1. Министерство высшего образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 13**

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

1. по дисциплине «Основы Информационной Безопасности»
2. Выполнил
3. студент гр. Белоконь Д. А.
4. <*подпись*>

Проверил Пахомов М. А.

1. <*подпись*>

1. Санкт-Петербург
2. 2023
3. **Цель работы**

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

Формулировка задания

Необходимо настроить и запустить web-сервер Apache, а также сконфигурировать доступ к определённым директориям различным пользователям и группам. Для этого также нужно подключить базовую аутентификацию.

Результаты

Для начала скачаем и настроим виртуальную машину. Работать будем через приложение VirtualBox с дистрибутивом Manjaro на базе Linux. В системе уже установлен базовый http сервер, который и представлен Apache2.

Далее, разрешим в главном конфигурационном файле изменение доступа к директориям локальными файлами, то есть файлами .htaccess. Кроме того, для дальнейшей работы, с помощью утилиты htpasswd создадим в главной папке документ .htpasswd, куда запишем 5 базовых пользователей с их паролями.

Затем в сами созданных в доступной для просмотра директории каталогах добавим файлы .htaccess, где пропишем доступ соответствующим пользователям, указанным в методическом материале:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | User1 | User2 | User3 | User4 | User5 | admins |
| secret | 1 | 0 | 0 | 0 | 0 | 1 |
| not\_secret | 1 | 1 | 1 | 1 | 1 | 1 |
| some\_secret | 1 | 0 | 1 | 0 | 1 | 0 |

Кроме того, разместим рядом с .htpasswd документ .htgroup, в котором обозначим принадлежность двух первых пользователей группе администраторов, а прочих — простых пользователей. Завершим работу с web-сервером добавлением в конфигурацию доступа директории secret необходимости принадлежности группе admins.

Практическая проверка доступности разных папок нужным пользователям оказывается успешной. Идём дальше.

Напишем на языке C программу, переводящую четырёхзначный пароль в base64 с последующей попыткой перебрать все возможные значения для обнаружения введённого. Воспользуемся уже написанной библиотекой base64. Прочий код представляет собой получение данных от пользователя, кодирование их и перебор четырьмя циклами for строки с переводом в base64 на каждой итерации. Когда полученный и сгенерированный коды совпадают, мы выводим сообщение об успехе.

В конце, скачаем из официального Github-аккаунта Apache репозиторий httpd. Откроем из него файл htpasswd.c и поищем в нём и во включённых в него файлах упоминание слов ввода пароля. После найденной строки вставим код проверки введённого пароля соответствующим критериям из методического материала, и в случае провала будем выводить информацию о допущенной ошибке и запрашивать ввод ещё раз.

Вывод

Лабораторная получилась достаточно простой благодаря тому, что системы на базе Linux легко настраиваются и имеют чёткие инструкции, следуя которым можно достичь результата. Никаких трудностей в процессе работы не возникло, зато были получены знания о том, что всё же представляют собой web-сервера.

**Приложение**

Приложение 1. Программа подбора пароля.

int main(int argc, char \*argv[]) {

if (argc < 2) {

fprintf(stderr, "No input file specified");

return -1;

}

FILE \*inputFile = fopen(argv[1], "r");

if (inputFile == NULL) {

fprintf(stderr, "Could not open input file");

return -2;

}

unsigned char controlSum;

char symbol;

while ((symbol = fgetc(inputFile)) != EOF) {

controlSum += symbol;

}

fclose(inputFile);

printf("Control sum for the file is: %d\n", controlSum);

return 0;

}

Приложение 2. Изменения в коде web-сервера Apache htpasswd.

int includesStrongLetters(char \*buffer) {

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

if (buffer[i] < 91 && buffer[i] > 64)

return 1;

}

return 0;

}

int includesWeakLetters(char \*buffer) {

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

if (buffer[i] < 123 && buffer[i] > 96)

return 1;

}

return 0;

}

int includesNumbers(char \*buffer) {

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

if (buffer[i] < 58 && buffer[i] > 47)

return 1;

}

return 0;

}

int includesSpecialSymbols(char \*buffer) {

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

if (buffer[i] == 95) return 1;

}

return 0;

}

int personalCheck(char \*buffer) {

if (strlen(buffer) < 10)

return 1;

if (!includesStrongLetters(buffer) || !includesWeakLetters(buffer))

return 2;

if (!includesNumbers(buffer))

return 3;

if (!includesSpecialSymbols(buffer))

return 4;

return 0;

}

int checkResult = personalCheck(buf);

while (checkResult) {

switch (checkResult) {

case 1:

puts("Password should be longer than 9 symbols.");

break;

case 2:

puts("Should include letters of different cases.");

break;

case 3:

puts("Should include numbers.");

break;

case 4:

puts("Should include special symbols.");

break;

}

if (apr\_password\_get("New password: ", buf, &bufsize) != 0)

goto err\_too\_long; checkResult = personalCheck(buf);

}