Discussion Part 1: Identifying internal stage compositions

Instrumentation

Participant

~~So for the instrumentation as a mentioned earlier, I would say it can have an impact to the data that you extract later because how a fuzzer goes about instrumenting the program can pretty much change: well, the data that you extract is of course based on the instrumentation that's there. So, I would say this is definitely related to the mutation and possibly also the execution because: what the point that I wanted to make there is it's simply that, OK, how the program is instrumented will determine, for example, how whatever data you extract, you extract from the from the fuzzing campaign would be represented. Because let's say one fuzzer goes through and instrumented, Jazzer for example it adds above 1000 of these identifiers to the program and then another program adds, I don't know, the half of it. The half and half it might be not a big deal, but then where in the program these the program is instrumented and so on can also make a difference: that's the thought that I have behind that.~~

~~And as I said before that I would put these internals initial seeds and stuff under mutation. So the fact that the instrumentation would affect the mutation then in this phase they would also be indirectly affected because, depending on how the program is instrumented then that would also impact the the seeds that are and the search strategy and so on., that is actually used to export the program.~~ (Stricken off because the participant later updates these steps after clarification)

So I would just say, it's just affects the mutations.

Interviewer

And not the other two (initial seeds and execution), OK, Yeah, because it just affected mutations. And then OK, we're done with instrumentation. Now we're within the mutation phase and then the mutation would then be affecting the other steps.

Initial Seeds

Participant

~~I mean, this is something more relevant for the mutation.~~
The search strategy, the power scheduler as well and also the execution.
~~Because see that the seeds that are generated, so the seed is pretty much gonna determine to an extent, OK, how this campaign will actually go through and then it affects the mutation.~~ (Stricken off because the participant later updates these steps after clarification)
And then by extension, how we actually search program and then when we execute it then whatever feedback that we get, it would then, this would also yeah, that's how I would say that one goes.

Participant

Well, it doesn't directly affect the mutation.

I mean, we're using it for the scheduling and more or less the schedule, the search strategy and in the end when we execute then, probably would not say it directly affects the: hold up directly, but it has the relationship with the execution for sure. OK.

Search Strategy

Participant

The search strategy, mutation, initial seeds, power schedule and execution as well. Because it's like, you're given a particular seed and the mutations that are done and how you should do all of that. So it's kind of related to all for those.

Power Schedule

Participant

Well, the execution for sure. So I'm just thinking, we have the power schedule this would impact execution because these are pretty much like you can say the jobs that are being done. But this would technically also affect all three of them: the mutation, the initial seeds and the search strategy.

Interviewer
Yeah, maybe justify them to mutations and to the initial seeds.

Participant
The initial seeds, OK, you have the seeds and then we know we just pretty much just schedule these jobs, OK, mutation should not be part of it.

Interviewer
So mutations are not part of it.

Participant
No, because the mutation. Yeah. So the power schedule: we're given a particular seed so well the seed actually impacts the the power schedule, but the other way around. But wait, does this relationship mean like it's affects the other phase or it's just that they for example they work together or somehow?

Interviewer
Both so working together is like affecting the other face.

Participant
So yeah, initial seeds for sure, because using these initial seeds then that's what we use to determine, OK, what we're gonna explore. The search strategy as well is also affected based on the the power schedule you and execution by default because it's it's in a loop.
So they will come again. The mutations: we get a different type so we have new data and then based on the seed gap.
So I would also say it's affected because yeah, given the.
What the assumption that we are in the mutation phase and then the power schedule is also part of it then yeah the relationship is there for sure.

Participant
Yeah, I mean, all of this is just based on my assumption that the steps (search strategy and power schedule) are part of the mutation and so of course there's a relationship with, them.

Mutations

Participant
The initial seeds, search strategy, power schedule and execution for sure. As I mentioned the mutation, assuming that these three are part of the mutation phase then of course there's a relationship to it, but also the instrumentation would also affect the mutation as well.

Interviewer
Instrumentation affects mutation. We have already done that. So, does mutations affect instrumentation? It's other way around. So no, you mean it should be just the one way, right?

Interviewer
Should we redo all the previous ones?

Participant
No instrumentation is done already, so, it's not affected.

Execution

Participant
Well, directly the mutation because given based on the feedback that we get, then, the mutation phase will be kickstarted and then indirectly these other phases will also be affected (search strategy and power schedule).

And yeah, it doesn't fit the instrumentation.

Interviewer
And all these three (mutations, search strategy and power schedule) are kind substages. So you say that affect, alright? So should we go through the other ones because you maybe would have answered because there are two relationships?

Discussion Part 2: Identifying visualization analysis tasks

Instrumentation

P2.1 AT. How the fuzzer instruments the code? Where is it instrumenting? Just for code coverage?

Participant

Not sure how this would be can be visualized, but I think it's important to make it clear how the fuzzer actually instruments the the code? So for example, where is it actually instrumenting the these identifiers that we will then use, for example, to try the code coverage? Is it for example just?

Because, like jazzer for example, it says critical points in the program, but sometimes I have the feeling like there's no specific rule to say that, OK, this is the reason that the identifiers that are here are not here, so it's I think it's an interesting and important to understand, OK, where are these identifiers added and what's the more listed general rule for how the program is instrumented?

P2.2 AT. How many mutations are generated per instrumented block of code?

Participant

For me, this relationship is interesting because OK, based on where how you instrument the program, the mutations that you're gonna generate are gonna be you're tying them to where the the instrumentation ID is in the program, so.

Participant

That relationship.

Yeah, that's where I extract that relationship, OK, this is where we instrumented the program, this is where we're doing some mutations and we need to tie this mutations to a particular block of the code. So you can get additional insights. The fuzzer was able to may be correctly generate the mutation, let's say to create a JSON file and yeah with.

P2.3 AT. How is the fuzzer mutating the information? These are the regular expressions that the fuzzer used.

How is the for the actually mutating the information to say so we can.
I mean, you don't listen to show all the things that are changed, but maybe even to say like somewhat at the high level, OK, this is the regular expressions that the further used to generate all of these mutations for example.

Interviewer
So do you mean how does each mutation phase perform when compared to other or?

Participant
No, it's not. It's not about the how it performs, but just the from me on exactly.

Interviewer
the history of mutation?

Participant
Exactly, so like for me it would be useful to understand OK at least as a developer for the fuzzer to say OK why did the fuzzer decide to go about it this way? And then if I can see, OK, what is it that the first there is generating or kind of like what type of approach taken to generate those mutations then that could possibly be helpful for him.

Interviewer
So is this what you were looking for? The lineage of each mutant. So how it was generated and at each step, what was the decisions that further took?

Participant
It doesn't really have to be at each step, it could also be summarized. If it's just text for example, maybe there's some way we could just show that OK, this is how the text actually mutated over time.

Interviewer
OK, I would move this to mutation, but I'll just let it be here for now.

P2.4 AT. Understanding how the program is instrumented? - Class level, function level... and has x% of instrumented identifiers - give information of complexity of the program

Participant

One last thing for the for the instrumentation would be to.

No idea how this can be done, but understanding where how the program is instrumented. If you want to, let's say, look at the the the program at the the class level or then you can say OK this class has, let's say, I don't know 50% of the instrumented identifiers. This could this could be also useful because then you would also gives some information about, let's say, maybe the complexity of the program, if we're instrumenting at these that these branching conditions, then we can see, OK, these are the classes of interest because yeah, that could that could be a useful metric I think.

Initial Seeds

P2.5 AT. How are the seeds generated? Is there a particular rule?
P2.6 AT. How does initial seed affect search strategy, power schedule, and execution? (In general, for all relationships)
P2.7 AT. What impact does a seed on other stages?

Participant

The generation of seeds is useful to figure that seed. How are the seeds generated? Are there some particular rules for seed generation?

How does the how does this affect, for example a search strategy or the scheduling that the fuzzer are actually does? But I guess this is also the question, I would say isrelevant for all the relationships that exist in general, given this particular data point, what relationship does it have on this thing to say? For example, we have a particular seed, what impact does this particular seed have on the scheduler, but I don't think it should have such an OK, maybe it does: what impact does it have there? I mean just capturing that relationship somewhere I'm not really sure how to go about that?

But yeah.

For yeah, for this one, I don't know the relationship I'm not too sure about what.

P2.8 AT. How does the energy vary from each initial seed to their particular mutants?

Interviewer
So for example, I can think about a question between initial seeds and execution. How does the energy vary from each initial seed to their particular mutants?

Participant
Yeah, exactly.

Search Strategy

P2.9 AT. What is the search strategy?

Participant
Well, I mean, I guess the first thing to ask is, OK, what is this search strategy?

P2.10 AT. What is the flow of the seed that lead of the interesting behaviors?

Yeah, just pretty much visualize in that somehow to show that, OK, this is how we go about the selected in the seats that are then used. I guess the this is like the common line, that's you'd have to try to show somehow where you have a particular: well, I mean, the fuzzer is gonna do this anyways, this is the particular value that led to this particular error but that would also be still be something useful to see that, irrespective of how you order the search strategy, the power schedule and the mutation, it would just be interesting to see the actual flow: we started with whatever value and how this value actually changed so that we could actually get to the the trigger in this particular error. So it'll be nice to kind of see the floor and the development there.

P2.11 AT. What effect does search strategy have in finding interesting behaviors in SUT?

Also, the relationship questions would be OK what's what effect does the search strategy have on the power schedule and?

Interviewer

Can you maybe think about a one level deeper? So basically, I'm asking you to think about effect right? So what effect I mean, what would you like to understand? Like for example and there's something like here, right? How does energy vary between initial sites and their mutants? Something like specific heuristic that you're interested in.

Participant
I would put it with the well, at least for the search strategy and the the the execution. But to be more specific, actually finding or not finding the like these interesting behaviors where what effect does the search strategy have on, for example, finding interesting behaviors in the in the program/ in the system under test.

Power Schedule

P2.12 AT. What approach is power schedule using? Any bias?

Participant
What algorithm or not necessarily algorithm, but what? What approach is the power scheduler using? Maybe understanding OK if there is a particular bias or?

P2.13 AT. How does it assign energy to a particular seed?

Some type of rule that is following to actually do this assignment. So yeah, it's pretty much getting some information on took it. What is the scheduling strategy that's used there now, and why does it decide to, for example, assign a certain amount of energy to a particular seed?

P2.14 AT. Relationship between energy assigned to the interesting behaviors?

So would be interesting to understand, OK, depending on the energy that's assigned to a particular seed, does this have an impact on the interesting behaviors that we find?

Mutations

P2.15 AT. How is the fuzzer mutating the information? These are the regular expressions that the fuzzer used.

How is the for the actually mutating the information to say so we can.
I mean, you don't listen to show all the things that are changed, but maybe even to say like somewhat at the high level, OK, this is the regular expressions that the further used to generate all of these mutations for example.

Interviewer
So do you mean how does each mutation phase perform when compared to other or?

Participant
No, it's not. It's not about the how it performs, but just the from me on exactly.

Interviewer
the history of mutation?

Participant
Exactly, so like for me it would be useful to understand OK at least as a developer for the fuzzer to say OK why did the fuzzer decide to go about it this way? And then if I can see, OK, what is it that the first there is generating or kind of like what type of approach taken to generate those mutations then that could possibly be helpful for him.

Interviewer
So is this what you were looking for? The lineage of each mutant. So how it was generated and at each step, what was the decisions that further took?

Participant
It doesn't really have to be at each step, it could also be summarized. If it's just text for example, maybe there's some way we could just show that OK, this is how the text actually mutated over time.

P2.16 AT. The lineage of each mutant and why did it choose a particular step in each step (summarized information) - dynamic slider

Participant
I think I also mentioned the understanding it's a high level.
What mutation approach is being used?

So let's say we're just working with some with with a string variable and to somehow. I just kind of think would be kind of cool to just have a slider and you see for example how the value is changing over the entire campaign. So it's a simple implementation I guess could be just see all the values that might be too much, but somehow even to summarize that, OK, this particular part changes, so maybe the first three characters are the same, so that's already there and then you visualize this somehow, OK, this is how the value changes. That could be probably helpful to understand how the fuzzer is going about mutating the values.

P2.17 AT. How long does it take for the mutator to switch between strategy? What made it change the strategy?

So you're using this high level representation of the data to also understand how quickly or how long does it take for the mutator to like maybe to switch its approach: so it has been switching using the same thing let's say for the last 10,000 trials and nothing happened. Like how long does it take before it's actually switched from? OK, let's throw this approach out of the way and try this.

So if something changes, like how long before it actually are the interesting changes in the mutation, like how long before it actually recognizes? And also maybe what could possibly cause you to change the approach?

Participant
For the mutation, then the search strategy. So given a particular mutant,
What strategy is what? What is the approach that is taken by this?
OK, I don't really know what should be the question there.

Interviewer

No problem. Maybe with the power schedule?

P2.18 AT. Reason for energy assignment to the mutants

Participant

So with all these different mutants that are created. You have a certain number of you have different mutants that you're working with, and then the power schedule is gonna determine, which one do we actually want to assign, as you said, this energy to? Even some insights into why or just a connection between, OK, you have these particular mutants and the poor schedule deciding this particular set of mutants or this particular mutant, should actually be given more energy in comparison to another one.

P2.19 AT. how did the fuzzer first of all get to this particular mutant maybe how long it took to get there and then from there it's like you can see, OK, what is the effect of the fuzzer getting to this mutant?

Interviewer

And about the execution. Mutation and execution.

Participant

So this is the what I mentioned also before where you have somehow you can visualize the entire process, OK, based on this mutant, this is how it actually led to this particular.

Interviewer

That's the first question I think.

Participant

Yeah, I would like to see the the particular exception being thrown, but I mean even there even there it could be you could take it to.

Participant

It doesn't have to be just OK, the fuzzer they also give you this information already, but when you're looking at it from a visual perspective, there's a lot of information that you could throw in there to say that, OK, you could add more information to you so that it's like, it's gives you a much better overview to see OK, how did the fuzzer first of all get to this particular mutant maybe how long it took to get there and then from there it's like you can see, OK, what is the effect of the fuzzer getting to this mutant? And then how does it like play out into actually generating the to throwing that exception at some point?

Interviewer
OK. Yeah, I think that's a lot of information. I will look again into the transcript and add that.

Execution

P2.20 AT. How many executions? Duration of executions?
P2.21 AT. How much execution improve the coverage?

Participant
Just basic questions are coming to my mind and I mean you have those already in terms of like how many executions and? the length of execution? Maybe something that interesting is to figure out the the progress of the of the execution: how much does an execution improve let's say, the code coverage or whatever coverage that we're using.
P2.22 AT. How does execution information help mutation?

Participant
So whatever information that we we gather from the execution: how does this information actually help us to improve, let's say, the mutation process and then by default the following processes (other stages)?

P2.23 AT. Pre-post execution relationship between the stages. What is recorded and how much of it is used in each execution loop?

So is this a cause and effect relationship that exists between these different stages, because this is also critical for this loop.
So first of all, understanding what causes, what led us to this particular execution,

and then after the execution is finished then also what are we using as a result of this execution to help us with the following execution and how does that actually impact the process? Does it improve it or does it make it worse? That type of thing!