ГОУ ВПО Российско-Армянский (Славянский) университет

Утверждено
Директор Института
Математики и Информатики
Дарбинян А.А.
«18» июня 2024, протокол №15

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ

Наименование дисциплины: Безопасность и надежность программного обеспечения

Авторы: ***канд.физ.-мат.наук Саргсян Севак Сеникович***
Матевосян Гагик Завенович

Направление подготовки: 01.04.02 Прикладная математика и информатика

ЕРЕВАН

Структура и содержание УМКД

1. **Аннотация**
   1.1. Взаимосвязь дисциплины с другими дисциплинами учебного плана специальности (направления)
   В рамках курса студенты будут изучать основы безопасной разработки программ. Будут изучаться известные программные ошибки и способы защиты программ от таких ошибок. Для прохождения данного курса необходимо базовые знания по C/C++.
   1.2. Требования к исходным уровням знаний, умений и навыков студентов для прохождения дисциплины (что должен знать, уметь и владеть студент для прохождения данной дисциплины)
   *Для удачного прохождения курса студенты должны **знать** язык программирования C, основы управления памятью, и основы создания веб-приложений. Студенты должны **уметь** писать программы на языке C и работать в среде операционной системы Линукс. **Навыки** работы с базами данных облегчит освоение курса.*
   1.3. Предварительное условие для прохождения (дисциплина(ы), изучение которых является необходимой базой для освоения данной дисциплины)
   Для прохождения этого курса необходимо пройти дисциплину алгоритмы и алгоритмические языки (язык C), программирование в среде Линукс.
2. **Содержание**
   2.1. Цели и задачи дисциплины
   Цель данного курса научить студентов основам безопасной разработки программ. Объяснить, причину появления программных ошибок, способы их эксплуатации, а также способы защиты программ от таких ошибок.
   2.2. Требования к уровню освоения содержания дисциплины (какие компетенции (знания, умения и навыки) должны быть сформированы у студента ПОСЛЕ прохождения данной дисциплины)
   После удачного прохождения курса студенты должны:
   - ***Знать** основные типы программных ошибок, причины появления уязвимостей в программах, существующие методы защиты на уровне ОС и компиляторов.*
   - ***Уметь** создавать безопасные ПО, анализировать причины ошибок при работе с памятью. Уметь создавать безопасные веб-приложения.*
   - *Владеть **навыками** разработки безопасной ПО.*

2.3.2. Распределение объема дисциплины по темам и видам учебной работы

Разделы и темы дисциплины	**Всего (ак. часов)**	**Лекции (ак. часов)**	**Практ. занятия (ак. часов)**	**Семина-ры (ак. часов)**	**Лабор. (ак. часов)**	**Другие виды занятий (ак. часов)**
1	**2=3+4+5+6+7**	**3**	**4**	**5**	**6**	**7**
Модуль 1.						
Введение	**2**	**2**				
Раздел 1. Безопасное программирование для языков программирования С/С++						
Тема1. Представление программы в памяти, Организация вызовов функций.	**2**	**2**				
Тема 2. Ошибка переполнения стека. внедрение вредоносного кода.	**4**	**4**				
Тема 3. Способы автоматической защиты программ.	**2**	**2**				
Тема 4. Переполнение буфера в куче. Ошибки форматной строки.	**4**	**4**				
Тема 5. Возвратно-ориентированное программирование.	**2**	**2**				
Тема 6. Разработка безопасного ПО.	**2**	**2**				
Раздел 2. Безопасность веб-приложений.						
Тема 7. Введение в веб. Модель клиент-сервер. Протокол http. Сессия и куки.	**2**	**2**				
Тема 8. Введение в базу данных. Ошибка sql-injection. Причины появления ошибки. Примеры эксплуатации ошибки. Защита программ от внедрения sql кода.	**2**	**2**				

Тема 9. Ошибка CSRF. Причины появления ошибки. Примеры эксплуатации ошибки. Защита программ.	**2**	**2**				
Тема 10. Ошибка XSS. Причины появления ошибки. Примеры эксплуатации ошибки. Защита программ.	**2**	**2**				
Тема 11. Классификация ошибок. Системы CWE, CVE. Организация owasp. Советы по безопасной разработки. Ограничение и валидация входных данных. Ограничение привилегий программ. Примеры.	**4**	**4**				
Раздел 3. Цикл безопасной разработки ПО						
Тема 12. Статический анализ программ. Чувствительный к потоку анализ. Чувствительный к контексту анализ. Taint анализ.	**4**	**4**				
Тема 13. Символьное выполнения. Решатели (solvers). Автоматическая генерация тестов.	**2**	**2**				
Тема 14. Динамический анализ программ. Фазеры. Метод белого, серого и черного ящика. Использование фазера в цикле безопасной разработки программ.	**2**	**2**				
ИТОГО	**36**	**36**				

2.3.3 Содержание разделов и тем дисциплины

Модуль 1

Введение

Раздел 1 Безопасное программирование для языков программирования C/C++

Тема 1. Представление программы в памяти, Организация вызовов функций.

Представление программы в памяти. Сегменты кода, стека и кучи. Помещение локальных переменных в стек сегменте. Организация вызовов функций. Передача аргументов. Работа в среде линукс. Gdb дебаггер.

Bryant, O'Hallaron Computer Systems [7]

Тема 2. Ошибка переполнения стека. Внедрение вредоносного кода

Ошибка переполнения стека. Примеры. Эксплуатация ошибки - изменение потока управления программы, вызов системных функций, внедрение вредоносного кода. Реальные примеры эксплуатации ошибок.

Secure coding in C++ [1-2]

Тема 3. Способы автоматической защиты программ.

Невыполняемая память. Канарейка стека. Метод ASLR. ASLR для 32 и 64 битных архитектур. Способы обходов существующие защиты.

Secure coding in C++ [1-2]

Тема 4. Переполнение буфера в куче. Ошибки форматной строки.

Переполнение ошибки в куче. Ошибки форматной строки. Эксплуатация ошибок на тестовых программах**.** Переполнение виртуальных таблиц C++. Ошибка двойного освобождения памяти. Использование памяти после освобождения. Атака unlink. Переполнение целых чисел.

Secure coding in C++ [3-4]

Тема 5. Возвратно-ориентированное программирование

Введение в возвратно-ориентированное программирование. Способы и ограничения. Метод слепой ROP атаки.

Тема 6. Разработка безопасного ПО.

Ограничение и валидация входных данных. Ограничение привилегий программ. Примеры.

Secure Programming Cookbook for C/C++ [1-3]

Раздел 2. Раздел 2. Безопасность веб-приложений.

Тема 7. Введение в разработку веб-приложений

Введение в веб. Модель клиент-сервер. Протокол http/htpps. Сессия и куки.

The Web Application Hacker's Handbook [1-3]

Тема 8. Ошибка sql-injection

Введение в базу данных. Ошибка sql-injection. Причины появления ошибки. Примеры эксплуатации ошибки. Защита программ от внедрения sql кода.

The Web Application Hacker's Handbook [8-9]

Тема 9. Ошибка CSRF

Причины появления уязвимости CSRF. Примеры эксплуатации ошибки. Защита программ

The Web Application Hacker's Handbook [10-13]

Тема 10. Ошибка XSS.

Ошибка XSS. Причины появления ошибки. Примеры эксплуатации ошибки. Защита программ.

The Web Application Hacker's Handbook [10-13]

Тема 11. Классификация ошибок

Классификация ошибок. Системы CWE, CVE. Организация owasp. Советы по безопасной разработки. Ограничение и валидация входных данных. Ограничение привилегий программ. Примеры.

Раздел 3. Цикл безопасной разработки ПО

Тема 12. Статический анализ программ

Статический анализ программ. Чувствительный к потоку анализ. Чувствительный к контексту анализ. Taint анализ.

Secure Programming with Static Analysis [1-4]

Тема 13. Символьное выполнения

Введение в символьное выполнения. Решатели (solvers). Автоматическая генерация тестов.

Тема 14. Динамический анализ

Динамический анализ программ. Фазеры. Метод белого, серого и черного ящика. Использование фазера в цикле безопасной разработки программ.

Fuzzing for Software Security Testing and Quality Assurance [1-4]

2.3.4 Краткое содержание семинарских/практических занятий и лабораторного практикума

Организация атак на тестовых программах. Примеры валидации входных данных

2.3. Материально-техническое обеспечение дисциплины

Компьютеры с доступом интернет