uOttawa

SOFTWARE SECURITY RESEARCH GROUP

In Collaboration With IBM

The Reconstruction of User-Sessions from HTTP Traces in RIAs

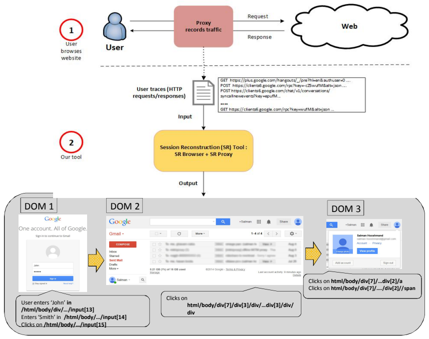
Sara Baghbanzadeh, Salman Hooshmand, Seyed M. Mirtaheri, Muhammad Faheem, Gregor v. Bochmann , Guy-Vincent Jourdan, Iosif Viorel Onut
School of Information Technology and Engineering - University of Ottawa

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:

- **Automatic testing**: replaying what a user has done
- **Debugging**: when a bug is reported, we can reconstruct what was actually done to automatically reproduce the fault
- **Automatic login**: Crawlers can learn how to login automatically to continue their work



GET https://plus.google.com/hangouts/_/pre?hl=en&authuser=0 ....
POST https://clients6.google.com/rpc?key=-cZ5wufM&alt=json ....
POST https://clients6.google.com/chat/v1/conversations/syncallnewevents?key=epufM...
....
GET https://clients6.google.com/rpc?key=wufM&alt=json ...

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**

Input and Output :

- **Input** is HTTP traces of user's previous session recorded by proxy.
- **Output** is a series of **DOMs** and the **XPath** of the elements on which the user has interacted, and inputs were provided by the user during the session

Background

Some methods have been proposed to capture and replay user's actions in JavaScript applications, e.g.

- **Mugshot** : logs sequence of JavaScript events executed in a browser to be sent to developers for debugging.
- **Timelapse** : records all events inside browser's web debugger, with ability to go back and forth for execution.
- **ClickMiner:** reconstructs user sessions from traces recorded by a passive proxy.

However, these have either require installation of additional software on user's machine (as in Mugshot and Timelapse) or has limited support for handling of JavaScript events and no ability to extract user-inputs (as in ClickMiner).

References

- Neasbitt, Christopher, et al. "Clickminer: Towards forensic reconstruction of user-browser interactions from network traces." *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014.
- Burg, B., Bailey, R., Ko, A. J., & Ernst, M. D. (2013, October). Interactive record/replay for web application debugging. In Proceedings of the 26th annual ACM symposium on User interface software and technology (pp. 473-484). ACM.
- Sampath, Sreedevi. "Advances in User-Session-Based Testing of Web Applications." Advances in Computers 86 (2012): 87-108.
- Richards, Gregor, et al. "Automated construction of JavaScript benchmarks." ACM SIGPLAN Notices 46.10 (2011): 677-694.
- Mickens, J. W., Elson, J., & Howell, J. (2010, April). Mugshot: Deterministic Capture and Replay for JavaScript Applications. In NSDI (Vol. 10, pp. 159-174).
- Atterer, Richard, and Albrecht Schmidt. "Tracking the interaction of users with AJAX applications for usability testing." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2007.

Methodology

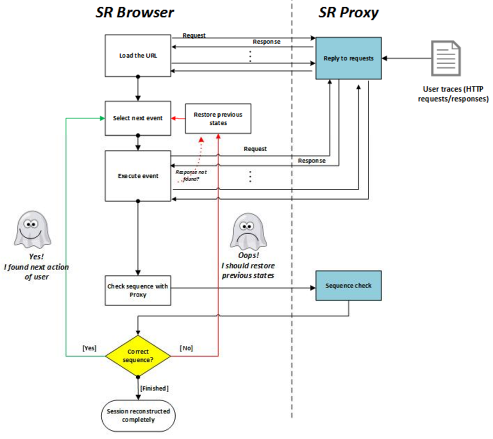
We have developed a **session reconstruction (SR) tool** which reconstructs user's session based on a set of previously recorded HTTP requests/responses.

The SR tool has two components:

1- SR proxy which responds to HTTP requests from the SR browser based on the traffic captured earlier. The SR proxy replaces the actual application server.

2- SR browser which loads a page, selects and executes events on the DOM and communicates with the SR proxy to rebuild the user session.

The user session is reconstructed using the following approach:



Implementation

Based on our methodology, we have used the following technologies to implement our SR tool:

- SR browser relies on PhantomJS to execute JavaScript events and get access to the current DOM of the application.
- SR proxy was developed using PHP. Fiddler was used to capture the user traces.

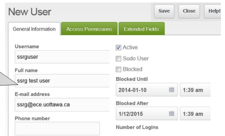
Handling AJAX

AJAX calls are asynchronous, how does the SR browser handle this?

- Our SR browser keeps track of sent requests and received responses.
- No event is selected/executed and no sequence check is done while we have pending requests.

Handling User Inputs

Users enter values in forms, can you detect these?



New User

Username: ssrguser
Full name: ssrg test user
E-mail address: ssrg@ece.uottawa.ca
Phone number:
☑ Active
☐ Sudo User
☐ Blocked
Blocked Until: 2014-01-10 1:39 am
Blocked After: 1/12/2015 1:39 am
Number of Logins

We try to extract possible values from traces, the SR browser asks the SR proxy which values should be used

Finding the Next User-Interaction

There are typically large number of possible events on each DOM, so a blind search is not practical. SR-Browser collaborates with SR-Proxy to find the most probable user action using following techniques:

- **Actionable Elements**
- **Explicit clues** in the next trace
- **Implicit clues**
  - Known JavaScript Libraries
  - Early click
  - Avoid non-existent click

Other challenges

- **Random parameters**: There are some random parameters in generated requests during replay.
  - Two instances of SR-Browser have been used to detect these parameters.
- **SSL encrypted websites**: The generated traffic is encrypted and SR-Proxy can not see the plain requests.
  - A MITM (man-in-the-middle) Proxy has been implemented to decrypt requests and responses
  - We assume that the recorded traffic is decrypted.

Experiments

We have tested our tool on several websites. It was able to handle relatively complex RIAs successfully.

Name	#Act.	#Req.	Time (hh:mm:ss)		Cost	
			Proposed method	Basic method	Proposed method	Basic method
OpenCard	150	325	0:10:26	76:10:45	3,221	1,808,250
OSCommerce	150	532	0:02:44	21:23:15	150	501,806
RTC	30	218	0:46:54	50:53:44	1,423	94,242
El-finder	150	175	0:14:55	07:24:40	12,533	376,820
Engage	25	164	0:31:13	01:47:02	7,834	17,052
TestRia	31	74	0:00:37	00:22:51	302	15,812
PeriodicTable	89	94	0:07:38	36:20:45	4,453	1,559,796
AltoroMutual	150	204	0:01:41	25:24:30	358	815,302
Joomla	150	253	0:48:20	15:24:30	344	2274

Conclusion and Future work

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

Acknowledgements

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.