Gmail **Paul Muntean <paulmuntean@gmail.com>**

typearmor paper

Xi Chen <xi.chen.chn@gmail.com> Thu, Jul 20, 2017 at 10:29 PM
To: Paul Muntean <paulmuntean@gmail.com>

Hi Paul,

I did not do the on paper editing because I found this paper is quite good writing and easy to follow, and I do not see any single points that need to correct but the problem is more about overall design.

From what I understand from paper, the major contribution from typeshield compared to typearmor is more precise type info on parameter (byte wide). However, from your evaluation, table V do not really show too much improvement on that (typeshield only have 10-20 target reduce which is around 5-10%). You do have a section in evaluation to show the upper bound of typeshield, but it is quite theoretical, and I think you need to show that in real world application, the wideness of parameter do matters. I basically shared the same view as reviewers that the work here is very solid but also quite incremental. My suggestion here is maybe try to use pointer type to further improve it. And maybe test with C++ to get better result. After all, COOP is mainly for C++.

Other two suggestion:

1. Compare your back-edge protection with shadow stack and show why you need to design a back-edge protection.
2. Show how you do address taken analysis, it can be only a paragraph, but at least talk about how you do that, and maybe a small table to show whether the result is promising. You can compare it with LLVM address taken analysis pass, I remember there are some address sanitizer pass in LLVM do that. But I can not recall precisely which one did that.

After all, I like the paper in general, but if you can not beat the "incremental" arguments, it will quite hard to get publish in my point-of-view.

Best regards and have a nice day

[Quoted text hidden]

--

Best Regards
X.Chen