СОДЕРЖАНИЕ ПРОЕКТИРОВАНИЯ

Проектирование системы защиты информации заключается в разработке на основе выработанной политики безопасности технических требований по системе защиты информации и архитектуре автоматизированной системы, а также разработке проектной документации системы защиты информации

Основная цель - разработка оптимальной комплексной системы, как по используемым технологиям, так и по наилучшему соотношению цены к получаемой степени безопасности.

Эффективность комплексной системы защиты информации может быть достигнута, если все ее компоненты представлены качественными решениями, функционируют как единый комплекс и имеют централизованное управление

СОСТАВ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ

- подсистема защиты от несанкционированного доступа
- подсистема управления учетными записями и правами доступа
- подсистема комплексной антивирусной защиты
- подсистема межсетевого экранирования
- подсистема обнаружения вторжений, контроля и анализа защищенности
- подсистема криптографической защиты
- подсистема управления средствами защиты информации
- подсистема обеспечения безопасности коммутируемой инфраструктуры и беспроводных сетей
- подсистема контроля использования информационных ресурсов
- подсистема управления событиями и инцидентами ИБ
- подсистема контроля эффективности защиты информации
- подсистема обеспечения непрерывности функционирования средств защиты

ОСНОВНЫЕ ШАГИ ПРОЕКТИРОВАНИЯ

- разработка решений по архитектуре системы защиты информации (СЗИ)
- разработка средств ЗИ и контроля
- разработка технического проекта СЗИ
- разработка рабочей документации СЗИ
- подготовка и оформление технической документации на поставку технических и программных средств для СЗИ
- проектирование помещений АС с учетом требований нормативных документов по защите информации
- разработка порядка сопровождения СЗИ в АС
- разработка порядка и этапов внедрения СЗИ, консультирование Заказчика при внедрении СЗИ

ПОДХОДЫ К ВСТРАИВАНИЮ СЗИ В АС

Первый подход - разработка и внедрение новых информационно-вычислительных систем (АС), в рамках которых решается весь комплекс проблем информационной безопасности. Этот подход является наиболее перспективным. Однако в процессе функционирования уже созданных ЗАС может понадобиться встраивание новых элементов защиты.

Второй подход - разработка подсистем защиты информации, объединение их в единую систему защиты, налагаемую на уже созданную АС, в которой изначально не предусматривался полный комплекс защитных механизмов. На практике данный подход до настоящего времени является достаточно распространенным, поскольку большое число широко используемых ОС, СУБД и других информационных систем изначально были созданы без должного учёта проблем защиты информации.

Основные конструкции структурного принципа

- **функциональный блок**
- **конструкция обобщенного цикла**
- **конструкция принятия двоичного решения**

Принцип модульного проектирования

Преимущества:

- **упрощается отладка программ, так как ограниченный доступ к модулю и однозначность его внешнего проявления исключают влияние ошибок в других, связанных с ним, модулях на его функционирование;**

- **обеспечивается возможность организации совместной работы больших коллективов разработчиков, так как каждый программист имеет дело с независимой от других частью программы;**

- **повышается качество программы, так как относительно малый размер модулей и, как следствие, небольшая сложность их позволяют провести более полную проверку программы.**

Ключевые требования «хорошей спецификации» (стандарт IEEE 830-1998)

Unambiguous (недвусмысленность) — отсутствие лексических, синтаксических и семантических ошибок

Complete (полнота) — должны быть описаны все значимые области требований

Verifiable (проверяемость) — должны содержаться только те требования, которые могут быть численно измерены

Consistent (целостность) — не должно быть конфликтов требований

Modifiable (модифицируемость) —спецификация должна быть легкой в использовании и организации ссылок между требованиями

Traceable (трассируемость) — спецификация должна позволять пошагово отслеживать (трассировать) от требований до предыдущих принятых решений, от документов, расширяющих спецификацию (проектировка и т.д.) к требованиям текущего состояния спецификации

Usable (возможность применения) — спецификация должна без излишних деталей описывать весь жизненный цикл системы

Рекомендуемая структура SRS (стандарт IEEE 830-1998)

- **Введение**
- **Общее описание**
- **Функциональность системы**
- **Требования к внешним интерфейсам**
- **Нефункциональные требования**
- **Прочие требования**

Основные подходы в определении спецификаций

- **спецификация как описание**
- **спецификация как предписание**
- **договорная методология**
- **спецификация как модель**

Основные отличительные черты моделей при описании системы

- **хорошее сочетание нисходящего и восходящего подходов к их разработке с возможностью выбора абстрактного описания;**
- **возможность описания параллельной, распределенной и циклической работы;**
- **возможность выбора различных формализованных аппаратов для описания систем.**

Формальное проектирование алгоритмов

Базируется на языках алгоритмических логик, которые включают высказывание вида:

$$Q \{S\} R ,$$

<u>читается следующим образом:</u>

"если до исполнения оператора S было выполнено условие Q, то после него будет R".

Здесь Q - предусловие,
R - постусловие.

Предусловие и постусловие - предикаты.

Преимущества представления алгоритма в виде преобразователя предикатов

Предоставляют возможности:

- **анализировать алгоритмы как математические объекты;**
- **дать формальное описание алгоритма, позволяющее интеллектуально охватить алгоритм;**
- **синтезировать алгоритмы по представленным спецификациям;**
- **провести формальное верифицирование алгоритма, т.е. доказать корректность его реализации.**

Методы формальной разработки и доказательства корректности алгоритмов

- **разработка алгоритма проводится методом последовательной декомпозиции, с разбивкой общей задачи, решаемой алгоритмом, на ряд более мелких подзадач;**

- **критерием детализации подзадач является возможность их реализации с помощью одной конструкции ветвления или цикла;**

- **разбиение общей задачи на подзадачи предусматривает формулирование пред- и постусловий для каждой подзадачи с целью их корректного проектирования и дальнейшей верификации.**

Функциональные возможности компонентов ТСВ

- осуществление взаимодействие с аппаратным обеспечением АС;
- обеспечение защиты памяти;
- реализация функции файлового ввода-вывода;
- обеспечение управление процессами.

Этапы разработки защищённой АС

- **определение политики безопасности;**
- **проектирование модели АС;**
- **разработка кода АС;**
- **обеспечение гарантий соответствия реализации заданной политике безопасности.**