1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ЛАБОРАТОРНАЯ РАБОТА № 13**

**«Защита web-сервера от несанкционированного доступа»**

1. по дисциплине «Основы информационной безопасности»
2. Выполнили
3. студенты гр. 5131001/40003 Веденеев А. С.
4. <*подпись*>  
     
    Поляков Д. Т.
   1. <*подпись*>
5. Преподаватель
6. ст. преподаватель Вагисаров В. Б.

<*подпись*>

1. Санкт-Петербург
2. 2025

# Цель и задачи

**Цель:** Приобретение практических навывок по созданию безопасной рабочей конфигурации web-сервера Apache, исследование функций безопасности web-сервера и механизмов контроля и управления доступом к страницам сайта.

**Задачи:**

## Установить на виртуальную машину web-сервер Apache, совместимый с версией операционной системы.

## Изучить и описать содержимое и значения параметров, указанных в файлах .htaccess и .htpasswd и созданных по умолчанию.

## Реализовать basic-аутентификацию пользователей web-сервера Apache. Для этого выполнить действия:

### **Создать файл с паролями .htpasswd.**

### **Создать пользователей user1, user2, user3, user4, user5 с непустыми паролями.**

### **Создать на web-сервере каталоги secret, not\_secret, some\_secret и разместить их в каталоге /var/www/html/.**

### **Разрешить доступ к каталогу secret только для пользователя user1.**

### **Разрешить доступ к каталогу not\_secret всем пользователям.**

### **Разрешить доступ к каталогу some\_secret пользователям user1, user3, user5.**

## Настроить групповой доступ. Для этого выполнить действия:

### **Создать файл .htgroup для работы с группами.**

### **В файле .htgroup прописать в строку: «admins:user1 user2.»**

### **В файле .htgroup прописать строку «users: user3 user4 user5».**

### **Привести файл .htaccess к нужному согласно пособию виду.**

### **Проверить доступ к папке secret.**

## Разработать программу подбора пароля пользователя, зашифрованного способом base64.

## Разработать программу, которая проверяет задаваемые пароли для пользователей web-сервера.

# ход работы

## На виртуальную машину был установлен web-сервер Apache, совместимый с Windows 10.

## При установке Apache файлы .htaccess и .htpasswd не были созданы, поэтому их содержимое не было изучено.

## Реализация basic-аутентификации пользователей web-сервера

### На том же диске, что и располагается Apache, были созданы категории «var/www/html». Далее в консоли Windows был запущен файл htpasswd.exe, после был создан файл паролей .htpasswd с помощью команды «-cp /var/www/html/.htpasswd user1» (рисунок 1).

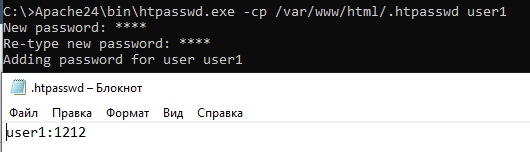


Рисунок 1. Содержимое файла .htpasswd после его создания

### С использованием все той же команды были созданы user2, user3, user4, user5 (рисунок 2).

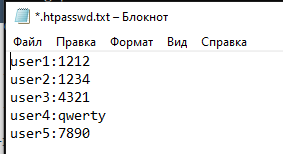


Рисунок 2. Содержимое файла .htpasswd после добавления еще 4 пользователей.

### В каталоге «/var/www/html/» были созданы 3 каталога с именами: «secret», «some\_secret», «not\_secret» (рисунок 3).

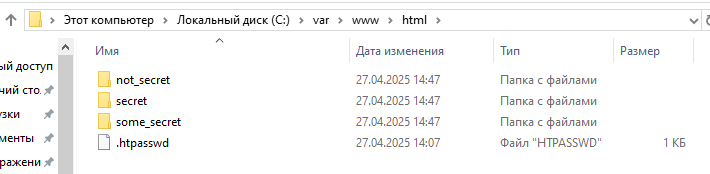


Рисунок 3. Каталоги web-сервера.

### Путем изменения файла .htaccess в каталоге «secret», был разрешен доступ к каталогу только для пользователя user1 (рисунок 4).

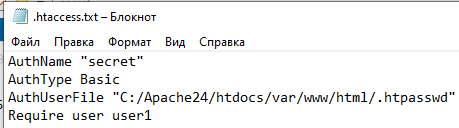


Рисунок 4. Файл .htaccess в каталоге «secret».

### Путем изменения файла .htaccess в каталоге «some\_secret», был разрешен доступ к каталогу для всех пользователей (рисунок 5).

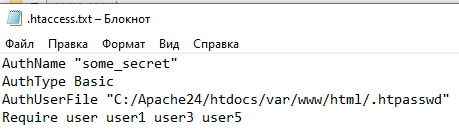


Рисунок 5. Файл .htaccess в каталоге «not\_secret».

### Путем изменения файла .htaccess в каталоге «some\_secret», был разрешен доступ к каталогу только для пользователей user1, user3, user5 (рисунок 6). Для того чтобы web-сервер работал, каталоги «var/www/html» и все их содержимое были перенесены в каталог «C:/Apache24/htdocs». Так же, в конфигурационном файле «httpd.conf» были изменены параметры «<Directory /, AllowOverride None, Require all denied» на «<Directory /var/www/html, AllowOverride All, Require all granted», а также параметр «AllowOverride None» в «<Directory "${SRVROOT}/htdocs">» на «AllowOverride All». Результат доступа к каталогам представлен в приложении 1.

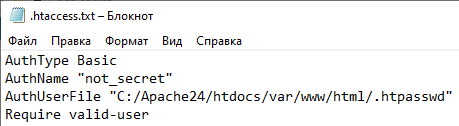


Рисунок 6. Файл .htaccess в каталоге «some\_secret».

## Настроить групповой доступ

### В каталоге «/var/www/html/» был создан файл .htgroup для работы с группами.

### В первую строчку файла .htgroup было прописано «admins: user1 user2».

### Во вторую строчку файла .htgroup было прописано «users: user3 user 4 user5» (рисунок 7).

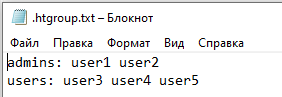


Рисунок 7. Содержимое файла .htgroup.

### Файл «.htaccess» в каталоге «secret» был изменен (рисунок 8).

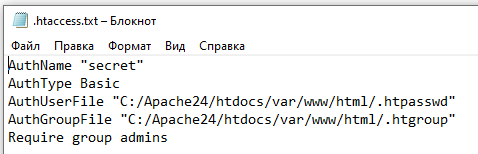


Рисунок 8. Содержимое файла «.htaccess» каталога «secret».

### Был проверен доступ к папке «secret» каждым пользователем, результат представлен в приложении 2.

## Была разработана программа подбора пароля пользователя, состоящего из не более 4 символов, букв английского алфавита, цифр и знака подчеркивания, зашифрованного способом base64, ее листинг представлен в приложении 3.

## Была разработана программа, которая проверяет задаваемые пароли для пользователей web-сервера на выполнение следующих условий: пароль состоит не менее чем из 10 символов, в пароле присутствуют как заглавные, так и строчные буквы латинского алфавита, а также содержит хотя бы один специальный символ и цифру. В случае, если пароль удовлетворяет условиям, то пользователь добавляется в файл «.htpasswd», листинг программы представлен в приложении 4.

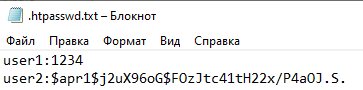


Рисунок 9. Результат работы программы для пункта 2.6.

# ответы на контрольные вопросы

## Какую роль web-сервер выполняет при работе с web-приложениями?

На web-сервере находится набор программ и данных, передаваемых клиентам, осуществляются вычисления, предоставляются ресурсы.

## Перечислите отличия web-приложения от обычного приложения

Web-приложения кроссплатформенны, независимы от программной и аппаратной среды, могут работать с большим количеством различных клиентов. Логика web-приложения реализуется на стороне сервера. Требуют наличие подключения к Интернету и web-браузера.

## Какие конфигурационные файлы используются при работе механизма контроля доступа к web-страницам сайта, размещенного на web-сервере Apache?

Конфигурационные файлы «.htpasswd» и «.htaccess» используются при работе механизма контроля доступа к web-страницам, первый из них содержит в себе информацию о том, какие пользователи и к каким каталогам имеют доступ, а второй содержит идентификационные и аутентификационные данные пользователей: логин и пароль.

## Опишите процесс basic-аутентификации со стороны клиента

Если пользователь не ввел пароль, или пароль неверен, то сервер отвечает заголовком «401 Unauthorized». Если web-браузер получает от сервера такое сообщение, то браузер должен отобразить пользователю диалоговое окно с требованием ввести имя пользователя и пароль. Как только имя и пароль введены пользователем, web-браузер должен повторить запрос к серверу и направить серверу имя пользователя и пароль. Web-сервер повторно проверяет имя пользователя и пароль и затем либо разрешает доступ, либо вновь отвечает сообщением «401 Unauthorized».

## Что позволяет нарушить безопасность web-сервера Apache при basic-аутентификации? Предложите и обоснуйте методы защиты от нарушений безопасности web-сервера Apache при basic-аутентификации.

Позволяет нарушить безопасность web-сервера хранение конфигурационных файлов в открытом виде. В случае, если злоумышленник получит доступ к данным файлам – даже если пароли представлены в виде хэшей, существует возможность их подобрать. Предлагаемые методы защиты – сложные пароли, тяжелые для полного перебора, установка пароля для конфигурационных файлов.

# Выводы

В ходе лабораторной работы были приобретены практические навыки по созданию безопасной рабочей конфигурации web-сервера Apache, были исследованы функции безопасности web-сервера и механизмов контроля и управления доступом к страницам сайта.

# Приложения

## Приложение 1

## Матрица доступа к каталогам «secret», «some\_secret», «not\_secret».

|  |  |  |  |
| --- | --- | --- | --- |
|  | secret | some\_secret | not\_secret |
| User1 |  |  |  |
| User2 |  |  |  |
| User3 |  |  |  |
| User4 |  |  |  |
| User5 |  |  |  |

## Приложение 2

**Матрица доступа к папке «secret» после настройки группового доступа**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **User1** | **User2** | **User3** | **User4** | **User5** |
| **Доступ** |  |  |  |  |  |

## Приложение 3

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include <windows.h>

const char base64\_list[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+\_";

char\* encode(char\* password) {

unsigned short num = 0; //число, которое будет получаться при делении нашего числа на 6 бит

unsigned short remainder = 0; //остаток от разделения на 6 бит

unsigned short bit6\_p = 2; //какое кол-во битов надо "отрезать"

unsigned short k = 0; //счетчик на какую позицию в хеше поставить символ

unsigned char hashpassword[9];

for (int i = 0; i < strlen(password); i++) {

num = password[i] >> bit6\_p;

num += remainder;

remainder = password[i] << (16 - bit6\_p);

remainder = remainder >> (8 + bit6\_p - ((i == 1) ? 2 : (i == 2) ? 4 : 0));

hashpassword[k] = base64\_list[num];

k++;

if (bit6\_p == 6) {

hashpassword[k] = base64\_list[remainder];

k++;

bit6\_p = 0;

remainder = 0;

}

bit6\_p += 2;

}

if ((strlen(password) == 1) || (strlen(password) == 4)) {

hashpassword[k] = base64\_list[remainder];

k++;

for (int i = 0; i < 2; i++, k++) {

hashpassword[k] = '=';

}

}

if (strlen(password) == 2) {

hashpassword[k] = base64\_list[remainder];

k++;

hashpassword[k] = '=';

k++;

}

hashpassword[k] = '\0';

return hashpassword;

}

void bruteforce(char\* hash) {

char password[5];

for (short i0 = 0; i0 < 64; i0++) {

password[0] = base64\_list[i0];

password[1] = '\0';

if (strcmp(hash, encode(password)) == 0) {

printf("Ваш пароль: %s", password);

return;

}

}

for (short i0 = 0; i0 < 64; i0++) {

for (short i1 = 0; i1 < 64; i1++) {

password[0] = base64\_list[i0];

password[1] = base64\_list[i1];

password[2] = '\0';

if (strcmp(hash, encode(password)) == 0) {

printf("Ваш пароль: %s", password);

return;

}

}

}

for (short i0 = 0; i0 < 64; i0++) {

for (short i1 = 0; i1 < 64; i1++) {

for (short i2 = 0; i2 < 64; i2++) {

password[0] = base64\_list[i0];

password[1] = base64\_list[i1];

password[2] = base64\_list[i2];

password[3] = '\0';

if (strcmp(hash, encode(password)) == 0) {

printf("Ваш пароль: %s", password);

return;

}

}

}

}

for (short i0 = 0; i0 < 64; i0++) {

for (short i1 = 0; i1 < 64; i1++) {

for (short i2 = 0; i2 < 64; i2++) {

for (short i3 = 0; i3 < 64; i3++) {

password[0] = base64\_list[i0];

password[1] = base64\_list[i1];

password[2] = base64\_list[i2];

password[3] = base64\_list[i3];

password[4] = '\0';

if (strcmp(hash, encode(password)) == 0) {

printf("Ваш пароль: %s", password);

return;

}

}

}

}

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char hash[15];

printf("Введите хеш:\n");

scanf("%s", hash);

bruteforce(hash);

}

## Приложение 4

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <string.h>

#include <windows.h>

const char small\_symbols[] = "abcdefghijklmnopqrstuvwxyz";

const char big\_symbols[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

const char digits[] = "0123456789";

const char unique\_symbols[] = "\*-+/.?,=\_)(&^%$#@!~}{><;:[]";

void check(char\* login) {

char password[100];

printf("Введите пароль:");

scanf("%s", password);

char \*flag\_small, \*flag\_big, \*flag\_digits, \*flag\_unique;

flag\_small = strpbrk(password, small\_symbols);

flag\_big = strpbrk(password, big\_symbols);

flag\_digits = strpbrk(password, digits);

flag\_unique = strpbrk(password, unique\_symbols);

while (strlen(password) <= 10 || flag\_small == NULL || flag\_big == NULL || flag\_digits == NULL || flag\_unique == NULL) {

printf("Пароль должен подходить под условия: состоять не менее чем из десяти символов, а также содержать:\nОдну заглавную букву латинского алфавита \nОдну прописную букву латинского алфавита \nОдну цифру и один уникальный символ!\nВведите пароль:");

scanf("%s", password);

flag\_small = strpbrk(password, small\_symbols);

flag\_big = strpbrk(password, big\_symbols);

flag\_digits = strpbrk(password, digits);

flag\_unique = strpbrk(password, unique\_symbols);

}

char buffer[300];

sprintf(buffer, "C:\\Apache24\\bin\\htpasswd.exe -b C:\\Apache24\\htdocs\\var\\www\\html\\.htpasswd %s %s", login, password); //ключ -b для создания/обновления пользователя без повторного ввода пароля

system(buffer);

printf("\nПользователь %s добавлен!", login);

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char login[100];

printf("Введите логин:\n");

scanf("%s", login);

check(login);

}