Mozilla Security Review

General Information

Project Name:	Festival Contest Donation Forms
Security Reviewer:	Matt Fuller (mfuller)
Date of Review:	9/12/2012
Bugzilla Bug #:	781385

Background Information:

In mid-September, we'll be launching a fundraising campaign for the Mozilla Festival - - basically offering folks a chance to win a trip to London for the festival in exchange for donating (or can enter without donating by signing up).

So this will just be a set of fundraising and signup pages built in Blue State Digital, similar to other pages that have been reviewed in the past.

Helpful URLs:

Pages: https://donate.mozilla.org/page/s/webmaker-sweepstakes
https://donate.mozilla.org/page/contribute/webmaker-sweepstakes

Access Information:

How is the application accessed?
Is the application internal or publicly available?
If it is internal, what mechanism prevents non-members from accessing it?
Are there links to user-only resources displayed to non-users?
Are login pages secured with SSL over HTTPS?
If HTTPS is used, is Strict Transport Security set?

The application is a single form allowing users to enter personal and payment information in order to sign up for a monthly billing donation to Mozilla webmaker. The application is external, available to the public and is hosted on a mozilla.org domain. The application is secured with SSL.

Infrastructure and Backends:

What languages do the applications use?
What database language is used if applicable?
Are the running versions up to date?
What server is it running on?

The page is a Python/Django application. The form is provided, managed, and sent from BSD.

Accounts and Passwords:

If the mechanism to prevent general access is a password, how is the signup process handled?
How is account information stored?

Are passwords properly stored within databases if applicable?
Is a password policy in place?
Are accounts locked-out after a number of invalid logins?
Are passwords 8 characters or greater?
Do passwords use both numbers and letters (and possibly symbols)?
Is there a blacklist of common passwords?
Do passwords expire after X days and require a reset?
Are invalid logins logged?
Is there a lockout after X invalid attempts?
Is the error message for lockout generic (does not include if user/pass is valid)?
How are password resets handled (i.e. email, security question, etc.)?
Do emails sent after signup/reset contain a session link? (should not)
Do email verification codes expire after one use or 8 hours?
Is password reuse limited/prevented?

There are no accounts or passwords.

Session Management:
How long are session cookies stored?
Are session tokens 128-bit or greater?
Is session ID token creation handled by the server (or cryptographically if locally)?
Do authenticated sessions timeout after 15 minutes?
Is the Secure Flag enabled for all set cookies?
Is the HTTP-Only flag used to disable malicious script access to the session ID?
Are new session ids created on login?
On logout, are session ids invalidated?

No login information is used.

Third-Party Resources:
Are third-party resources used (i.e. JavaScript libraries, images, CSS, etc.)?
Can those resources be trusted / are they from reputable sources?
Is there a chance the resource could be compromised?
Is it possible to host the resources locally to mitigate risks?
Is a third-party responsible for storage of user data?
Does the application connect with services like Facebook, Twitter, etc?

JQuery API from Google, image off Amazon aws, BSD hosted page itself JavaScript and stylesheets loaded from https://dnwssx4l7gl7s.cloudfront.net/

Data Handling:
What kind of data is transferred between the user and the application?
Is this data generated by the user or generated automatically?
Can the data be trusted?
How is the data sent to the application (i.e. JSON format, plain-text, GET/POST, etc.)?
How is the data handled by the server/application?
Can the data be manipulated in transit?
What happens if the data is altered?
What is done with the data once it is received (i.e. stored in a database, displayed to users)?
Is any data storage done via cookies? If so, what kind of data is stored via this method?

The only data submitted is the user's email, name, and country. This is provided by the user. Although sensitive, it is less sensitive than other forms collecting payment data.

Uploaded Data:
Can the user upload data to the application?
Are extensions, file size, file type (not only based on extension), etc. checked?
Are files renamed upon upload?
Is the correct content-type used?

No uploaded data.

Data Sensitivity:
What kind of data is being stored and/or manipulated by the application?
Does this data need to be encrypted in transit? In storage?
What is the impact if this data is lost/stolen?
Is secure/sensitive data sent over SSL?

Sensitive data is name, email, and it is sent over SSL. Impact is moderate.

Application Administration:
Is there an administration console?
Can it be accessed publicly?
How is it secured if so?
Are correct methods used to prevent admin actions from being performed outside of the admin console (i.e. using CSRF tokens)?
Are there any configuration pages that should not be made public?

No admin page related to forms.

Security Coding Flaws:
Have all user inputs been sanitized?
Is a maximum size for data (input or uploads) defined?
Do all URL variables pass through sanitization?
Is data from databases escaped properly?
Are CSRF tokens used to prevent POSTs from outside websites?
If a database is used, are protections against SQL injection in place?
Is validation done on the server side (not just client-side)?
Is output encoded in addition to sanitization of input?
Does the user ever send data to the OS level?
Are x-frame options sent to "deny" or "sameorigin"?
Is debug mode disabled?

User inputs properly filtered
CSRF tokens not used.
X-Frame set to sameorigin
Cookies secure
STS set

Testing:

List all tests performed on the application

1. Checked input for XSS/SQL injection
2. Checked for CSRF / use of nonce/token
3. X-Frame-Options
4. Look for STS
5. Fuzzed params
6. Is SSL used?

Results:

List the results of the above tests

1. Properly escaped
2. No CSRF protection
3. Set to SAMEORIGIN
4. STS Set
5. Params escaped properly
6. SSL used

Meetings and Notes:

CSRF is not used on BSD forms.