P = Participant

I = Interviewer

I

So very quickly before the actual interview: about your background, could you first of all tell me your job description?

P

Okay. I'm a security analyst and manager at [company].

I

Okay, so how many years of security analysis experience do you have?

P

Yeah, it will be four years.

I

Okay.

P

So previously I was working on sensor networks and no security.

I

All right. So any software development experience all in all? How long is that?

P

Yes, yes, yes. Well, how long? My previous job, I spent seven years and one year before. Yeah, maybe I would say I wouldn't consider myself as a developer at [company].

I

Okay.

P

That's not primarily what I do. I do security evaluations and project management. And occasionally, I can write some code as well. But I would say that the developer background is about, let's say, six years, because at my previous job, I was more of a researcher, not an explicit developer, I was a researcher.

I

Okay. But still, that's some experience, right?

P

With software development, not just proof-of-concept but production-ready code.

I

Okay. As you know, just some background information to report how expert you are. All right. Let's have a look at the...

P

But I need to state that I was not developing Java code.

I

Yes, okay.

P

And I was not familiar with the Spring framework. Basically I got familiar with Spring during what we did with the Piggy Metrics project.

I

Ah, okay. So from a security analysis aspect, also no Java at all or just no Spring at all?

P

Well, a little Java. I mean, of course at the university, I use Java and made works with Java, but during my work, I didn't use Java for development. In the security realm, I looked at Java code and (the animations)? for Java code for Android applications. And yeah, that was it.

I

Okay, thanks. So you had two tasks where you did the manual analysis first and then two tasks where you had the tool output. As a first general question, was one of the ways easier than the other one?

P

Well, of course with less background in Spring, it's easier to start with something instead of doing it manually. So I would say of course it was easier to start with the tool output.

I

Okay.

P

It gave me useful pointers, which I can start looking at.

I

Okay. So let's say if we talk about speed or efficiency, it was quicker to get started?

P

Yes, yes, yes.

I

Okay. So then how did you use the tool output that I gave you? What exactly did you do with it?

P

Well, in the evaluations when I got the tool output first -I would speak about that scenario- I, of course, went through the traceability links that showed me -- of course, first I made sense of the rule itself, what was exactly the rule which was used. And it was, of course, in line with the description of the task. So, no surprises there. But you know, I always check for consistency, these kind of things. And then I went through what endpoints or nodes have the tool selected and what was the verdict for each. I mean, was it included or was it not, in the part of the rule? I mean, was it used for input or was it disregarded? And then took a look at the actual references to the code. So the traceability information. I clicked on the links and opened the files and it was very good that it was line level information, not just (?) level information.

I

Yeah.

P

And yeah, in most of the cases, it was very straightforward. What I ... . In some cases, I think it was basically because of my lack of knowledge in spring, that it led to a bit of confusion that for one particular case, I had two different outputs for the same question, basically, or two different traceability information for the same question. But I have to look it up. I wrote it down.

I

Okay. Did you write it in the answer sheet?

P

Yes, yes, yes, yes, yes.

I

Ah okay, then I can see there.

P

I will check in the meantime. Okay. Yeah, so authorization is attributed to “@EnableResourceServer” annotation. In task one, the tool output attributed is to “@EnableOAuth2SSO”. Yeah, so that's- -

I

Okay, I see.

P

For the same annotation, I got different traceability.

I

I see.

P

It's confusing for me, but maybe it's okay because Spring has multiple ways to do the same thing. So it might be absolutely correct. I'm not sure.

I

Well, but it's good to know. I mean, as you can see, it's the OAuth framework behind it. And of course there are multiple components and they all do authorization, but since they're different parts of the process, they have different annotations. But that's good to know.

Yeah, so you already said you used the traceability. Would you say that it was sufficient to support the annotations or the evidence?

P

Yes, yes, yes, yes, I really like that feature. I think it's essential to be able to trust in the user. We have to show how that, where it comes from. So what the decision comes from, we have to show traceability.

I

Okay. So you mentioned trust. In general, how much did you trust the tool output that I gave you?

P

Yeah, so I would say because of the lack of expertise in Spring, I would, you know -- one to five scale, I would say, probably three and a half / four, something similar to what I would trust, how I would trust, for example, a StackOverflow answer, meaning that I'm not an expert, someone says this, okay, maybe true. But okay, yeah, I knew how the tool works and I know how -- I would say I would trust the tool. Maybe I would question the model in some cases.

I

Yeah, right.

P

So it's the overall score I gave, not just for the tool. It's 3.5, 4. It's not for the tool, but the overall end-to-end working. So with the model, I would say.

I

Yeah, all right. Did it change over the progress of the session? So you did four tasks, right? Did you trust it more in the end?

P

Yeah, sure. At the first time, it's "OK, what can I see here?". I saw that in some cases no traceability was provided. We discussed this that this would happen previously before the experiment. So I knew that it's OK. It will be nice if you are able to make progress in this regard. So this feature can be extended for including more and more traceability. I would say that's very, very beneficial for security evaluators. And in general, I would say that I would make the DFD analysis as part of upcoming evaluations in microservice architecture kind of evaluations, because it's very straightforward to use and can be deduced semi-automatically. So I would say that this tool gave me the idea or the enthusiasm to build this into our evaluation workflow.

I

Okay. Just to understand again, you're saying that the whole process with DFD and rule checking, the whole thing you would include in your work or...

P

Yes, I think so.

I

Okay.

P

It shows potential for a level that I would say it's an efficient thing to do. So the speed up with using the model or using the tool is worth it. I mean, and the capability of automating it is also very, you know, promising or has potential.

I

So then, um, you said you give it a 3.5 five or something in trust -

P

Yes what's basically because of these issues of how some annotations or how some traceability information differed in some cases and maybe the fact that I was not aware what should be considered as infrastructure component, external component or internal. If I knew that better, I would trust it more probably.

I

So I wanted to know about any suggestions that you have for improving trust in the tool. So one is seemingly training, better understanding. Anything else? Is it just knowing it better or is there anything else we can improve? Do you have any idea?

P

Maybe, I would say that the trust in the tool is significantly bigger than the trust in the model.

I

Okay.

P

So if we somehow are able to, I would not say verify the model, but something like that. I mean, some kind of trust for the model would benefit the whole process. Is it possible to create models automatically or is it always a human assessment and is it completely human or are you able to automate it, semi-automate it? Yeah, this is, sorry, this is just my curiosity.

I

Okay, I see. Yeah, so we [anonymized because it reveals authors, no relevant information contained].

P

Great

I

Okay, but I understand. So it's more on the model side where the issues lie.

P

Yeah, I would say.

I

All right.

P

Yeah, the tool is capable of creating the rules necessary for showing or creating the outputs which are necessary to prove the rules or make verdict on the rules. I would say that logical combination and filtering is what we need here to make it generic and you're able to express basically arbitrary rules in this setting. So it was nice to see that directions are also handled. So in the graph of the DFD, you have directed edges. So that was good to see that these directions are understood and used by the tool.

I

Okay, all right. So we talked about the traceability quite a lot now. About the evidence: you know the difference, right? So basically if you just think about what I gave you, the table, the list, without the links to GitHub. What did you think about that? The structure with the step-by-step… was everything in there that you needed to understand? Did you trust this one, this listing of nodes and so on? Anything you could tell me about that?

P

Maybe it would -

I

So, -sorry- first of all: of course, this is just a crappy HTML file. Let's assume we have a better UI, it looks a bit nicer and so on, but in general, the idea of presenting it.

P

The information presented.

I

Yeah.

P

I would say that if I do a security evaluation, even before taking a look at the rules themselves, it would be interesting to see only the nodes and the edges. So basically the DFD, but so that if the tool is able to list all the nodes and mark them, which are infrastructure nodes, which are external, which are services? Just as is, I mean, without filtering or without any operations on them, that's even giving me the sense that the tool is able to cover the whole source code or the whole model, basically, so that everything is considered, that there is nothing left out from the scope. Just to make sure that in the following steps, when I apply the rules, I can trust the tool that it considered all the components of the TOE. So the Target Of Evaluation. And that would, I think, give me even more trust to the tool, if I'm able to extract this information. So I assume that it's possible to query all the nodes from the tool. We didn't use the tool, just the tool outputs. But that is something that I would do if I get access to the tool and, okay just “list me all the nodes” and I can make sure that all the components that I thought of are covered by the tool and then I trust that, okay, the picture I see is complete and the reasoning I would say is based on the whole project and not just maybe parts of it if something would be omitted.

I

Okay, good to know. But in that case, would you prefer the actual DFD, the graphical visualization, or also something like this table -- a list?

P

I would say both are relevant and useful for me. The DFD would show me the interrelationships or the connections as well, that's also useful. But I would say, a list is also useful to ensure the completeness or the exhaustivity of the...

I

I see, okay.

P

...of the analysis. And probably if I create a report, I would list all the components anyway. So that's a good... or useful output also from the tool, if I can extract all the components.

I

Okay, I see. If you create such a report, do you usually do something like the traceability? So giving evidence to why you list these nodes, for example?

P

Yes, I would. Yeah, I would mention that, okay, because everything is listed in -I don't know- Docker Compose or everything. Or every class annotation with, I don't know, whatever class. And these are the things which match this pattern. And then I list those.

I

OK. And just going back: for such reports, do you think the traceability that we provide would also be useful or enough?

P

Yes, yes, yes.

I

Okay, nice. That's good.

P

Traceability would be enough and some reasoning can be added. Or even generated based on the rule. I would say that if you have this traceability, you are able to generate a few sentences with the knowledge of the tool. You already know what to say. It's even possible to generate parts of the report. Basically, for the rules that you are able to measure or validate, then you are able to generate basically the whole report based on this information for those rules.

I

Yeah, okay, all right. So my final question field basically is if you would use this in your job and how. We've touched on this already, could you maybe -- just as a summary: how could you imagine using the tool in your job, if at all, and if so, how specifically? For what tasks and… yeah, maybe just talk a bit about what you find useful.

P

I would definitely use it as an automated toolset. I would love to generate a DFD and then run intelligent rule checking on it. The DFD is -- I would not say it's not useful as is, but with rule checking, that's much, much better.

I

Okay.

P

So I definitely see value in the tool beyond the DFD. I mean, the tool itself is valuable. And if I'm able to automatically generate models, which can be input for the tool, then I would definitely use this as part of the assessment or the evaluation process. I will introduce this tool to the microservices evaluations if we do something like that.

I

Okay. And more like you did now as a primer, as a starting point that you then check yourself afterwards or would you do it manually first...

P

Yes

I

Okay.

P

Yes, probably with some random testing of validity of the model, I would say. So I would list all the nodes with the tool and check whether all the components are there, which I would see as a component. Maybe I would go through their classification. Are they external? Are they infrastructure nodes or... what type? And if I think that covers the whole source code, then I would trust the model or trust the tool to obtain information from the model.

I

All right. The tasks that you did -- if I give an example, the "all entry points have a load balancer" -- that's a rather simple thing to check, right? The one with "do the edges between external and service, are they encrypted?", that's already quite more complex. Do you think the tool is better suited for the one or the other? Or is it the same? Does it help the complex ones more?

P

I would say if... so once you have basically the evidence, then you can prove both, more complex rules as well. So I think not just easy tasks can be performed and the question is maybe for -- I would say that it's not just the problem of the model but the problem of the tool how to handle things that are missing.

I

Yes.

P

So for example, no TLS, no certificates are found. How do you treat this? So you already have to know exactly what you're looking for. Is it there in the model? Maybe not. The DFD I would say not deals with such level of information. So maybe the tool can be extended with other information helping to reason the verdict and include maybe other type of evidence like even if it's as simple as a grep. Things that would be useful for deciding the rule and not obtainable from the model would be very interesting to be incorporated into the tool.

I

Do you have an example?

P

Yeah, I think that's a good example to mention that there is no certificate in the model, or I expect there would be no certificates in the DFD.

I

Yes.

P

So to prove that TLS is used, then you can rely on another check whether you find these certificates in the source code or not, outside of the model.

I

Okay, so you would basically show that the tool searched for the keyword “certificate”, and didn't find any. So that's your proof that there is no starting point.

P

Yeah, maybe for the format of the certificate, like "start ()?" or "beginRSA", or I'm not sure. So things like that. We also... giving more information and augment what you can extract from the DFD.

I

All right. That's a nice idea. Yeah, as you know, we always had this problem of how do you show something that's not there, right?

P

Yeah, maybe you can go beyond the DFD and take a look at the source code with routes like this, which are automatable and the tool is able to perform some quick checks, predefined for each rule and not just rely on the DFD.

I

All right. Last detailed question: so you didn't write your own rules in the session on Monday. But you saw the queries in the beginning of the output. And also you know the language structure a bit... do you think it's -- do you think you could use it? Do you think it's easy to use and do you think it would help you? Or would you always just use the ones that are in the tool?

P

To create new rules.

I

Okay.

P

Yeah, I understood the filters, I understood the logical operations. I would say it was very easy to check whether the rule text meant the same as the query.

I

Okay. So it's readable ...

P

...readable and easy to generate or easy to create.

I

Since you know the rules that are in the tool, do you think you would -- these are enough or in your daily work, would you write your own ones that you always run on models? What do you think? I mean, you don't have to remember the rules now, but would you have certain checks that -- for example, you said you would always list all nodes?

P

Yes, yes, yes. I would create such an output or create such a rule. And yeah, I think I would think about how to use the tool for other types of rules, maybe, or further types of requirements. So for example, let me check whether I can give you an example -- Yeah, probably you already have something about logging. So in the Piggy Metrics project, PII was logged. And I think that if you introduce things like not just relying on the DFD, but being able to search in the code with some, maybe with some parameters, I would say, then this is something that we… that's something similar that we use in [tool developed by company] as well. So we're able to search for patterns or search for not just as a simple grep, but combine information in a way that is useful for us. So nearby, is there something like this? So I would, I think use the tool to determine which nodes are logging. That's basically easy to do, I think, then use this traceability information to take a look at exactly what is logged because I don't expect the model to include that level of information of what exactly was logged, like PII. I'm not sure if this assumption is true, but I expect it's not. And then that would speed things up compared to a manual search. I can even list automatically all the... write out all the lines which include it according to the tool logging capability and then just take a look at a glance I see that there is -I don't know- “address” there is “telephone number” and your requirement is failed.

I

I really like this. This is like using only part of the tool. So don't using these property checks, only the scoping to make something with it that the tool itself cannot do. Yeah, because as you said, the level of detail is just not there. That's very hacky, typical developer. Yeah, it's a really good idea. I like that. All right. I'm basically done. Do you have anything else that we didn't talk about that you would like to add? Anything positive, negative, feedback, improvements...?

P

I think generally I was happy using the tool and saw the potential and would introduce it if automatic model generation is something that we can add and that would be very useful for us.

I

Okay.

P

Maybe not just for microservices, but introduce this concept to other types of evaluations.

I

Okay. Yeah, that would be interesting as well. Many things you can do with it. Like the advisory board also said for cloud infrastructure. So there are more use cases where this can be useful.

P

Yeah.

I

All right. Okay, thanks a lot, [name]. That was good insight.

P

I hope you will be able to use our whole evaluation process or expert review that not just me but my colleagues did. I hope it's useful for you. And for us, it was, I think, at least for me, it was very, very useful too, to -you know- get the bug in your ear to try to use more like this to automate evaluations, introduce new tools and new concepts. Yeah.

I

Okay. Nice. I like that. Yeah, I definitely do think that I will have good insights. So I had an interview with [name] yesterday already. Other than that, you're the second. And he was also very open and talked a lot. It was nice and said good things. So I think it'll be very valuable. All right. Thanks a lot.

P

Thank you.

I

Have a nice day and we see each other.

P

All right. Take care.

I

Bye-bye.

P

You too.