Recovering user-browser interactions from HTTP logs of Rich Internet Applications

Salman Hooshmand, Gregor v. Bochmann , Muhammad Faheem, Guy-Vincent Jourdan, Russ Couturier,
, Iosif Viorel Onut
School of Information Technology and Engineering - University of Ottawa

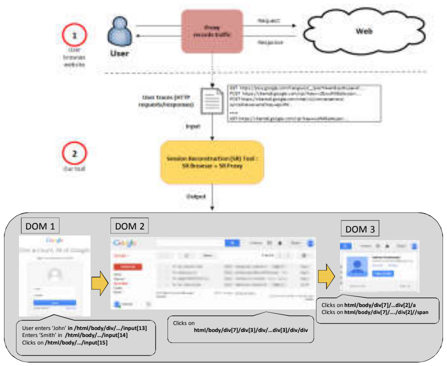
uOttawa
SOFTWARE SECURITY RESEARCH GROUP
In Collaboration With IBM

Introduction

In a Web Application, each user-session generates a series of HTTP requests and responses regardless of technology/device used.

It is beneficial to reconstruct user's session from HTTP traces for several reasons, including:

- **Forensics Analysis**: Analysis of usage logs of a security incident to find out how the attack happened.
- **Debugging**: Reconstruction of what user has done to reproduce the fault automatically after a user reports a bug.
- **Automatic Login**: Automatic learning of login action for crawlers.



User enters 'John' in **/html/body/div/.../input[13]**
Enters 'Smith' in **/html/body/.../input[14]**
Clicks on **/html/body/.../input[15]**

Clicks on **html/body/div[7]/div[3]/div/...div[3]/div/div**

Clicks on **html/body/div[7]/...div[2]/a**
Clicks on **html/body/div[7]/..../div[2]//span**

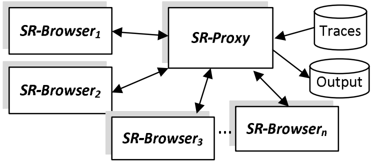
Methodology

We have developed **D-ForenRIA**, a session reconstruction (SR) tool which reconstructs user's session based on a set of previously recorded HTTP requests/responses.

D-ForenRIA has two components:

1- SR-Proxy: Responds to HTTP requests from SR-Browsers based on the traffic captured earlier. The SR-Proxy replaces the actual application server**.**

2- SR-Browsers: A set of browsers where each browser loads a page, selects and executes events on the DOM, and communicates with the SR-Proxy to rebuild the user session.



D-ForenRIA

Implementation:

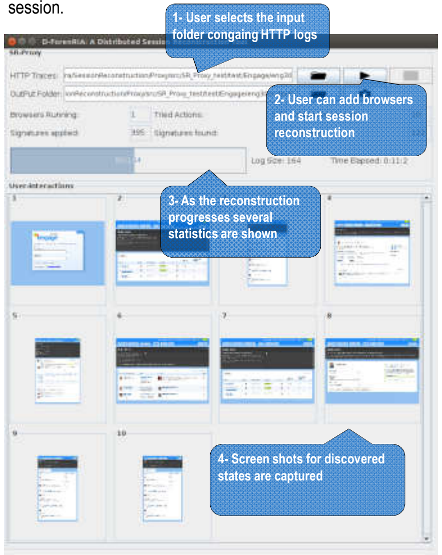
Based on our methodology, we have used the following technologies to implement D-ForenRIA:

SR-Browser relies on **Selenium** to execute JavaScript events and to get access to the current DOM of the application.

SR-Proxy was developed using **Java.**

Input and Output:

- **Input** is HTTP traces of user's previous session (Captured using Fiddler).
- **Output** is a series of DOMs and the XPath of the elements with which the user has interacted and provided inputs during the session.



1- User selects the input folder congaing HTTP logs

2- User can add browsers and start session reconstruction

3- As the reconstruction progresses several statistics are shown

4- Screen shots for discovered states are captured

Challenges and Solutions

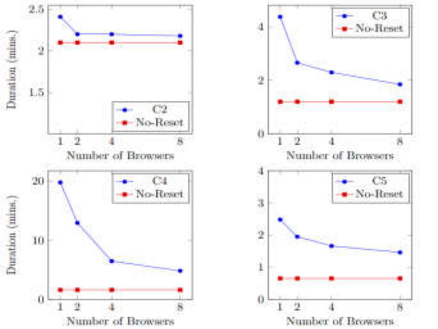
- **Finding the Next promising actions**: Considering a large number of possible events on each DOM, so a blind search is not practical. D-ForenRIA prioritizes "Actionable Elements " and it learns the "Signature" of Actions .
- **Random Parameters:** The SR-Proxy asks the SR-Browser to repeat the execution of actions generating random parameters in requests**.**
- **Timers :** The SR-Browser detects the existence of timers, timer handlers are being executed at the appropriate time.
- **JSON based User-inputs**: user-input interactions that encode data using JSON are detected by performing actions using sample data.
- **SSL Encrypted Websites:** A "man-in-the-middle" proxy has been added to decrypt requests and encrypt responses.
- **AJAX calls**: SR-Browser keeps track of sent requests and received responses. No event is selected/executed while we have pending requests.

Experiments

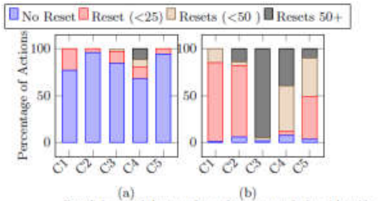
We have tested our tool on several Websites. Experimental results have shown that D-ForenRIA was able to handle different RIAs successfully.

Subject applications and characteristics of the recorded user-sessions

ID	Name	#Requests	#Actions	URL
C1	Elfinder	175	150	https://github.com/Studio-42/elFinder
C2	AltoroMutual	204	50	http://www.altoromutual.com/
C3	PeriodicTable	94	45	http://ssrg.site.uottawa.ca/apr5/success1/
C4	Engage	164	25	http://engage.calibreapps.com/
C5	TestRIA	74	31	http://ssrg.eecs.uottawa.ca/testbeds.html



Scalability of *D-ForenRIA* in different RIAs compared to the no-reset time.



Breakdown of the number of resets needed to identify a user-browser interaction in *D-ForenRIA* (a) and in the basic solution (b).

A demonstration of several experiments including sample inputs/outputs of the tool can be found on :

http://ssrg.site.uottawa.ca/sr/demo.html

Conclusion and Future Work

- We have presented a tool to reconstruct user-sessions from HTTP traces. It includes the ability to fill forms and works with SSL encrypted sites.
- In the future, we plan to improve the performance of D-ForenRIA and connect it to crawlers and testing tools.

Acknowledgments

This work is supported by Natural Science and Engineering Research Council of Canada and Center for Advanced Studies, IBM Canada.

DISCLAIMER
The views expressed in this poster are the sole responsibility of the authors and do not necessarily reflect those of the Center for Advanced Studies of **IBM**.