**25. Анализаторы программного кода.**

Для обнаружения уязвимостей защиты в программах применяют следующие инструментальные средства:

Статический анализ кода - это анализ программного обеспечения, который производится над исходным кодом программ и реализуется без реального исполнения исследуемой программы.

Динамические отладчики. Инструменты, которые позволяют производить отладку программы в процессе её исполнения.

Статические анализаторы указывают на те места в программе, в которых возможно находится ошибка. Эти подозрительные фрагменты кода могут, как содержать ошибку, так и оказаться совершенно безопасными.

Программное обеспечение часто содержит разнообразные уязвимости из-за ошибок в коде программ. Ошибки, допущенные при разработке программ, в некоторых ситуациях приводят к сбою программы, а, следовательно, нарушается нормальная работа программы: при этом часто возникает изменение и порча данных, останов программы или даже системы. Большинство уязвимостей связано с неправильной обработкой данных, получаемых извне, или недостаточно строгой их проверкой.

Классификация уязвимостей защиты в зависимости от программных ошибок:

Переполнение буфера (buffer overflow). Эта уязвимость возникает из-за отсутствия контроля за выходом за пределы массива в памяти во время выполнения программы.

Уязвимости (tainted input vulnerability). Уязвимости могут возникать в случаях, когда вводимые пользователем данные без достаточного контроля передаются интерпретатору некоторого внешнего языка (обычно это язык Unix shell или SQL).

Ошибки форматных строк (format string vulnerability). Данный тип уязвимостей защиты является подклассом уязвимости . Он возникает из-за недостаточного контроля параметров при использовании функций форматного ввода-вывода

Уязвимости как следствие ошибок синхронизации (race conditions). Проблемы, связанные с многозадачностью, приводят к ситуациям, называемым : программа, не рассчитанная на выполнение в многозадачной среде

**26. Утилиты автоматической сборки проекта.**

Автоматизация сборки — этап процесса разработки программного обеспечения, заключающийся в автоматизации широкого спектра задач, решаемых программистами в их повседневной деятельности.

Включает в себя такие действия, как:

-компиляция исходного кода в Объектный модуль

-сборка бинарного кода в исполняемый файл

-выполнение тестов

-разворачивание программы на производственной платформе

-написание сопроводительной документации или описание изменений новой версии

Преимущества

Улучшение качества продукта

Ускорение процесса компиляции и линковки

Избавление от излишних действий

Минимизация «плохих (некорректных) сборок»

Избавление от привязки к конкретному человеку

Ведение истории сборок и релизов для разбора выпусков

Экономия времени и денег благодаря причинам, указанным выше

Базовые требования:

Частые или ночные сборки для своевременного выявления проблем.

Поддержка управления зависимостями исходного кода

Обработка разностной сборки

Уведомление при совпадении исходного кода с имеющимися бинарными файлами.

Ускорение сборки

Отчет о результатах компиляции и линковки.

Дополнительные требования:

Создание описания изменений и другой сопутствующей документации.

Отчет о статусе сборки

Отчет об успешном/неуспешном прохождении тестов.

Суммирование добавленных/измененных/удаленных особенностей в каждой новой сборке

**27. Отладчики программ.**

Отла́дчик — компьютерная программа, предназначенная для поиска ошибок в других программах, ядрах операционных систем, SQL-запросах и других видах кода. Отладчик позволяет выполнять трассировку, отслеживать, устанавливать или изменять значения переменных в процессе выполнения кода, устанавливать и удалять контрольные точки или условия остановки и т.д.

**28. Программы создания инсталляторов.**

Практика разработки коммерческого программного обеспечения показывает, что далеко не все пользователи умеют работать с архивами. Поэтому программы рекомендуется поставлять в виде исполняемых файлов, которые автоматически создают необходимые папки в файловой системе, копируют туда файлы программы, создают необходимые файлы настроек или ключи в реестре, а также пункты меню запуска программы и ярлыки на рабочем столе. Для упрощения создания инсталляторов существует много специализированных программных продуктов.

**29. Редакторы ресурсов.**

Создают и обрабатывают файлы ресурсов, которые после обработки могут быть скомпилированы и включены в исполнимый модуль. Эти программы специфичны для платформы Win.

Характеристики:

- состав поддерживаемых ресурсов;

- возможность работы с нестандартными ресурсами;

- возможности импорта и экспорта ресурсов.

Представители:

Borland Resource Workshop, eXe Scope, Resource Builder , Resource Explorer

**30. Программы профилировщики.**

Профилирование — сбор характеристик работы программы, таких как время выполнения отдельных фрагментов (обычно подпрограмм), число верно предсказанных условных переходов, число кэш-промахов и т. д. Инструмент, используемый для анализа работы, называют профилировщиком или профайлером.Обычно выполняется совместно с оптимизацией программы.

**31. Программы поддержки версий.**

Программы поддержки версий

Дают возможность централизованно отслеживать изменения в проекте, создавать независимые новые версии программных модулей, автоматически разносить изменения по копиям проекта программной системы и т.д.

Представители: Visual Source Safe (Microsoft Cor.), Nexus Safe Source; Subversion.

**32. Программы создания файлов помощи.**

Позволяют создавать файлы помощи, автоматизировать документирование.

Характеристики:

- форматы поддерживаемых выходных файлов (hlp, chm, html, pdf, …);

- средства, необходимые для работы с файлами документации;

- возможность конвертирования из других распространенных форматов;

- возможность структурирования информации в файле помощи;

- возможность организации поиска по документации;

- возможность интеграции в существующие среды (например, для библиотеки COM – объектов возможность вызывать справку по ней при нажатии F1, если курсор стоит на объекте из этой библиотеки);

- возможность автоматической генерации помощи (или заготовки) по описанию библиотеки, COM-объекта.

Представители: RoboHelp (), Anet Help Tool, Help and Manual, Visual CHM, Adobe Acrobat (Adobe); HTML HelpWorkshop.

**33. Программы отслеживания активности и изменений в системе.**

Позволяют отслеживать действия программ по изменению реестра, файловой системы, вызовов системных сервисов и т.д. Следят за загруженностью системы в целом.

Характеристики:

- тип отслеживаемых изменений/активности;

- возможность протоколирования (логирования);

- возможность фильтрации получаемой информации;

- возможность уведомления.

Представители:

Microsoft: Spy++, Process Viewer, ApiMon, SysMon (для Win2000/XP – ActiveX компонент для mmc).

Winternals Systems (Mark E. Russinovich): RegMon, FileMon, HandleEx.

**34. Программы-вериферы и контейнеры.**

Создают виртуальную среду для отдельных классов программ, в которой можно исследовать поведение программы).

Представители: Driver Verifier, ActiveX Control Test Container (Microsoft Corp.)

**35. История развития CASE-средств.**

cредства автоматизации разработки программ (CASE-средства) — инструменты автоматизации процессов проектирования и разработки программного обеспечения для системного аналитика, разработчика ПО и программиста. Первоначально под CASE-средствами понимались только инструменты для упрощения наиболее трудоёмких процессов анализа и проектирования, но с приходом стандарта ISO/IEC 14102 CASE-средства стали определять, как программные средства для поддержки процессов жизненного цикла ПО.

**36. Классификация CASE- средств.**

Типичными CASE-инструментами являются:

-инструменты управления конфигурацией;

-инструменты моделирования данных;

-инструменты анализа и проектирования;

-инструменты преобразования моделей;

-инструменты редактирования программного кода;

-инструменты рефакторинга кода;

-генераторы кода;

-инструменты для построения UML-диаграмм.

средства анализа — предназначены для построения и анализа модели предметной области;

средства проектирования баз данных;

средства разработки приложений;

средства реинжиниринга процессов;

средства планирования и управления проектом;

средства тестирования;

средства документирования.

**37. Инструментальные средства Telelogic.**

Инструментальное средство Telelogic DOORS предназначено для автоматизации управления требованиями в жизненном цикле программных средств.

Telelogic TAU предназначено для автоматизации работ и задач процесса разработки программных средств и систем. Данное семейство предоставляет визуальную среду разработки, основанную на использовании универсального языка моделированияUML, с возможностью последующей автоматической код генерации.

Инструментальное средство Telelogic SYNERGY предназначено для автоматизации управления изменениями и конфигурацией программных средств. Его использование позволяет упростить организацию коллективной работы над проектом и обеспечить контроль текущего состояния проекта с учетом возможных изменений в требованиях заказчика, в функциях разрабатываемых программных продуктов, исправлений ошибок в продуктах.

Инструментальное средство Telelogic DocExpress предназначено для поддержкипроцесса документирования жизненного цикла программных средств (см. подразд. 1.2 данного обучающего курса).Telelogic DocExpress позволяет автоматизировать поступление из различных источников исходных данных для документирования, поддерживать исходные данные и документацию в актуальном состоянии, выполнять автоматическое форматирование документов в соответствии с действующими стандартами на документацию, упрощает сопровождение документов.

**38. Инструментальные средства Computer Associates.**

Компания Computer Associates (www.ca.com) предлагает комплексы инструментальных средств поддержки различных процессов ЖЦ ПО:

AllFusion Modeling Suite - интегрированный комплекс CASE-средств [16], включающий следующие продукты:

AllFusion Process Modeler (BPwin) - функциональное моделирование;

AllFusion ERwin Data Modeler (ERwin) - моделирование данных;

AllFusion Component Modeler (Paradigm Plus) - объектно-ориентированный анализ и проектирование с использованием UML и возможностью генерации кода;

AllFusion Model Manager (Model Mart) - организация совместной работы команды разработчиков;

AllFusion Data Model Validator (ERwin Examiner) - проверка структуры и качества моделей данных.

AllFusion Change Management Suite - комплекс средств управления конфигурацией и изменениями.

AllFusion Process Management Suite - средства управления процессами и проектами для различных типов приложений.

**39. Генераторы документации. Автоматизированное создание программной документации**

Генератор документации — программа или пакет программ, позволяющая получать документацию, предназначенную для программистов (документация на API) и/или для конечных пользователей системы, по особому образу комментированному исходному коду и, в некоторых случаях, по исполняемым модулям (полученным на выходе компилятора).

Обычно генератор анализирует исходный код программы, выделяя синтаксические конструкции, соответствующие значимым объектам программы (типам, классам и их членам/свойствам/методам, процедурам/функциям и т. п.). В ходе анализа также используется метаинформация об объектах программы, представленная в виде документирующих комментариев. На основе всей собранной информации формируется готовая документация, как правило, в одном из общепринятых форматов — HTML, HTMLHelp, PDF, RTF и других.

**40. Средства непрерывной интеграции. Непрерывная интеграция изменений программного обеспечения. Учет изменений.**

Непрерывная интеграция — практика разработки программного обеспечения, которая заключается в постоянном слиянии рабочих копий в общую основную ветвь разработки (до нескольких раз в день) и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем. В обычном проекте, где над разными частями системы разработчики трудятся независимо, стадия интеграции является заключительной. Она может непредсказуемо задержать окончание работ. Переход к непрерывной интеграции позволяет снизить трудоёмкость интеграции и сделать её более предсказуемой за счёт наиболее раннего обнаружения и устранения ошибок и противоречий, но основным преимуществом является сокращение стоимости исправления дефекта, за счёт раннего его выявления.

Обратимся к принципам методологии гибкой разработки программного обеспечения и выделим те, на которые непосредственно влияет или может повлиять процесс непрерывной интеграции:

- удовлетворение потребностей заказчика благодаря ранней и регулярной поставке программного обеспечения;

- изменение требований даже на поздних этапах разработки;

- учащение выпуска обновлений продукта;

- повышение мотивации команды за счет повышения технологичности процесса разработки;

- стабильный и работающий продукт – основной показатель прогресса;

- устойчивый и постоянный ритм процесса разработки;

- постоянное внимание к техническому совершенству и качеству проектирования.

**41. Средства автоматизированного тестирования. Программы автоматизированного тестирования приложений, систем, баз данных**

IBM Rational Robot – универсальное средство автоматизации тестирования общего назначения для команд разработчиков, выполняющих функциональное тестирование клиент-серверных приложений. Дает возможность обнаруживать неполадки в ПО благодаря расширению сценариев тестирования средствами условной логики, позволяющей целиком охватить тестируемое приложение. Robot позволяет создавать сценарии тестирования с вызовом внешних библиотек DLL или исполняемых модулей.

IBM Rational Performance Tester – инструмент нагрузочного и стрессового тестирования, с помощью которого можно выявлять проблемы системной производительности и их причины. Позволяет создавать тесты без написания кода и не требуя навыков программирования. Обеспечивает гибкие возможности моделирования и эмуляции различных пользовательских нагрузок. Выполняет сбор и интеграцию данных о серверных ресурсах с данными о производительности приложений, получаемыми в режиме реального времени.

IBM Rational Functional Tester – набор средств автоматизированного тестирования, позволяющих выполнять функциональное и регрессионное тестирование, тестирование пользовательского интерфейса и тестирование, управляемое данными. Инструмент применяет технологию ScriptAssure (бесшовная проверка достоверности динамических данных) и функции поиска соответствия по шаблону, позволяющие повысить устойчивость сценариев тестирования в условиях частых изменений пользовательских интерфейсов приложений. Тестировщики могут выбрать язык сценариев для разработки и настройки тестов: Java в среде Eclipse или Microsoft Visual Basic .Net в среде Visual Studio .Net.

IBM Rational Quality Manager – решение для реализации процессов управления тестированием и качеством, поддерживает сотрудничество участников групп по разработке программных продуктов, предоставляя им возможность обмениваться информацией, применять средства автоматизации для сокращения графиков выполнения проектов, а также составлять отчеты по проектным показателям для принятия обоснованных решений. Rational Quality Manager может быть дополнен средством управления ресурсами тестирования Rational Test Lab, обеспечивающим учет ресурсов тестирования (серверов), их бронирование, автоматизацию развертывания тестовой среды на сервере и запуск скриптов тестирования, а также отчетность по использованию ресурсов тестирования.

Rational Quality Manager и Rational Test Lab созданы на базе открытой платформы Jazz, которая предоставляет стандартные интерфейсы и удобные возможности для интеграции с решениями партнеров и других производителей.