

The infrastructure required to run the artifact is set up on a VirtualBox virtual machine (<https://www.virtualbox.org>). The VM can be downloaded from <http://people.cs.umass.edu/~brun/ICSE2019SwamiVM> and can be imported using the VirtualBox software as described below. If you are asked for a password to log into the VM, that password is **swami**

**NOTE:** VirtualBox can conflict with Docker or any other hypervisor tool that is running on the host machine. Before running VirtualBox, stop all other hypervisors.

**Steps to import the VM using VirtualBox software:**

1. Download VirtualBox from <https://www.virtualbox.org/>
2. Download the swami.ova file from the above url.  
Please note that this is a large file (8GB) and may take a long time to download.
3. Open VirtualBox
4. Click on **File -> Import Appliance**
5. In the import appliance wizard, select the **swami.ova** file downloaded in step 2.
6. Click **continue**
7. Click **import**. This step may take 2-3 minutes to execute.

To ensure that the artifact is set up properly, execute the following commands sequentially in the terminal:

```
cd swami/src
./reproduceRQ1.sh
```

The script reproduceRQ1.sh will take around 10 minutes to finish the execution and will print results to the terminal, as shown in Figure 1 below. If the output obtained by you matches the output shown in Figure 1, the installation package is successfully installed.

```

STEP1: Extract Relevant sections

Extracting relevant sections from: ../data/ECMA-262_v8.txt
begin extracting relevant sections .....
Extracting Header Progress: | 100.0% Complete
Extracting Relevant Specifications Progress: | 100.0% Complete
Writing Relevant Sections to the File Progress: | 100.0% Complete
Writing Relevant Sections to the File Progress: | 100.3% Complete
Total number of relevant sections extracted = 367
Output is available in: ../results//ECMA-262_v8_relevant_sections.txt

STEP2: Generate Templates for extracted Relevant sections Node.js

Reading relevant sections from existing file.....
attempting to generate templates for 367 relevant sections
Generating Test Templates Progress: | 100.0% Complete
Total number of test templates generated = 83
Generated templates are available in file: ../results//ecma262_templates.js

STEP3: Instantiate generated Templates for Node.js

Reading relevant sections from existing file.....
Generating Executable Tests Progress: | 100.0% Complete
Test files generated for Node.js are available in: ../results//Node_ECMA262_Tests
Total #tests generated: 83000

STEP4: Run and analyse generated Tests on Node.js

Total #tests generated: 83000
#Innocuous tests: 50086
#Non-Innocuous tests: 32914

#failing tests: 1533
Manual analysis of 1533 failing tests to identify false alarms reveals
that 998 tests of test-get-arraybuffer-prototype-bytelength.js test file are not
false alarms and expose a bug in the implementation of ArrayBuffer.byteLength.
The remaining 532 tests are false alarms caused because of the existing overloaded methods

#Good tests: 32379
#Bad tests: 535

Percent of Good tests (out of total): 39.01%
Percent of Bad tests (out of total): .64%
Percent of Innocuous tests (out of total): 60.34%

Percent of Good tests (out of non-innocuous): 98.37%
#####

```

Figure1: Output of the reproduceRQ1.sh script.



