Программно-аппаратные средства обеспечения информационной безопасности

Лекция № 4

Безопасное взаимодействие в АС. Идентификация и аутентификация.

Рахманенко И.А.

План

- Безопасное взаимодействие в АС.
- Идентификация и аутентификация
- Типовые схемы аутентификации
- Аутентификация до загрузки ОС
- Контроль и управление доступом

Безопасное взаимодействие в АС

- Для создания АС с гарантированно выполненной ПБ необходимо реализовать МБО, поддерживающий эту ПБ и МБС, гарантирующий ПБ, и замкнуть каким-либо образом субъекты АС в ИПС.

Безопасное взаимодействие в АС

- С точки зрения оптимизации трудозатрат на реализацию защитных механизмов целесообразно максимально использовать средства, которые уже реализованы в КС, в необходимых случаях усиливая и дополняя их.

Проблема сопряжения штатных и дополненных средств защиты

Проблемы

- Сопряжение субъектов ОС, осуществляющих аутентификацию с дополняемыми субъектами, получившими информацию от модулей аутентификации
- Гарантии передачи параметров между модулями аутентификации и модулями реализации и гарантирования ПБ (МБО и МБС) без нарушения условий корректности субъектов
- Организация структур, обеспечивающих хранение данных для работы модулей аутентификации и сопряжения с другими объектами КС без нарушения ПБ.

Вопросы

1. Формализация процедур аутентификации пользователей АС и описание ее характеристик.
2. Формализация процедур сопряжения субъектов (для решения задач передачи параметров от модулей аутентификации к модулям реализации и поддержания ПБ).
3. Описание процедур сопряжения различных аутентифицирующих объектов.
4. Формализация и описание процедур использования внешних субъектов для поддержания защищенности АС.
5. Методика анализа попарной корректности субъектов.

ПРОЦЕДУРА ИДЕНТИФИКАЦИИ И АУТЕНТИФИКАЦИИ

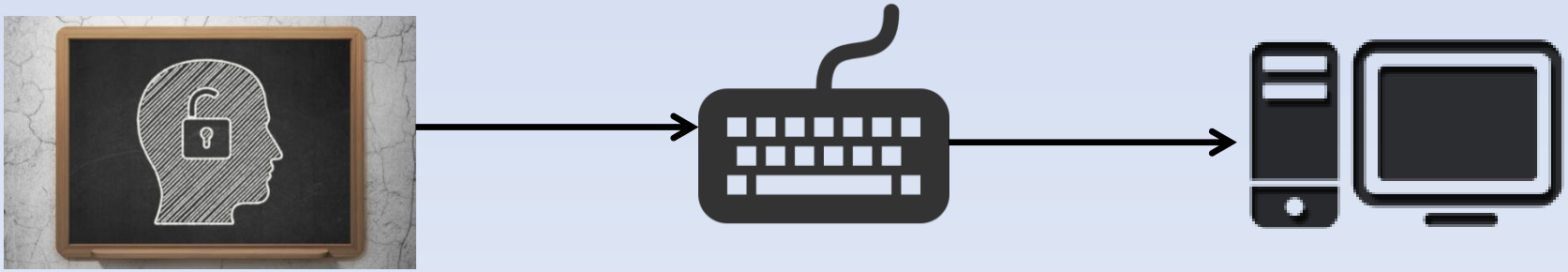
Объекты, аутентифицирующие пользователя

- Внешний аутентифицирующий объект, не принадлежащий АС
- Внутренний аутентифицирующий объект, принадлежащий АС, в который переносится информация из внешнего объекта

Дополнительно : субъект переноса информации от внешнего к внутреннему объекту (пример - драйвер клавиатуры).

Пример

- Символьный пароль для входа в систему - в “памяти пользователя”
- Путем набора на клавиатуре переносится в буфер программы запроса пароля (объект оперативной памяти ПЭВМ).



Содержание аутентифицирующего объекта

- $\mathbf{ID_i}$ – неизменяемый идентификатор i-го пользователя, который является аналогом имени и используется для идентификации пользователя
- $\mathbf{K_i}$ – аутентифицирующая информация пользователя, которая может изменяться и служит для аутентификации.

Первичная аутентифицирующая информация

- Совокупную информацию в аутентифицирующем объекте будем называть *первичной аутентифицирующей информацией* i-го пользователя.
- Описанная структура соответствует практически любому устройству, служащему для опознания пользователя

Примеры

- Touch Memory (TM) имеет 8 байт неперезаписываемого неповторяющегося серийного номера, который однозначно характеризует кокретную TM и некоторый объект перезаписываемой памяти, который может содержать Ki

Примеры

- Пластиковые карты - выделяется неизменяемая информация первичной персонализации пользователя, соответствующая ID_i, и объект в файловой структуре карты, содержащий K_i
- Аналогично: eToken.

Требования

- Внутренний аутентифицирующий объект не должен существовать в КС длительное время (большее времени работы конкретного пользователя).
- Для постоянного хранения необходимо использовать некую преобразованную информацию от первичной (хэш).

ТИПОВЫЕ СХЕМЫ АУТЕНТИФИКАЦИИ

Эталон для идентификации и аутентификации

- $E_i = F (ID_i, K_i)$, где F – функция, для которой можно качественно описать свойство "невосстановимости" Ki по E_i и ID_i.
- "Невосстановимость" Ki описывается некоторой пороговой трудоемкостью T_o решения задачи восстановления аутентифицирующей информации по E_i и ID_i, ниже которой не должна опускаться ни одна оценка трудоемкости нахождения K_i для всех известных алгоритмов данной задачи.

Объект-эталон для схемы 1

	Информация для идентификации	Информация для аутентификации
1	ID_1	E_1
2	ID_2	E_2
…	…	…
n	ID_n	E_n

Схема 1

1. Пользователь предъявляет свой идентификатор (имя) ID.
2. Если ID не совпадает ни с одним ID_i, зарегистрированным в АС, то идентификация отвергается – пользователь не допущен к работе, иначе идентификация проходит.
3. У пользователя субъектом аутентификации запрашивается аутентификатор K.
4. Субъектом аутентификации вычисляется $Y=F(ID_i,K)$.
5. Субъектом аутентификации производится сравнение E_i и Y. При совпадении фиксируется событие " пользователь аутентифицирован", информация о пользователе передается в МБО, считываются необходимые для реализации заданной ПБ массивы данных, в противном случае аутентификация отвергается.

Объект-эталон для схемы 2

	Информация для идентификации	Информация для аутентификации
1	ID_1, S_1	E_1
2	ID_2, S_2	E_2
…	…	…
n	ID_n, S_n	E_n

Ei=F(Si, Ki), Si – случайный вектор, заданный при создании пользователя

Схема 2

1. Пользователь предъявляет свой идентификатор (имя) ID.
2. Если ID не совпадает ни с одним ID_i, зарегистрированным в АС, то идентификация отвергается.
3. По ID_i выделяется S_i.
4. У пользователя субъектом аутенфикации запрашивается аутентификатор K.
5. Субъектом аутентификации вычисляется $Y=F(S_i,K)$.
6. Субъектом аутентификации производится сравнение E_i и Y. При совпадении фиксируется факт успешной аутентификации, информация о пользователе передается в МБО, считываются необходимые для реализации заданной ПБ массивы данных.

Пример

- Вторая схема - в ОС Unix.
- ID - имя пользователя (запрошенное по Login)
- Ki – пароль пользователя (запрошенный по Password)
- F - алгоритм шифрования DES
- Эталоны для идентификации и аутентификации содержатся в файле Etc/passwd.

Аутентификация на основе пароля

- Часто пароля выбираются из множества осмысленных слов
- Необходимы жесткие требования к паролям
- Для повышения надежности защиты необходимо добавить аппаратную аутентификацию (добавляется 2-ой фактор)

АУТЕНТИФИКАЦИЯ ДО ЗАГРУЗКИ ОС

Аутентификация до загрузки ОС

Возможна в 2 вариантах:

- На уровне расширений BIOS
- На уровне загрузчика ОС

Уровень расширений Bios

- В ЭВМ на платформе Intel, первичная активизация вычислительных ресурсов компьютера производится кодом процессора, хранящемся в основном Bios.
- При включении питания код основного Bios "проецируется" в область памяти F000 и управление передается на точку входа, определенную производителем Bios

Уровень расширений Bios

- Код Bios производит тестирование оборудования, инициализацию векторов прерываний, активизацию видеосистемы и др., зависящие от специфики Bios
- В состав кода Bios входит типовая процедура поиска так называемых расширений Bios (Bios Extention).
- Управление передается в ходе процедуры поиска расширений

Поиск расширений

- Сканирование с шагом 512 байт области памяти с C000 до F000 с целью нахождения двухбайтовой сигнатуры 55AA.
- После нахождения этой сигнатуры анализируется следующий (третий начиная с 55) байт, который указывает область расширения Bios в 512-байтных страницах (или блоках).
- Если в указанной позиции находится число, отличное от 0, то вычисляется арифметическая байтовая контрольная сумма от области памяти с байта 55 на длину, указанную в третьем байте.
- В случае совпадения этой суммы с 0 на четвертый (от первого байта 55) байт передается управление.

Уровень расширений Bios

- С учетом того, что объем расширения Bios не может быть очень большим, то на этом уровне может быть реализован достаточно небольшой объем значимых для безопасности функций.

Функции, реализуемые в расширениях

- Идентификация и аутентификация пользователя (возможно, с использованием специфического аппаратного носителя;
- Запрет несанкционированной загрузки ОС с избранных носителей (например с CD-ROM);
- Контроль неизменности или целостности аппаратной или программной компоненты ЭВМ.

Программирование расширений Bios

- Первый расширенный Bios, код которого будет исполнен, – это расширение, проецируемое видеокартой (VideoBios).
- Возможно программирование расширенного Bios.
- Программирование - на языке низкого уровня
- Изменение состояний переменных программы при ее неизменном размещении в ПЗУ невозможно -> нужно перемещение в ОЗУ.

Программирование расширений Bios

- Существует платы с местом размещения ПЗУ или флеш (например сетевые карты)
- Средства защиты (например, плата АККОРД), которые дают возможность перепрограммироваия кода расширений Bios)
- Можно заложить необходимый механизм парольной идентификации и аутентификации пользователей (например, троекратный запрос пароля).

АУТЕНТИФИКАЦИЯ ДО ЗАГРУЗКИ ОС. УРОВЕНЬ ЗАГРУЗЧИКОВ ОС

Уровень загрузчиков ОС

Возможно решение задач:

- “Ранняя" идентификация и аутентификация пользователей (при отсутствии аппаратных средств защиты)
- Защита от несанкционированной загрузки ОС
- Получение специального вида загрузочных носителей.

“Ранняя" идентификация и аутентификация

- Не всегда процедуры идентификации и аутентификации удается реализовать на этапе инициализации аппаратной компоненты компьютера (в частности, невозможно реализовать указанные процедуры в расширении Bios).
- Удается выполнить идентификацию и аутентификацию на ранней стадии сеанса работы пользователя (Программно).

Защита от несанкционированной загрузки ОС

- Используют тонкости обработки загрузки с внешних носителей либо преобразуют (например, шифруют) информацию на несъемных носителях компьютера.
- В первом случае загрузка с внешних носителей операционной системы невозможна физически
- Во втором – даже при успешной загрузке с несанкционированной копии ОС информация недоступна.

Получение специального вида загрузочных носителей

- Загрузка ОС возможна только при наличии данного носителя.

Решение

- В общем случае - программирование модифицированного загрузчика (или загрузчиков) операционной системы.

Необходимые операции

- Заместить первичный код загрузчика собственным фрагментом;
- Сохранить исходный код загрузочного сектора (в случае необходимости его выполнения);
- С учетом необходимости размещения первичного загрузчика по тому же адресу, что и модифицированного, обеспечить корректное перемещение модифицированного загрузчика в другую область памяти без потери управления.

КОНТРОЛЬ И УПРАВЛЕНИЕ ДОСТУПОМ. ПРОИЗВОЛЬНОЕ УПРАВЛЕНИЕ ДОСТУПОМ

Контроль и управление доступом

- Основная задача - ограничение операций, выполняемых зарегистрированными пользователями в системе.
- Два основных механизма управления доступом – дискреционный (произвольный) и мандатный (нормативный).

Произвольное управление доступом

- Основа - матрица прав доступа
- Строки - субъекты (пользователи, процессы)
- Столбцы – объекты (файлы, каталоги и т.п.)
- В ячейках - права доступа субъектов к объектам

Матрица доступа

Файлы\ Пользователи	F1	F2	F3	F4	F5
Лапин		R		R	
Котов	RW		R		
Волков		RW			
Майоров	C	C	C	C	C

R – права доступа пользователя по чтению;

W – права доступа пользователя по записи;

C – управление доступом для других пользователей

Варианты реализации

- Списки прав доступа
- Биты доступа
- “Парольная” защита.

“Парольная” защита

- Пользователь использует отдельный пароль для доступа к каждому объекту в системе
- Неудобства, связанные с запоминанием паролей.

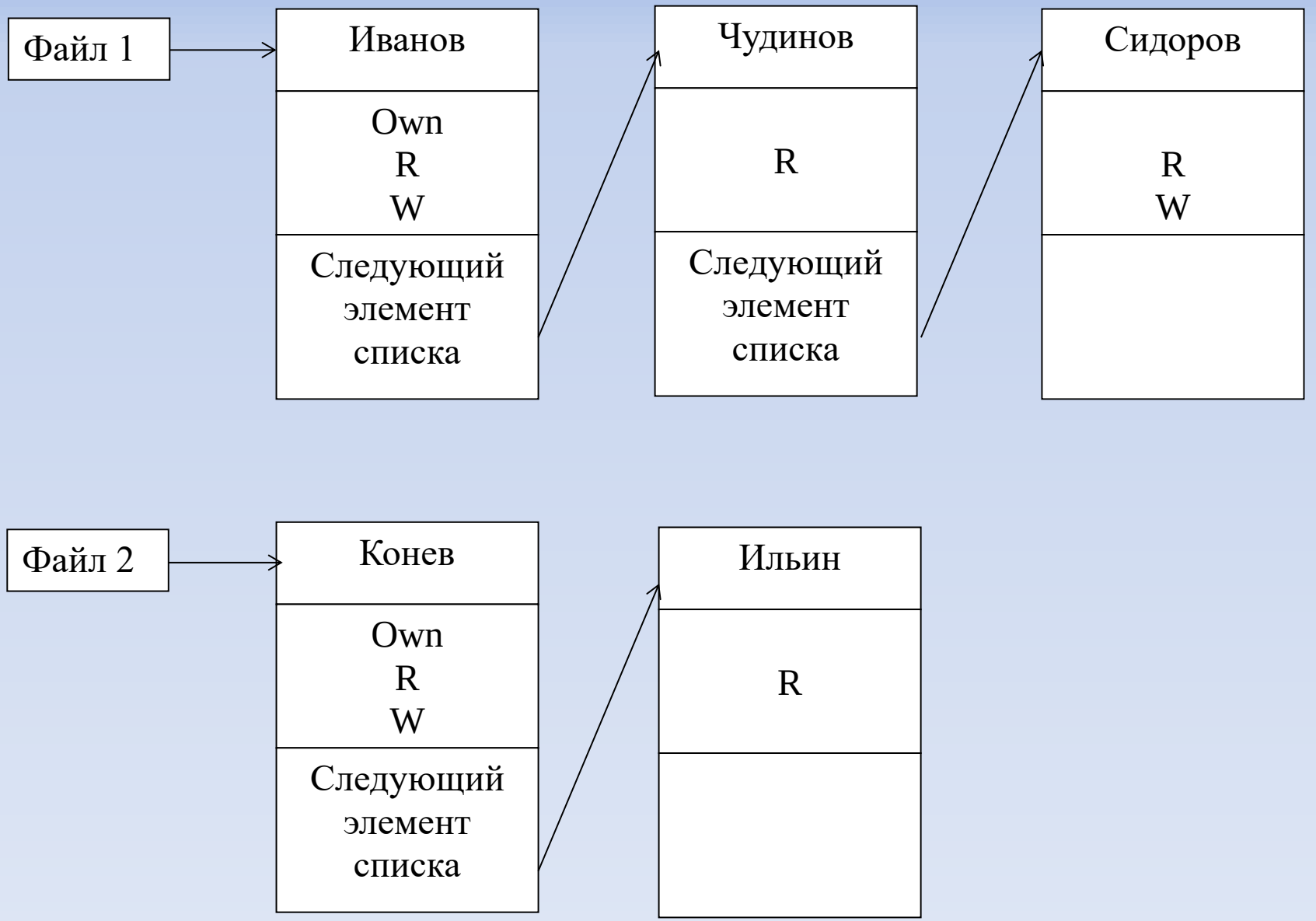
Списки прав доступа

- Access Control List – ACL;
- С каждым объектом ассоциируется список пользователей с указанием их прав доступа к объекту;
- При принятии решения о доступе, соответствующий объекту доступа ACL проверяется на наличие прав, ассоциированных с идентификатором пользователя, запрашивающего доступ, или его группы.

Списки прав доступа

Основные недостатки:

- Большие временные затраты на обработку списков (по сравнению с битами защиты)
- Необходимость разрешения противоречий между отдельными элементами списка.


Файл 1
Иванов
Own
R
W
Следующий элемент списка
Чудинов
R
Следующий элемент списка
Сидоров
R
W
Файл 2
Конев
Own
R
W
Следующий элемент списка
Ильин
R

Биты защиты

- Привязаны к объектам.
- Указывают права доступа для трех категорий пользователей (ОС Unix): все пользователи (world), члены группы владельца (group) и владелец (owner).
- Может изменять только владелец объекта и администратор.

Биты защиты

Владелец			Группа			Все пользователи		
Чтение	Запись	Выполн	Чтение	Запись	Выполн	Чтение	Запись	Выполн
1	2	3	4	5	6	7	8	9

1…9 – номер бита

Проверка доступа

1. Проверяется, является ли субъект собственником объекта. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если идентификаторы не равны, то осуществляется переход ко второму шагу алгоритма.
2. Проверяется, входит ли субъект в группу владельца. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется. Если идентификаторы не равны, то осуществляется переход к третьему шагу алгоритма.

Проверка доступа

3. Сравниваются полномочия, предоставленные всем пользователям системы с запрашиваемым типом доступа. Если запрашиваемый тип доступа присутствует в соответствующем поле, то доступ предоставляется.

Биты защиты

- Недостатком использования механизма битов защиты является неполная реализация произвольного контроля доступа, т.к. доступ к объекту нельзя разрешить или запретить для отдельных пользователей.
- В современных системах часто используются комбинации списков контроля доступа и битов защиты.

КОНТРОЛЬ И УПРАВЛЕНИЕ ДОСТУПОМ. НОРМАТИВНОЕ УПРАВЛЕНИЕ ДОСТУПОМ

Нормативное управление доступом

- Полностью запрещает передачу прав доступа между пользователями. Полномочное – нет.
- Позволяет решить проблему "троянских коней”
- Модели Белла-Лападула и Биба.

Нормативное управление доступом

- Объектам задается метка секретности, субъектам – уровень доступа.
- Запрещается запись в объекты более низкого уровня и чтение из объектов более высокого уровня, чем уровень доступа субъекта.

Пример

- ОС UTS MLS (доработанная Linux)
- Информация о метке секретности объекта содержится в битах доступа файла

Владелец	Метка секретности	Все пользователи

- Одновременно работает произвольный доступ (поле все пользователи)

Основные защитные механизмы ОС семейства Windows

- идентификация и аутентификация пользователя при входе в систему;
- разграничение прав доступа к файловой системе (дискреционная модель доступа);
- аудит (регистрация событий).

Недостатки защитных механизмов ОС семейства Windows

- невозможно обеспечить замкнутость (целостность) программной среды;
- не в полном объеме реализуется дискреционная модель доступа (системный процесс);
- невозможно встроенными средствами гарантированно удалять остаточную информацию;
- нет возможности контроля целостности файловой системы.

СЗИ от НСД

- ПАК Аккорд
- Secret Net
- Панцирь
- СЗИ Аура
- Dallas Lock

Мобильная система Вооружённых Сил

ОС МСВС — защищённая операционная система общего назначения. Предназначена для построения стационарных защищённых автоматизированных систем. Принята на снабжение в ВС РФ в 2002 году.

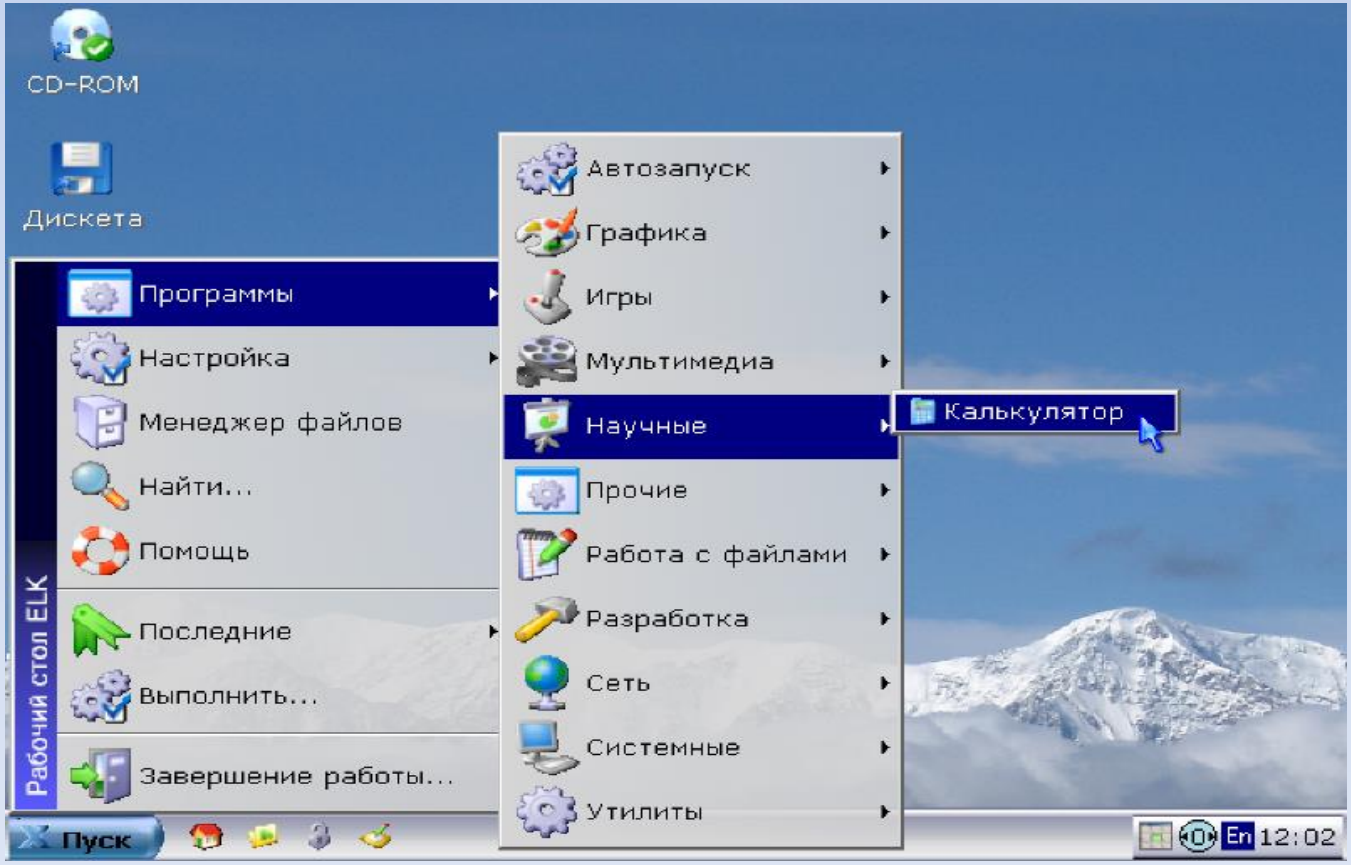
МСВС 5.0 создана на основе Red Hat Enterprice Linux

Мобильная система Вооружённых Сил

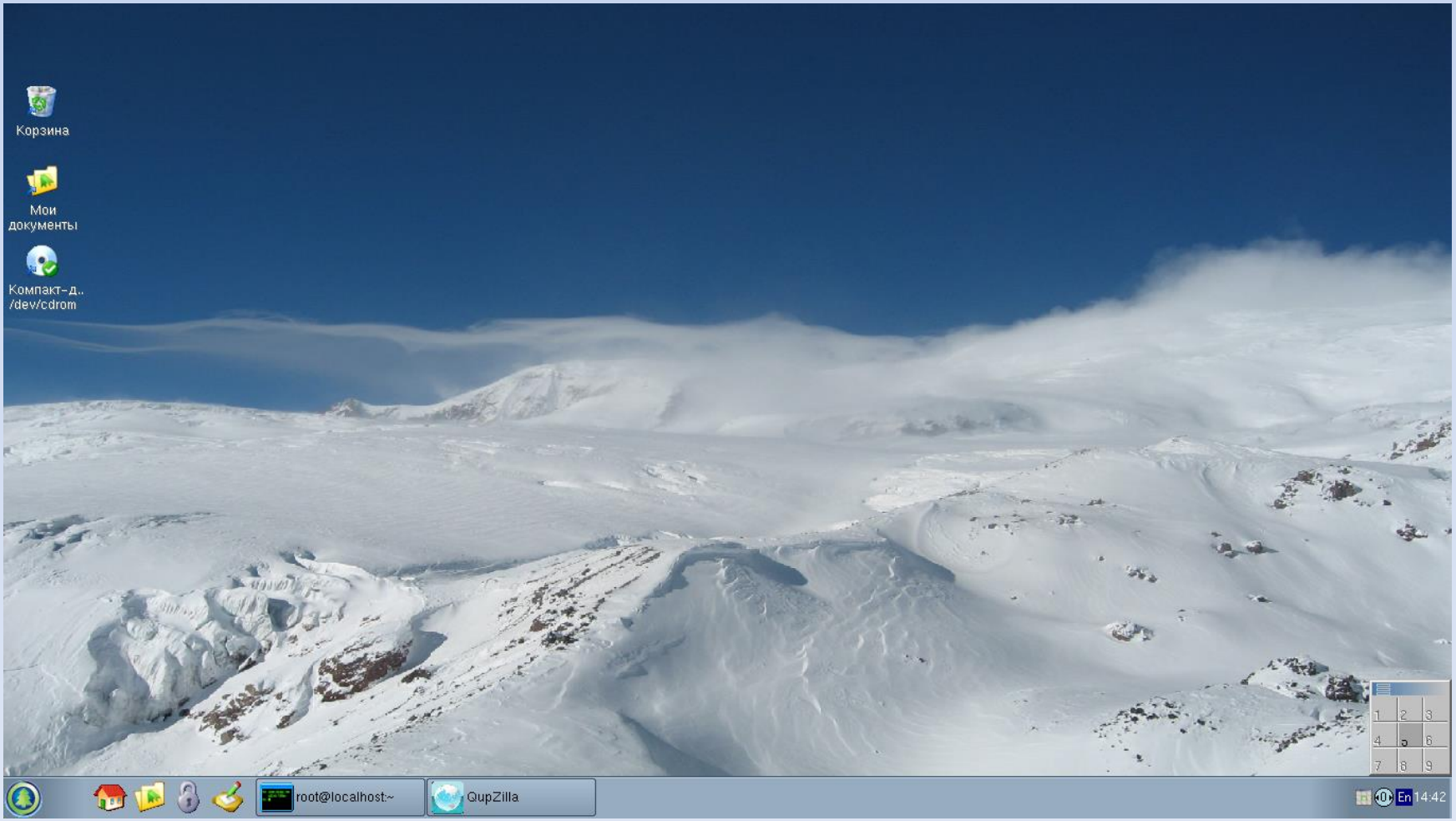
ОС МСВС 3.0 сертифицирована по требованиям безопасности информации:

- 2 класс защищённости информации от НСД
- 1 уровню классификации контроля отсутствия недекларированных возможностей

Мобильная система Вооружённых Сил



Мобильная система Вооружённых Сил



Рассмотренные вопросы

- Безопасное взаимодействие в АС.
- Идентификация и аутентификация
- Типовые схемы аутентификации
- Аутентификация до загрузки ОС
- Контроль и управление доступом