Score Forgery Report for Kongregate.com

Alec Thomson Alan Huang Casey McNamara Chris Billie

1 Introduction

Kongregate is one of the most popular Flash Game portals on the web. In addition to providing a hub for Flash games, Kongregate also provides services such as a shared-content server, in-app purchases, chat functionality, and a "stats" service which is regularly used to record players' high scores. For the purposes of this project, we focused exclusively on the stats service to see what measures are taken to prevent players from submitting fake stats (i.e. scores).

The APIs and security measures provided by Kongregate are outlined in Section 2. After analyzing these, we used a game owned by one of the authors to construct an exploit that posts a fake score to the Kongregate servers. This exploit is described in Section 3. In Section 4, we propose some possible additions to the Kongregate design to allow for even greater security of the stats API.

2 Kongregate's Security Design

Kongregate offers two APIs for games to access the Kongregate services, a *client API* and a *server API*. The client API allows game code to talk directly to the Kongregate servers to post stats and scores. The server API allows a developer owned server to communicate to the Kongregate servers on behalf of a client game. Since the client API is easier to use and does not require the game developer to maintain an external server, it is much more widely used than the server API, but potentially less secure against forgeries.

2.1 Client API

A Flash game wishing to access the client API is given an "API path" parameter as a load variable when it is embedded onto the Kongregate page. The API path points to a Kongregate-owned SWF file. When the game connects to the API, it loads the API SWF via an HTTP GET request and sets up cross-domain scripting via ActionScript's "Security.allowDomain()" method. The API SWF is then able to authenticate the game and user by extracting additional parameters embedded in the game SWF at load time, including the player's username, a game identification token, a user authentication token, etc. Once authenticated, the game can make RPC calls to the API SWF to post stats and scores to the Kongregate servers.

2.2 Server API

The Kongregate Server API is designed to allow a game to communicate to a developer owned server which can then communicate on behalf of the game with the Kongregate servers to post stats and scores. When a developer owned server communicates with Kongregate, it signs all of its messages with a unique "Game API Key" provided by Kongregate, allowing secure communication between the two as long as the key remains a shared secret between the developer and Kongregate. While this API was primarily created for payment verification and other sensitive tasks, it can also be used to allow a developer to perform server-side verification of scores before they are submitted to Kongregate.

Specifically, when a developer sets up the available game stats on the Kongregate website, they have the option of allowing certain stats to be submitted by the server API only, which gives the developer full control over the security of score submission. For the purposes of our project, the verification strategies we use to implement a forgery-resistant score server can be integrated with Kongregate's server API to achieve the same level of security as our implementation when posting scores to Kongregate.

3 The Exploit

To post a fake score to the Kongregate servers, we attacked the game "Roller Derby Riot Queens" (RDRQ) which is owned by one of the authors and uses the client API. The exploit consists of an ActionScript project available in our github repository under the "Kongregate_Exploit" directory. The project compiles a SWF file that simply connects to the Kongregate client API through the method described in Section 2.1 and posts a fake score. The primary trick is that the page embedding the exploit SWF applies the same authentication parameters to the SWF at load time as those embedded in the actual game. These parameters were easy to replicate as they can be retrieved from the actual game's embedding page via the Firefox web console.

4 Possible Improvements

While Kongregate's anti-forgery security seems to rely almost entirely on developer-specific implementations via the server API, we noticed there were a few steps that could be taken to improve the security of the client API slightly by making it significantly more difficult for a casual attacker to post a fake score. While we acknowledge some of these measures are impractical, particularly since Kongregate has many other goals beyond anti-forgery security, we hope they might be useful for implementing our own forgery-resistant score server.

- Instead of embedding the authentication information in the game at load-time via javascript on the embedding page, Kongregate could potentially embed the authenticating information in the game on the server prior to serving the game to the embedding page. While the information would still be available to the attacker inside the game SWF, it would be significantly more difficult to retrieve. As a further step, the Kongregate server could also use a "Flash Obfuscator" to further protect the authentication information from the attacker.

- Kongregate already hosts many duplicate API SWFs to ease load on any single file. One possible improvement would be to only allow a game to connect to a unique API SWF generated when the game is loaded. This SWF would only accept API connections from the specific game and would refuse any additional connection requests after connecting to the game SWF. With this defense, an attacker would have to prevent the game SWF from connecting to the API (which is usually done immediately on load) before the exploit could be run.

- When the API SWF is requested from the embedding Kongregate site, a "Referer" header is included in the HTTP request which identifies the Kongregate site the game is hosted on. This "Referer" will not be the same for any HTTP requests from the exploit SWF as it will not be hosted on the Kongregate site. Using this information, the Kongregate site could refuse to serve the API SWF if the referer is not correct. Since Actionscript does not allow SWFs to modify headers when making HTTP GET requests (POSTS are a little different), an attacker would have to send the request through a proxy server that modifies the referer header to actually access the API.