РОССИЙСКАЯ АКАДЕМИЯ НАУК

Федеральное государственное бюджетное учреждение науки
Институт системного программирования
Российской академии наук

«УТВЕРЖДАЮ»

Директор ИСП РАН
академик РАН,
д.ф.-м.н., профессор
_____ В.П.Иванников
«___»_____ 2012 г.

РАБОЧАЯ ПРОГРАММА

УЧЕБНОЙ ДИСЦИПЛИНЫ
«Статический анализ программ и проблема обнаружения дефектов в ПО»

для подготовки аспирантов по специальности
05.13.11 - Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Москва 2012

1. АННОТАЦИЯ

Проблема обнаружения дефектов в ПО в настоящее время является одной из наиболее актуальных. В курсе дается обзор методов статического анализа программ и рассматривается применение этих методов для обнаружения уязвимостей и критических ошибок. Некоторые типы ошибок, обнаруживаемых статическими анализаторами: неопределённое поведение (неинициализированные переменные, обращение к NULL-указателям); типичные сценарии, приводящие к недокументированному поведению (стандартная библиотека языка Си известна большим количеством неудачных технических решений: такие функции, как например, **`gets`**, в принципе небезопасны, функции **`sprintf`** и **`strcpy`** безопасны лишь при определённых условиях); переполнение буфера - когда компьютерная программа записывает данные за пределами выделенного в памяти буфера. В Интернете доступны многочисленные публикации, слайды и учебные пособия (на английском языке), но они требуют знания основ статического анализа, без которых невозможно продуктивное изучение этих материалов.

2. ЦЕЛИ И ЗАДАЧИ КУРСА

Цель курса **–** изучение статического анализа потоков данных и его применения для обнаружения дефектов программ, которые могут привести к нарушению безопасности не только рассматриваемой программы, но и всей вычислительной системы, на которой она выполняется. Такие ошибки встречаются на редко выполняемых («нетипичных») трассах, что затрудняет их обнаружение в процессе тестирования и отладки.

Задачами данного курса являются:

- освоение слушателями базовых знаний по статическому анализу потоков данных - локальному, глобальному (в пределах процедуры) и межпроцедурному;
- освоение слушателями эвристических методов статического анализа, обеспечивающих нахождение дефектов в программах ;
- освоение слушателями принципов проектирования, реализации и сопровождения системных программных средств обнаружения дефектов как на стадии аудита уже существующего ПО, так и на стадии разработки нового ПО;
- оказание консультаций и помощи аспирантам в проведении собственных исследований и разработок в областях, использующих подходы, рассматриваемые в курсе.

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ПОСЛЕВУЗОВСКОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ (АСПИРАНТУРА))

Дисциплина «Статический анализ программ и проблема обнаружения дефектов в ПО» относится к дисциплинам по выбору учебного плана подготовки аспирантов по научной специальности 05.13.11 «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей».

Для успешного изучения курса аспиранту необходимо знать общесистемное программное обеспечение современных компьютеров и основы компиляторных технологий, а также уметь работать с персональной ЭВМ.

Основные положения дисциплины будут использованы при подготовке к кандидатскому экзамену по научной специальности 05.13.11 «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей», в научно-

исследовательской работе и при выполнении диссертации на соискание ученой степени кандидата физико-математических наук.

4. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ СОДЕРЖАНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины «Статический анализ программ и проблема обнаружения дефектов в ПО» обучающийся должен:

Знать:

- фундаментальные понятия современных компиляторных технологий;
- структуру и состав современных оптимизирующих компиляторных сред (примеры – GCC, LLVM и др.);
- цели, задачи и методы машинно-независимого статического анализа программ для выявления в них дефектов и для выполнения других видов программной инженерии (например, понимания программ);
- принципы разработки и применения сред статического анализа программ.

Уметь:

- разрабатывать, обосновывать и реализовывать новые методы и алгоритмы статического анализа исходного и бинарного представлений программ;
- разрабатывать, обосновывать и реализовывать новые промежуточные представления программ;
- использовать существующие и разрабатывать новые компиляторные среды как основу для решения различных задач обратной инженерии, защиты программного кода, обнаружения дефектов в программах и др.

Владеть:

- навыками самостоятельной работы в Интернете;
- культурой разработки и реализации системного программного обеспечения современных компьютеров;
- навыками грамотной разработки новых языков программирования и их программного обеспечения.

5. Содержание и структура курса

5.1 Содержание разделов курса

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Введение. Проблема безопасности ПО и статический анализ	Проблема безопасности ПО как проблема программной инженерии. Необходимость выявления дефектов ПО (в частности, уязвимостей и ошибок в работе с динамической памятью). Возможности статического анализа по решению проблем безопасности. Обзор курса.	Т

2	Методы статического анализа	Основные этапы статического анализа. Промежуточное представление программ и его построение: лексический анализ, синтаксический анализ, контекстный анализ. Структура таблицы символов. Атрибуты и их вычисление. Аннотации. Построение базовых блоков: выявление «лидеров», анализ достигающих определений, анализ живого кода. Анализ базовых блоков (ориентированный ациклический граф, нумерация значений). Вычисление и уточнение атрибутов с помощью анализа потока данных в пределах процедуры. SSA-форма. Выделение областей для ускорения анализа. Межпроцедурный статический анализ. Граф вызовов и способы его обхода. Понятие контекстно-чувствительного анализа. Анализ указателей.	Т
3	Применение методов статического анализа для аудита ПО	Классификация дефектов. Чекеры и их применение. Компромиссы между точностью анализа, его глубиной и масштабируемостью. Понятие стиля программирования (выделение безопасного подмножества языка программирования). Применение статического анализа для выявления нарушений принятого стиля.	Т
4	Выявление уязвимостей и других дефектов в анализируемом ПО (на примере уязвимости «переполнение буфера»)	Анализ уязвимости «переполнение буфера». Чем опасно возможное переполнение буфера? Понятие злонамеренного кода. Примеры. Стратегии выделения буферов (статические и динамические), исключающие возможность переполнения. Стратегии отслеживания размеров буфера (явные и неявные). Анализ интервалов. Наследование уязвимостей. Опасные функции **`gets()`**, **`scanf()`** и др. Анализ примеров. Ограниченные операции со строками. Наиболее распространенные ошибки при использовании ограниченных функций. Последствия отбрасывания «лишних» значений. Другие похожие уязвимости. Форматные строки и связанные с ними ошибки. Классическая атака с использованием уязвимостей форматной строки. Методы обнаружения и предотвращения переполнения буфера. Использование ограничений стиля для исключения уязвимостей.	Т

5	Архитектура системы обнаружения уязвимостей и других дефектов ПО (на примере среды ***Svace***)	Обзор систем обнаружения уязвимостей. Классификация систем. Наиболее успешные системы. Архитектура и особенности реализации системы ***Svace.***	Т
6	Заключение	Дефекты в программном обеспечении и ограничения, связанные с методикой статического анализа исходного кода	Т

5.2 Структура курса

Общая трудоемкость курса составляет 2,5 зачетные единицы (90 часов).

Вид работы	Трудоемкость, часов
	2 курс
Общая трудоемкость	**90**
Аудиторная работа:	**32**
Лекции (Л)	32
Практические занятия (ПЗ)	-
Лабораторные работы (ЛР)	-
Самостоятельная работа:	**58**
Самостоятельное изучение разделов	-
Самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, выполнение практических заданий)	58
Вид итогового контроля (зачет, экзамен)	**Кандидатский экзамен**

Трудоемкость отдельных разделов курса.

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Вне-ауд. работа
			Л	ПЗ	ЛР	СР
1	Введение. Проблема безопасности ПО и статический анализ	4	2	-	-	2
2	Методы статического анализа	22	8	-	-	14
3	Применение методов статического анализа для аудита ПО	16	6	-	-	10

4	Выявление уязвимостей и других дефектов в анализируемом ПО (на примере уязвимости «переполнение буфера»)	22	8	-	-	14
5	Архитектура системы обнаружения уязвимостей и других дефектов ПО (на примере среды *Svace*)	22	6			16
6	Заключение	4	2			2
	Итого:	90	32	-	-	58

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы аспирантов

Форма контроля знаний:
- кандидатский экзамен по специальности.

Контрольно-измерительные материалы

На кандидатском экзамене аспирант должен продемонстрировать знания в объеме основной программы кандидатского экзамена по специальности 05.13.11 «Математическое обеспечение вычислительных машин, комплексов и компьютерных сетей», а также дополнительной программы, в которую, в зависимости от выбранной аспирантом специализации, могут входить вопросы, рассматриваемые в данном курсе.

Перечень контрольных вопросов для дополнительной программы:

1. Возможности статического анализа по решению проблем безопасности.

2. Построение абстрактного синтаксического дерева и таблицы символов: лексический, синтаксический и контекстный анализ. Построение промежуточного представления.

3. Построение и предварительный анализ графа потока управления.

4. Построение SSA-формы промежуточного представления.

5. Межпроцедурный статический анализ. Граф вызовов и способы его обхода. Анализ указателей.

6. Понятие контекстно-чувствительного анализа. Методы обеспечения требуемого уровня чувствительности к контексту.

7. Классификация дефектов. Компромиссы между точностью анализа, его глубиной и масштабируемостью.

8. Понятие стиля программирования (выделение безопасного подмножества языка программирования).

9. Применение статического анализа для выявления нарушений принятого стиля.

10. Анализ уязвимости «переполнение буфера». Чем опасно возможное переполнение буфера? Понятие злонамеренного кода.

11. Стратегии выделения буферов (статические и динамические), исключающие возможность переполнения.

12. Стратегии отслеживания размеров буфера (явные и неявные). Анализ интервалов.

13. Наследование уязвимостей. Опасные функции **`gets()`**, **`scanf()`** и др.

14. Ограниченные операции со строками и наиболее распространенные ошибки при их использовании. Последствия отбрасывания «лишних» значений.

15. Форматные строки и связанные с ними ошибки. Классическая атака с использованием уязвимостей форматной строки.

16. Методы обнаружения и предотвращения переполнения буфера.

17. Классификация и сравнительный анализ систем обнаружения уязвимостей. Наиболее успешные системы.

18. Архитектура и особенности реализации системы

19. Способы расширения возможностей***.***

20. Сравнение системы ***Svace*** с другими системами обнаружения дефектов.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КУРСА

7.1 Рекомендуемая литература

7.1.1 Основная литература

1. Brian Chess and Jacob West. Secure Programming with Static Analysis. Pearson Education, Inc. 2007, ISBN: 0-321-42477-8.

7.1.2 Дополнительная литература

1. Aho, Alfred V., Ravi Sethi, Jeffrey D. Ullman, Monica Lam. Compilers: Principles, Techniques, and Tools, 2nd Edition. Boston, MA: Addison-Wesley, 2006. (Есть русский перевод, 2007)
2. Chess, B. “Improving Computer Security Using Extended Static Checking.” // Proceedings of the 2002 IEEE Symposium on Security and Privacy (Oakland, CA, 2002), 118–130
3. Kratkiewicz, K. “Evaluating Static Analysis Tools for Detecting Buffer Overflows in C Code.” Master’s thesis, Harvard University, // 2005. http://www.ll.mit.edu/IST/pubs/050610_Kratkiewicz.pdf.

7.1.3. Пособия и методические указания.

1. Слайды лекций

8. Материально-техническое обеспечение курса

Для получения необходимой информации и самостоятельной работы аспирантов используются web-ресурсы Интернет и локальная библиотека электронных материалов.

В компьютерных классах ИСП РАН (ауд. 109) аспиранты могут самостоятельно ознакомиться с программным обеспечением, используемым для обнаружения уязвимостей и других дефектов ПО.

Программу составил к.ф.–м.н., доцент Гайсарян С.С.

Программа принята на заседании Ученого Совета ИСП РАН

протокол № 2012-5 от 23 мая 2012 г.