Cloud Service Privacy in a Pervasive Monitoring Landscape

John Mattsson, Stefan Håkansson, Ericsson Research

Abstract. There is an ongoing transformation towards third-party cloud services for storing and managing information. Trust in global cloud services is fundamental for the further development of the Internet economy. The market potential for enterprise and government cloud services and web applications are held back by privacy and security concerns. For information security and privacy reasons, W3C should work on general mechanisms where the browser encrypts and decrypts information in such a way that the web application cannot access cleartext data in any way. Cloud services and web applications should be fully trustworthy also in a pervasive monitoring landscape.

1. Introduction

There is an ongoing transformation towards third-party cloud services and web applications for storing and managing information, both for individual and enterprise purposes. In many ways this is very positive, as cloud services are flexible, cost efficient, and developer-friendly, thus enabling fast deployment of new services and giving end users access to data everywhere. Cloud services are therefore essential for the future of the Web and the Web of Things.

The drawback is that large amount of information gets aggregated in global data centers, making them attractive targets. From the recent mass surveillance revelations, we know that both data centers and fiber communication between such centers have been breached. The IAB/W3C STRINT (Strengthening the Internet Against Pervasive Monitoring) workshop [1] discussed the pervasive monitoring threats [2] and potential mitigations for different applications such as cloud services [3]. IETF has stated that pervasive monitoring is an attack on the Internet and that it should be mitigated [4].

To maintain the benefits of third-party cloud services and web applications while still providing information security and end user privacy, end-to-end encryption is needed. To enable storage of encrypted and integrity protected data, protection needs to be done on information objects (files, form data, JSON, XML, EXI, etc.). Sensitive data should be protected in such a way that the service provider cannot access keys nor cleartext. In this way, cleartext data is only accessible by the individual or enterprise that protected it in the first place, or someone selectively given authorization to access the data.

Not only would such an approach protect the information against the service provider, it would also protect the information in the case of data breaches, as well as making service providers unable to comply with government demands to share customer data. While this can be implemented in native applications, web applications (and therefore the third-party service provider) have control of the JavaScript runtime environment and therefore access to cleartext content before the content is encrypted and after it is decrypted.

In light of the mass-surveillance revelations, the Cloud Storage and Protected Documents Exchange use cases [5] should be brought up again and extended. For information security and privacy reasons, the user must be able to protect sensitive information also from a web application. Doing so in a browser without relying on third party applications implies that it should be possible to make sensitive content inaccessible to JavaScript controlled by the web application.

Similar thoughts have already been pursued in WebRTC [6] [7], where the web application is not trusted with the session encryption keys. Another example is Isolated Media Streams [8], which are inaccessible to the content JavaScript. A further example in this direction is the Encrypted Media Extensions [9] that allows the web application to control the rendering of media without having access to the unencrypted media.

2. Example Use Cases

Cloud storage

Cloud storage like Dropbox are natural targets for pervasive surveillance. There is currently no easy way to use cloud storage in a secure way. Using a web application or native app for accessing the cloud storage gives the service provider access to keys or cleartext. Tech savvy persons can download a third-party application to do the encryption but that is not for average people, and there is no easy to validate the trustworthiness of third-party applications. W3C should work on standards and APIs for secure File Input/File Download where cleartext is not accessible by the web application/JavaScript runtime environment.

HTML forms

Data entered in forms is available to the web application in cleartext. In situations when the application cannot be trusted this is problematic. One way to handle this issue could be to introduce forms where the data is not accessible to the application in any other form than encrypted. One example of secure forms is Google End-to-End [10], a Chrome extension to add OpenPGP support to Gmail.

3. Conclusions

The market potential for enterprise and government cloud services and web applications are held back by privacy and security concerns. Trust in global cloud services is fundamental for the further development of the Internet economy. W3C should take action to make cloud services and web applications trustworthy also in a pervasive monitoring landscape.

While host-based security has served the Web ecosystem well [11], and will likely continue to do so in many cases, this security model alone is not enough in a pervasive monitoring landscape. To fight pervasive monitoring while still enabling the many benefits of cloud services, W3C needs to take action and work on general mechanisms where the browser encrypts and decrypts data in such a way that the web application cannot access keys nor cleartext.

While technical solutions to accomplish this seem to be available, a challenge might be the user interaction – how can the user verify that the information given is not accessible by the application?

References

[1] STRINT - A W3C/IAB workshop on Strengthening the Internet Against Pervasive Monitoring
http://www.w3.org/TR/WebCryptoAPI/

[2] Barnes et al. “Pervasive Attack: A Threat Model and Problem Statement”
http://tools.ietf.org/html/draft-barnes-pervasive-problem

[3] Cooper, Jennings, “The Trust-to-Trust Model of Cloud Services”
https://www.w3.org/2014/strint/papers/30.pdf

[4] Farrell, Tschofenig, IETF RFC 7258, “Pervasive Monitoring Is an Attack”
https://tools.ietf.org/html/rfc7258

[5] W3C, “Web Cryptography API”
http://www.w3.org/TR/WebCryptoAPI/

[6] W3C, “WebRTC 1.0: Real-time Communication Between Browsers”
http://www.w3.org/TR/webrtc/

[7] IETF, “Real-Time Communication in WEB-browsers”
http://tools.ietf.org/wg/rtcweb/

[8] W3C, “Media Capture and Streams”
http://www.w3.org/TR/mediacapture-streams/#isolated-media-streams

[9] W3C, “Encrypted Media Extensions”
http://www.w3.org/TR/encrypted-media/

[10] Google, “End-To-End”
https://code.google.com/p/end-to-end/

[11] Halpin, “The W3C Web Cryptography API: Design and Issues”, 2014
http://ws-rest.org/2014/sites/default/files/wsrest2014_submission_11.pdf