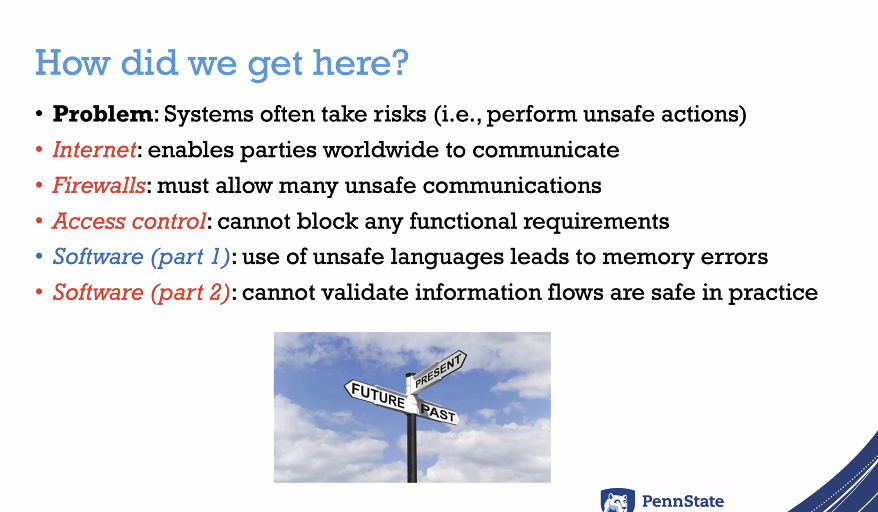
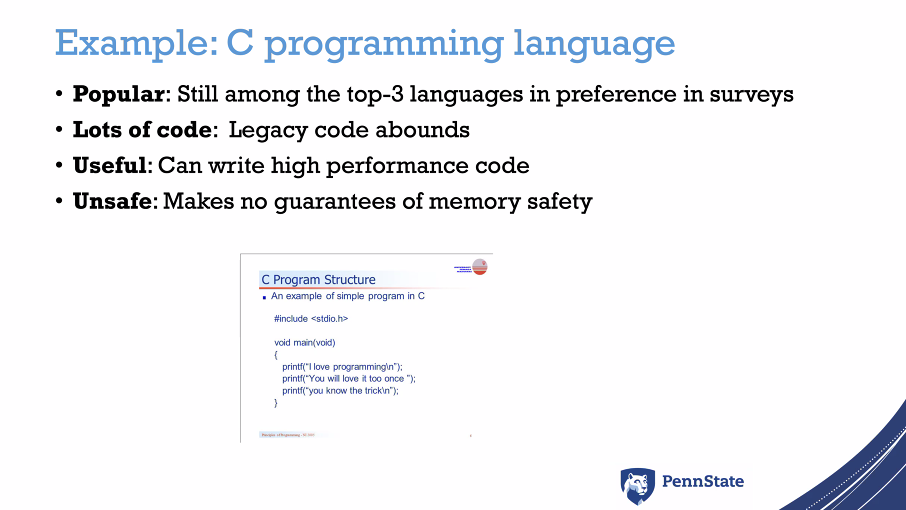
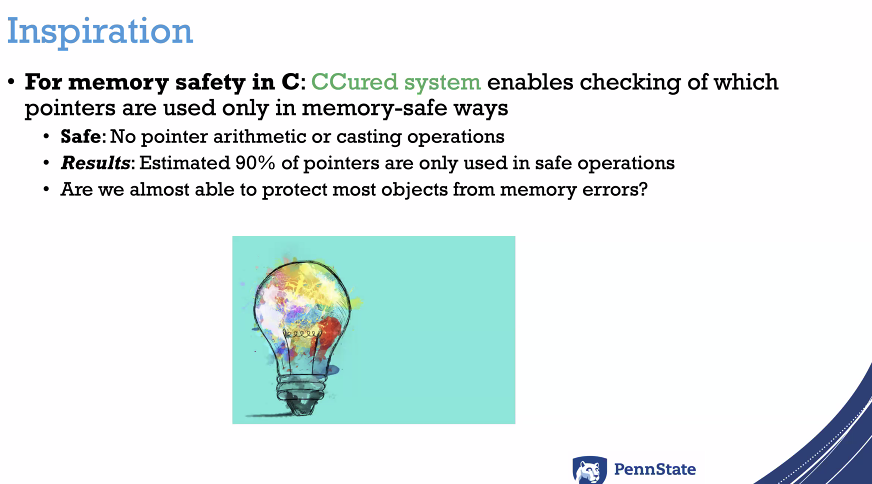
Presentation Notes

Developing techniques to validate particular objects that are placed on the stack, and utilize that to provide more protection at a fairly low cost

Seem that intrusion detection has become too heavily relied upon. Could there be other ways?

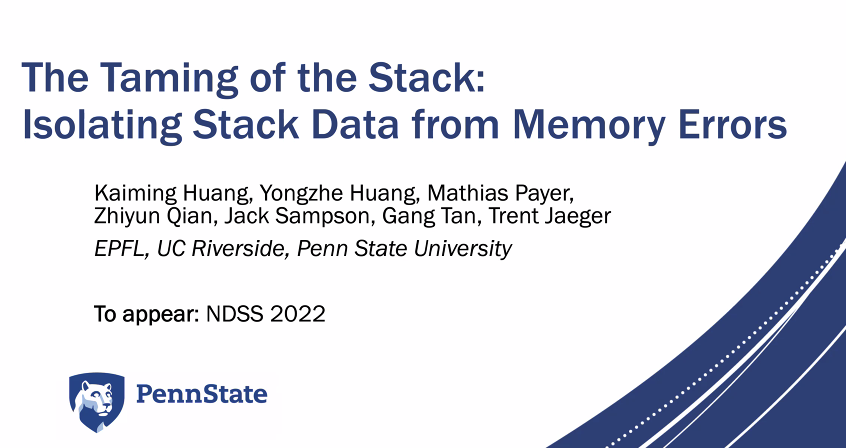




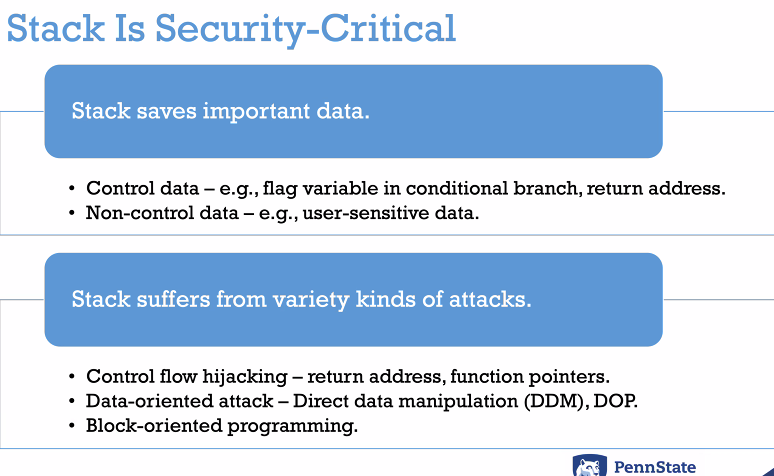


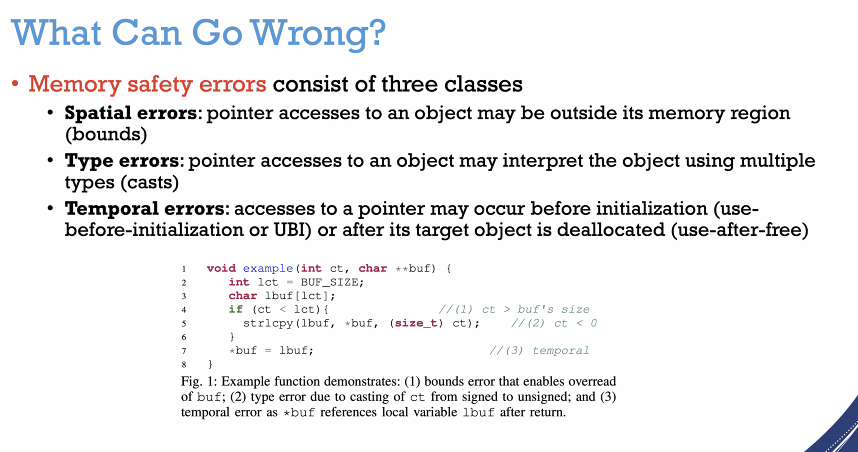
Come up with a static technique that pointers used in programs were safe and would not violate memory safety.



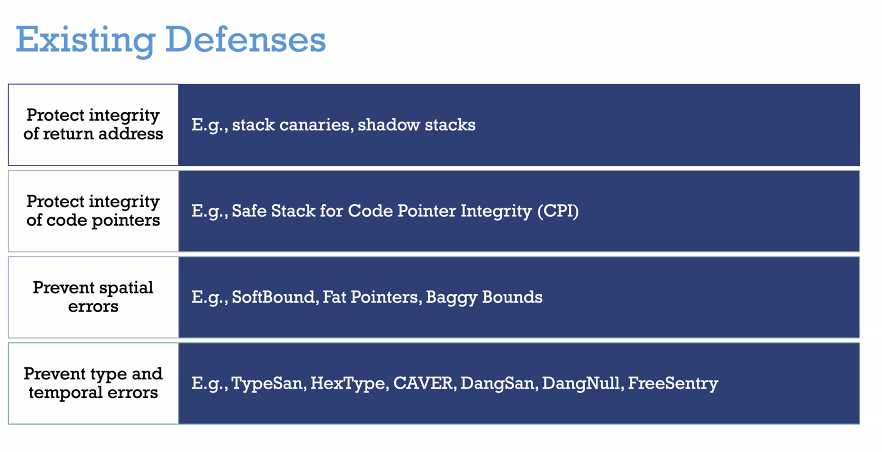


Just released a research paper. Focuses on stack memory errors to prove

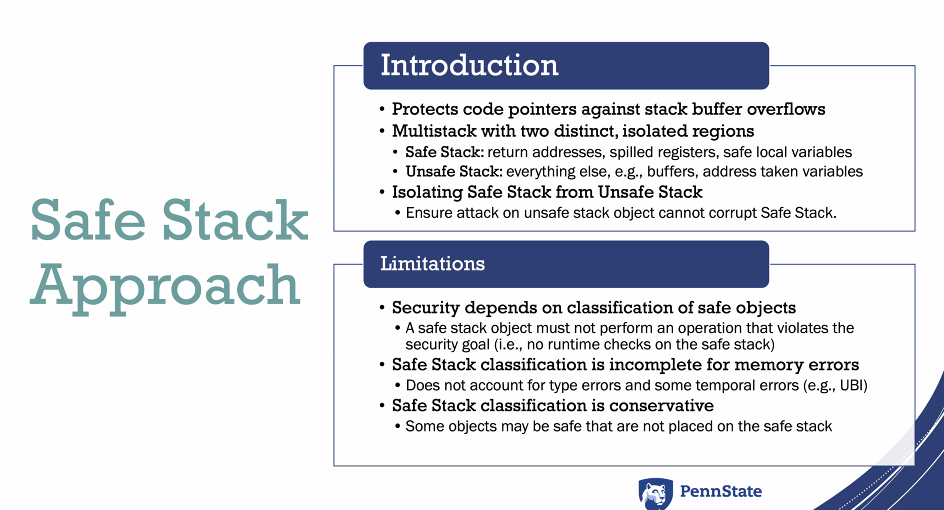




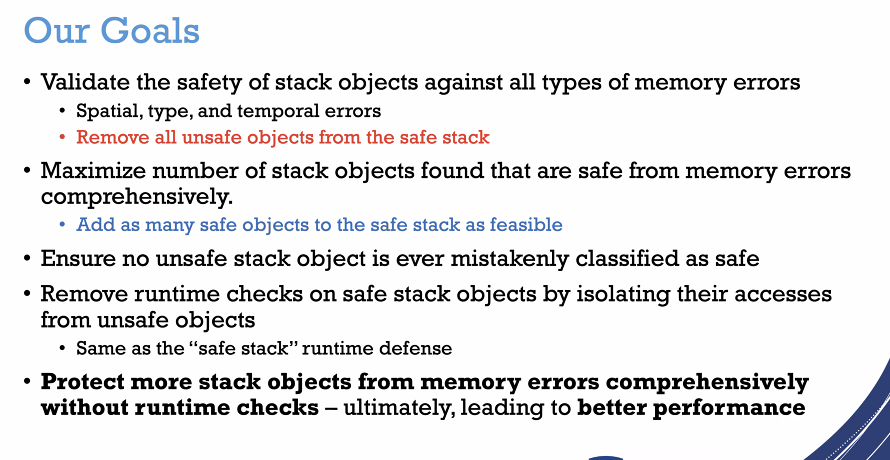
Spatial error -> Buffer Overflow



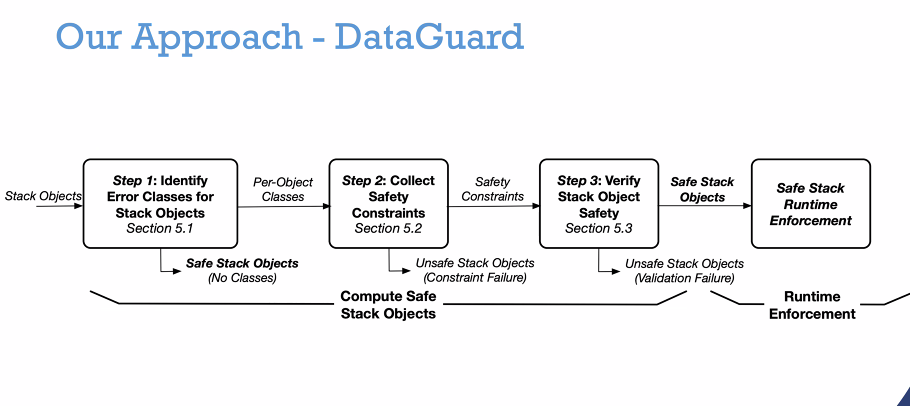
The problem with each of these measures is that they’re expensive to use all together.

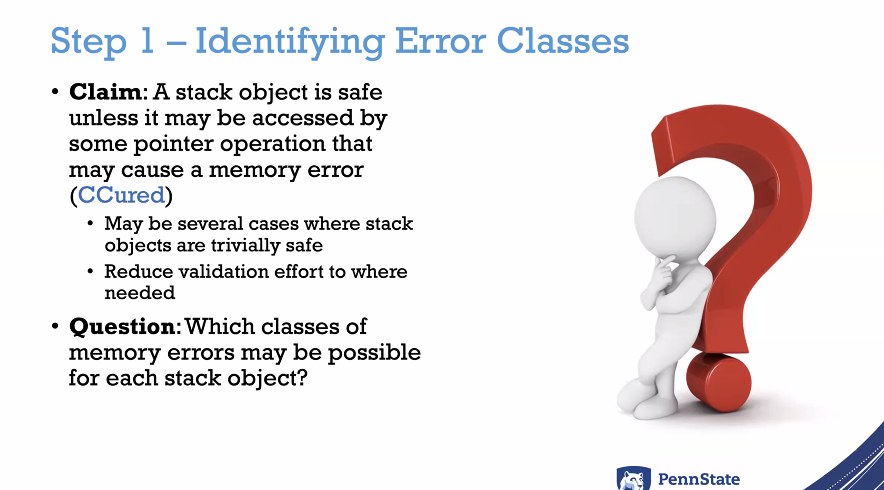


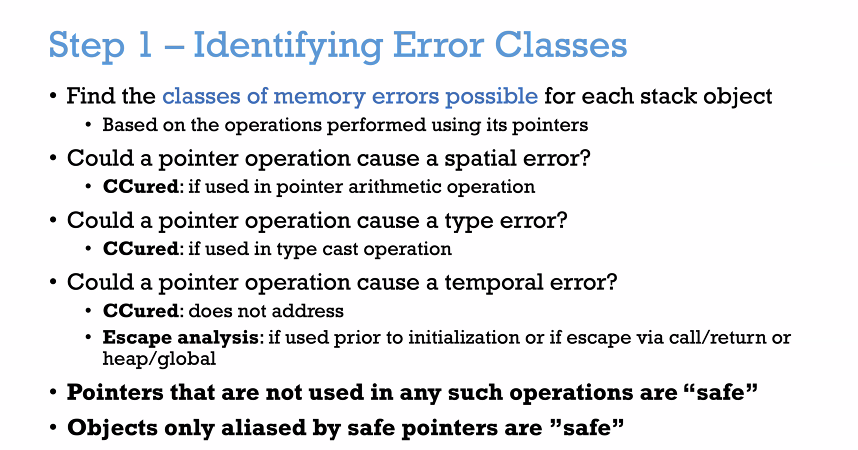
The safe stack divides the original stacks into two stacks.

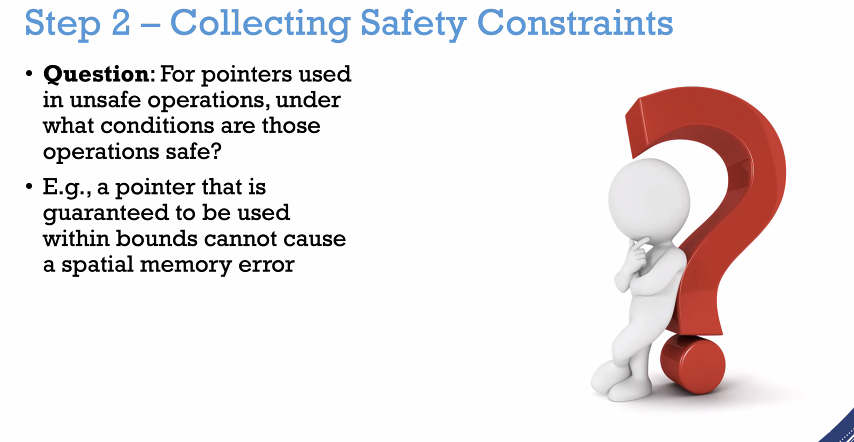


The safe stack scans the objects before runtime.



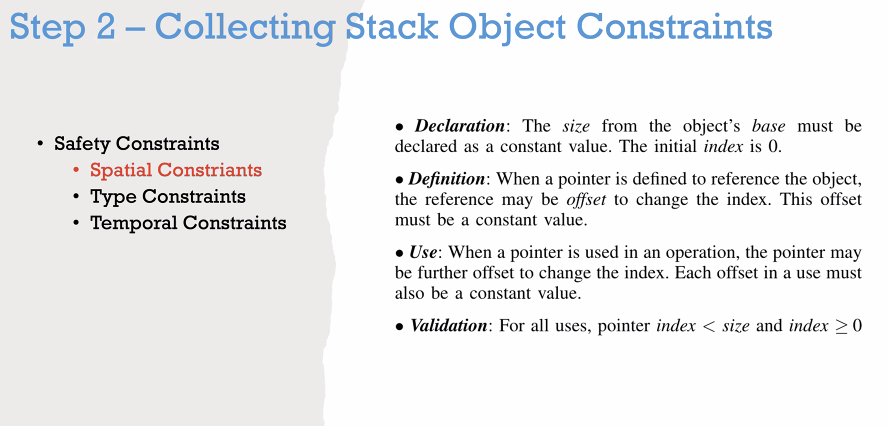




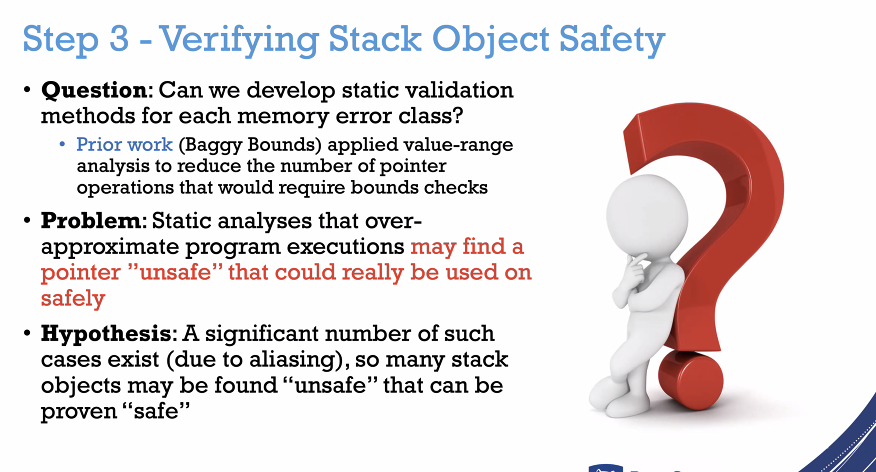




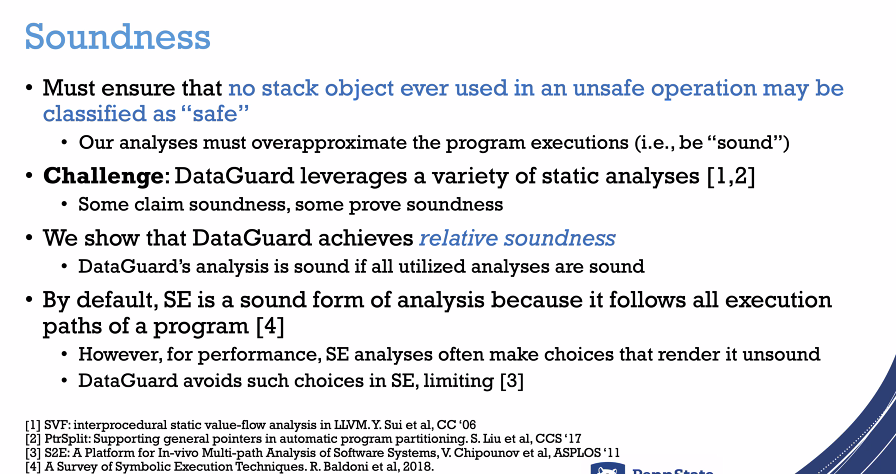
In LLVM each pointer is tagged be SSA which makes fi

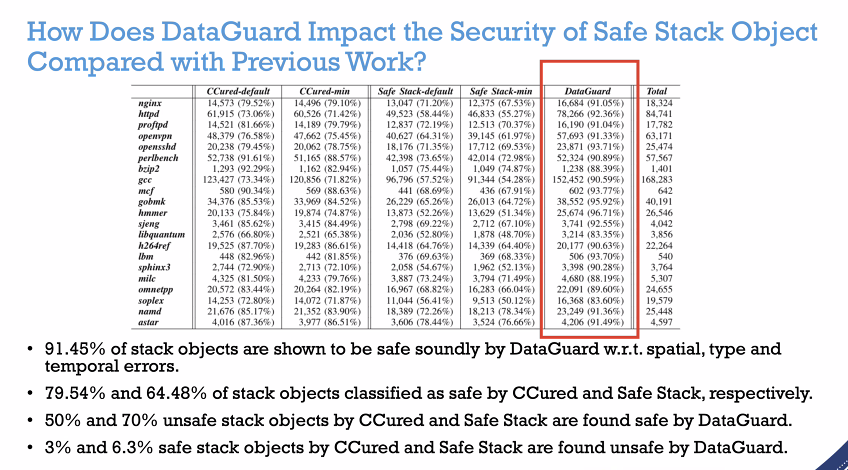


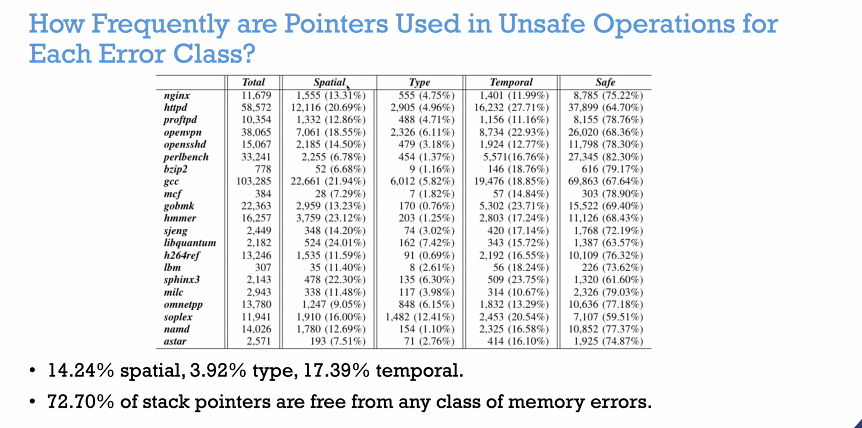
They did not invent these, they just restrict whats necessary to valide them

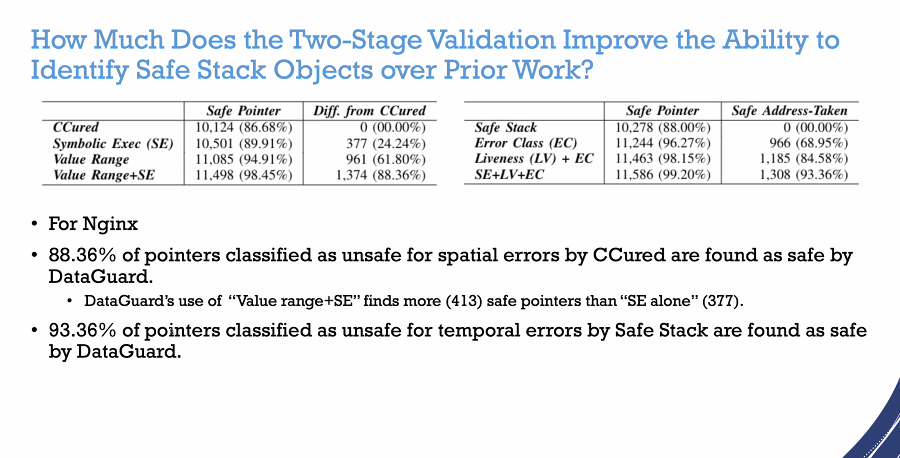


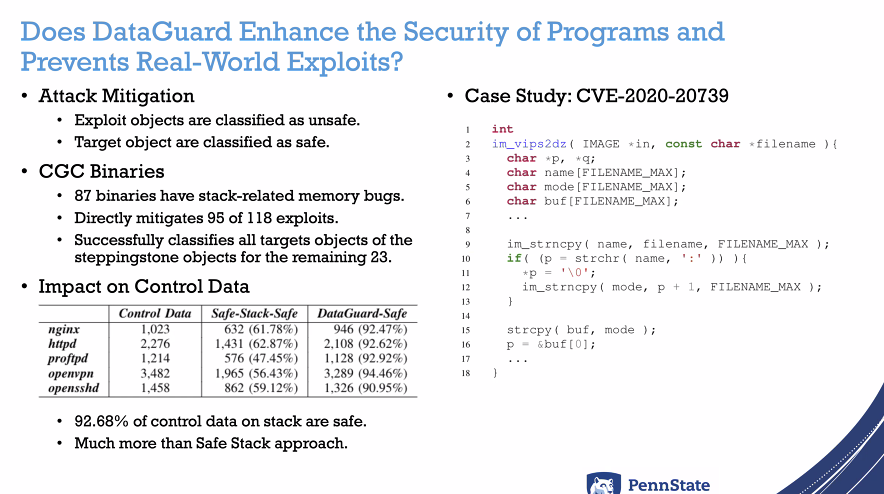












The objects that can be targeted after the attacker exploits unsafe objects cannot be used to continue out an advanced multi step attack.

