

CsFire: browser-enforced mitigation against CSRF

Lieven Desmet and Philippe De Ryck

IBBT-DistriNet Research Group, Katholieke Universiteit Leuven, Belgium

Cross-Site Request Forgery (CSRF) is a web application attack vector that can be leveraged by an attacker to force an unwitting user's browser to perform actions on a third party website, possibly reusing all cached authentication credentials of that user. In 2007, CSRF was listed as one of the most serious web application vulnerability in the OWASP Top Ten. In 2008, Zeller and Felten documented a number of serious CSRF vulnerabilities in high-profile websites, among which was a vulnerability in the home banking website of ING Direct.

One of the root causes of CSRF is the abuse of cached credentials in cross-domain requests. A website can easily trigger new requests to web applications in a different trust domain without any user intervention. This results in the browser sending out cross-domain requests, while implicitly using credentials cached in the browser (such as cookies, SSL certificates or login/password pairs). From a server point of view, these implicitly authenticated requests are legitimate and are requested on behalf of the user. The user, however, is not aware that he sent out those requests, nor that he approved them.

Currently, a whole range of techniques exist to mitigate CSRF, either by protecting the server application or by protecting the end-user (e.g. via a browser extension or a client-side proxy). Unfortunately, the server-side protection mechanisms are not yet widely adopted. On the other side, most of the client-side solutions provide only limited protection or cannot deal with complex web 2.0 applications, which use techniques such as AJAX, mashups or single sign-on (SSO). As a result, even the most cautious web user is unable to appropriately protect himself against CSRF, without compromising heavily on usability. Therefore, it is necessary to construct more robust client-side protection techniques against CSRF, capable of dealing with current and next-generation web applications.

In this talk, we will presents three interesting results of our research: (1) an extensive, real-world traffic analysis to gain more insights in cross-domain web interactions, (2) requirements for client-side mitigation against CSRF and an analysis of existing browser extensions and (3) CsFire, our newly developed FireFox extension to mitigate CSRF. More details can this research be found in [1].

Traffic Analysis

To identify the nature of nowadays web interactions, and to be able to find an appropriate balance between usability and security for cross-domain requests, we conducted an extensive, real-world traffic analysis.

We have collected real-life traffic from about 15 volunteers over a 3 months time span, resulting in a total of 750334 requests. The analysis of this traffic has revealed a number of interesting properties that can be used to determine a secure cross-domain policy. Among others, the user interaction, the implicit credentials and the parameters were recorded.

	GET	POST	Other	Total
Cross-domain requests (strict SOP)	320529	1793	0	322322
Cross-domain requests (relaxed SOP)	242084	1503	0	243587
All requests	722298	28025	11	750334

In addition, we extended the experiment to 50 grad students over a 10 weeks period, resulting in over 4M requests. The new data set confirms the identified patterns, and the full analysis of will be presented at the conference in June.

Requirements for client-side mitigation against CSRF

Next, based on an extensive traffic analysis, we propose the following requirements for a client-side CSRF solution in a contemporary web 2.0 context.

R1. The client-side protection should not depend on user input. A substantial fraction of web requests in an average browsing session is cross-domain. It is infeasible for the user to validate requests. Furthermore, users cannot be expected to know which third parties a web application needs to function correctly. Therefore, a transparent operation is essential.

R2. The protection mechanism should be usable in a web 2.0 context, without noticeable service degradation. The solution should support the dynamic interaction behavior of today's applications (i.e., 'mashups'), and embrace current and future web technologies.

R3. Secure by default. The solution should have minimal false negatives using its default configuration.

CsFire: a firefox extension against CSRF

Finally, we define CsFire, an autonomous client-side protection policy, which is independent of user-input or server-provided information. This policy determines which cross-domain traffic is considered harmful and enforces the appropriate action (e.g. blocking the request or removing the implicit authentication credentials from the request). To do so, the policy records user interactions, and combines information about the request originator as well as access to the raw HTTP requests.

CsFire is available as an extension for the Firefox browser via the Mozilla Add-on repository (AMO) [2]. CsFire is extensively evaluated via a CSRF testbed consisting of over 50 CSRF scenarios, and an in-field evaluation by 50 test users over a 3 months time span. The proposed policy works well without noticeable functional degradation, except for a limited set of companies spanning multiple top-level domains (e.g. Google and Yahoo!).

In addition, to counter the small set of situations where no automatic approach can differentiate between legitimate requests and CSRF attacks, CsFire can be extended by server-specific refinements.

References

[1] Philippe De Ryck, Lieven Desmet, Thomas Heyman, Frank Piessens, and Wouter Joosen, CsFire: Transparent Client-Side Mitigation of Malicious Cross-Domain Requests, In: Proceedings of 2nd Symposium on Engineering Secure Software and Systems (ESSoS 2010), LNCS 5965, pp. 18-34, 2010.

[2] <https://addons.mozilla.org/firefox/addon/58189>

Biography

Lieven Desmet is Research Manager on Secure Software at the Katholieke Universiteit Leuven (Belgium), where he coaches junior researchers and leads a research team on web application security. His main interests are in software verification and security of middleware and web-enabled technologies. Lieven is actively engaged in OWASP and is board member of the OWASP Chapter Belgium.

Philippe De Ryck is a PhD researcher of the DistriNet Research Group at the Katholieke Universiteit Leuven (Belgium). Philippe graduated last year with his MSc thesis entitled "Non-repudiation Middleware for Web-based Architectures". His research interests are in the area of web application security, focused on cross-domain interaction (CSRF attacks and mitigation, Mashup security, ...).