

UNIVERSITY OF PADOVA

COMPUTER AND NETWORK SECURITY

insert-title-here

Students:

Davide Trevisan

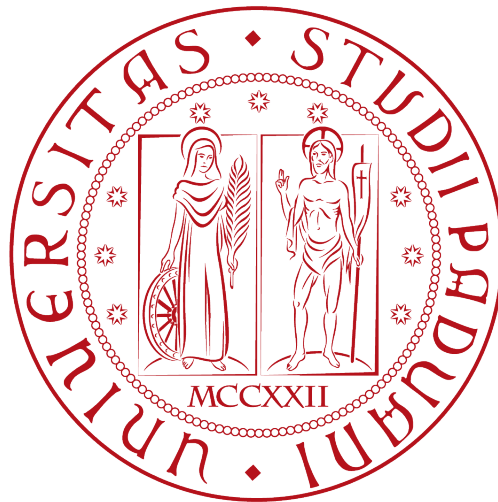
Andrea Multineddu

Registration number:

1070686

MATRICOLA

Thursday 25 August 2016



1 Stealth Screenshot Extension

The source code of the extension we developed is downloadable [here](#)

should be aware of what is happening; not triggering any event returns only null. Once triggered, the extension schedule all the future screenshot through the

1.1 Scope of the extension

the scope of the stealth screenshot extension is to create a small prototype of an extension that catch screenshot in a regular interval with the minimum possible user interaction and obviously giving it back.

1.2.2 Screenshot retrieval

The screenshot can be collected in two ways in the wxextension we developed: through a combination of key for retrieving all screenshots (we programmed it on Ctrl+Shift+Y) or writing "show" in the omnibox, pressing "tab" and writing in the omnibox the number of screenshot to show (starting from the last one).

1.2 How the extension work

1.2.1 Activation and screenshot collection

The extension uses the relatively new tab APIs given by chrome too take screenshot. The extension makes use of the "storage", "tabs", "all_urls;", "unlimitedStorage", "activeTab" permissions in his manifest file. The extension works in a simple way: it triggers on a browser event, in this case the click on the icon of the extension, but there are other possibilities: we tested that there are no simple ways to take screenshot without no interaction, because the security policy of the API doesn't allow to take a screenshot without some kind of events, presuming that the user

1.2.3 Performance

We tested that the extension is able to take screenshot with a interval of 1 minute for more than 2 hours without losing any screenshot. The application occupy less of 50MB of ram for an hour of screenshots taken every minute. The impact on the CPU is negligible. Screenshot retrieval through the key combo however can crash chrome, although it never happened in our PC because of the great performance (we had an Intel i5 6400, with 8GB of DDR4 memory and SSD), but it got freezed for some seconds.

1.3 Limitations

The major limitation of this extension is that the screenshot only lives until the extension is active: this implies that closing all the windows of chrome completely deletes the screenshot taken. This is a consequence of how the API and the chrome sandbox works. We at the moment found no way to get around it, but we are pretty confident that is possible to save those screenshots, but for our lack of knowledge we are not able at

the moment to demonstrate it.

1.4 Future work

Possible future works will focus on find a way to save those screenshot. The application should also be rewritten without the tab API, to avoid of the limitation of it. Javascript inject the code for the screenshot or use it for simulate the right events should do the work, but the time and the train needed to do it made impossible for us to test it for this paper.