Mozilla Security Review

General Information

Project Name:	Mozilla Monthly Donation Form
Security Reviewer:	Matt Fuller (mfuller)
Date of Review:	7/7/12
Bugzilla Bug #:	771674

Background Information:

We are launching a new donation campaign, asking folks to donate monthly to fund our webmaker work, and would love a sec review to make sure it's good to go before we launch next Wednesday.

This is similar to the last monthly page that was reviewed in Bug 723734 (https://donate.mozilla.org/page/contribute/monthlymozillatshirt) with two major differences:
--There is a new wrapper that fits into our webmaker project's visual brand (just surface/image background changes)
--There is a mobile version of the page that appears if you're on a mobile browser

Helpful URLs:

The form: https://donate.mozilla.org/page/contribute/support-webmaker-monthly

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?

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?

The jQuery library is loaded from Google's server
Several scripts and stylesheets loaded from mozilla.org directly
Image loaded from amazon aws.
JavaScript and stylesheets loaded from https://dnwssx4l7gl7s.cloudfront.net/
If the user selects Paypal as a payment method, he is redirected to Paypal's login.

All 3rd party resources were confirmed by devs to be owned by BSD.

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 data is entered by the user and includes personal and payment details (cc number, address, etc.). It is submitted to https://donate.mozilla.org/page/cde/Contribution/Charge via POST. The data is transmitted via HTTPS and should not be altered in transit.

The data is saved by BSD.

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 uploads are allowed.

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?

All data is sent over SSL. The risk if it is lost/stolen/misused is high as it contains credit card and personal data.

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 console.

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 are checked on client and server side. Input size is defined on client side, but “long string of a’s” appears to return a 500 Internal Server Error. Possible DOS attack point. Data from database is not echoed back to the user at any point. No commands to OS level Debug disabled x-frames sameorigin

Testing:
List all tests performed on the application

1. Standard testing for XSS 2. Check for loading of 3rd party content and origin 3. Check for Strict Transport Security 4. Check for x-frame info 5. Manipulated JS to allow for submission of altered form data not matching validation checks (to make sure done on server side too) 6. Checked for CSRF issues

Results:
List the results of the above tests

1. No XSS found 2. Confirmed - all owned by us, Google’s JQuery script, or BSD 3. Enabled 4. Set to sameorigin 5. Returned error. Good. 6. Pass.

Meetings and Notes:
