `input`, `select`, and `button` elements.

AllTextFields returns a list of every `input` element with the `text` type.

RandomLinks returns a list of approximately 30% of the links on the page, randomly chosen each time.

ArbitraryLinks returns a list of approximately 20% of the links on the page, randomly chosen but consistent across viewings. This is achieved by computing the sum of each character value in the link's string representation, and then including that link if the sum modulo 100 is less than 20.

UnencryptedInputs returns the AllInputs list, but only if the page is not encrypted (accessed with `https`).

As the design is improved, defined semantics could associate each of these targets with one or more permanent visualizations.

SafeCrawler is designed to be extensible. Targets are defined in `targets.js`, and visualizations are defined in `visualizations.js`. The options panel is dynamically generated, and each target and visualization follows a basic interface: targets have a `list()` method that returns their list, and each visualization has an `initialize()` method and a `update()` method. SafeCrawler is thus implemented in an object-oriented fashion. Upon page load, each target's list is computed, and then the appropriate visualization for that target is initialized. Afterwards, each visualization is updated in a main timer loop (currently operating at 20 fps). More work could easily establish plugins that define additional visualizations.

Visualizations can define their own method of display, but the primary method is to use a globally-defined canvas overlay. This canvas element covers the entire screen, and can be drawn upon by any visualization (which ends up being fairly efficient). All mouse events for the canvas are disabled, which restricts visualizations from interfering with the function of a web page.

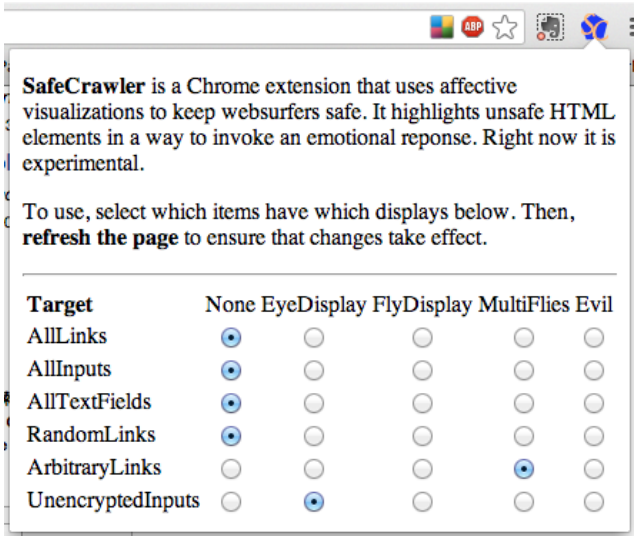


Figure 2: SafeCrawler popup user interface and options.

4.1 Limitations

SafeCrawler is experimental software, and not yet ready for deployment. There are several limitations that must be considered. First, web pages that dynamically expand (such as Facebook or Twitter's news feed) do not update to have visualizations on the added content. Second, the Evil visualization requires defined PNG images, and must connect to them from the content script, which runs in the web page's namespace. Because connecting to these images is not secure, the Evil visualization is disabled when viewing pages loaded with SSL/TLS.

5. USER FEEDBACK

Although formal validation or investigation are unreasonable at this stage, I sought informal and brief feedback from 6 participants about the first iteration of SafeCrawler. Reception was generally positive. Both the Eyes and MultiFlies visualizations were dubbed "creepy", suggesting that both achieved an affective response. Furthermore, I observed participants investigating into whether the webpage was secure: one participant double-checked the address bar to ensure that `www.facebook.com` was connected through `https` after seeing eyes on the input elements, and investigated into whether links on `www.wikipedia.org` that were covered with flies were accurate or not. This suggests that these visualizations might be promising to promote caution in users. Finally, the movement of both the eyes and the flies were well received, and one participant commented that he liked the Evil figure.

However, there were some lessons that should inform future design of SafeCrawler. First, some visualizations were deemed too difficult to see, or not salient enough - particularly the Evil figure, but also the flies (both MultiFlies and FlyDisplay). More use of colour might be important to improve the saliency, as long as we can avoid banner blindness [4]. Some participants thought that the visualizations were too busy where there were a lot. Future visualizations might best be normalized to adapt for how many are on the screen. Also, a visual saliency map of the page might help guide the location of the visualizations.

Second, the semantic meaning of each visualization was important. Participants were unsure what the eye visualization meant - was the server seeing the inputted data, or a third party? If the former, then that is to be expected; if the latter, then it is a possible threat. Part of this confusion manifested itself in the eye's appearance, as they were deemed to be not very menacing by some participants (although others described them directly as "menacing"). Participants did suggest that they would find these visualizations useful if they knew the semantics.

Finally, trust in the extension is an important consideration. Participants mentioned that the visualizations would be helpful if they depicted a change in a site they frequent, and suggested that they would heed the warning in that situation (suspecting an attack). However, if the extension appeared to act at random, trust in the system would rapidly deteriorate and no warnings would be heeded. As well, participants suggested that a white list might be useful for disabling sites that the user trusted.

Ultimately, affective visualizations show promise. Affective response looks to be likely with the chosen visualizations of eyes and a swarm of flies. However, several design

iterations, including a planned semantic link between visualizations and their meaning, need to be done before we have a truly usable and useful system.

6. CONCLUSION

In this paper, I have given rationale for including affect in the planning of online browser warnings. To that end we have built SafeCrawler, an extensible Chrome extension designed to explore this space, and implemented several visualizations. User feedback suggests that affect can be communicated and invoked through such visualizations.

However, there remains much work to be done. A wider variety of visualizations would be valuable to investigate. Once more are designed, an interpretivist inquiry (*e.g.*, grounded theory of interviews and observations) into how users react to the visualizations could give us a deep understanding of how these mechanisms work. Upon further design, we could test these visualizations against a control (no warning) and the state-of-the art (existing warnings in browsers) to examine how it influences behaviour in a larger user study. Role-playing has been suggested to be valid in these situations [12], and so a role-playing exercise might suffice. Longer-term evaluation can happen later.

I am excited to see the results of these visualizations. The visceral feeling had me personally turning it off before logging into websites, despite the fact that I knew every line of source code. Though this promising approach needs more investigation, I look forward to see where these visualizations might be used in the future.

7. ACKNOWLEDGMENTS

A big thank you to Dr. Bill Aiello for his feedback, guidance, and enthusiasm during the course! I am also grateful for the time and effort of all my informal participants.

8. REFERENCES

[1] A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, Dec. 1999.

[2] D. Bandyopadhyay, V. S. C. Pammi, and N. Srinivasan. Role of affect in decision making. *Progress in brain research*, 202:37–53, Jan. 2013.

[3] M. E. Bayles. Designing online banner advertisements. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*, page 363, New York, New York, USA, Apr. 2002. ACM Press.

[4] J. P. Benway. Banner Blindness: The Irony of Attention Grabbing on the World Wide Web. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 42(5):463–467, Oct. 1998.

[5] J. C. Brustoloni and R. Villamarín-Salomón. Improving security decisions with polymorphic and audited dialogs. In *Proceedings of the 3rd symposium on Usable privacy and security - SOUPS '07*, page 76, New York, New York, USA, July 2007. ACM Press.

[6] S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, pages 1065–1074, New York, New York, USA, Apr. 2008. ACM Press.

[7] B. J. Fogg. Persuasive technology. *Ubiquity*, 2002(December):2, Dec. 2002.

[8] B. J. Fogg, J. Marshall, O. Laraki, A. Osipovich, C. Varma, N. Fang, J. Paul, A. Rangnekar, J. Shon, P. Swani, and M. Treinen. What makes web sites credible? A report on a large quantitative study. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '01*, pages 61–68, New York, New York, USA, Mar. 2001. ACM Press.

[9] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop - NSPW '09*, pages 133–144, New York, New York, USA, Sept. 2009. ACM Press.

[10] A. M. Isen and B. Means. The Influence of Positive Affect on Decision-Making Strategy. Jan. 2011.

[11] J. S. Lerner and D. Keltner. Fear, anger, and risk. *Journal of personality and social psychology*, 81(1):146–59, July 2001.

[12] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators: an evaluation of website authentication and the effect of role playing on usability studies. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 51–65. IEEE, May 2007.

[13] D. Shin and R. Lopes. An empirical study of visual security cues to prevent the SSLstripping attack. In *Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC '11*, pages 287–296, New York, New York, USA, Dec. 2011. ACM Press.

[14] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: an empirical study of SSL warning effectiveness. In *Proceedings of the 18th USENIX Security Symposium*, pages 399–432, 2009.

[15] M. E. Zurko and R. T. Simon. User-centered security. In *Proceedings of the 1996 workshop on New security paradigms - NSPW '96*, pages 27–33, New York, New York, USA, Sept. 1996. ACM Press.