CSE 509 -- Fall 2016

Project 4: Malicious browser extension with server backend

Since more and more native applications move to the web and to the cloud, attackers have realized that there is a wealth of private information collected in websites. Because of this, the last years, we have seen a surge of malicious browser extensions that steal the user's data without the knowledge of the user.
In this project, you are asked to design and implement a basic malicious extension for the browser of your choice (either Mozilla Firefox or Google Chrome). This extension, after being installed by the user (user convinced that the extension does something useful) should:

Generic features

- Leak the browsing history of a user to the attacker's server
- Steal usernames and passwords from forms as the user writes them
- Steal cookies from outgoing HTTP requests
- Stop the user from visiting security websites and reroute them to random websites (Depending on the browser you choose, you may need to be creative as to how you are going to do that). This list must be dynamically updatable.

All collected data should be sent out to a website that the attacker controls. The attacker needs to have a WebUI that shows him all the collected data from each individual users.

Dynamic features

Every time that an infected user opens her browser, your extension needs to check-in with a remote attacker-controlled server. That remote server should have a Web UI that shows the infected users that are currently online. The attacker must have the ability to perform the following actions on per-user basis:

- **Personalized Phishing.** The attacker must be able to instruct the malicious extension of an infected user to change the DOM of specific pages according to the attacker's desires. This can allow the attacker to convince the victim that a popular site wants her to download a specific executable or needs more information from her
- **JavaScript execution.** The attacker must be able to instruct the malicious extension of an infected user to run arbitrary JS code within the context of victim website. E.g. execute the following script "<script>alert("Hello");</script>" within the context of stonybrook.edu. The attacker can use this to steal very specific information from custom websites of his interest.

YOU MUST implement all the project by yourself. **DO NOT** use an existing malicious extension. **ANY** code fragments that you find and use must be documented in your report.