ПЛАН

проведения занятий по программе повышения квалификации «Техническая защита информации. Основы инструментальной и технологической поддержки проведения фаззинг-тестирования программного обеспечения»

на период с 18 апреля по 25 апреля 2023 г.

Продолжительность обучения: 40 часов.

Форма обучения: сочетание заочной и очной форм обучения. Заочный этап обучения продолжительностью 24 часа в период с 18 по 20 апреля 2023 г. (самостоятельная работа слушателей в соответствии с Методическими рекомендациями по организации учебного курса (приложение 1)). Очный этап обучения продолжительностью 16 часов 24 и 25 апреля 2023 г. (проведение занятий с преподавателем и итоговая аттестация).

Место проведения очного этапа обучения: г. Москва, ул. А. Солженицына, д. 25.

Перечень изучаемых тем

№ п/п	Изучаемая тема	Количество часов
1.	Фаззинг-тестирование в жизненном цикле программного обеспечения (ПО)	6
2.	Исследование безопасности кода ПО средствами динамического анализа	7
3.	Базовые методы фаззинг-тестирования	14
4.	Способы улучшения фаззинг-тестирования	4
5.	Применение фаззинг-тестирования в процессах сертификации ПО	3
6.	Итоговая аттестация	6

Расписание занятий

Заочный этап обучения

№ п/п	Вид и тема занятия	Время, час
18 апреля		
1.	Лекция (видео). Фаззинг-тестирование: основные понятия, типовая архитектура фаззера. Качество фаззинг-тестирования. Разновидности фаззинг-тестирования, способы улучшить качество тестирования. Место фаззинг-тестирования в жизненном цикле ПО, взаимодействие с другими видами проверок кода	1
2.	Лекция (видео). Методы и средства детектирования ошибок во время динамического анализа и фаззинг-тестирования.	2

№ п/п	Вид и тема занятия	Время, час
	Работа с исходным кодом – флаги компилятора и санитайзеры, внедрение датчиков срабатывания ошибок. Работа с бинарным кодом – инструменты Appverifier, Valgrind и отладочные версии менеджера динамической памяти	
3.	Семинар (видео). Инфраструктура разработки ПО. Системы непрерывной интеграции и доставки, примеры таких систем: Gitlab, Gitea и Jenkins	1
4.	Семинар (видео). Контейнерная виртуализация, работа с docker-контейнерам. Применение контейнеров в системе сборки ПО	1
5.	Практическое занятие (самостоятельное выполнение задания и отчетность за 1 учебный день). Настройка автоматизированной сборки ПО с применением контейнерной виртуализации. Настройка CI/CD, создание контейнера, развертывание необходимых библиотек, настройка компилятора для получения релизной и отладочной сборок, сборка с внедрением датчиков срабатывания ошибок	3
19 апреля		
6.	Лекция (видео). Типы ошибок в программном коде. Переполнение буфера на стеке. Переполнение буфера на куче. Неинициализированные переменные. Целочисленные переполнения. Использование памяти после освобождения	1
7.	Семинар (видео). Сбор покрытия кода при тестировании. Средства сбора покрытия в компиляторах gcc и llvm. Средства визуализации покрытия кода	1
8.	Семинар (видео). Фаззер afl++, его практическое применение для тестирования Linux-приложения на архитектуре x86-64. • начальный набор образцов для мутации (корпус входных данных); • интерфейсы ввода данных (файл, сеть, пользовательский ввод); • перманентный (persistent) режим работы фаззера	2
9.	Лекция (видео). Фаззинг-тестирование в сертификационных испытаниях, уровни доверия, требования к технологическому уровню применяемых фаззеров. Оформление протоколов в части фаззинг-тестирования	1
10.	Практическое занятие (самостоятельное выполнение задания и отчетность за 2 учебный день). Интеграция тестирования в CI/CD, запуск функциональных тестов, сбор и анализ покрытия кода	3
20 апреля		
11.	Лекция (видео). Фаззинг-тестирование библиотек, инструмент Libfuzzer	1
12.	Лекция (видео). Фаззинг-тестирование сетевых протоколов без состояния и с состоянием	1
13.	Семинар (видео). Распараллеливание фаззинг-тестирования: • изоляция ПО при параллельном фаззинг-тестировании при помощи виртуальных машин QEMU/KVM;	1

№ п/п	Вид и тема занятия	Время, час
	• параллельный запуск фаззинг-тестирования на нескольких виртуальных машинах, обеспечение идентичности условий работы путем тиражирования образа QEMU, содержащего настроенный образец тестируемого ПО	
14.	Лекция (видео). Подходы к фаззинг-тестированию интерпретируемых языков	2
15.	Практическое занятие (самостоятельное выполнение задания и отчетность за 3 учебный день). Сборка тестируемого ПО со статическим инструментированием и датчиками ошибок, подготовка корпуса входных данных, запуск фаззера в несколько потоков, анализ прироста покрытия	3

Очный этап обучения

№ п/п	Вид и тема занятия	Время, час
	24 апреля	
16.	Семинар. Инструмент Crusher: • пре- и пост-обработка данных в фаззере; • состояния программы и их учет в ходе фаззинг-тестирования	1
17.	Практическое занятие. Практическая работа по фаззинг-тестированию программ, написанных на интерпретируемых языках: • фаззинг-тестирование программы на python; • фаззинг-тестирование программы на C#; • фаззинг-тестирование программы на java	2
18.	Лекция. Основы динамического символьного исполнения. Математический аппарат символьного исполнения: символьные переменные, выражение семантики кода, символьные ограничения. Предикаты пути. Затраты на построение трасс (замедление выполнения). Чистое символьное исполнение и гибридное конкретно-символьное выполнение (concolic execution)	1
19.	Семинар. Инструменты динамического символьного выполнения Sydr и sydr_fuzz	1
20.	Семинар. Фаззинг-тестирование программного кода, выполняющегося в привилегированном режиме: • особенности фаззинг-тестирования ядра ОС Linux; • инструмент Syzkaller	2
21.	Практическое занятие. Практическое фаззинг-тестирование ядра ОС и драйверов: • фаззинг-тестирование ядра и драйверов ОС семейства Linux	1
	25 апреля	
22.	Лекция. Особенности фаззинг-тестирования, проводимого в технологическом центре исследования безопасности ядра Linux	2
23.	Итоговая аттестация Зачет с оценкой, практическое занятие	6

Приложение 1

Методические рекомендации по организации учебного курса

Обучение состоит из заочного и очного этапов.

На заочном этапе обучения слушатели самостоятельно изучают переданные им учебно-методические материалы (видеозаписи лекций и семинаров, слайды в формате PDF), выполняют задания в рамках практических занятий. Материалы для изучения заблаговременно передаются слушателям посредством электронной почты либо выкладываются для скачивания на Интернет-ресурсах. Удаленное консультирование ведется через закрытую телеграмм-группу.

Практические занятия заочного этапа используются для текущего контроля. Каждый слушатель получает индивидуальное задание, представляющее собой сквозной пример: модельный объект оценки, отражающий характерные свойства ПО СЗИ. Каждый день заочного обучения завершается практическим занятием, успешное выполнение которого демонстрирует усвоение теоретических и практических знаний. Слушатель должен направить по окончании практического занятия полученные результаты на почту fuzz@ispras.ru. Сквозной пример используется в трех практических занятиях, где последовательно от занятия к занятию добавляются манипуляции с объектом оценки, в итоге приводящие к выполнению фаззинг-тестирования объекта оценки в типовом окружении промышленной разработки ПО.

Практическое занятие 1.

В виртуальной машине требуется развернуть и настроить CI/CD систему Jenkins.

Требуется подготовить образ docker-контейнера, развернуть в нем пакеты-зависимости.

Включить в конфигурацию Jenkins сборку объекта оценки в docker-контейнере в виде релизной и отладочной сборок.

Настроить опции компилятора, обеспечивающие внедрение датчиков срабатывания ошибок, добиться успешной компиляции, при необходимости внося исправления в исходный код объекта оценки и скрипты сборки.

Подготовить отчетные материалы в соответствии с требованиями полученного задания: описать ключевые особенности подготовленного стенда, описать внесенные в код ПО и скрипты сборки изменения.

Практическое занятие 2.

Используется стенд, подготовленный на практическом занятии 1.

В конфигурацию конвейера CI/CD системы добавляется этап функционального тестирования.

Отладочная сборка дорабатывается с целью включения в собираемое ПО механизма сбора покрытия кода (gcov/lcov, llvm-cov).

Обеспечить выполнение штатных для объекта оценки функциональных тестов со сбором покрытия кода. Проанализировать собранное покрытие с целью определения качества применяемого набора функциональных тестов.

Подготовить отчетные материалы в соответствии с требованиями полученного задания: описать ключевые особенности сбора покрытия, описать особенности текущего покрытия кода тестами.

Практическое занятие 3.

Используется стенд, доработанный на практическом занятии 2.

Конфигурация конвейера CI/CD системы дорабатывается с целью получения отладочной сборки для фаззинг-тестирования, в которую встроен инструментальный код нескольких механизмов: датчики срабатывания ошибок, сбор покрытия, обратная связь с фаззером.

В конфигурацию конвейера CI/CD системы добавляется этап фаззинг-тестирования.

На основе штатных функциональных тестов подготавливается корпус образцов данных для фаззинг-тестирования.

В docker-контейнере разворачивается инструмент фаззинг-тестирования (afl++, …), обеспечивается выполнение тестирования в несколько потоков отладочной сборки ПО.

Подготовить отчетные материалы в соответствии с требованиями полученного задания: описать ключевые события процесса фаззинг-тестирования, при выявлении сработавших ошибок – включить в отчетные материалы соответствующую диагностическую информацию:

- последовательность всех команд, выполненных из Linux-консоли;
- снимки экрана по результатам сборки с инструментированием фаззера, снимки экрана с графическим интерфейсом запущенного фаззера с открытыми путями, участка графического представления отчета фаззера с покрытием;
- набор входных образцов;
- набор выходных образцов;
- покрытие в формате html-отчета lcov.

После завершения заочного этапа освоение программы продолжается очно на базе ИСП РАН.

Завершается обучение итоговой аттестацией в форме зачета с оценкой, предусматривающего выполнение практического задания.