Interviewer
What have been your motivations for learning Rust and for choosing to use it?

Participant
I've been interested in using Rust just because it's interfacing with unsafe code. So it's performance when it comes to critical code. So when dealing with unsafe abstractions, it's greater at the end. It really helps me out. And also when dealing with writing sometimes low-level code, I just like to use it as a trust. Yeah, I think it works better for me.

Rust Performs Well

16:... Low-Level Access

Interviewer
Gotcha. So are you transitioning, like with Rust, did you have a previous language that you generally used for this type of stuff and then you moved to Rust?

Participant
I think, okay, I learned programming from someone else who is actually an expert. And so he introduced me to Rust because that's what he was using. So I don't maybe have alternative languages, unless maybe if you talk about ...

Interviewer
Oh, sorry, I don't think I can hear you. I think you got cut off there. I think you got cut off in the middle of your last answer. I didn't hear the last half of your answer.

Participant
Oh, yeah, okay. I think I've specialized with Rust, although prior to maybe using Rust, I was using C++ and also Go or something like Python or maybe Kotlin or something. But for now, I just specialize with Rust, although sometimes I use Kotlin or maybe Swift. I just integrate something, so it's just kind of maybe a task that I have.

Interviewer
Gotcha. Okay. And what do you use unsafe Rust for, generally? Okay.

Participant
I just maybe limiting some scope or maybe of unsafe code or maybe to review unsafe code or maybe to just document and communicate.

Interviewer
Could you say that last part again? I'm not sure if I understood.

Participant
Can you hear me? I think I'm having some ...

Interviewer
Yeah, I think the end of your answer, it got cut off again, I think. Could you repeat that last answer? Sorry.

Participant
Oh, yeah. I mean, to document and maybe to communicate, maybe I have an example called documentation and I want to create someone or maybe someone we're working with, I can just create it with Rust. I think sometimes the people I work with, they just understand it better.

Interviewer
Gotcha. Gotcha. For documentation, is it like there are particular conditions that people who call these APIs have to maintain in order for it to be safe?

Participant
Unless maybe there's some structured code base they can maybe be using, then I think that's the base.

Interviewer
Sorry. Sorry. I don't think I made out that last answer.

Participant
Can you hear me out clearly?

Interviewer
I think now it's working fine.

Participant
Okay. I think we're having some, maybe you can address your question and maybe you can start over because I think maybe there's a point where we lost each other.

Interviewer
Gotcha. I think my question is, I guess you mentioned using Unsafe for Rust as having Unsafe be documenting certain functions when you're in a larger code base. What are some of the reasons specifically why you'd choose to label something as unsafe?

Participant
Okay. I would say maybe for example, when writing a low-level code, such maybe an operating system would be running device drivers or embedded systems, I can use, I mostly use unsafe Rust because I feel like it's necessary to use it. And maybe especially the scenarios when they maybe want to infer things like memory manipulation, interacting

16:3 Okay. I...

No Other Choice

with hardware, or maybe performing operations that are not covered by other safety guarantees of the language. So I basically use it in such instances.

16:...

No Other Choice

Interviewer
Gotcha. So with the embedded code, it's mostly just the necessity, right? There's just no way to do that in a safe context. What are some of the cases where you've worked around Rust guarantees or Rust's limitations like with memory that you mentioned?

Participant
Okay. Let's say maybe when I'm dealing with some kind of unsafe abstractions, when you say I'll say this can go maybe, you know, writing some kind of safe API when it's actually surrounded by some kind of unsafe implementation. So in such sense, then I would just use unsafe Rust.

16:4 Okay. L...

Easier or More Ergono

Interviewer
Gotcha. Gotcha. So I guess which, could you give an example of some specific APIs that you've worked with like that? Okay.

Participant
I'll be letting you finish because I'm here for you to help me.

Interviewer
I'm sorry, I don't think I made out that last part.

Participant
You're asking of ...

Interviewer
Oh, I guess like what are some, you mentioned like this, a choice between using like safe Rust to encapsulate unsafe code or just using unsafe code. Could you give an example of specific situations that you've encountered with that?

Participant
40 Yeah. So maybe when we are doing some, maybe we've developed it, but maybe contributed some kind of API have used it in work, you know, that is in web development and also in a text web or maybe even rocket. Maybe I'm taking a task on networking programming, then I can use, maybe I have used it in [library]. and also called ASINs and it matters with the nature in the kind of tasks that I'm taking on. It comes to maybe database access, I can use something like this, I can use unsafe rust, maybe to take this on a VPN, maybe work on some implementations, maybe, you know, some kind of assessing the security vulnerabilities or something. So basically, yeah, just layout, but a few.

Interviewer
Wait, so are you contributing to these crates or is it just that you're using them and you know that there's unsafe beneath the surface, but you're not actually writing the unsafe?

Participant
I think I just, it matters.

Interviewer
Say that again.

Participant
It matters with the nature of the task or maybe with the security vulnerabilities of what is required or maybe the kind of task I'm running.

Interviewer
So with, I guess when you are calling, I guess, has, has performance ever been a consideration in, in the unsafe that you've used? Like have you ever used, used unsafe to increase performance or had like a performance issue maybe with safe Rust that you used unsafe to get around?

Participant
Yeah, yeah, absolutely. So maybe, or when maybe dealing with the performance critical code, certain performances, then, you know, I have to look at the critical sections of my code and then it's key for me to, you know, to look for the unnecessary or necessary points where I can use unsafe for us just to achieve fine-grained control over memory allocation or maybe do the occasional perform low-level optimizations. So it's important to profile and benchmark your code to ensure that, you know, the performance gains outweigh the risk associated with unsafe code just because it's unsafe, that's it. Yeah.

16:5 So maybe, or when...

Increase Performance

Profiling

Interviewer
So you mentioned like allocation as being one example. Could you give a more specific example of where using like the unsafe allocator versus one of Rust types gave you better performance?

Participant
Maybe when checking out for compatibility or maybe portability, yeah. So I can maybe learn as that one.

Interviewer
Gotcha. Gotcha. So it's more about just having better low-level control over the, like, formatting out of the memory that you're getting?

Participant
Yeah, that's it.

Interviewer
Gotcha. Okay. And then when you are, like, do you generally write documentation for any unsafe APIs that you expose to, like, other users?

Participant
Yeah, sometimes.

Interviewer
Gotcha. What, like, in writing that documentation, what are you describing? Like what, what aspects of the unsafe do you focus on?

Participant
Okay, maybe mostly I'll be laying out, if it matters, if it's web development, then I might be laying out some kind of instructions for maybe somebody who want to scrap, maybe, you know, sometimes we leave it, but also it can enable other programmers to use their code. So I just leave it like instructions just because I am considerate that, also I benefit from resources online from other programmers. I just leave instructions there for, just for anybody else who might want to use the unsafe, unsafe Rust. So I just leave it for maybe other users.

16:13 So I ju...

Engaging with...st Com

Interviewer
Gotcha. In, like, so are you describing, like, are there a particular, like, what's an example of something that you have documented then for users? Like what's the type of, what's an instruction that you might give to someone when using one of these APIs?

Participant
Some of the instructions, like, just telling them about the, for example, I mentioned from them about the security vulnerabilities that that code has, or maybe the level of compatibility or some kind of talk about compilers and everything. So it matters really how I have confirmed the code and maybe how I think it's better for somebody to better use it. So I just can lay out in a kind of a way that it's easy to understand it doesn't be a problem to maybe the next user. So just to make it as easy and communicating, of course.

16:6 Some of the instr...

Documented...ract or I

Interviewer
Gotcha. So it's, it's not just like potential errors that you could run into or just what you need to do to actually make it run, like avoiding segfaults, it's also, you need to be careful about this or else you could add like a security vulnerability.

94 **Participant**
Yeah, yeah, yeah.

95

96 **Interviewer**
Are there particular like classes or patterns of vulnerabilities that you have seen as being like common issues that people have run into when using unsafe in the ways that you've used it?

97

98 **Participant**
How you mean like challenges?

99

100 **Interviewer**
Yeah.

101

102 **Participant**
I think there's maybe some I can highlight. And some of this may be a lack of maybe compilers and type safety checks. I think that that's some kind of a common problem which I personally deal with and I make sure that they check out that it's okay and it runs because sometimes you can run problems, you're trying to run it and then are you getting the type safety checks is not aligning to whatever setting you're trying to make it up to align to. And also I think, you know, especially when you're sharing unsafe Rust to maybe colleagues or maybe people you're working with, maybe a particular project, you know how it is, you just don't take, sometimes you're working as a group of people, particular project. And so when you share it, some people are having trouble trying to understand the code and also maintenance of the code is also kind of, it's complex and it's harder to reason about compared to safe code, you know. And also, you know, the movement of low level operations, raw pointers and intricate memory manipulations. I think those are some maybe points that I can highlight and say they are quite a challenge. So maybe when it comes to compatibility and portability, I think it's a problem. Sometimes you're sending code to somebody and it's not compatible to whatever, whatever maybe you're trying to use them for, it's not compatible, it's not working. And also one of the most, it's also the security vulnerabilities of unsafe rust, it's vulnerable. So when you are using unsafe rust, you'll be sure to get problems like low buffer overflows, or some memory corruption, or maybe you'll be sure to be able to collect a lot of external resources, or maybe you can run into maybe careless or maybe incorrect usage of unsafe rust, it can compromise your security system. I actually compromised one of the processes, we had to work for [redacted] and we had to run unsafe rust and it was totally compromised. So we had to like work it for about 72 hours to recover it, it was so hectic.

16:7 And also I think, you kn...

Unsafe is Diffi...to Unde

16:8 So when you are...

Integer Bound...Overflo

Security Vulnerability

103

104 **Interviewer**
Wow, wow, yeah, that sounds like a huge issue. And then have you had any particular challenges relating to memory with, like I noticed you mentioned that you use Rust's intrinsics as well as calling system nodes. Have there been any issues with memory safety and unsafe surrounding

those two features?

Participant
So probably I can highlight, but to do with memory manipulations, yeah, it's been a sort of issue. And also it makes it difficult to detect and fix bugs, so yeah.

16:14...

Unsafe is Diffi...to Und

Interviewer
Gotcha. I guess, what in particular makes bugs difficult to detect with those features?

Participant
I think it's the complexity.

Interviewer
Oh, sorry, I think you're getting cut off there again. Oh, I think it, so it says you're muted. Can you hear me? I can hear you now. Sorry about that. Yeah.

Participant
So can you repeat your question a bit?

Interviewer
Yeah, sure. So you said that bugs related to intrinsics and system calls are tricky to detect. So I was curious about what makes them tough to find.

Participant
I think it's a lot of concurrency issues.

Interviewer
So is it that, oh, I see, so you're using, or are you using like intrinsics to implement concurrency primitives and then those like are just difficult to reason about because concurrency is difficult?

Participant
Yeah, that's it.

Interviewer
So then you mentioned using several different Rust like memory container types with unsafe or like in unsafe contexts. So in particular,
104 you mentioned using arc, unsafe cell, ref cell, as well as manually drop
lazy cell and one cell on the survey. Could you talk a bit about, I guess, starting with unsafe cell, like where, where would you use an unsafe cell and then the code bases that you've worked on?

125

Interviewer
Hello? Hello. I can hear you now, did you hear my question? No, I think, I don't know who's having maybe network programs. I don't know if it's, it's me or if it's you, maybe I can check my security, my, sometimes like I'm getting like, getting lost along the way and getting like some breaks in between your sentences, I don't know. Gotcha. We can, we can keep going. If you want to reschedule too, and figure out what the connection issues are, that's fine as well. Feel free to interrupt me to whenever I start breaking up. Okay.

Interviewer
So I think my, just to sort of go back to my question, where would you use an unsafe cell in, in the code that you've worked on?

Participant
Okay. When it comes to an unsafe cell, maybe let's say, have a code maybe from, I want to borrow a code from maybe a C++, or maybe from, from Python or something else to take it and maybe compile, maybe assess its vulnerabilities, then, just because, you know, it's a lightweight code, I can use it for, for some kind of maybe, just to, just to borrow maybe a code from, from maybe another SafeRust, or maybe another SafeRust, or maybe just to integrate it, just to compile it or something.

Interviewer
Gotcha. So you're, you have memory coming in from UnsafeRust, or from like a, a C or C++ call, and you want to borrow against that and access it in SafeRust, and then you're, you're wrapping it in an unsafe cell to allow that?

Participant
Yes, yes, yes, so to do.

Interviewer
Gotcha. And then with Arc, where would you use an Arc?

Participant
Come again?

Interviewer
So, like where, where would you use like a reference counted container then to like, when, when containing like,

Participant
maybe when we have a thread, I want to provide a safe thread, or maybe dealing with some maybe shared ownerships, then I can use that to ensure synchronization, just to access maybe underlying data.

Interviewer
Gotcha. Yeah. Gotcha.

Participant
So you have data that's shared between threads, so you'd need it to live for as long as each of those threads are living, and you might have concurrent access to the, the like, concurrent mutation maybe to the object, but you want that to be synchronized, so you'd wrap something in the object.

Interviewer
Yeah. Gotcha.

Interviewer
And then with, I was curious about RefCell, I know that that can cause some performance impact because of the checking, has that ever been a problem with you, or has the performance difference there been fairly minimal?

Participant
I think that, that's why I think it's a, it's a common problem with the outside. I think it's a common problem, because even the, my, my training just doesn't really be controlled by that, so there's like a concurrent maybe, it's a common problem.

Interviewer
Gotcha. Gotcha. And then when would you use manually drop, I guess, is like, what, what use cases would you have for that? Are they generally in, with like Rust allocations, or are there situations where you've used that with allocations coming in from like another language?

Participant
Yeah, yeah. Maybe, it can be, maybe in session, or maybe when you want to do some, okay, removal of elements. Gotcha.

Interviewer
And then the last question, I was curious about LazyCell and OnceCell, so are you using those with like static variables, or are there like, like what's, what's the, at least the use case for those two that I'm most familiar with are things that you would need to have globally available and have initialized like right at the start of the program. Is that generally where those are used, or do you have like other use cases for them?

Participant
I mean, I have tried, maybe I tried to maybe use the team, increase, increase of vector just to help with resizing, you know, re-write structures if I have them. So I can use that to resize, at least that is.

Interviewer
Gotcha. Gotcha. Yeah. And then, so you mentioned you doing FFI with Rust between C++ as well as JavaScript and TypeScript. Would you describe, like Rust has pretty particular assumptions that it makes about memory, and those are pretty strict compared to what JavaScript or C++ would assume. So have you had any issues in handling or accessing or managing memory across the language boundaries just because of how things are different?

Participant
I think [unknown] has this problem. So when it comes to C++, when it comes to, in reference to memory, I think it lacks safety guarantees and also some features of Rust is more modern than C++. So that's just like of, I'll say C++ is more for kind of analog. So when it comes to when you're using modern language features of, you know, like presenting Rust currency, then it leads to more bugs and vulnerabilities if you're not careful.

16:9 So when it co...

Feature Disparity

Interviewer
Gotcha. Are there any particular vulnerabilities you can think of?

Participant
Maybe, okay, I don't know. I don't have any other, can we be recall of, I think kind of somewhere, you know, I think that's, that's, yeah.

Interviewer
Gotcha. So, yeah. And is there like, do you find with JavaScript that it's a similar problem or is that either like easier or maybe difficult for different reasons to handle memory that's passing like across that, that foreign boundary?

Participant
I think with, okay, I'm not a fan of JavaScript as much, but I think with JavaScript for, you know, for the, I say, as for Java, I think I've got maybe some positive feedback from them.It's like, it works better for them, we don't have any much problems with Java.

Interviewer
Gotcha. And when using, so the, you've, you've mentioned using a few different tools to help create like bindings, bindgen, windows-bindgen, SwiftBridge, and then also just like the, using, using WebAssembly, I'm assuming with your JavaScript and TypeScript, could you describe like the, the usability that you've had with these tools? Like have, have they been easy to use? Have there been any problems that you've encountered with them? Oh, I think you're muted. I'm not sure if I, I might be breaking, I might be breaking up, or I might've broken up there. Sorry. It looks like you're muted right now.

Participant
Oh yeah, sorry, I was, I was talking to someone, sorry, sorry, yeah, I think I'm done.

Interviewer
So I guess the question was, when you use tools like bindgen and windows bindgen to create bindings between Rust and other languages, have there been any problems that you've encountered using them?

Participant
Yeah. Yeah. Yeah. Of course. I think, you know, computation is a common problem. Sometimes it takes longer time to compile, and especially with a larger code bases, or maybe projects with many dependencies, the process is, is too slow. And I feel like it's too slow, like you need some work to be done. So and also, I can maybe, I guess, or maybe be sure about it, is when it comes to building configurations, it's also where you can get some, some challenges in building configurations. Also things like compile time errors, it's also a problem that I deal with. Yeah, I think those are the major ones I can talk of. That's right. So, so maybe I can say first positive, positives in learning can be learning curve that is, can be a bit tricky or maybe trouble to you.

16:11 So...
Limitation of Binding T

16:10 So...
Bindings are a Fido

Interviewer
Gotcha

Participant
So it's, it's like the, the main issue is that it takes a long time to compile these. And then also you end up getting a lot of compile, like errors that you then have to go and, and fix manually. So it's, it's not like, it seems to require a lot of configuration. Yeah, that's it.

Interviewer
Gotcha And is it that like, are the results once you've finished compiling correct? Like have you had any issues with like mismatched types maybe, or like the interfaces that it generates being incorrect in some way, or are the results that you get generally correct after you finish doing the compilation and configuration?

Participant
I think once I'm done compiling everything, and when there are, maybe I would fix them or maybe the bug or whatever errors you have here, I think once you're done, that's it. I have never made, or maybe have post, maybe post challenges or maybe first errors, I think so it just works.

16:15 I...
Generation VS Validati

Interviewer
Gotcha. Gotcha. So not, not like a typical issue that you'd find.

Participant
Yeah.

Interviewer
So then you mentioned using Clippy as well as address sanitizer, thread sanitizer and memory sanitizer with code bases containing unsafe rust. So I guess let's focus on the sanitizers first. What errors do those typically find in code that you're running? Like, are there particular errors that tend to pop up with unsafe rust that the sanitizers can detect for you?

Participant
Yeah, I think they're workable because they can be able to maybe, maybe to track uninitialized memory usage and it can also report when uninterested memory is not reading. So I think, yeah, it works for me and I like using it because it makes the work easier to detect memory, issues with memory or maybe with bugs and everything. And so actually the best thing is that for example, the UBSan is able to detect potential bugs, potential bugs even without them occurring. So you can be able to fix that even before the actual bug happen. And also as some kind of defined behavior, they can also be able to detect that just to get rid of unexpected program behavior. That's really something.

16:16 An...

UBSAN

Interviewer
Gotcha. Are there any like patterns that you've implemented in Rust or like certain operations that you've done in Rust that you've found have tended to cause a lot of errors that these sanitizers have detected for you?

Participant
Maybe, which one I can, I don't know, I think I'm off, I've talked a lot, maybe if something hits up my mind, I can just image you.

Interviewer
Yeah. Yeah, sure. That'd be great. Yeah. Feel free to send anything. And then I guess my, so then with those tools, my last question, I have a couple extra questions. The first is, I guess, what's the most challenging bug that you've faced that involves unsafe Rust or Rust's FFI and how did you end up discovering it?

Participant
Actually, to the initial performance, we were trying to detect it as something, so it was actually a very, it was a huge code and some features of unsafe Rust. So I think there was a guy who actually, you know, he messed up. So he's kind of interviewed both safe Rust and unsafe Rust at the same time and actually gave him the access to actually compile his code. And I think the issue was somebody, somebody need to create them up since something came up. Just a minute.

Interviewer
Sounds good. Yeah. No, no, no, no worries whatsoever. Okay. Okay.

Participant
Oh yeah, sorry for that, I'm back.

Interviewer
Oh, hello. No worries.

Participant
Sorry.

Interviewer
Yeah. Okay. So my question was, or sorry, there's something about, we were talking about a bug related to a large code base, I think both unsafe and safe Rust. There's an issue someone was having with compilation. Yeah. Yeah. I think it was both refactoring and an amazing issue, it was a huge one. We had to scramble the code one by one, it was a lot of work. So it did crumble like, I'll say about 40% of the code was free. So I didn't know how it can do it eventually to such a big calm, but I think it was an issue with compilation and with refactoring. So it was a mess, actually it was a mess, but it was a mess from somebody, from a programmer, but also unsafe Rust deep-prayer also had a very significant part of it. And I think unsafe Rust should be used in a very cautious way, or maybe when it's necessary, because it can cause problems sometimes.

16:17...
Preference for Safety

Interviewer
Gotcha. So was it that the output of the code is compiling correctly, or the code was compiling, but the output was incorrect and causing issues, and then you need to do substantial refactoring to figure out what you had done there.

Participant
Yeah. Yeah. Yeah.

Interviewer
Gotcha. That sounds like it could be related to some of the problems that Miri can detect with Rust code bases in terms of undefined behavior using pointers. Have you used Miri in any code bases, or has that been tricky because of the FFI stuff that you do?

Participant
Yeah. Yeah. I know of somebody who's using code bases, uh, uh, but maybe like, uh, I use it as much. But I have some significant experience.

Interviewer
And then, so I guess my last question is, do your development tools and practices, are they helpful for addressing all of the problems that you find in working with unsafe, or have there been issues that you think the Rust community could do a better job of tackling with using unsafe?

Participant
I think, as I say, I just use unsafe Rust when it's necessary. And I will say that I have maybe much confidence when using unsafe Rust, I just use it when it's necessary. So it's something that I can see the tool that I use, it's a language that I like using as much. So I'm very cautious when using it, and I make sure that I do some moderating just to make sure that in case of problems, I would have a bigger problem.

16:12 I just use...
No Other Choice
16:18...
Preference for Safe

Interviewer
Gotcha. So because you minimize the extent to which you use it, you generally don't, like there aren't any major problems that you face in using it.

Participant
Come again?

Interviewer
So like, is it that because you only very rarely use it, that there aren't like significant problems that you're facing, because it's just something that you try to use as little as possible?

Participant
Yeah, I think that's it.

Interviewer
Gotcha. Gotcha.

Participant
Yeah.

Interviewer
So I think that that's all the questions I have. Thanks for chatting with me. Sorry about the connection issues, but I really appreciate your time. And so it's been great hearing about your experiences.

Participant
Thank you for having me today. I appreciate your time. And also maybe...

Interviewer
Sorry.

Participant
I can't get maybe, what are you maybe trying to, what's your goal by doing this kind of...

Interviewer
Yeah, so I think that my goal is, so my research is in sort of two parallel directions right now.

Participant
Okay.

Interviewer
One is I'm looking at Rust's type system and trying to figure out ways of verifying properties about Rust code. And then the other is more experiential, trying to figure out what current challenges Rust developers are facing in particular with unsafe. So I'm sort of trying to figure out what are the challenges that people are facing with on safe so that I can think of ways to apply like formal verification to help solve them. So the goal of this interview studies is to hear from as many developers as I can get to about how they use them, what are the problems that they've encountered with it?

Participant
All right.I think that's cool. And just hope you get all the information that we can forward to and yeah, good luck with us.