Fuzz-Dojo - post survey

Your email address will not be recorded unless you provide it below. Please choose a unique but nonidentifying pseudonym for
pwn.college if you register an account for this project. De-identified data collected will be shared with others (e.g., investigators or industry partners) for future research or other uses.

Estimated survey length: 15 minutes

All questions of the survey other than your ID are optional. If you do not have an opinion about a particular question, please leave it blank.

* Indicates required question

1. What is your pwn.college pseudonym? (non-identifying unique name) *

Task Review

Please describe how you accomplished the following activities:

2. What Fuzz Dojo and OSS-Fuzz projects did you work on?

3. What functions in the project did you try before achieving success and why did you choose them? What fuzz driver files (ie: new_fuzzer.cc) did you modify?

4. 2. Creating/modifying fuzz driver code and debugging compiling issues

5. 3. Compiling the project and using different sanitizers

6. 4. Resolving fuzzing issues such as slow operations and low coverage

7. 5. Running the tools to check for improved code coverage

8. 6. How would you compare the usability of the Fuzz Dojo challenges compared to the OSS-Fuzz challenges?

9. Beyond compiling and checking for code coverage, what Fuzz Dojo features and tools did you find useful?

10. What actions did you take outside the platform? (for example: webpages, other tools)

11. Which features did you use?

Check all that apply.

□ "LOC" coverage calculators
□ Fuzz Introspector
□ Corpus
□ Dictionary
□ Address Sanitizer
□ Memory Sanitizer
□ Undefined Behavior Sanitizer
□ Thread Sanitizer

□ Other: ____________________

12. Did you find any crashes when using the platform?

Mark only one oval.

○ Yes

○ No

13. How would you rate the usability of the Fuzz Dojo?

Mark only one oval.

	1	2	3	4	5	
very	○	○	○	○	○	very good

14. How would you rate the training videos, links, and exercises?

Mark only one oval.

	1	2	3	4	5	
very	○	○	○	○	○	very good

Types of Difficulties Encountered

During this study only (do not reference your prior experience with fuzzing) how frequently did you encounter each type of challenge? 5-level answer ("Always" "Often" "Sometimes" "Rarely" "Never" and "I don't know")

15. Different fuzzing environments lead to unstable/unreliable results. When these machines have different environments, the consistency of the fuzzing output can be affected or disrupted. Examples: different Python versions and different Linux builds. (F1)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

16. Incompatibility between the fuzzing environment and the project leads to build failures. Compiler updates and versioning, specifically, can break fuzzing builds and require complex manual intervention from the developers to fix the problem. (F2)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

17. Bad fuzz drivers lead to fuzzing failure or inconsistent results. (F6)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

18. Issues with fuzzing test code lead to crashes/build failures. (F7)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

19. Bugs in the fuzzer lead to build failures or abnormal / inconsistent fuzzing behaviors. This includes fuzzers not being able to detect known positive cases, generating false positives, reporting the wrong type of failure, or simply crashing. (F8)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

20. Incorrect use of the corpus leads to fuzzing failures or low coverage. (F9)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

21. Issues with the build tools or external dependencies lead to crash/build failures. For example, docker images have grown too big over time. (F10)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

22. Issues in the corpus (such as unreadable or bad data) lead to build failures or unreliable results. (F11)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

23. Developers have difficulties generating the correct inputs or targeting specific parts of the software. For example, a fuzz driver achieving only shallow coverage or being unable to pass strict pre-condition checks. (F15)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

24. Developers have difficulties in setting up, building, or using the fuzzer. This includes breaking the build, not fuzzing the right fuzz target, causing the fuzzer to use too much memory, and many other issues. (F16)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

25. Documentation on how to use the fuzzer is missing/ insufficient. For example, finding information on how to utilize optional features of a fuzzing tool. (F17)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

26. Messages/information generated by fuzzers is missing / confusing/unhelpful. For example, crushing outputs are found but not reported. (F18)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

27. Bad code design limits the ability to fuzz. This includes insufficient segmentation/ separation of the target code, monolithic design, and assumptions about intended behavior, often requiring tests be done on the full system. (F20)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

28. Developers have difficulties writing/understanding and debugging fuzzing code. (F21)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

29. Developers have difficulty deciding which parts of the software to fuzz. (F22)

Mark only one oval.

○ 1 - Always

○ 2 - Often

○ 3 - Sometimes

○ 4 - Rarely

○ 5 - Never

○ I don't know

30. Have you encountered other challenges not mentioned in this survey?

31. Do you have any general suggestions for how the platform or the fuzzing process could be improved?

Google Forms