

Лабораторная работа №3

Дискреционное разграничение прав в Linux. Два пользователя

Якушевич Артём Юрьевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	11
5	Ответы на контрольные вопросы	12
6	Список литературы	14

Список иллюстраций

3.1	Функция, шифрующая данные	7
3.2	Результат работы функции, шифрующей данные	8
3.3	Функция, дешифрующая данные	8
3.4	Результат работы функции, дешифрующей данные	9
3.5	Сравнение ключей	10

Список таблиц

1 Цель работы

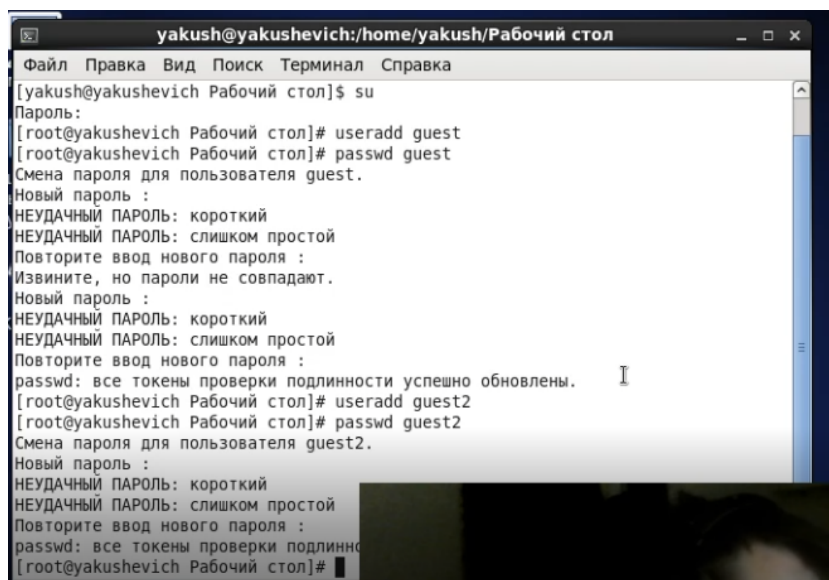
Освоить на практике применение режима однократного гаммирования [1].

2 Задание

1. Написать программу, которая должна определить вид шифротекста при известном ключе и известном открытом тексте
2. Также эта программа должна определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста

3 Выполнение лабораторной работы

1. Написал функцию шифрования, которая определяет вид шифротекста при известном ключе и известном открытом тексте “С Новым Годом, друзья!”.
Ниже представлены функция, шифрующая данные (рис - @fig:001), а также работа данной функции (рис - @fig:002).



```
yakush@yakushevich:/home/yakush/Рабочий стол
Файл Правка Вид Поиск Терминал Справка
[yakush@yakushevich Рабочий стол]$ su
Пароль:
[root@yakushevich Рабочий стол]# useradd guest
[root@yakushevich Рабочий стол]# passwd guest
Смена пароля для пользователя guest.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
Повторите ввод нового пароля :
Извините, но пароли не совпадают.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
Повторите ввод нового пароля :
passwd: все токены проверки подлинности успешно обновлены.
[root@yakushevich Рабочий стол]# useradd guest2
[root@yakushevich Рабочий стол]# passwd guest2
Смена пароля для пользователя guest2.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
Повторите ввод нового пароля :
passwd: все токены проверки подлинности успешно обновлены.
[root@yakushevich Рабочий стол]#
```

Рис. 3.1: Функция, шифрующая данные

```

[root@yakushevich Рабочий стол]# useradd guest2
[root@yakushevich Рабочий стол]# passwd guest2
Смена пароля для пользователя guest2.
Новый пароль :
НЕУДАЧНЫЙ ПАРОЛЬ: короткий
НЕУДАЧНЫЙ ПАРОЛЬ: слишком простой
Повторите ввод нового пароля :
passwd: все токены проверки подлинности успешно обновлены.
[root@yakushevich Рабочий стол]#

```

Рис. 3.2: Результат работы функции, шифрующей данные

2. Написал функцию дешифровки, которая определяет ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста (рис - @fig:003). А также представил результаты работы программы (рис - @fig:004).

```

passwd: все токены проверки подлинности успешно обновлены.
[root@yakushevich Рабочий стол]# gpasswd -a guest2 guest
Adding user guest2 to group guest
[root@yakushevich Рабочий стол]#

```

Рис. 3.3: Функция, дешифрующая данные

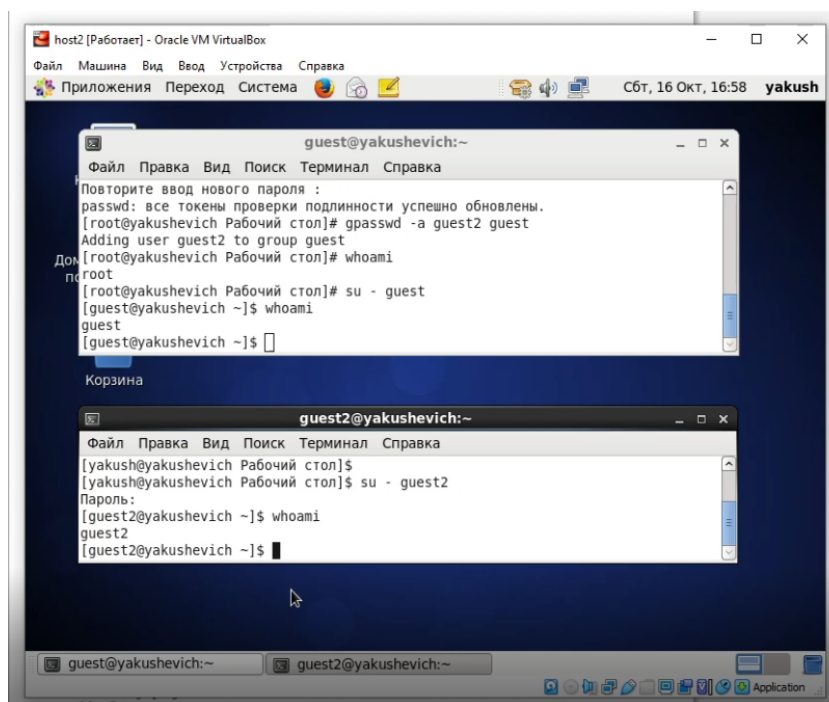


Рис. 3.4: Результат работы функции, дешифрующей данные

Сравнение ключей, полученных с помощью первой и второй функций (рис - @fig:005).

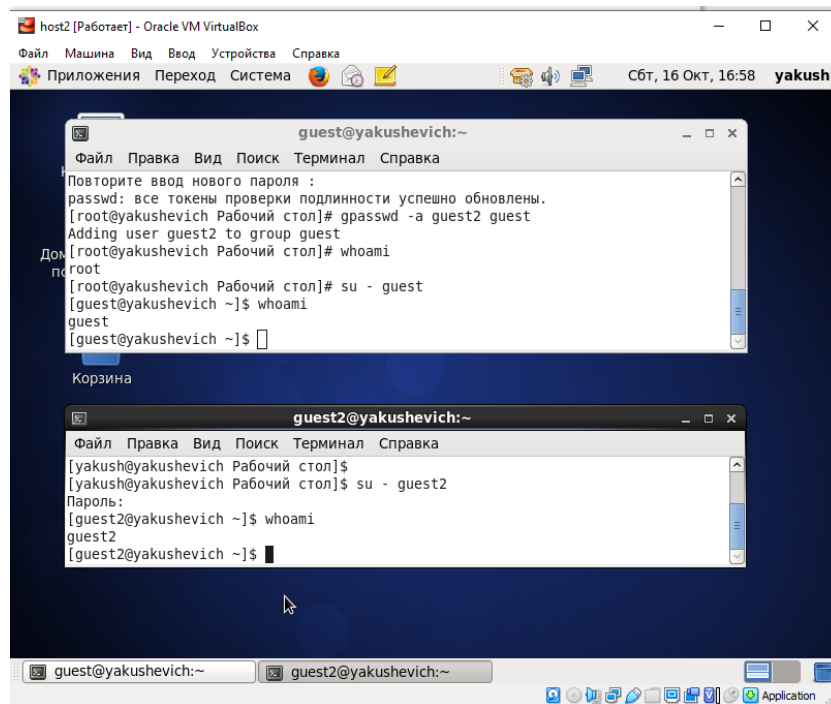


Рис. 3.5: Сравнение ключей

4 Выводы

Освоил на практике применение режима однократного гаммирования.

5 Ответы на контрольные вопросы

1. Одократное гаммирование - выполнение операции XOR между элементами гаммы и элементами подлежащего сокрытию текста. Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.
2. Недостатки однократного гаммирования: Абсолютная стойкость шифра доказана только для случая, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения.
3. Преимущества однократного гаммирования: во-первых, такой способ симметричен, т.е. двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение; во-вторых, шифрование и расшифрование может быть выполнено одной и той же программой. Наконец, Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении C все различные ключевые последовательности K возможны и равновероятны, а значит, возможны и любые сообщения P .
4. Длина открытого текста должна совпадать с длиной ключа, т.к. если ключ

короче текста, то операция XOR будет применена не ко всем элементам и конец сообщения будет не закодирован, а если ключ будет длиннее, то появится неоднозначность декодирования.

5. Операция XOR используется в режиме однократного гаммирования. Наложение гаммы по сути представляет собой выполнение побитовой операции сложения по модулю 2, т.е. мы должны сложить каждый элемент гаммы с соответствующим элементом ключа. Данная операция является симметричной, так как прибавление одной и той же величины по модулю 2 восстанавливает исходное значение.
6. Получение шифротекста по открытому тексту и ключу: $C_i = P_i \oplus K_i$
7. Получение ключа по открытому тексту и шифротексту: $K_i = P_i \oplus C_i$
8. Необходимы и достаточные условия абсолютной стойкости шифра: полная случайность ключа; равенство длин ключа и открытого текста; однократное использование ключа.

6 Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 7. Элементы криптографии. Однократное гаммирование.