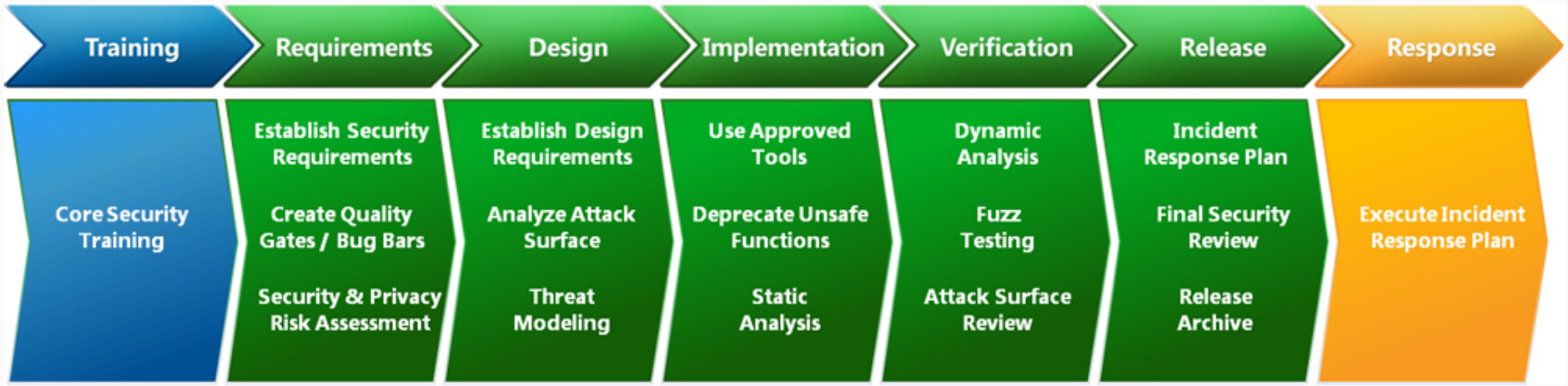
Динамический анализ для безопасного цикла разработки ПО

Алексей Вишняков, м.н.с

23 марта 2022 г.

ИСП РАН

Безопасный цикл разработки (SDL)



- Безопасный цикл разработки ПО — стандарт индустрии
- Мы применяем динамический анализ для повышения надежности и безопасности ПО
  - Поиск ошибок
  - Фаззинг (Sydr-fuzz)
  - Символьное выполнение (Sydr)
  - Анализ аварийных завершений (Casr)

OSS-Sydr-Fuzz: Поиск ошибок в Open Source

Trophies

- FreeImage:
  - https://sourceforge.net/p/freeimage/bugs/343/
  - https://sourceforge.net/p/freeimage/bugs/344/
  - https://sourceforge.net/p/freeimage/bugs/345/
- Tarantool:
  - tarantool/tarantool#6614
  - tarantool/tarantool#6662
- unbound:
  - NLnetLabs/unbound#637
- xlnt:
  - tfussell/xlnt#592
  - tfussell/xlnt#593
  - tfussell/xlnt#594
  - tfussell/xlnt#595
  - tfussell/xlnt#596
  - tfussell/xlnt#597
  - tfussell/xlnt#598
  - tfussell/xlnt#616

Фаззинг (Sydr-fuzz)

- Исследование и разработка методов интеграции фаззинга и динамического символьного выполнения
- Интеграция Sydr с AFL++ и StdFuzzer (libAFL)
- Фаззинг Python
- Фаззинг REST API
- Сравнение и разработка методов бенчмаркинга фаззеров
- Применение современных методов фаззинга для поиска ошибок в ПО с открытым исходным кодом
- Поддержка интеграции с fork mode libFuzzer

Динамическое символьное выполнение (Sydr)

- Исследование и разработка методов повышения эффективности фаззинга с использованием динамического символьного выполнения
- Применение компиляторных оптимизаций для упрощения формул (Triton)
- Разработка символьных трансляций для новых архитектур набора команд

Анализ аварийных завершений программы (Casr)

- Анализ и кластеризация аварийных завершений от интерпретируемых языков (Python)
- Анализ исключений в C++ и Rust
- Исследование и уточнение методов оценки критичности аварийных завершений

Спасибо за внимание! Вопросы?

education.at.ispras.ru/sdl

e-mail: vishnya@ispras.ru

Telegram: @SweetVishnya