

Secure Visualization of Authentication Information: A Case Study

Sean Cannella
Brown University

Daniel J. Polivy*
Microsoft

Michael Shin*
Goldman Sachs

Christian Straub*
Oracle

Roberto Tamassia†
Brown University

Abstract

The open nature of the Web makes it possible to create spoofed Web pages which, to the casual observer, are indistinguishable from authentic pages. The defense against Web spoofing attacks is a challenging problem since most users are unwilling (or unable) to follow complex interactive authentication procedures (e.g., inspect a Web server certificate). In this paper, we present a visual scheme that protects users against Web spoofing attacks while requiring minimal interaction.

1. Introduction

Ensuring the integrity of Web content has become a fundamental issue in information security. Even with a cryptographically secure system, we need to present the relevant authentication information to the user in a way that is clear, concise, and useful.

In the standard HTTP protocol, there are many opportunities for an attacker to intercept and manipulate Web pages. Such content-spoofing attacks are relatively easy to perform [2, 4] and present a serious security problem. The user interface components of the Web browser (including the toolbars and the status bar) are also vulnerable to spoofing attacks that leverage standard scripting languages such as JavaScript [5, 10]. A common remedy to this problem is to simply disable client-side scripting and other basic browser features (such as the ability to launch new windows.) However, this approach penalizes legitimate uses of scripting technology, such as the validation of forms.

The transport layer security protocol (TLS) and its predecessor SSL are widely used today for protecting the transmission of sensitive information over the Internet. However, their security is undermined by the simplicity of the inter-

face provided by most browsers. The closed-lock icon indicates that an encrypted channel has been established, but can offer a false sense of security if the subject of the SSL certificate is not scrutinized. Moreover, the icon and certificate inspection window can be easily forged [2, 4, 10].

In this paper, we present a visual scheme that protects users against Web spoofing attacks while requiring minimal interaction. As a case study, we use *prooflets*, a distributed architecture for the end-to-end integrity of Web content [8]. Using prooflets, we develop a user interface for presenting authentication information in a tamper-resistant fashion, without sacrificing the user experience. We also report on the results of a preliminary user study comparing our prooflets approach with a previous approach [10].

2. Previous Approaches

Ye and Smith [10] propose a technique called *synchronized random dynamic (SRD) boundaries*. The main idea is to use a browser window whose border alternates between two colors, blinking in synchrony with a trusted window. User studies indicate that the SRD boundaries approach is an effective passive approach that does not require user interaction. However, this approach does not allow for modular verification of portions of a Web page since the unit of authentication is an entire page. Also, the modifications of the browser required to implement SRD boundaries go beyond plugins and are thus suited for open-source browsers.

GeoTrust's *True Site* product [7] places a "smart icon" on authenticated Web pages, which is dynamically generated when the page is requested. The icon includes a timestamp and a watermark to prevent spoofing. Clicking on the icon allows the user to view detailed authentication information. Since the typical user experience will generally involve a quick glance to check for the icon's presence, this technique is also a passive scheme. However, a savvy attacker could still try to forge the smart icon (with its associated timestamp and watermark) and the detailed authentication information.

*The work by this author was performed while he was at Brown University.

†Contact author, rt@cs.brown.edu.

3. Our Approach

The principle of least effort [6] dictates that the casual user does not need to be presented with the full details of authentication information at all times. While a sizable amount of information is necessary to counter certain attacks, the tradeoff between security and ease of use must be weighed carefully.

In a *passive* authentication visualization approach, significant security events are displayed to the user without interaction. The closed-lock icon indicating an SSL-encrypted session, the SRD boundaries and the True Site smart icon can be viewed as examples of the passive approach. The main problem with the passive approach is that the user may overlook or ignore the visual cues.

In order to provide a stronger level of security against forgery attacks, we introduce a controlled model of user interaction where the system provides to the user a mechanism for expressing interest in the authentication process at various levels. Our approach does not guarantee full protection against spoofing attacks. However, it increases the likelihood that the user will notice such an attack. Although we require some level of user interaction, our model strives to keep this interaction as small as possible.

4. Prooflets: A Case Study

Prooflets [8] are a scalable architecture for authenticating Web content based on authenticated dictionaries (see, e.g., [9]). A *prooflet* tag is a specific HTML span tag delimiting a portion of an HTML (or XML) document that can be authenticated. Prooflet-aware browsers recognize such tags and query the authenticated dictionary to determine the authenticity of the data enclosed by prooflet tags. In an abstract sense, a prooflet can be considered as a remote procedure call (RPC) to the authenticated dictionary system, where the content enclosed within the prooflet tags can be viewed as the parameter of the RPC.

Prooflets provide authentication by extending, rather than supplanting, the current Web content delivery infrastructure. Our prototype implementation of the prooflets architecture employs a plugin *toolband* for Microsoft Internet Explorer, called the *prooflets toolband*, that includes action buttons and a status indicator.

For a casual and passive visualization of prooflets, we can specify different styles for valid content (e.g., green background) and invalid content (e.g., red background). However, this scheme may cause confusion (e.g., does green always associate with valid?) and lends itself to spoofing attacks. For increased protection, we employ a minimal level of user interaction (usually a single click), as discussed below.

In order to distinguish reliably between authentic visual elements (created by the prooflets user interface) and spoofed visual elements (created by the attacker), we employ a visual portfolio recognition system, based on a visual hashing scheme [3]. When prooflets are installed, a user randomly selects four images out of a large collection of images. These four images form the user's *visual portfolio*, which will be displayed on any visual element purporting to originate from the prooflets user interface, including the prooflets toolband. Assuming the user has undergone an initial training period, visual recognition can be very successful.

In order to validate a specific prooflet, the user can just hover the mouse over its content. The status indicator on the prooflets toolband will display a check-mark if the content is validated, a cross if the content is not validated, and a question mark if the content is not a prooflet. Recall that the visual portfolio is also displayed on the toolband to counter spoofing attacks that try to emulate the toolband.

We also provide the *prooflets view* of a Web page, a mechanism that allows the user to visually inspect all the prooflets on the page. Upon request by the user, all content that is not associated with a prooflet will slowly fade away from the screen (though not completely) with a transition effect [1]. What the user is left with then is purely prooflet content. Since the rest of the content is still slightly visible, the user can see the context of each prooflet and its relative position on the page.

At any point, the user can take a snapshot of all the prooflets in a document and view complete authentication information about an individual prooflet. Both the snapshot window and the individual prooflet windows are authenticated using the visual portfolio.

Finally, we support the ability to visually connect prooflets to each other. A typical use for this feature is the validation of key-value associations. For example, in a Web page listing stock quotes, we would like to validate the associations between each stock symbol (key) and its price (value). An attacker may try to rearrange the position of the symbols and prices on the page to induce the user to make an incorrect key-value association. To counter this attack, when the user enters the prooflets view of the document, keys and their associated values are shown using the same background color. The user can then immediately connect like colors and make inferences based on the relative locations of the prooflets.

5. Preliminary User Studies

We have conducted two preliminary user studies, the first on effectiveness and the second on usability, to compare our prooflets approach with the SRD boundaries approach [10]. For these studies, we have used a simplified



Figure 1. Simplified prooflets interface that uses a personal identifier.

version of the prooflets interface that uses a personal identifier (a.k.a. “magic key” or “MAC phrase”) instead of a visual portfolio, as shown in Figure 1.

Our user group for the effectiveness study consisted of eleven people: eight doctoral students, two masters students, and one undergraduate student. About half of these users had experience in user interface design. Each user was exposed to different visual scenarios. After showing each scenario, the screen was blanked momentarily so that visual cues would not be given away while the visual environment was modified for the next scenario. The subjects were exposed to the following three scenarios: valid unspoofed, invalid unspoofed, and valid spoofed. All the users were able to recognize a valid unspoofed SRD window. In addition, all the users were able to detect a spoofed SRD window with incorrect timing. However, when presented with a spoofed SRD window with correct timing, ten of the users still thought that the page was authentic. Only one participant was able to detect that the page was spoofed. For prooflets, all the users were able to detect the valid and invalid prooflets, as well as detect the spoofed version of the prooflets user interface.

The usability study was conducted by three doctoral students who did not participate in the effectiveness study. When testing with the SRD-enabled browser, they all found the flashing windows to be distracting for casual Web surfing and eventually stopped observing the reference window. After several minutes, one user depended solely on the flashing main window for authentication, without even taking note of the reference window. All participants noted that they began to ignore the pace of the flashing windows

as time progressed.

For prooflets, all the users were willing to do an interactive check of the validity of the prooflets that were of high interest to them by hovering the mouse over the content and looking at the status indicator on the toolband. Two of the users noticed the correctness of the personal identifier all the time (after an interactive check), while the third user noticed the identifier most of the time.

Acknowledgments

We would like to thank Robert Cohen, Michael Goodrich, Seth Proctor, Nikos Triandopoulos and Danfeng Yao for useful discussions. Our work benefited from a previous project by David Emory, who developed an earlier end-to-end Web integrity system based on authenticated dictionaries. This work was supported in part by the National Science Foundation under grants EIA-0303577, DUE-0231202 and IIS-0324846, and by a research gift from Sun Microsystems.

References

- [1] B.-W. Chang and D. Ungar. Animation: from cartoons to the user interface. In *Proc. ACM Symp. on User Interface Software and Technology*, pages 45–55. ACM Press, 1993.
- [2] F. De Paoli, A. L. Dos Santos, and R. A. Kemmerer. Vulnerability of “secure” Web browsers. In *Proc. National Information Systems Security Conf.*, pages 476–487, 1997.
- [3] R. Dhamija and A. Perrig. Déjà Vu: A user study — using images for authentication. In *Proc. USENIX Security Symp.*, 2000.
- [4] E. Felten, D. Balfanz, D. Dean, and D. Wallach. Web spoofing: An Internet con game. In *Proc. National Information Systems Security Conf.*, 1996.
- [5] A. Fox, S. D. Gribble, Y. Chawathe, A. S. Polito, A. Huang, B. Ling, and E. A. Brewer. Orthogonal extensions to the WWW user interface using client-side technologies. In *Proc. ACM Symp. on User Interface Software and Technology*, pages 83–84. ACM Press, 1997.
- [6] A. Morse. Some principles for the effective display of data. In *Proc. Conf. on Computer Graphics and Interactive Techniques*, pages 94–101, 1979.
- [7] J. Rosenberg. True Site. White paper, GeoTrust, Inc., 2001. http://www.geotrust.com/resources/white_papers/TrueSiteWP.pdf.
- [8] M. Shin, C. Straub, R. Tamassia, and D. J. Polivy. Authenticating Web content with prooflets. Technical report, Center for Geometric Computing, Brown University, 2002. <http://www.cs.brown.edu/cgc/stms/papers/prooflets.pdf>.
- [9] R. Tamassia. Authenticated data structures. In *Proc. European Symp. on Algorithms*, volume 2832 of *Lecture Notes in Computer Science*, pages 2–5. Springer-Verlag, 2003.
- [10] Z. E. Ye and S. Smith. Trusted paths for browsers. In *Proc. USENIX Security Symp.*, pages 263–279, 2002.