Discussion Part 1: Identifying internal stage compositions

Instrumentation

Participant
OK, so first of all, let's say instrumentation. What it does, right? So we try to inject some code and we use that code to collect some information later, correct?

For example, if it's code coverage, then usually what we do is we add some kind of markers in the code and every time we execute something with some random input, we see how many markers have been executed and then accordingly we can collect the code coverage, right?

So considering that one, how does it relate to, for example, initial seeds?

Yeah, I would say for the initial seeds, the relationship is always the entry point, right? So in the program, we always have this entry points and using initial seeds, we can actually determine which entry points have been executed right. So I would say one is the entry points.

Interviewer
So you're talking about instrumentation or initial seeds?

Participant
The relationship between instrumentation and the initial seeds, right? So we have this instrumentation right? Let's say we inject all the markers for the code coverage, right? And then in the program there are multiple entry points, right?

Now, let's say that we took one seed and when we executed we found out that with that seed the entry 1 got executed. So we can clearly say that, OK, this seed is always goes to this entry 1. And then we take another seed and we found out that, oh, OK, the entry .2. So I would say how this instrumentation relates to initial seeds is that for example, this entry point, we can decide this entry point based on the seed.

Interviewer
Does instrumentation affect initial seeds? I mean, that's the important question that we have to ask here, right?

Participant
Affect. No, no, I would say no.

Interviewer
And then I would more I'll be more interested in knowing if instrumentation is affecting other stages.

Participant
What else? Oh, definitely search strategy, right? Depending on what we inject or what we instrument and what data that we collect, we can improve the search strategy. So that is one thing.

I would also say it also affects the power schedule because if you have a bad instrumentation and for example, let's say, we instrument try to identify the power schedule. Let's say there is a for loop and we collect some kind of data from the instrumentation and if we collect the data wrongly, we might execute more number of times and we might spend more power. So I would also say instrumentation also affects the power schedule depending on what we do with the instrumentation.

Mutation? Oh, definitely. Mutation also gets affected, right? Because, let's say uh, we code coverage says that 60% and we found out that one particular seed is covering more code, we tend to spend more in that mutant, for example, so instrumentation also affects mutation.

Does it affect execution? I don't think so because we execute and we collect data, but by instrumenting OK?

Interviewer
The more you instrument, the more time it might take.

Participant
Yeah, that's what I was about to say. Conceptually, it should not affect right. Instrumentation should not affect the execution conceptually, but, practically, yes, it does affect right. For example, it might take more time because we instrument a lot of things and also there can be some bugs, right? Have experienced this! For example you instrument something and then, the instrumentation framework had some kind of problem. We could not instrument or you instrument then bytecode got corrupted and then when you try to run you get a completely different exception or error which is not at all related to input or the application but it is related to the instrumented code. So I would say practically yes, but with my knowledge I would say conceptually no I guess.

Initial Seeds

Participant
Does initial seeds affects the instrumentation? OK, I need to think in different ways here. So when we take an initial seeds, do we need to instrument differently based on the seeds? I would say no. I think initial seeds does not affect instrumentation.

Does initial seeds affect the search strategy? Depending on the initial seeds, the output of the search strategy will affect. But does it affect the search strategy algorithm itself? Which one are you interested in the first one or the second one?

Interviewer
I mean you can answer both.

Participant
Depending on the seed that you pick will definitely affect the output of the search strategy. If you have really bad initial seeds it doesn’t matter how good your search strategy is your output of the search strategy is going to be bad. In terms of output, yes, but in terms of the algorithm itself, probably yeah, because, if you have some random input and if you have algorithm that, let's say, splicing!

Interviewer
Splicing is mutation.

Participant
Oh, sorry, sorry. Splicing is a mutation, right?

Can you give me one example of search strategy?

Interviewer
First, in, first out, last in, first out.

Participant
With the knowledge I have, it's hard to answer this question, but what I think is it does affect. Let's take an example. Let's say we have you have a 10 initial seeds, OK, And the chances of having a more a very good mutated input from the first initial seed is, let's say 90%, and then the second one is like 80% and then it keeps decreasing right? It will affect the search strategy also, because with the first itself you get lot of input then you are going to add lot of seeds to the your seed pool and then it keep decreasing your search strategy. Also like you know changes but if you have the other way round now let's say you have, the first one is like 20% chances of having a very good mutated input and then it keeps increasing, first time your search strategy, for example first and first out, it's gonna take a lot of time to go to the last seed, which gives more chances. So yeah, I would say search the initial seeds does affect the search strategy.

Interviewer
OK, how about power schedule?

Participant
It's tricky to answer this question with the same example that I explained just now. It does affect the power schedule, but is it because of just initial seed? Not necessarily right? Because it's combination with the search strategy and initial seed. As I said, the first seed gives you 90% and you have a first in first out with very less power. You have already a very good input. But the other way around… I would say still it affects the power schedule.

And then we have the mutation.

Yes it does. For example, let's take SQL injection. You have a initial seeds which imitate the read and the delete operation of the SQL injection right? Whichever the mutation algorithm that you take, depending on the mutation algorithm, you can improve the input domain now, right? If you have something like just, I don't know. Let's say splicing, it might not be good for SQL injection because your initial seed does not have the insert operation for example, even if you take the grammar based one there also you don't it doesn't work because you just have for delete and read, but you do not have for insert. I don't know. Maybe there is some kind of algorithm which randomly picks many things that might work here. So it also effects there.

Does initial seeds affects the execution? Yes it does, because depending on what you're trying to achieve and if your initial seed is already close to the goal that you're trying to achieve, then your execution is very faster.

Interviewer
So it depends on the goal what you set.

Participant
Yes, depends on the goal.

Interviewer
What if conceptually, if I ask the same question.

Participant
I would still say yes because as I said, right, so depending on the initial seed, whatever you're trying to achieve, if it's far then it's going to take more time. That means you're going to execute a lot of time or lot of iteration. If it's already close to the goal that we are trying to achieve or whatever we are trying to achieve then we are going to execute very quickly and it's going to stop the fuzzer very quickly.

Search Strategy

Participant
OK does search strategy affects the instrumentation? The way depending on how you pick the input, that is the first strategy, does it affect? No, it doesn't.

Does it affect the initial seeds? No. Nope, no. Doesn't make sense that it affects.

So the power schedule. Yeah, definitely. Right. For some scenarios we might execute faster or some scenario we might take more energy, right? So it does affect definitely.

And search strategy, does it affect mutation? So let's take an example. Let's say we have first in, first out. And we have, I don't know, splicing for example or randomly delete some character for example. Depending on what we pick, does it affect mutation? It does not. Nope. At least for me it does not make sense.

What about execution? Yes, it does. Depending on the search strategy, it does affect the execution, yes.

Interviewer
In what way?

Participant
For some input seeds and you pick some search strategy, it might have less number of iteration of execution. For example first in first out might give you in some scenario less number of iteration of execution. In some scenario it might give you more number of iteration of execution.

Interviewer
Number of iterations is taken care of a power schedule!

Participant
But the power schedule depends on the input, right? The initial seeds.

Interviewer
You're saying the search strategy also effects execution because, there might be multiple executions of the selected seed. And if we just consider direct relationship from search strategy and execution, the number of iterations is actually not taken by care by search strategy. It is taken care by the power schedule.

Participant
So direct, then I would say no. Indirectly, yes, but yeah, directly I would say no then.

Power Schedule

Participant
So definitely the execution as we already discussed.

Does power schedule affects the mutation? So let's say what power schedule the power schedule will decide how many number of iteration of execution right or the energy that we want to spend for this particular seed?

Let me tell you what I'm thinking. You decide. Is it correct or not so. For example, if power schedule says that hey 100 is the energy that you can spend for this seed. And 100 is actually not enough, let's say then mutation might not give the desired result that we want. If power schedule gives less than, it's not enough. If the power schedule gives more, then mutation is keep on trying to mutate, which might be useless. So in this context, Yeah, I would say, then power schedule affects the mutation.

So power schedule, right, so does the power schedule affects the search strategy like depending on how much of iteration that you give, does it really affect the what strategy that you pick? No, I don't think so. And also the initial seeds for example.

Interviewer
So just to just to clarify, search strategy is just one thing, there are no multiple strategies. There is one strategy based on some heuristic. So it the strategy can be the mutant that gives me more coverage, I select that, or the mutant that gives me less coverage, or a mutant that covers more function, more basic blocks that could also be a search strategy so. More often it is in a fuzzer there'll be just one kind of search strategy.

Participant
It uses 1 heuristic OK. What power schedule does? Does it affect the heuristic or the search strategy that are currently being implemented in the fuzzer A for example?

Interviewer
Yeah, it both are in the same fuzzer for power, schedule and search strategy, so.

Participant
And I mean, I don't see the other way around because by picking what power schedule does, I don't think so. It affects the search strategy other way around, yes, by picking search strategy, it affects the power schedule. But by picking what or what power schedule does, I don't think so It affects the search strategy.

And also the initial seeds. I don't think so. Again, the other way around, yes, but not the.

This way power schedule to initial sets.

And instrumentation, no, I don't think so. Again, other way around, yes.

Mutations

Let's move to mutations.

0:42:57.610 --> 0:43:2.850
So for the mutation, yes, definitely execution, right. The way that the heuristics or the algorithm that you pick it affects the execution. Definitely because if one particular algorithm can give you a faster than less execution right.

Does mutation affect the power schedule? It's trying to use some algorithm again. I really don't think so. It affects anything the top. Even the search strategy. Initial seeds. Yeah, I don't think so. At least with my knowledge, I don't think so.

Execution

OK. Does execution affect the instrumentation? I do think, for some reason this I do think that execution does affect the instrumentation, actually. Because as I said, the way the execution goes, we might need to change the instrumentation. This is mainly because, as I said earlier, the instrumentation affects the execution, right? Because the practically, when you think about it, depending on the technology that we use, the instrumentation framework that we use, there might be some bug in the instrumentation framework which gives some exception in order to keep the instrumentation continue, I would say we need to re-instrument and so that we remove that particular marker or the injected code and try to execute. So I would say execution affects the instrumentation.

Conceptually, execution does effects initial seeds, right? OK, let me, let me explain this. So depending on the output of the execution, we got something really good input. We tried to add it to the seed pool. Do you consider that as affecting the initial sets? No, right?

Interviewer
No initial seeds starts before.

Participant
So yeah, so whatever happens later, we do not consider that as the initial seeds for the next mutation, right? Then I would say no. It does not affect the initial seeds.

Does it affect the search strategy. No. Yep.

Interviewer
OK, how about power schedule and mutations?

Participant
I would still say no.

Depends depending on the algorithm, like if you if you want to make a very good power schedule algorithm. Depending on how execution happen for this seed, we can change the power schedule something like interactive or something like adaptive for example.

Let's say for this initial seed with this power schedule, we had less number of whatever that iteration is. Then we can say that, OK, let's keep the power schedule as it is and we found out that for this initial seed it was not working. This is this is a different concept like adaptive. If you consider that, then we can keep changing, yes.

Mutation, I don't know, I don't think so.

Discussion Part 2: Identifying visualization analysis tasks

Instrumentation

P1.1 AT. What percentage of instrumented code is syntactically correct?

P1.2 AT. Where is the failed instrumented code?

Participant
So the first thing that comes to my mind is for the instrumentation is number of code statements that are successfully instrumented and executed.

Interviewer
What do you mean by successfully executed?

Participant
Yeah, this is again something to do with what I explained earlier, right? So when I tried instrumentation before. And let's say I instrumented some 100 lines or 100 lines of code and I found out that in some particular cases the one particular code statement that I instrumented failed when I run, when it reaches there it fails for some reason. Maybe it is because of the framework I used before. I can give you an example. So I worked on Java right. I tried Java assist and the ASM. And I know that in some scenarios when I instrument I get a weird errors like verification error for example or things like this and when you instrument something, if there is some kind of compilation error then you know you will also get error. So it could help me to understand the number of lines of code that I instrumented, how much is it actually successfully correct statement that I instrumented and it is actually executable. Or if it fails somewhere, I would like to know OK where it actually failed.

Did you get it?

Interviewer
Half actually. I mean if you can put it into the sentence, that would be great.

Participant
OK.

What percentage of instrumented code is syntactically correct and what percentage of in code is Executed or failed. Does it make sense? Like what percentage of instrumented code, this actually just gives me on a higher level, right? Like, OK, 90% of them are syntactically correct, 10% of them are somewhat syntactically wrong somewhere. I don't why! Or 90% of them successfully executed and 10% of them failed. That's good. But the deeper one would be where actually.

Where is the failed Instrumented code?

P1.3 AT. Can search strategy bypass the failed instrumented code?

Participant
For the search strategy, I said right. So in by the way you instrument, it affects the search strategy because. So let's say you are concentrating on code coverage. And I got to know that 10% of them actually failed. And now for the search strategy, the way you pick the input seeds, let's say you picked one input seed and then we got to know that it is reaching that particular failed code statement. Probably we can we change the search strategy in between?

Interviewer
I don't think so.

Participant
But anyway it does affect, right, because we know that, OK, this seed is not working and if it's still big that search strategy and it search strategies keeping same way, then we are probably wasting our time because we know that OK, this part of the instrumented code is already failed and we cannot use that information to compute code coverage so that is one thing and for the power schedule. I don't. I can't think of any. What kind of questions we can ask.

Interviewer
That's still fine.

P1.4 AT. What information we need to collect dynamically to improve the mutation? How much of that data is successfully extracted?

Participant
So in order to have a good mutation, what data that we want to collect dynamically, that is from the grey box fuzzing and how much of the data is actually successfully extracted in order to improve the mutation? For example, let's say code coverage and for the code coverage we take, what is the

current code coverage that we have and accordingly we try to mutate. Now, as I said earlier, we failed to instrument some part of the code, right. Like let's say 10% of them. That means we already know that the maximum code cover coverage that we can have is the 90. But let's say our mutation algorithm is trying to achieve that 95% of them. How much of this data that we want to implementation is actually successfully extracted that percentage would help me understand that, OK, is it good for mutation or not?

Initial Seeds

P1.5 AT. Percentage of similarities between the initial seeds.

Percentage of similarities between the initial seeds. And the reason because, OK, let's say we have 10 initial seeds and we have initial seed one and initial seed two. OK. And it has a similarities of 90%? Right now, now the question is, do we really need to take the initial seed two? Or should I go to the next initial seed? It's empirical study like by running and seeing. Should we pick the most similar one? So should we pick the less similar one? But I think that this answers the question, does it affect the search strategy or not like percentage of similarities between the initial seeds?

P1.6 AT. What percentage of assigned power is used for that particular initial seed?

And for the power schedule I would say, What percentage of assigned power is used but that particular initial seed? And with the combination of the 1st and the 2nd, we can also decide something here, right? Let's say we have two initial seeds with the 90% similarities and when we use the first one, we found out that, it used like I don't know 50% of the assigned power. That means we already know that the second one is also fifty. It's going to be close, like around 50-ish. So we can we can kind of tell that, OK, which next coming seed is going to take how much and we can decide maybe to increase or to decrease. So this might help there.

P1.7 AT. How many of the mutants are from the initial seed and how much of the coverage?

If you have a very good initial seeds, our mutation can help in getting better inputs are not. What question I need here?

This might be, yeah.

Interviewer
Something like what I had before or what I thought here was.

How does the mutations for initial set of seeds look like how many of them were generated?

Participant
That's it. That's actually good question.

Interviewer
And how many of these mutants from the initial seeds and what is the coverage or what? How does they perform in execution?

Participant
That's a good question.

The second question actually kind also answers the execution right. What percentage of power is used for the particular initial seed is actually indirectly even to the execution. It gives me that OK 50% or 40%. So I would say it's the same question or at least that one question for sure both execution and power schedule.

Search Strategy

P1.8 AT. How is the graph look like for the distribution of the power from the selected seed?

It's actually kind of instead of question, it's actually kind of how is the graph look like for the distribution of the power from the selected seed, know what I mean like so. Let's say we have a 10 seeds. Our search strategy took the second seed as first and I want to see that OK. This is the first seed that was picked and this is the amount of power assigned and this is the second seed and this is the amount third seed I want to see how that varies.

Interviewer
And what do you plan to infer from this variation?

Participant
So let's say that we found out this particular strategy for the goal whatever we are trying to achieve, every time we use this search strategy, we found out that OK, initially it assigns more energy and then we find out that after we go towards the end of the seeds, OK, it gives the less energy, for example and in return, we also found that the less energy those inputs are, the one kind of giving me more input, then probably we should not pick that search strategy. Maybe we should take some other search strategy to find and understand which actually gives the better power schedule by looking at how this distribution of the power depending on the seed that we pick.

Power Schedule

P1.9 AT. What percentage of assigned energy is not being used? on higher level: average of all the selected seeds would be nice, but for the deeper insights we need percentage of energy not being used for each of the selected seed.

Participant

The task of the power schedule is to identify the amount of power that needs to be given to this particular selected seed. It is actually kind of similar question that I asked before for one of the stage. What percentage of assigned energy is not being used. The reason because, let's say let's say we have 10 seeds, right? 10 times we selected a seat and for the first one 20% of the given energy is not being

used second time 30, 40, 50, 60 and we take all this percentage and we average it out and we found out that on an average like 30% of the energy being is not used. Then probably next time we need to decrease the power that we provide so that we can improve. So what percentage of energy is not being used.

On a higher level I would say, average of all the selected seeds would be nice. But the deeper insights we need percentage of energy not being used, but each of the selected seed. Like because by looking at the seed then we can maybe we can infer something, OK, that particular seed, it's not so good to for the energy or something like that, you know.

Mutations

P1.10 AT. How many mutations did it take for a selected seed to be a valid ones?

Participant
Yeah, this is very interesting actually for me, because I tried once a kind of grammar based right for the SQL injection and but before I tried grammar based I was trying a random. You know, pick some random characters and try to change it to different characters. I don't know what this algorithm is, maybe somebody came up with something. Yeah, but what I realized is that it's really, really horrible, because for the SQL injection, we already know the pattern, right? Pattern is like OK, we are select and we are start from and then some something so. That took lot of execution in order to achieve what I want to achieve.

But when I selected the a grammar based I gave my first saying that hey, OK, the pattern is like this select and then we have some kind of you know regular expression and then we have from which is fixed and then we have again some kind of regular expression and then I have something like a set of characters which you can use it to create combination like and or. I found out that the output I got the output very quickly like really quickly than the really hacky way of my implementation.

So thinking this, I think the question that I would like to ask is, how fast we achieved for the selected seed for the given mutation algorithm. Does that make sense?

Interviewer
Do you mean, how many mutations that it took for the seed to achieve to be a valid ones?

Participant
Yes, exactly.

Interviewer
And then maybe I will just rephrase that.

Participant
The idea is that how fast it could achieve? Like, you know, if we are trying to get some exception. Let's take this example. So we have my hacky way of implementation right where I took some random character and then I replace it with something. So first iteration I replaced something, nothing happened, second iteration nothing happened, third iteration nothing happened. fourth iteration I got the proper input so it took 4 iterations!

OK, but when I use this grammar based, let's say it took only two iteration. So that is exactly what I want to look. To see that graph you know like, OK, this algorithm for this seed, it took this much. And for this algorithm for this seed it took this much. And we can correlate and see that, OK, probably for this kind of seeds, this particular algorithm is better than the other one.

And the reason because it also depends on what we achieve right? For the SQL injection as I said, grammar based might work, but let's say I'm trying to check the reliability of some command for example, I would say the first one is far better because it creates lots of random generation. We might achieve very faster rather than looking into some pattern.

Interviewer
The first question that it's written there is it just mutation or it is just for both mutation and execution relationship.

Participant
I think it is the mutation and the relationship, right, because depending on the number of mutation, we perform the number of executions. This question answers me that OK for this particular set with this particular algorithm we had 10 mutation. That means we also executed 10 times, right?

For me, when you say mutation, I would consider both, but if you want user to know, I would explicitly specify that is mutation/execution.

Interviewer
OK, that's a good suggestion. I will take that into account now.

Execution

P1.11 P1AT. What percentage of instrumented code is executed/failed?

Participant
So actually some of the questions that I asked, I might also ask here. For example, what percent actually the good thing, what we can do is in the first stage we can keep: what percentage of instrumented code is syntactically correct in the instrumentation stage?

And we can take the second question, what percentage of instrumented code is executed or failed in the execution stage?

Because that's where we get to know, right? It's not in the instrumentation stage. We get to know during the execution stage. So let me take that and put it there.

Interviewer
OK, no problem and execution and instrumentation. You already said something just with execution. Just with execution, do you want to know something?

Participant
Actually I'm not.

This affecting, I'm not sure because if I remember correctly I said right, for the affecting, if it is something like adaptive then we can decide for the next iteration. Oh, OK. Should I give more power or should I give less power because we got to know that only 50% of the assigned power is being used.

Interviewer
You would know that indirectly. The power schedule might not know from which initial seed this mutant is from, but it will know how much of the interesting behavior it did trigger.

Participant
That's the thing this the top stages for some reason since maybe I don't have much experience I'm not able to see how it affects the higher stages, so I wouldn't.

Interviewer
That's OK K so, but this is just for the execution part, right?

Participant
Yes, yes.