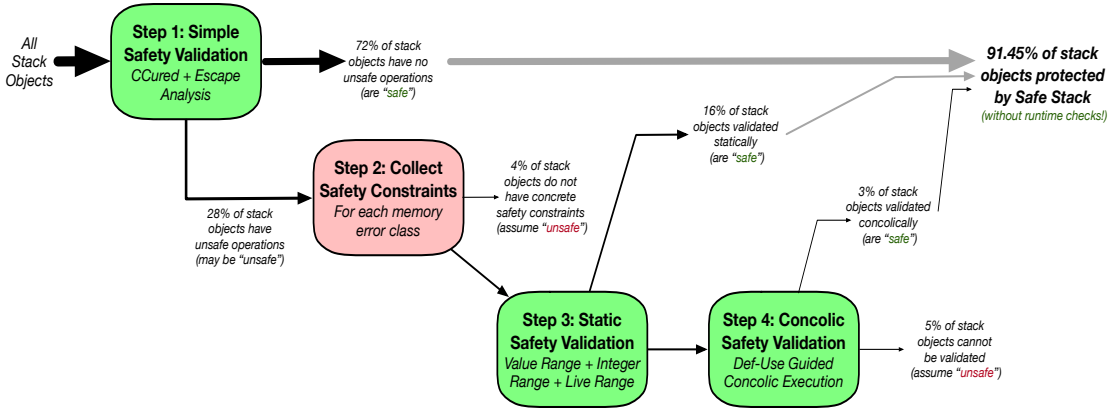
DataGuard System – "The Taming of the Stack" – NDSS 2022

Challenge:

- Current defenses provide **very limited protection** from **memory safety errors**
- Proposed runtime defenses **incur overheads** that have **limited their use**
- Researchers have proposed that **many accesses are safe** from memory errors, but no technique provides **validation for all classes of memory errors**



Solution:

- **Safety validation** identifies stack objects whose accesses cannot cause any spatial, type, and temporal memory errors ***comprehensively***
- Validation using **static analyses** and **guided concolic execution**
- **Isolate safe objects** using the Safe Stack defense ***without runtime checks***

CNS-1801534: Threat-Aware Defenses - Trent Jaeger (Penn State), Gang Tan (Penn State), Mathias Payer (Purdue/EPFL), Dongyan Xu (Purdue)

Scientific Impact:

- ***>90%*** of stack objects are safe from spatial, type, and temporal errors **comprehensively**
- ***3%*** and ***6.3%*** of stack objects found safe by CCured and Safe Stack, respectively, are actually **unsafe** – ***reduce attack surface***
- Safe Stack **overhead reduced** from 11.3% to ***4.3%*** for SPEC 2006 benchmarks
- Applicable to **real-world programs** and prevents **real exploits** – CVEs and CGC binaries

Broader Impact and Participation:

- Shows that using **safety validation to provide a foundation for low-cost runtime protection** is feasible
- Automatic **program hardening protects** over 90% of stack objects from memory errors by construction
- Analyses facilitate automation of **kernel-driver isolation** - ***KSplit*** - see our poster
- Extending safety validation for **heap objects**
- Exploring **hardware-assisted approaches** for secure isolation
- **Teaching** in our Software Security course