Mozilla Security Review

General Information

Project Name:	Affiliates Facebook Application
Security Reviewer:	Matt Fuller (mfuller)
Date of Review:	8/30/2012
Bugzilla Bug #:	772287

Background Information:

A Facebook application is developed to allow users to spread links for downloading Firefox, create banners for their pages, send messages to friends, and track their own referrals.

Note: This application is loaded within Facebook's frame. Many of the security questions below will be handled by Facebook rather than our site. For example, the app uses Facebook accounts, so all questions regarding account policies are N/A.

Helpful URLs:

FB App: https://apps.facebook.com/fxaffiliates/
Dev Site: https://affiliates-fb-dev.allizom.org/

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 accessed through Facebook. Users must install the app, accept permissions, then load the page. Loading the framed page directly, redirects to the non-FB version.

Public.

All accessed via HTTPS.

STS set.

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?

Python

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?

All access is through Facebook. Facebook handles account verification. Can link to an affiliates account, which is handled on the non-FB page, not a part of this review.

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?

All handled by FB.

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?

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 mostly handled by Facebook but the user provides a comment for their Firefox Download link. All data is handled properly (XSS escaped etc)

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 - although it does pull the FB profile image

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?

FB account information ^ all above is handled by Facebook.

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?

Admin console is handled by Facebook (the developer center). CSRF handled by Facebook.

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 sanitized, no URL variables, data escaped, csrf by Facebook, x-frame options deny, debug disabled, sensitive Facebook keys are in local settings

Testing:
List all tests performed on the application

1. Loaded page outside Facebook
2. Checked input for XSS
3. Checked for CSRF
4. Facebook keys in local not public
5. Loaded tracking URL directly

Results:
List the results of the above tests

1. Redirects to non-FB version
2. Properly escaped
3. Handled by Facebook - good
4. In local, good
5. Causes click counter to increment (see notes)

Meetings and Notes:

Talked about spoofing click counter by directly loading URLs. Mkelly said it is a known issue and they'll be fixing it in future releases.