Browser Security Problem Set

Description

The goal of this problem set is to perform a client-side code injection against a vulnerable web application, and modify the application to defend against the class of vulnerabilities.

To complete the problem set, you will need to ssh to your container at `$user@amplifier.ccs.neu.edu:$port`, where `$user` is your gitlab username and `$port` is your assigned ssh port (https://seclab-devel.ccs.neu.edu/snippets/6). Authentication is performed using any of your uploaded ssh public keys in gitlab.

You will also need to clone the problem set repository located at `git@seclab-devel.ccs.neu.edu:softvulnsec/prset05.git`.

Important Information	
Available	Wed 01 Apr 10:00 EST
Submission Deadline	Tue 07 Apr 18:00 EST
Gitlab URL	https://seclab-devel.ccs.neu.edu/softvulnsec/prset05.git (https://seclab-devel.ccs.neu.edu/softvulnsec/prset05.git)

Vulnerability Identification

The web application you have cloned makes use of a third-party advertisement network to monetize the site's content.

Examine the source code to the web application to identify a client-side injection vulnerability.

Exploit the Vulnerability

Assume that you are a malicious ad network, or have compromised the ad network. Develop an exploit for the vulnerability that you've identified. As a proof-of-concept, you should leak the value `document.secret` to a server that you control.

Patch the Vulnerability

Using the knowledge of how you exploited the vulnerability, develop a patch. In particular, your patch should consider both the issue of avoiding untrusted code execution and origin endpoint restrictions.

Harden the Application

In addition to patching the specific vulnerability you've found, you can also harden the application against other possible vulnerabilities. Do this by developing a CSP ruleset that is *as tight as possible*. Points will be deducted for overly-permissive rulesets. In particular, you should not use `unsafe-inline` or `unsafe-eval`.

Modify the application to indicate that the browser should enforce your ruleset.

Answer Submission

Fork the repository for this problem set in gitlab. On the master branch, commit both your patch for the vulnerability as well as your modifications to enforce a strong CSP ruleset.

In addition, commit your proof-of-concept exploit to `exploit/`. Finally, commit a `README.md` that describes your solution *as precisely as possible*.

Updated Fri 07 Aug 2015 10:38 EDT
Revision master/52d826a