Poster: Preventing SSLstripping Attack using Visual Security Cues

Rodrigo Lopes and Dongwan Shin
Computer Science & Engineering Department
New Mexico Tech
Socorro, New Mexico, USA
{rodrigo,doshin}@nmt.edu

1. INTRODUCTION

The Secure Socket Layer (SSL) protocol has been the most widely used security mechanism enabling safe web browsing. A new attack, called SSLstripping, reported by Moxie Malinspike at the Blackhat conference in 2009 [2], effectively defeats the SSL security by exploiting either users' browsing habits or websites' SSL policy, rather than a technological flaw in the protocol. For the former, most users do not write in the address bar the full address of a website that they want to visit securely, instead relying on their browser and the website to redirect them to a proper secure location. For the latter, many websites do not support SSL by default, only having login forms use a secure connection. As a type of man-in-the-middle (MITM) attack, the SSLstripping attack has the potential to affect tens of millions of online users that login to those websites protected by SSL. Facebook.com is one of the vulnerable websites.

Two solutions have been proposed that could be used to address the SSLstripping attack. The first one, ForceHTTPS [1] makes the websites notify a user's web browser that they require a secure connection to operate, and therefore the browser will always establish a secure connection with those websites that required so. The problem is that many websites do not require HTTPS, and if the attack is launched before the website is first contacted, the browser will never get the notification. Another solution, HProxy [3] relies on the browser's history information to compare the current and past security mechanisms used by a website already visited. Once again, this solution will not work if the attack is deployed before a browsing history is established or if the history does not exist.

We present a novel approach to addressing the SSLstripping attack through the use of visually augmented security. Motivated by the design of typical traffic lights, we introduce a set of visual cues aimed at thwarting the attack. The visual cues can be used to boost the user's trust against her browser when sensitive credentials need to be entered and submitted to websites for the purpose of authentication.

Our contributions are as follows: we propose visual cue based solutions that help address the SSLstriping attack; we propose a better solution to inform users about websites that request sensitive login credentials through an insecure channel by design. Users can then be constantly aware of websites with secure and insecure login, and make informed decisions on how they choose and use their credentials; and we also conduct a user study to explore whether our approach is more effective and promising than the existing pop-up method.

2. APPROACH

We developed two visual cue based solutions to both prevent a successful SSLstripping attack and help users identify web pages that are insecure by default. The first one, called the security status light (SSLight), is based on a three color design resembling a traffic light, as shown in Figure 1, while the second uses a blinking red background in the login input boxes as an alternative approach to inform users of their security status when they need to submit sensitive credentials to a website. The traffic light metaphor was adopted for its simple and intuitive design for most users, who do not have the technical background or depth to determine whether the web page that they are visiting is secure or not by looking at its complex source code.

In order to evaluate the security status of a web page, we developed an algorithm that compares the address of the web page with the login form's action data in the web page. This algorithm was then implemented as a browser extension on Google Chrome web browser using Javascript, HTML and CSS.

More specifically, the first step is to identify if the page loaded by the browser is already being accessed over SSL. If this is the case, we just need to verify that the action on the login form belongs to the same domain that is already secure, a situation that is true if the action is a relative path. This means the form will submit the login request to an address in the scope of the current secure connection. Hence, the SSLstripping is not possible and we return a positive evaluation (**Green light**). If the current page is not on a secure connection and the form action URL is an insecure address, we immediately return a negative evaluation (**Red light**), which means it is unsafe to submit the login request. In the scenario where the current page is not secure, but the form action is under the HTTPS protocol, we proceed with another round of analysis. Further analysis will first assess the certificate of the secure location referenced in the form action. If this is a self-signed or expired certificate, we will return a negative evaluation. Next, if the certificate proves not to be invalid, we compare the domain in the form action with the domain of the loaded page. If these two domains match, we return a positive evaluation, otherwise, we check whether the domain of the login form action is in a list of trusted login entities. If we cannot white-list the URL, we will issue an uncertain assessment (**Yellow light**) and delegate to the user, showing them the domain where they will be submitting their form. In an attack situation, a warning would appear in the webpage that previously did not raise any issues.

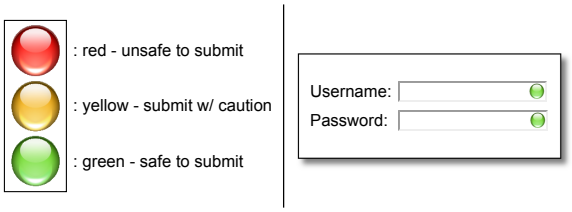

Figure 1: Our approach: security status light

3. EXPERIMENT

We conducted a user study[1] not only to test the effectiveness and efficiency of our solutions, but also to compare it against an existing solution, which is the classic pop-up window introduced by Netscape and currently present only as an opt-in feature in Firefox and Internet Explorer. We also wanted to test how all of these solutions work against the absence of any user warning. We recruited 100 test subjects, 25 of which were assigned to each of four test groups. Each group was exposed to the SSLstripping attack; three groups were given a specific warning (SSLight, blinking red background, and pop-up window), and the fourth group was not warned at all. To avoid the framing effect, we did not want the users to be aware that we are testing their login behavior and their reaction to a security warning. Also, we wanted the users to be exposed to the warning as an abnormality or exceptional condition. To achieve these goals, we asked users to perform different sets of tasks on their Facebook accounts. This required them to use their actual credentials and accomplish a set of tasks that would realistically simulate their behavior. The hypotheses we wanted to test were on (1) the user awareness of an insecure form submission, (2) the efficacy of the SSLstripping attack, (3) the fact that users would ignore the pop-up window and (4) that our proposed visual cue solutions would be more effective than the pop-up window. Finally we wanted to verify that (5) the two different visual cue solutions proposed would have similar results.

The users were asked to act as if they were using their own machines, in that all decisions they made should be the same as if they were being made on their own private computers. Security was never explicitly mentioned. Although we initially thought the "make all decisions as if this was your machine" statement could bring focus into security issues, the results showed us that this was not the case.

4. SOME USER STUDY RESULTS

Each user was asked to take an exit survey, from which we are able to infer that most users are aware of the danger of submitting data over an HTTP connection. However, from the results of our hypothesis testing, we find that without any added security mechanisms the SSLstripping attack is highly effective against even technologically sophisticated users. We demonstrate, furthermore, that the pop-up window is as effective as having no warning at all. Conversely, the results obtained for the proposed solutions (SSLight and red blinking background) proved more efficient than the pop-up window. The SSLight solution led 9 out of 25 users to not submit their login credentials, while the red blinking background helped 17 out of 25 users decide not to submit their credentials.

[1]This user study was approved by our university's IRB in Fall 2010 and conducted for one week at NM Tech campus.

5. DISCUSSION AND FUTURE WORK

Some of the interesting results we found are the discrepancy between the personal opinion users have about a warning method and the warning efficacy. There is no significant difference between the ratings attributed by the test subjects to each method. Even when the subjects are only asked to select their favorite warning method, we can find no significant difference on the fraction of the subjects that selected each of the alternatives. However, there is a statistically significant difference between the effectiveness of the different methods. This is a strong indicator that designing a system based on user opinion may not be the best approach. Studies where users' normal interactions are surveyed seem to be the best source of information that can successfully guide the development of usable and useful security.

Our empirical study clearly shows that the proposed solutions are more effective and efficient in preventing the SSLstripping attack than the classic pop-up window. However, our approach is by no means complete. For our immediate future work, we will investigate how to improve the effectiveness of the SSLight solution. Specifically, the proposed SSLight is based on one factor only, which is color. This makes the solution ineffective to the color-blind. We will study how to address this by adding additional factors, such as symbol or text. Another future work is related to our experimental design. Our study used a sample consisting mainly of higher education students, a demographic that does not represent the average user accurately. Gathering a more representative sample poses a bigger challenge that we are trying to address. Finally, there is also the fact that the results obtained by our approaches could stem directly from its novelty alone. To circumvent this, we are working on another round of data collection that will require longer and more frequent interactions to exclude the novelty as a factor for the good results, thereby studying the effect of user habituation.

ACKNOWLEDGEMENT

This work was partially supported at the Secure Computing Laboratory at New Mexico Tech by the grant from the National Science Foundation (NSF-IIS-0916875).

6. REFERENCES

[1] C. Jackson and A. Barth. Forcehttps: protecting high-security web sites from network attacks. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 525–534, New York, NY, USA, 2008. ACM.

[2] M. Marlinspike. New tricks for defeating ssl in practice, 2009. http://www.thoughtcrime.org/software/sslstrip/.

[3] N. Nikiforakis, Y. Younan, and W. Joosen. Hproxy: Client-side detection of ssl stripping attacks. In C. Kreibich and M. Jahnke, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, volume 6201 of *Lecture Notes in Computer Science*, pages 200–218. Springer Berlin / Heidelberg, 2010.