Instructions to crawl input data

These instructions refer to the replicability package available at the following URL https://bit.ly/MT2020

The proposed approach takes sequences of actions as inputs. These inputs could be automatically generated by using a modified version of the Web crawling tool "crawljax". Please refer to the submitted paper for an overview of the changes that we applied to crawlajx. In the following paragraph, we describe how to crawl input data using the modified version of "crawljax".

1. Setup of crawljax

A prepared crawljax package is available from the location "/Security-MT-ICST-2020/Software/crawljax.zip". Decompress this zip file, you will receive the folder “crawljax”, which contains:

- crawljax.jar: the package of the modified version of crawljax.
- crawlConfig.json: a configuration file including the parameters required to crawl the Jenkins system provided in our replicability package. In the following we describe the parameters that should be set:
    - systemUrl: indicates the index URL of the SUT (e.g., http://192.168.56.102:8080).
    - time: determines the length of the crawling period in minutes (e.g., 300).
    - userParam: specifies the username parameter in the login page of the SUT (e.g., “j_username” for the case of Jenkins).
    - passwordParam: specifies the password parameter in the login page of the SUT (e.g., “j_password” for the case of Jenkins).
    - username: specifies a username to log into the SUT (e.g., “admin”).
    - password: specifies a password to log into the SUT (e.g., “adminPass”).
    - path: indicates the folder path to store the crawled data.
    - loginButtonText: determines the text of login button in the login page of the SUT (e.g., “log in”).
    - ignoredUrls: lists of URLs should be avoided during the crawling period (e.g., URLs to shut down the SUT, or to upgrade the SUT).

In order to execute crawljax you need to download and install "chromedriver", a tool that enables the use of the Chrome browser through Selenium HQ, which is in turn used by crawljax.

Please download the chromedriver from the following URL
https://chromedriver.storage.googleapis.com/index.html?path=2.44/

The chromedriver should be copied in "/usr/local/bin/". We tested it using MacOS.

2. Crawling the system under test

Before crawling the SUT, be sure that the SUT has been started. In the case of Jenkins, see the instructions in "/Security-MT-ICST-2020/EmpiricalData/RQ2/InstructionsToReplicateRQ2.pdf" to configure and start the system.

To start crawling SUT using our modified version of crawljax please perform the following actions:

- Open a Terminal.

- Change directory to the folder containing crawjax.jar and crawlConfig.json files.
- Invoke the FullCrawlerWithParams class from the crawljax.jar:
  java -cp crawljax.jar com.crawljax.examples.FullCrawlerWithParams ./crawlConfig.json

During the execution of crawljax, the crawled data is stored in the folder indicated by the parameter "path" in the crawlConfig.json file. The main output of our version of crawlajx is the file "inputs.json", which is created once crawljax terminates, and contains the source inputs to be used for metamorphic testing.

3. Additional processing of the collected source inputs

Despite the file "inputs.json" might be used "as is" to successfully execute most of our metamorphic relations, we further process the data in "inputs.json" to (1) collect "the background URL requests" (page 6 of the paper) and (2) collect all the outputs that should be observed when executing the source inputs, which is used by the Web-specific function "userCanRetrieveContent" (Table 2 of the paper).

Despite this analysis might be integrated in crawljax, we have decoupled it from crawljax to simplify debugging, given the prototypical nature of crawljax.

To run the analysis please unzip the compressed archive "/Security-MT-ICST-2020/Software/improveInputs.zip". The archive contains the jar smrl.jar (which is available also in the MT workspace) and a few configuration files. Please execute the following commands within the given uncompressed folder.

- To collect background URL requests, please execute
  java -cp smrl.jar smrl.mr.language.AugmentInput ./jenkinsSysConfig_withProxy.json

  This command simply replays all the requests in the source inputs and additionally monitors all the background requests using a proxy. The file "jenkinsSysConfig_withProxy.json" contains the SUT configuration parameters with the proxy's parameter (IP address, port, certificate and API key). Please note that you have to update the path of the source inputs to be augmented.

  To run this function, you should download and start the OWASP ZAP proxy[1]. Monitoring of background requests might be alternatively performed with hooks injected in the chromedriver used by crawljax, we relied on ZAP to simplify the implementation of the prototype.

  As output, the command above generates a new json file with the augmented inputs, which will be named "inputs_AUGMENTED.json".

- To collect the outputs generated by all the source inputs (i.e., to enable proper functioning of the Web-specific function "userCanRetrieveContent "), please execute

  java -cp smrl.jar smrl.mr.language.OutputDBFiller ./outputStore ./jenkinsSysConfig.json

  This command simply replays all the requests in the source inputs and record the output files. The parameter "./outputStore" indicates the path where to store all the outputs (this should be consistent with the parameter "outputStore" in the config file

[1] https://github.com/zaproxy/zaproxy/wiki/Downloads

"./jenkinsSysConfig.json"). Please note that you have to updated the path of the file inputs_AUGMENTED.json, if necessary.